Algorithmic Game Theory	Problem Set 3
CS 684 Fall 2005	Due Friday, November 18, 2005

We will maintain a FAQ for the problem set on the course Web page. You may use any fact we proved in class without proving the proof or reference. However, you may not use published papers.

You are expected to attempt all problems. If you cannot solve a problem, write down how far you got, and why are you stuck.

Cooperation in developing answers is encouraged. However, each student must write down all answers separately.

(1) We know that in a game with a finite set of players, where each player has a finite set of pure strategies, the game has a Nash equilibrium. Our algorithm was based on Brouwer's fixed point theorem. Unfortunately, the proof of the fixed point theorem is not algorithm. It is a major outstanding open problem whether a Nash equilibrium in such a game can be found in polynomial time. In this problem, we explore if one can at least do this in finite time (maybe exponential). Consider the special case with two players. To be more formal, assume that there are 2 players, and player i chooses between n_i pure strategies. Assume that the game is given by listing the payoff for each player for each $n_1 \times n_2$ possible plays (this is the traditional payoff matrix that we called matrix A and B in class on Wednesday, Nov 2nd).

(a) Give a polynomial time algorithm to check if there is a Nash equilibrium strategy for the game in which each player mixes between at most two strategies.

(b) Give a finite algorithm for finding a Nash equilibrium for general games with two players. Your algorithm may run in exponential time.

(2) Consider a two player game with with two reward matrices A and B as also used in the previous problem, and assume that both players have n possible strategies (so A and B are n by n matrices. Assume that the matrix A and B has random entries, say all entries in the range $[0, 1]$ filed out uniformly independently at random. Show that the provability that this random game has a pure (deterministic) Nash equilibrium is roughly $1 - 1/e$ if n is large. You may use the fact that for large n we have that $(1 - 1/n)^n \approx 1/e$.

Warning. You may want to compute the probability that a pair of strategies (i, j) forms a Nash. Unfortunately, these events are **not** independent!

(3) We have seen that finding a Nash in a 2-person 0-sum game is significantly easier than general 2 person games. Now consider a 3 person 0-sum game, that is a game when the reward of the 3 players always sums to 0. Show that finding a Nash equilibrium in a 3 person 0-sum game is at least as hard as a 2-person general game.

(4) Consider an n person game where each player has only two strategies. This game has 2^n possible outcomes (for the 2^n ways the n people can play, so even giving the game in the above matrix form is rather problematic (takes exponential information). One special class of games that one can consider is called graphical games. The idea is that the value (or payoff) of a player can only depend on a subset of players. We can define a dependence graph G whose nodes are the players, and we put an edge between two players i and j if the payoff of player i depends on the strategy of player j or vica versa. This way if node i has only some k neighbors, than his payoff only depends on his own strategy and the strategy of his neighbors. If the degree is a nice is bounded by 3 then his payoff values can be given by only $2^4 = 16$ entries. Now consider a game where the players each have 2 pure strategies, and assume that the graph G is a tree with maximum degree 3. Give a polynomial time algorithm to decide if the game has a pure Nash equilibrium. (Note that there are 2^n possible pure strategies.)

(5) Consider an n person game where each player has 2 strategies. For this problem, think of the strategies as "on" and "off", i.e., the strategy is either to participate, or not to participate. Further assume that all players have the same payoff functions, and that the payoff for a player only depends on whether this player is "on", and the number of people playing strategy "on". So the game is defined by $2n$ values: $u_{on}(k)$ and $u_{off}(k)$ that denote the payoff for playing the "on" and "'off" strategy assuming k players chose to play "on" with $k = 0, \ldots n$ (where $u_{on}(0)$ and $u_{off}(n)$ is meaningless). Give a polynomial time algorithm to find a correlated equilibrium for such a game. Note that the input to this problem consists of the $2n$ numbers above, so polynomial means, polynomial in this input length. You may use that linear programming is solvable in polynomial time.