# CS364A: Problem Set #5/Take-Home Final

Due to the TAs by noon on Friday, December 13, 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 30

(15 points, plus 10 more extra credit) Recall atomic selfish routing games with parallel edges (two vertices, a common source $s$ and a common sink $t$, and $m$ parallel edges from $s$ to $t$). This problem considers the generalization in which players can have different cost functions: each player $j$ incurs a cost $c_i^j(k)$ on the $i$th edge if it is one of $k$ players using the edge. Assume that for each fixed $j$ and $i$, $c_i^j(k)$ is nondecreasing in $k$.

Consider the version of best-response dynamics in which you begin at an arbitrary outcome, and as long as the current outcome is not a (pure-strategy) Nash equilibrium, an arbitrary player who can benefit by deviating is allowed to switch its strategy (i.e., its chosen edge). Assume that a deviating player switches to its best (i.e., cost-minimizing) strategy. You can assume that there are no ties, with all $c_i^j(k)$'s distinct.

Solve two of the following four problems for full credit. You will receive extra credit for each problem after the first two 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 edges, then best-response dynamics converges to a pure-strategy Nash equilibrium.

## Problem 31

(10 points) Show by example — an explicit (non-zero-sum) two-player game — that the time-averaged history of joint play generated by no-regret dynamics need not converge to a mixed Nash equilibrium.

## Problem 32

(10 points) Show that each of the following problems can be formulated directly as a linear programming problem.

(i) Given the payoff matrix of a two-player zero-sum game, compute a mixed-strategy Nash equilibrium.

(ii) Given a $k$-player game in "normal form" — a list of the payoffs $\pi_i(\mathbf{s})$ for every player $i$ and outcome $\mathbf{s} \in S_1 \times \cdots \times S_k$ — compute a coarse correlated equilibrium.

(iii) Given a $k$-player game in normal form, compute a correlated equilibrium.

## Problem 33

Consider a two-player, non-zero sum game, specified by payoff matrices $A$ and $B$, in which both players have $n$ strategies. Assume that all payoffs are rational numbers between 0 and 1.

(a) (4 points) The *support* of a mixed strategy is the set of pure strategies that are played with strictly positive probability. Suppose you were promised that there was a mixed-strategy Nash equilibrium of $(A, B)$ with row and column supports $S$ and $T$, respectively. Show how you can then recover such an equilibrium in polynomial time.

[Hint: you may use the fact that linear programs can be solved in polynomial time.]

(b) (3 points) Give a $2^{O(n)}$-time algorithm to compute an exact mixed-strategy Nash equilibrium of a two-player game.

(c) (9 points) Consider a mixed-strategy Nash equilibrium $(x^*, y^*)$ of $(A, B)$. Suppose we draw $K = (s \log n)/\epsilon^2$ independent samples $(r_1, c_1), \ldots, (r_K, c_K)$ from the product distribution $x^* \times y^*$, where $\epsilon > 0$ is a parameter and $s > 0$ is a sufficiently large constant. Form the corresponding empirical distributions $\hat{x}$ and $\hat{y}$, where components are the frequencies of play in the sequence. Prove that, with high probability, $(\hat{x}, \hat{y})$ is an $\epsilon$-approximate Nash equilibrium in the sense that every strategy in a player's support has expected payoff within (additive) $\epsilon$ of that of a best response.

[Hint: if you don't know about them, read up about Chernoff Bounds.]

(d) (5 points) Use (an extension of) (a) and (c) to give an $n^{O((\log n)/\epsilon^2)}$-time algorithm for computing an $\epsilon$-approximate mixed-strategy Nash equilibrium of a two-player game.

(e) (4 points) Extend your algorithm and analysis to compute, with the same running time bound, an $\epsilon$-approximate mixed-strategy Nash equilibrium with expected total payoff close to that of the best (maximum-payoff) exact Nash equilibrium.

## Problem 34

We continue the study of algorithms for approximate Nash equilibria in bimatrix games, in the sense of the previous problem. However, we'll use a weaker definition of approximate equilibria than in Problem 33: now, we'll call $(\hat{x}, \hat{y})$ an *$\epsilon$-approximate Nash equilibrium* if $\hat{x}^T A\hat{y} \geq x^T A\hat{y} - \epsilon$ for all row mixed strategies $x$ and $\hat{x}^T B\hat{y} \geq \hat{x}^T By - \epsilon$ for all column mixed strategies $y$.

(a) (5 points) Prove that there is a $\frac{1}{2}$-approximate Nash equilibrium in which one player plays a pure strategy and the other randomizes uniformly between two strategies.

(b) (10 points) Consider the following idea for improving on (a). Suppose we form the matrix $A-B$, view it as a zero-sum game, and solve for an exact mixed-strategy Nash equilibrium $(x^*, y^*)$. (By Problem 32, this can be done in polynomial time.) Perhaps $(x^*, y^*)$ is an $\epsilon$-approximate Nash equilibrium for $(A, B)$ for a satisfactory value of $\epsilon$; if not, we generalize (a) and have one player play a pure strategy while the other randomizes (not necessarily uniformly) between a pure strategy and its optimal strategy for $A - B$.

Can you use these ideas (with appropriate parameter choices) to improve upon the result in (a)? What is the minimum value of $\epsilon$ for which your algorithm computes an $\epsilon$-approximate Nash equilibrium in polynomial time?

## Problem 35

Consider again the problem of computing some exact mixed-strategy Nash equilibrium of a (non-zero-sum) bimatrix game.

(a) (5 points) A bimatrix game $(A, B)$ is *symmetric* if both players have the same strategy set and $B = A^T$. (Rock-Paper-Scissors and the Prisoner's Dilemma are both examples.) Nash's theorem (or the Lemke-Howson algorithm) can be adapted to show that every symmetric game has a *symmetric* Nash equilibrium, in which both players employ the same mixed strategy. Prove that the problem of computing a mixed Nash equilibrium of a general bimatrix game polynomial-time reduces to that of computing a symmetric Nash equilibrium of a symmetric bimatrix game.

(b) (5 points) Prove that the problem of computing a mixed Nash equilibrium of a general bimatrix game polynomial-time reduces to that of computing *any* Nash equilibrium of a symmetric bimatrix game.

(c) (5 points) Prove that the problem of computing a mixed Nash equilibrium of a general bimatrix game polynomial-time reduces to that of computing an *asymmetric* Nash equilibrium of a symmetric bimatrix game, if one exists.