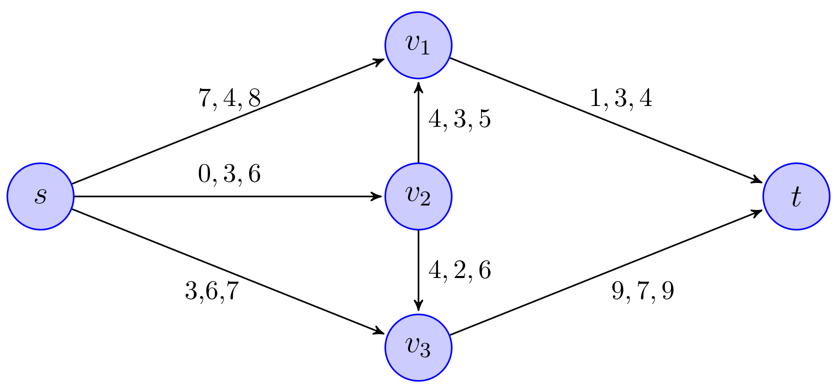
AGTA Tutorial 5 Solutions

Lecturer: Aris Filos-Ratsikas Tutor: Charalampos Kokkalis

March 18, 2025

Exercise 1. Consider the *atomic network congestion game*, with three players, described by the following directed graph.



In this game, every player i (for $i = 1, 2, 3$) needs to choose a directed path from the source s to the target t. Thus, every player i's set of possible actions (i.e., its set of pure strategies) is the set of all possible directed paths from s to t.

Each edge e is labeled with a sequence of three numbers (c_1, c_2, c_3). Given a profile $\pi = (\pi_1, \pi_2, \pi_3)$ of pure strategies (i.e., s-t-paths) for all three players, the *cost* to player i of each directed edge, e, that is contained in player i's path π_i, is c_k, where k is the total number of players that have chosen edge e in their path. The total cost to player i, in the given profile π, is the sum of the costs of *all* the edges in its path π_i from s to t. Each player of course wants to minimize its own total cost.

Compute a pure strategy Nash Equilibrium in this atomic network congestion game.

Solution 1. As we are working with a congestion game, we can find a pure Nash Equilibrium by starting at any pure strategy profile, and iteratively improving it until we cannot improve any further. To get a concrete starting point, let's say all players take the route $s \to v_3 \to t$. Then we can do iterative improvements for example[1] as follows:

A. Player 1 switches to $s \to v_2 \to v_1 \to t$

B. Player 2 switches to $s \to v_2 \to v_1 \to t$

C. Player 3 switches to $s \to v_1 \to t$.

D. Player 2 switches to $s \to v_1 \to t$

[1] At many stages in the process there is more than one option on who improves and how.

At **D.** no further improvements can be made, so we reached the following PNE:

Player 1: $s \to v_2 \to v_1 \to t$
Player 2: $s \to v_1 \to t$
Player 3: $s \to v_1 \to t$

Note that in the above sequence we were not done at stage **C**, even though every player had switched once. Other starting points will take through other sequences of steps, and they might end up in a different PNE, although it turns out that in this game all pure Nash equilibria send two players via the route $s \to v_1 \to t$ and one via $s \to v_2 \to v_1 \to t$, differing only in which player chooses the path $s \to v_2 \to v_1 \to t$.

Note that there are "different" pure PNEs in this game for a trivial reason: we can arbitrarily rename the players in any permutation. However, this is only a different pure PNE in a trivial sense, by renaming which player plays which strategy. It turns out that in this game there is no other "fundamentally different" PNE, meaning a PNE consisting of an entirely different set of 3 paths from s to t.

In general however, such an atomic network congestion came can have multiple genuinely different PNEs, where the players have completely different costs in the different PNEs.

Exercise 2. Consider the class of linear network congestion games, i.e., network congestion games with linear cost functions $c_e(x) = a_e \cdot x + b_e$. For these games, the cost functions can be represented by providing a_e and b_e for each edge $e \in E$ by its binary representation. The strategy sets can be represented *implicitly*, by using the graph G as an input (e.g., represented as an adjacency list) and the origin-destination (source-target) pair (o_i, d_i) for each player i.

- Explain why the best response dynamics algorithm for finding a pure Nash equilibrium of such a game is still a pseudopolynomial time, under this representation of the cost functions.
- Now that the strategies are implicitly represented rather than explicitly, can the best response of an agent be computed in polynomial time? Justify your answer.

Solution 2. Before we discuss linear network congestion games, let us first discuss the representation of the inputs in the general class of congestion games. The input consists of the number of player n and the number of resources m. Since we need to specify strategies for each agent and costs for each resource, we need to be able to iterate over n and m, so these will not be represented in binary. So an algorithm that runs in time $\text{poly}(n, m)$ is indeed polynomial. The strategy set of each agent is given explicitly in the input as a set $S_i \subseteq 2^m$, i.e., it could be exponential in the number of resources m. Still, an algorithm that runs in time $\text{poly}(m, n, \max_i |S_i|)$ is polynomial.

Furthermore, the cost functions are part of the input, and these need to be represented as well. How do we represent a function? If the function is *general*, i.e., it does not have any specific structure that allows us to represent it more succinctly, we will have to represent it as a map (e.g., a two-dimensional array) by explicitly stating for each possible input (i.e., each number $n_i \in 1, 2, \ldots, n$) the corresponding output, i.e., a number c_r, such that $c_r = c_r(n_i)$, where c_i is represented in binary. If we look at one entry of the map, the input is bounded by n, and the output is bounded by the binary representatio of c_r, i.e., the number of bits needed to represent c_r. In terms of space, this is $\Theta(\log c_r)$. So, an algorithm that runs in time $\text{poly}(m, n, \max_i |S_i|, \log \max_r c_r)$ is polynomial in the size of the input, but an algorithm that runs in time $\text{poly}(m, n, \max_i |S_i|, \max_r c_r)$ is exponential (and in fact, pseudopolynomial, since c_r are the numerical parameters of the input).

- Now let's look at linear congestion games. Now the cost functions have a more explicit representation: for each resource (edge) e, we instead give a_e and b_e in the input; crucially, these are again represented in binary. Therefore, a polynomial time algorithm would need to run in time $\text{poly}(m, n, \log \max_e(a_e \cdot n + b_e))$. The best-response dynamics algorithm however runs in time $\text{poly}(m, n, \max_e(a_e \cdot n + b_e))$ and thus it is only pseudopolynomial. The following fact is easy to verify, by looking at the maximum value of the potential function in Rosenthal's potential.

- The best response of an agent can still be computed in polynomial time. The potential issue here is that we no longer have an input representation of the strategies that is exponential in the number of resources m. In other words, we no longer provide all possible paths in the input, but only the origin and the destination for each player. Note that this (origin, destination) pair in itself does not specify a strategy, but the strategy can be recovered by this pair as well as a path from the origin to the destination. To compute the best response, a player would need to be able to verify whether a given strategy profile is an equilibrium, i.e., to find a (o_i, d_i) pair that minimises her cost, fixing the strategies (paths) of the other players. This boils down to a minimum-cost path computation on the network, which can be done e.g., using the Bellman-Ford algorithm.

Exercise 3.

A. Consider any linear congestion game (i.e., a congestion game with linear cost functions), and any strategy profile $s \in S_1 \times \ldots \times S_n$ in this game. Let $\Phi(s)$ be the value of Rosenthal's potential function on input s, and let $\mathrm{SC}(s)$ be the social cost of s. Show that

$$\frac{1}{2}\mathrm{SC}(s) \leq \Phi(s) \leq \mathrm{SC}(s)$$

Recall that Rosenthal's potential function is defined as:

$$\Phi(s) = \sum_{r\in R} \sum_{j=1}^{n_r(s)} c_r(j),$$

where $n_r(s) = n_s$ is the number of players that use resource r under strategy profile s.

B. Consider a congestion game for which we have the following guarantee: The cost functions c_r are such that no resource is every used by more than λ players. Use the Potential Method (and Rosenthal's potential function) to show that the Price of Stability of any such game is at most λ.

Solution 3.

A. Recall that the social cost and Rosenthal's potential are defined, respectively, by

$$C(\mathbf{s}) = \sum_{r\in R} n_r c_r(n_r) \qquad \varPhi(\mathbf{s}) = \sum_{r\in R} \sum_{j=1}^{n_r} c_r(j),$$

where for simplicity we are denoting $n_r = n_r(\mathbf{s})$ to be the set of players using resource r at strategy $\mathbf{s}$.

For the right-hand side inequality first, since the cost functions of the resources are nondecreasing, we have

$$\varPhi(\mathbf{s}) = \sum_{r\in R} \sum_{i=1}^{n_r} c_r(i) \leq \sum_{r\in R} \sum_{i=1}^{n_r} c_r(n_r) = \sum_{e\in E} n_r c_r(n_r) = C(\mathbf{s}).$$

For the remaining inequality, observe that for any natural number n and for $k = 0, 1$, it holds that

$$\frac{1}{k+1} n^k \leq 1^k + 2^k + \cdots + n^k = \sum_{j=1}^{n} j^k. \tag{1}$$

Since each cost function can be expressed as a linear function $c_r(x) = a_r \cdot x + b_r$ we can upper bound the social cost by:

$$
\begin{aligned}
C(\mathbf{s}) &= \sum_{r\in R} n_r c_r(n_r) \\
&= \sum_{r\in R} n_r(a_r n_r + b_r) \\
&= \sum_{r\in R} (a_r \cdot n_r^2 + b_r \cdot n_r) \\
&\leq \sum_{r\in R} \left(2a_r \sum_{j=1}^{n_r} j^2 + b_r \sum_{j=1}^{n_r} j \right), && \text{due to (1),} \\
&\leq 2 \sum_{r\in R} \sum_{k=0}^{1} a_{r,k} \sum_{j=1}^{n_r} j^k && \text{letting } a_{r,1} = a_r \text{ and } a_{r,0} = b_r \\
&= 2 \sum_{r\in R} \sum_{j=1}^{n_r} \sum_{k=0}^{1} a_{r,k} \cdot j^k \\
&= 2 \sum_{r\in R} \sum_{j=1}^{n_r} c_r(j) \\
&= 2\Phi(\mathbf{s}).
\end{aligned}
$$

B. Let s be a strategy profile and let $SC(s)$ be the social cost of that strategy profile. It holds that

$$
\begin{aligned}
SC(s) \quad &= \quad \sum_{i=1}^{n} \text{cost}_i(s) = \sum_{i=1}^{n} \sum_{r\in s_i} c_r(n_s) = \sum_{r\in R} \sum_{i:r\in s_i} c_r(n_s) = \sum_{r\in R} n_s \cdot c_r(n_s) \\
&\leq \quad \lambda \sum_{r\in R} c_r(n_s) \leq \lambda \sum_{r\in R} \sum_{i=1}^{n} c_r(i) = \lambda \Phi(s).
\end{aligned}
$$

where the first inequality follows from the fact that at most λ players are using resource r in any profile. We also have that

$$
\Phi(s) = \sum_{r\in R} \sum_{i=1}^{n_s} c_r(i) \leq \sum_{r\in R} \sum_{i=1}^{n_s} c_r(n_s) = \sum_{r\in R}, n_s c_r(n_s) = SC(s)
$$

Now we start from a profile s^* with minimum social cost. If s^* is a pure Nash equilibrium, then the Price of Stability is 1 and we are done. Otherwise, we let the players deviate and best-respond until the reach a pure Nash equilibrium s. Since in each step some player decreases her cost and Φ is a potential function, we know that $\Phi(s) \leq \Phi(s^*)$ and we have

$$
SC(s) \leq \lambda\Phi(s) \leq \lambda\Phi(s^*) \leq \lambda SC(s^*),
$$

and the Price of Stability bound follows.

Exercise 4. Consider the class of *singleton congestion games*, i.e., congestion games in which the strategies of the players consist of single resources. Show that in singleton congestion games, a pure Nash equilibrium can be computed in polynomial time.

Hint: Starting from a singleton congestion game G, construct an "equivalent" game $\tilde{G}$ with $\tilde{c}_{\max} = poly(n, m)$, and argue that the best response dynamics algorithm converges in $\tilde{G}$ in polynomial time.

Solution 4. Fix a singleton congestion game $\mathcal{G} = (N, E, \{S_i\}, \{c_e\})$. Our goal is to construct an "equivalent" game $\tilde{\mathcal{G}} = (N, E, \{S_i\}, \{\tilde{c}_e\})$ with $\tilde{c}_{\max} = \mathsf{poly}(n, m)$. Then, we can deploy the best-response dynamics convergence theorem to get convergence in polynomially many steps, concluding our proof.

Let $K = \{c_e(j) \mid e \in E, j \in [n]\}$ be the set of all possible resource-cost values of $\mathcal{G}$. Notice that $|K| \leq nm$. We define modified cost functions $\tilde{c}_e$, for all resources $e \in E$, via

$$\tilde{c}_e(j) = k \quad \Longleftrightarrow \quad |\{\kappa \in K \mid \kappa < c_e(j)\}| = k - 1. \tag{2}$$

In other words, $\tilde{c}_e(j)$ gives the position of the value $c_e(j)$ within K (when taken ordered from smallest to largest). By this, we can immediately deduce that indeed $\tilde{c}_e(j) \leq nm = \mathsf{poly}(n, m)$ for all resources e.

The equivalence of $\mathcal{G}$ and $\tilde{\mathcal{G}}$ can now be taken to mean that the two games preserve the better-response incentives of the players. Formally, we need to show that for all outcomes $\vec{s}$, every player i and any deviation s'_i:

$$C_i(s'_i, \mathbf{s}_{-i}) < C_i(\vec{s}) \quad \Longleftrightarrow \quad \tilde{C}_i(s'_i, \mathbf{s}_{-i}) < \tilde{C}_i(\vec{s}).$$

Making use of the fact that our games are singleton, and denoting with e, e' the (single) resource in strategies s_i and s'_i, respectively, this condition is equivalently written as

$$c_{e'}(n_{e'} + 1) < c_e(n_e) \quad \Longleftrightarrow \quad \tilde{c}_{e'}(n_{e'} + 1) < \tilde{c}_e(n_e), \tag{3}$$

where for simplicity we are using the shortcut notation $n_e = n_e(\vec{s})$, $n_{e'} = n_{e'}(\vec{s})$. Now it is not hard to establish the validity of (3), directly by observing that the definition of the modified costs $\tilde{c}_e$ in (2) preserves the relative order of the cost values.

Exercise 5 (Cut Games)**.** Consider the following class of games, called *cut games*. We have an undirected graph $G = (V, E)$ in which each player i controls a vertex $v_i \in V$. Each edge $e \in E$ is associated with a *nonnegative* weight w_e. A *cut* is a partition of the set of vertices into two disjoint sets LEFT and RIGHT. Each player selects the set in which his contorolled vertex will be, i.e. the strategy space of each player i is $s_i = \{\text{LEFT, RIGHT}\}$ and let $s = (s_1, \ldots, s_n)$ be the corresponding strategy profile.

Let $\text{CUT}(s)$ be the set of edges that have one endpoint in each set, and let N_i be the set of neighbours of v_i in the graph G. The utility of player i is defined as

$$u_i(s) = \sum_{e \in \text{CUT}(s) \cap N_i} w_e.$$

A. Design a cut game in which the Price of Anarchy is 2.

Hint: Use a graph with four vertices and unit weights.

B. Prove that the Price of Anarchy of cut games is at most 2.

Hint: First argue that in a pure Nash equilibrium, the total weight of the neighbouring vertices of a vertex v_i in the cut is at least half the total weight of all the neighbouring vertices of the vertex.

Solution 5.

A. Consider a graph with 4 vertices a, b, c, d and edges $(a, b), (c, d), (a, c)$ and (b, d) with unit weights, i.e. for each edge e, it holds that $w_e = 1$. Consider the profile in which vertices a and b are on $LEFT$ and vertices c and d are on $RIGHT$. This is a pure Nash equilibrium. Consider some player, for example player i that is associated with vertex a and observe that her payoff is 1, as only edge $(a, c) \in CUT(a) \cap N_i$.Assume that i decides to claim strategy $RIGHT$ instead of $LEFT$; her payoff then would still be 1 as now only edge $(a, b) \in CUT(a) \cap N_i$ and the player has no incentive to deviate. The arguments for all other cases are symmetric and similar. The total payoff of this pure Nash equilibrium is 4 (each player has a payoff of 1). On the other hand, the profile in which a and d are on $LEFT$ and b and c are on $RIGHT$ has a total payoff of 8, since all players have a payoff of 2. The Price of Anarchy is 2.

B. Let S_{eq} be a pure Nash equilibrium in which some players (possibly none) choose $LEFT$ and some players choose $RIGHT$. Since we are at a pure Nash equilibrium, a player does not wish to deviate, which means that the total weight of the edges that are adjacent to her associated vertex u that are in the cut must be at least half of the total weight of the edges adjacent to u (as otherwise the player would have an incentive to deviate to the other part of the partiion). Formally, this means that

$$u_i(S_{eq}) = \sum_{e \in CUT(S_{eq}) \cap N_1} w_e \geq \frac{1}{2} \sum_{e \in N_i} w_e$$

By summing up over all players we get that:

$$\sum_{i=1}^{n} u_i(S_{eq}) \geq \frac{1}{2} \sum_{i=1}^{n} \sum_{e \in N_i} w_e = \sum_{e \in E} w_e,$$

where the last equality holds because when suming up over all the players, we count each edge (u, v) *twice*, once for the player that is associated with vertex u and once for the player that is associated with vertex v. The optimal total payoff can not be higher than twice the total weight of the edges and therefore the Price of Anarchy guarantee of 2 follows.