UPenn NETS 412: Algorithmic Game Theory
Midterm Practice Problems

Instructor: Bo Waggoner

Problem 1

Here we'll analyze a game where each player has infinitely many choices of actions.

Alice and Bob are working on a project together. They both value their free-time, which they give up for working on the project. Let x_a represent the fraction of the day (expressed as a real value in $[0, 1]$) Alice spends on the project and x_b the fraction of the day Bob spends. The value of the outcome of the project is given by $4x_1x_2$, and Alice and Bob split this quantity equally. The payoff to each player is the value of the outcome of the project minus the effort that player put it.

Part a Write Alice's payoff as a function of x_a and x_b.

$$u_a = \cdots\cdots\cdots$$

Solution. We have

$$u_a = \frac{1}{2}4x_ax_b - x_a = 2x_ax_b - x_a = x_a(2x_b - 1)$$

All of these are equivalent.

Part b Write Bob's payoff as a function of x_a and x_b.

Solution. Symmetrically,

$$u_b = \frac{1}{2}4x_ax_b - x_b = 2x_ax_b - x_b = x_b(2x_a - 1)$$

Part c If Bob puts in effort $x_b = .75$, what is Alice's best-response?

Solution. If $x_b = .75$, we have $u_a = x_a(2(.75) - 1) = 0.5x_a$. The derivative with respect to x_a is $\frac{du_a}{dx_a} = 0.5$. So Alice's utility only gets bigger the more we increase x_a, so Alice should choose x_a as large as possible. Thus she should pick $x_a = 1$.

Part d Now, if Alice is putting in effort $x_a = 1$, what is Bob's best-response?

Solution. If $x_a = 1$, $u_b = x_b(2(1) - 1) = x_b$. Again, this only gets bigger as we increase x_b, so Bob should pick $x_b = 1$.

Part e Why is $x_a = x_b = 1$ a Nash equilibrium of this game?

Solution. We just saw that $x_b = 1$ is Bob's best-response to $x_a = 1$, and by symmetry $x_a = 1$ is Alice's best-response to $x_b = 1$.

Part f Now, if Alice is putting in effort $x_a = .25$, what is Bob's best-response?

Solution. We have $u_b = x_b(2(.25) - 1) = x_b(-.5)$. This only gets smaller as we increase x_b, or equivalently only gets larger as we *decrease* x_b, so Bob wants to pick x_b as small as possible, so he should pick $x_b = 0$.

Part g What is Alice's best-response to Bob's choice of $x_b = 0$?

Solution. If $x_b = 0$, then we have $u_a = x_a(2(0) - 1) = -x_a$. Again, this only gets bigger as we decrease x_a, so Alice should pick $x_a = 0$.

Part h Why is $x_a = x_b = 0$ a Nash equilibrium?

Solution. We just saw that $x_a = 0$ is Alice's best-response to $x_b = 0$. By a symmetric argument, $x_b = 0$ is Bob's best-response to $x_a = 0$.

Part i What value of x_b makes Alice indifferent between all of her possible choices of x_a? Hint: Alice is indifferent between all of her options when her payoff is the same regardless of her choice of x_a.

Solution. Alice's utility is $u_a = x_a(2x_b - 1)$. So $\frac{du_a}{dx_a} = 2x_b - 1$. Observe that if Bob sets $x_b = .5$, then $2(x_b) - 1 = 0$. In this case, regardless of her choice of x_a, Alice's utility doesn't change (in fact, her utility equals zero for any choice of x_a.) So $x_b = 5$.

Part j Why is $x_a = x_b = .5$ a Nash equilibrium?

Solution. We just saw that at $x_b = .5$, Alice is indifferent between all of her choices of x_a, so every one of her choices, including $x_a = .5$ is a best-response to $x_b = .5$. Symmetrically for Bob, he is indifferent between all of his choices when $x_a = .5$, so every choice of x_b, including $x_b = .5$ is a best-response. Therefore, $x_a = x_b = .5$ is a point where Alice and Bob are best-responding to each other.

Problem 2

Traffic is a congestion game played by n players, all starting from a source node S and traveling along routes represented by directed edges to a common destination node T. Each player's action set consists of all paths from A to T.

The delay along any edge (u, v) is equal to the number of players who chose to travel that edge, which we will write $n(u, v)$. Each player i's cost is equal to the sum of delays along all edges in her path.

For parts a and b, consider the following graph. Here, $n(S, A)$ refers to the number of players who travel along edge (S, A), and so on.

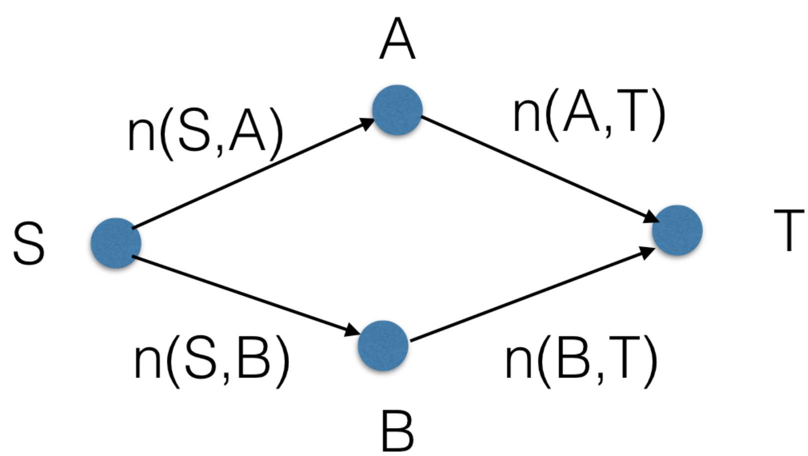

Figure 1: Graph for parts a and b

Part a Suppose n is even. Find a pure strategy Nash equilibrium, and prove that it is an equilibrium.

Solution. Notice that any path that begins with (S, A) must follow through (A, T), and any path that begins with (S, B) must follow through (B, T). Hence, the only important choice for a player to make is whether to start with (S, A) or (S, B).

Suppose n_A players start with (S, A) and n_B start with (S, B). Then a player starting with S_A has total cost

$$\begin{aligned} n(S, A) + n(A, T) &= n_A + n_A \\ &= 2n_A. \end{aligned}$$

Similarly, a player starting with (S, B) has total cost $2n_B$. So a player's best response will be to switch if there are fewer players on the other path than hers.

So if n is even, we claim that $n/2$ agents choosing (S, A) and $n/2$ agents choosing (S, B) is an equilibrium. In this case $n_A = n_B = n/2$, so all players have a payoff of n, and any player who switched to the other path would do worse.

Part b Suppose n is odd. Find a pure strategy Nash equilibrium, and prove that it is an equilibrium.

Solution. One pure strategy equilibrium is for $n_A = (n+1)/2$ players to choose the top path and $n_B = (n-1)/2$ players to choose the bottom path. Each player choosing the top path has cost $2n_A = (n+1)$. If she switched to the bottom path, then she would have payoff $2(n_B + 1) = n + 1$, so she is indifferent between switching and staying. (Note that it is $2(n_B + 1)$ because there would be one more person along the bottom path.)

Similarly, if any player on the bottom path switched to the top path, she would get worse utility (should would get $n + 2$ instead of $n - 1$). So this is an equilibrium.

Part c For this part, consider the following graph. Outline the argument that this game has a pure strategy Nash equilbrium. (You do not have to find it!) You can assume in your argument that this is a congestion game (it is actually a slight generalization).

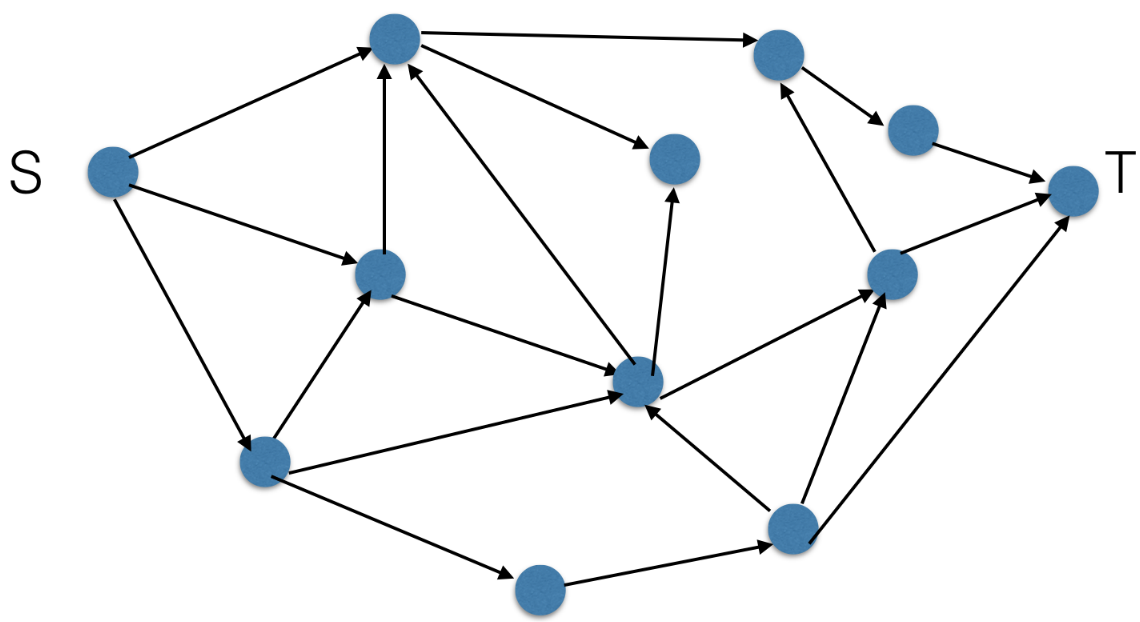

Figure 2: Graph for part c

Solution. Despite the complexity of this graph, this is basically the same as parts a and b. Notice that as written, *Traffic* mirrors the very first example of congestion games we saw (but did not solve) - network routing. Here is an outline of the argument:

- We can run BRD on a congestion game starting from any strategy profile.
- If BRD halts, it must be on a strategy profile that is a pure strategy Nash equilibrium.
- We can give a potential function for this game, and it decreases monotonically as players improve their responses to the current strategy profile.
- Since there are only a finite number of actions, the potential function has only a finite number of values. By this and the previous point, BRD must halt.

Since BRD must halt, and only halts on a pure strategy Nash equilibrium profile, a PSNE must exist.

Problem 3

In class, we proved that the polynomial weights algorithm achieves "no regret" – that is, for arbitrary sequences of losses, it guarantees that the difference between the average loss achieved by the algorithm, and the average loss achieved by the best expert in hindsight is $o(1)$ – tending to zero as $T \to \infty$. Recall that the polynomial weights algorithm is randomized. Here we show that no deterministic algorithm can obtain the same guarantee.

Part a Consider the algorithm "Follow the Leader" that always picks the expert that has the lowest cumulative loss so far – i.e. at day j it picks expert k such that:

$$k = \arg\min_i \sum_{t=1}^{j-1} \ell_i^t.$$

(Suppose for concreteness that if there is a tie, the algorithm picks the min-loss expert with the smallest index). Show that Follow the Leader is not a no-regret algorithm. i.e. exhibit a sequence of losses such that for all T, the regret of the algorithm is $\Omega(T)$.

Solution. Consider the case of $k = 2$ experts and the sequence of loss vectors $\ell^1 = (1, 0)$, $\ell^2 = (0, 1)$, $\ell^3 = (1, 0), \ldots$ where in general, $\ell^j = (1, 0)$ if j is odd and $\ell^j = (0, 1)$ if j is even. Follow the leader picks expert 1 on odd days and expert 2 on even days, and so experiences loss 1 every day, for a cumulative loss of $L_{FTL}^T = T$. On the other hand, each fixed expert has cumulative loss only $L_1^T = L_2^T = T/2$. Hence, the total (unnormalized) regret of the algorithm is $T/2$, and the total normalized regret is $1/2$ – i.e. Follow the Leader is not a no regret algorithm.

Part b Prove that no *deterministic* experts algorithm can achieve $o(1)$ regret – i.e. that randomization is necessary to achieve a guarantee like that of the polynomial weights algorithm.

Hint: Consider any fixed deterministic algorithm, and then place yourself in the role of an adversary who is trying to foil it. Can you, knowing which expert the algorithm is going to pick next, design a sequence of losses so that after T rounds, the best expert always has cumulative loss that is lower than the algorithm's loss by at least $T/2$?

Solution. Observe that for any deterministic algorithm, the expert i_t that the algorithm chooses at round t is entirely determined by the sequence of losses $\ell^1, \ldots, \ell^{t-1}$ realized on days $< t$. Using this fact, an adversary can inductively construct a series of loss vectors to guarantee a large gap between the loss of the algorithm and the loss of the best expert in hindsight. Here is one simple way: At day t, construct the loss vector ℓ^t such that $\ell^t_{i_t} = 1$, and $\ell^t_j = 0$ for all experts $j \neq i_t$. By construction, the algorithm experiences loss 1 every day and so has cumulative loss $L^T_A = T$. However, the average cumulative loss of all experts is only T/k (since only 1 of k experts has nonzero loss each day), and so by averaging, the best expert in hindsight j^* must have cumulative loss $L^T_{j^*} \leq T/k$. Hence, the unnormalized regret of the algorithm is at least $(1 - 1/k)T$, and the normalized regret is at least $1 - 1/k$ – i.e. not tending to zero with T.