ETH Zürich
Andrei Ivanov,
Lorenzo Laneve,
Varun Maram,
Paolo Penna

November 19, 2021

Deadline: Beginning of next lecture

Algorithmic Game Theory

Autumn 2021

Exercise Set 9

Your solutions to this exercise sheet will be **graded**. Together with the other three graded exercise sheets, it will account for 30% of your final grade for the course.

You are expected to solve the exercises carefully and then write a nice complete exposition of your solution (preferably using **LaTeX** or similar computer editors – the appearance of your solution will also be part of the grade). You are welcome to discuss the tasks with your fellow students, but we expect each of you to hand in **your own** individual write-up. Your write-up should list all collaborators.

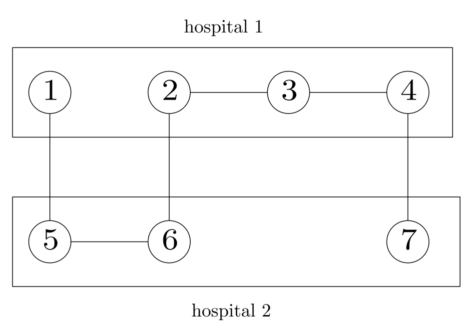
Please submit your solutions via moodle before the beginning of next lecture (**November 26, 10:00 am**). If you cannot use moodle, please send solutions by **email** to agt-course@lists.inf.ethz.ch.

Exercise 1: (4 Points)

Let us consider **matching mechanisms** applied to the **Kidney exchange** problem involving **two hospitals** (players). Recall the setting:

1. We are given an indirected graph where an edge (u, v) represents mutual compatibility between donor-patient u and donor-patient v (they can be matched).
2. The possible solutions (alternatives) are the matchings over the graph, and the social welfare is the number of matched nodes.
3. Each player i corresponds to a subset of nodes (nodes are partitioned across the players) and the utility of a player is the number of his/her nodes that are matched.

Consider the following example:

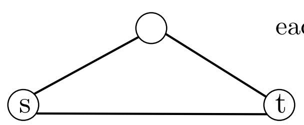


A player (hospital) can **hide** some of its nodes to the mechanism (not reporting them) and also do some **internal matching** involving nodes that are hidden. For instance, hospital 1 could hide nodes 2 and 3, match them internally; In this scenario, the mechanism gets in input the subgraph induced by the remaining nodes (all edges involving 2 and 3 are no longer present). We say that a mechanism is **truthful** if no player can increase its own utility by hiding some nodes (and matching them internally) to the mechanism (the utility is given by the nodes matched by the mechanism plus those matched internally).

Your task: Show that no deterministic truthful mechanism can have an approximation guarantee better than 2 (an α-approximation mechanism returns a matching whose social welfare is at least OPT/α, where OPT is the optimum social welfare; the social welfare of a matching equals the number of nodes that are matched).

Exercise 2: (2+1 Points)

Consider the following scenario:



and we want to sent T **units of traffic** from s to t. The traffic can be divided arbitrarily between the two paths. Moreover:

1. Each player i has some **private** cost t_i and we know

$$t_i \in \{L, H\}$$

   where t_i is the cost per unit of traffic, and $H > L$ (intuitively, these two values represent $L = Low$ and $H = High$ cost per unit of traffic).

2. We give a **fixed compensation** per unit of traffic:

$$F \cdot w_i$$

   is the payment to player i when he/she gets w_i units of traffic.

Your task (2+1 points):

1. Model this game as a **single-peaked preferences** when $L < F < H$. (Explain what are the outcomes, and why the preferences of each of the three players are single-peaked.)

2. Which outcomes are selected by the **median voter**? (Explain how the traffic is allocated in our original problem if we run the median voter mechanism for the corresponding single-peaked domain. The median voter is also described in the AGT book edited by Nisan, Roughgarden, Tardos, Vazirani.)

Exercise 3: (2 Points)

Consider the following special case of (bipartite) stable-matching problem. Players are partitioned into n **interns** and n **hospitals**:

- Hospitals have a **common (same) order** of interns,

$$i_1 \succ i_2 \succ \cdots \succ i_n \tag{1}$$

- Interns rank hospitals differently (e.g., based on salary, location, etc.). So each intern has his/her own order $\prec_i$ over the hospitals.

Your task: Consider running the **sequential-dictator** mechanism which asks the interns (in some order) the top-preference among the still available hospitals (hospital must passively accept the assigned intern).

- Suppose we know the hospital preferences (1). Show that in this case you can choose the order in which we ask interns in the serial-dictator mechanism so that it returns a stable matching.

Exercise 4: (2 Points)
Consider the **generic** mechanism design problems with private **costs** (see Lecture 5, Section 2.2).

Show that, if a mechanism (A, P) is truthful, then the algorithm A must satisfy the following **monotonicity** condition:

Algorithm A is **monotone** if, for all i, for any two inputs that differ only in i's report, that is,

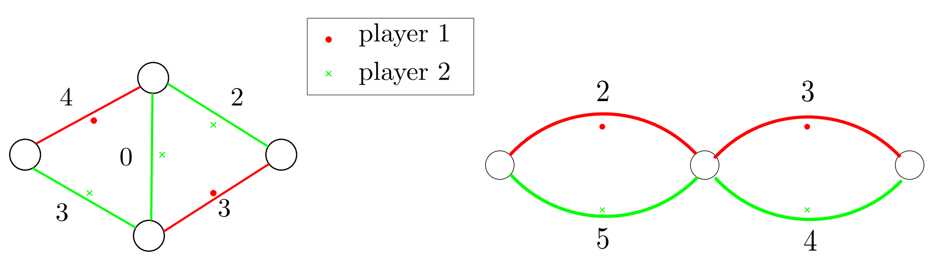
$$c = (c''_1, \ldots, c''_{i-1}, c_i, c''_{i+1} \ldots, c''_n) \qquad c' = (c''_1, \ldots, c''_{i-1}, c'_i, c''_{i+1} \ldots, c''_n)\ , \tag{2}$$

the algorithm satisfies the following:

$$c_i(A(c)) + c'_i(A(c')) \leq c_i(A(c')) + c'_i(A(c))\ . \tag{3}$$

Note: Recall that all these c''_j, c'_i and c''_i are functions.

Exercise 5: (4 Points)
Consider the following multi-parameter mechanism design problem. We are given a graph $G(V, E)$, and each edge has a private cost which is known to the player owning this edge. We consider the case in which a player may own **several edges**. Here are two examples with two players:



The feasible solutions consist of the **trees** spanning all nodes of the graph, and the cost for a player is given by the sum of the costs of his/her edges in the chosen tree. For example, the minimum spanning tree in the left example above consists of only the green edges. This costs 0 to player 1 (red) and 5 to player 2 (green).

Here however we do not want to find the minimum spanning tree. We instead consider a different optimization goal, namely, we want to minimize the **maximum cost** among the players. In the left example above, this would mean that we choose the tree consisting of the two edges of cost 3 and the edge of cost 0; this results in a cost of 3 for both players (the maximum cost is 3 instead of 5).
Any solution a costs to player i

$$t_i(a) = \sum_{e \in E_i \cap a} t_i^e$$

where E_i is the subset of edges owned by i, a is a spanning tree (set of edges), and t_i^e denotes the true cost of edge $e \in E_i$. We are interesting in minimizing the **maximum cost** among the players:

$$maxcost(a, t) := \max_i t_i(a)$$

Your task:

- Prove that no monotone algorithm (in the sense of the previous exercise) can minimize the maximum cost.

Note: In this problem, each player i reports some cost c_i^e for each of his/her edges $e \in E_i$. Based on the reported costs c the mechanism selects a solution $A(c)$ and payments $P_i(c)$.