Algorithmic Game Theory

Autumn 2021
Exercise Set 9

Elisabetta Fedele

Contributors
Elisabetta Fedele, Massimo Albarello, Julian Gleissner, Davide Maioli, Ravi Srinivasan

Exercise 1

Let us consider matching mechanism applied to Kidney exchange problem involving two hospitals, where we consider the hospitals as players. Given that we define a mechanism truthful if no player can increase his/her utility by hiding some nodes, our goal is to prove that there is no deterministic truthful mechanism with an approximation guarantee better than 2.

Let us first recall the definition of α-approximation for a mechanism and analyze what does it mean to have an approximation guarantee better than 2.
We know that if a mechanism has an approximation guarantee better than 2, it must hold that:

$$SW(M) \geq \frac{OPT}{\alpha} \text{ with } \alpha < 2$$

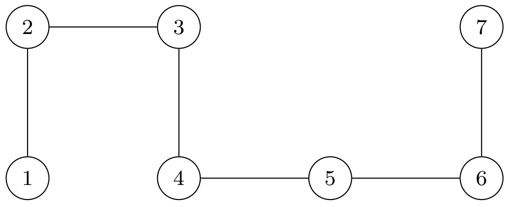
Where OPT is the optimum social welfare. Hence, rearranging the terms, we obtain:

$$\frac{OPT}{SW(M)} \leq \alpha < 2$$

As a result, it must always hold that

$$\frac{OPT}{SW(M)} < 2$$

If such a mechanism exists it should work in every possible configuration. It worth noticing that the exercise consider the mechanisms which are deterministic, so that fixed an input, they returns always the same output. Thus, we assume that all the players know *a priori* the choice that will be made by the algorithm before reporting their costs.
Consider now the following example:



Where the nodes 2, 3 and 7 belongs to the hospital 1 and the other nodes to the hospital 2.
Here, the maximum matching has $OPT = 6$, thus my mechanism should return a matching whose cardinality is higher than $\frac{OPT}{2} = 3$, and thus at least of 4.
These are 16 possible matchings that can be returned by an α-approximation with $\alpha < 2$. We can divide these 16 cases in 4 groups:

1. 1-2, 3-4 and 5-6 and their subsets (1-2 and 3-4, 1-2 and 5-6, 3-4 and 5-6)
2. 1-2, 3-4 and 6-7 and their subsets (1-2 and 3-4, 1-2 and 6-7, 3-4 and 6-7)
3. 1-2, 4-5 and 6-7 and their subsets (1-2 and 4-5, 1-2 and 6-7, 4-5 and 6-7)
4. 2-3, 4-5 and 6-7 and their subsets (2-3 and 4-5, 2-3 and 6-7, 4-5 and 6-7)

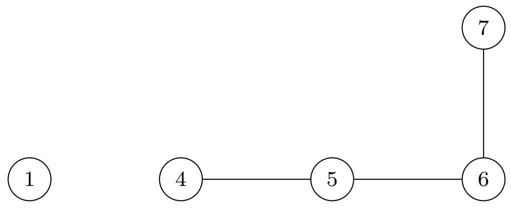
Now we will analyze these solutions one by one and we will show that there is not an α-approximation mechanism (having $\alpha < 2$) that is able to return **deterministically** a truthful matching for this example and thus that in general there is no deterministic truthful mechanism with an approximation guarantee better than 2.

Case 1: 1-2, 3-4 and 5-6 and their subsets

The utilities of the two players are the following:

- utility(player 1)= 2
- utility(player 2)= 4

Let us consider what would happen if player 1 hides the nodes 2 and 3 and matches them internally.



The mechanism would now have to find an allocation on the graph above. There we have $OPT = 4$, so the truthful mechanism that satisfy the approximation required should return a social welfare $SW > 2$ and, thus it should match at least 4 nodes (an odd number of nodes cannot be matched). In the only feasible solution (4-5 and 6-7) the utility of player 1 deriving from the nodes matched by our mechanism would be exactly 1.
Thus, we would have:

$$\text{utility(player 1)} = 2 + 1 = 3$$

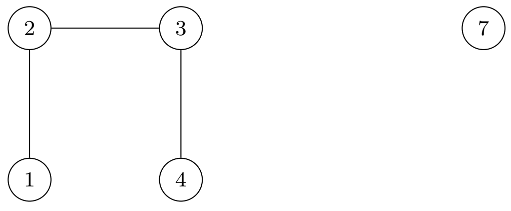
Thus, the player 1 would increase his/her utility by hiding the node 2 and the node 3. We can thus conclude that this matching (1-2, 3-4 and 5-6) cannot be returned by any truthful algorithm.
Furthermore, we can observe that considering only one subset of this solution, the utility of player 1 can only decrease and, thus, he would always prefer to hide the node 2 and 3.
Hence, all the four solutions we are considering in this case cannot be returned by any deterministic truthful algorithm.

Case 2: 1-2, 3-4 and 6-7 and their subsets

The utilities of the two players are the following:

- utility(player 1)= 3
- utility(player 2)= 3

Let us consider what would happen if player 2 hides the nodes 5 and 6 and matches them internally.



The mechanism would now have to find an allocation on the graph above. There we have $OPT = 4$, so the truthful mechanism that satisfy the approximation required should return a social welfare $SW > 2$ and, thus it should match at least 4 nodes (an odd number of nodes cannot be matched). In the only feasible solutions (1-2 and 3-4) we would have that the utility of player 2 deriving from the nodes matched by our mechanism would be exactly 2.
Thus, we would have

$$\text{utility(player 2)} = 2 + 2 = 4$$

Thus, the player 2 would increase his/her utility by hiding the node 5 and the node 6. We can thus conclude that this matching (1-2, 3-4 and 6-7) cannot be returned by any deterministic truthful algorithm.
As we have already discussed in the case 1, the same argument can be repeated for the three subsets.
Hence, even in this case, all the four solutions we are considering cannot be returned by any deterministic truthful algorithm.

Case 3: 1-2, 4-5 and 6-7 and their subsets

The utilities of the two players are the following:

- utility(player 1)= 2
- utility(player 2)= 4

As we have already seen in the first case, the player 1, by hiding both the nodes 2 and 3, can obtain a guarantee utility of 3. Thus, the player 1 would increase his/her utility by hiding the node 2 and the node 3. We can thus conclude that this matching (1-2, 3-4 and 6-7) cannot be returned by any deterministic truthful algorithm.
As we have already said in the previous cases, the same argument can be repeated for the three subsets. Hence, all these four solutions cannot be returned by any deterministic truthful algorithm.

Case 4: 2-3, 4-5 and 6-7 and their subsets

The utilities of the two players are the following:

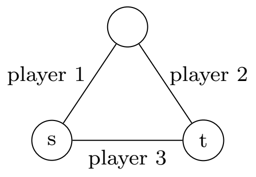
- utility(player 1)= 3
- utility(player 2)= 3

As we have already seen in the second case, the player 2, by hiding both the nodes 5 and 6, can obtain a guarantee utility of 4.
Thus, the player 2 would increase his/her utility by hiding the node 5 and the node 6. We can thus conclude that this matching (2-3, 4-5 and 6-7) cannot be returned by any deterministic truthful algorithm.
As we have already said in the previous cases, the same argument can be repeated for the three subsets. Hence, all these four solutions cannot be returned by any deterministic truthful algorithm.

Since we have proven that there is no deterministic truthful mechanism can have an approximation guarantee better than 2 in this particular example, we can conclude that in general such an algorithm does not exist.

Exercise 2

Consider the following scenario in which the two edges of the top path are respectively player 1 and player 2 and the bottom edge is the player 3.



Part 1

Our first task is to model this as a **single-peaked preferences** and to explain what are the outcomes given by the players, explaining for every situation why their preferences are single peaked.
In order to model this game as a sinle-peaked preferences we decide that every player has to express his/her preference giving a number included in the interval $[0, T]$, which represents the units of traffic sent from **s** to **t** using the upper path. Furthermore, we assume that the remaining units will be sent using the lowest one.
Since the cost of the other players are private, every player has a preference which depends only on his/her own cost. Hence, we should consider the following feasible situations.

cost 1	cost 2	cost 3	preference player 1	preference player 2	preference player 3
L	L	L	T	T	0
L	L	H	T	T	T
L	H	L	T	0	0
L	H	H	T	0	T
H	L	L	0	T	0
H	L	H	0	T	T
H	H	L	0	0	0
H	H	H	0	0	T

All the player always has an utility of $u = (F - t_i) \cdot w_i$, where w_i are the number of unit of traffic he/she has got and t_i is his/her private cost. In particular, we can notice that

1. If the player i has private cost H, his utility $u = (F - H) \cdot w_i$ will be always negative if $w_i \neq 0$ as we know that $F < H$. Thus, the player i will always express the preference that leads 0 units toward the path he is contained in (0 if $i = 1$ or $i = 2$, T if $i = 3$) and his preference will be single-peaked as preferences are *decreasing* as he/she moves away from this peak.
2. If the player i has private cost L, his utility $u = (F - L) \cdot w_i$ will be always non.negative for the values of w_i in the range $[0, T]$. In particular the utility would be maximize with $w_i = T$. Thus, the player i will always express the preference that leads all the T units toward the path he is contained in (T if $i = 1$ or $i = 2$, 0 if $i = 3$). Also in this situatuion, his preference will be single-peaked as preferences are *decreasing* as he/she moves away from this peak.

Part 2

Our second task is to describe the outcomes that are selected by the median voter, depending on the players' vote. If we run the median voter we would have the following outcomes:

preference player 1	preference player 2	preference player 3	outcome
T	T	0	T
T	T	T	T
T	0	0	0
T	0	T	T
0	T	0	0
0	T	T	T
0	0	0	0
0	0	T	0

We remark that if:

- outcome = T: T units are sent from s to t using the upper path
- outcome = 0: T units are sent from s to t using the lower path

Exercise 3

Let us consider the given configuration of a bipartite stable matching problem.
Our task is to define an order in which the **sequential dictator** mechanism should be run in order to obtain a stable matching.

We will start by defining this order and then we will prove that the matching obtained using that order is a stable one.
Let us consider the following (common) hospital order of interns:

$$i_1 \succ i_2 \succ \ ... \ \succ i_n$$

We propose to ask the interns the top preference starting from i_1 and following the order given above until the last one, that is i_n.

We have now to prove that the matching obtained is a **stable matching**, thus that it does not contain any blocking pair.
For the sake of contradiction consider a intern i not matched to a hospital h. There are two possible reasons for i not being matched to h. Either i is matched to another hospital h' which he preferred over h, or h was not among the still available hospitals when the intern i had to choose.
In the first case, as we have already said, i must be matched to an hospital higher than h on i's list. But then the pair (i, h) is not blocking.
In the second case, h is matched to an intern who was able to choose the hospital before i. But as we asked the interns to peak their choice following the hospitals' list of preferences over interns, h is matched to an intern which is higher than i on the common list of preferences. Hence, also in this case (i, h) is not a blocking pair.

We have thus proven that, with the order we have chosen, the serial-dictator mechanism returns a stable matching.

Exercise 4

A mechanism is a pair (A, P) which get on input the cost $c = (c_1, ..., c_n)$ reported by the players and outputs:

- A solution (an allocation) $A(c) \in \mathcal{A}$
- A payment $P_i(c)$ for each player i

The corresponding utility for each agent is

$$u_i(c|t_i) := P_i(c) - t_i(A(c))$$

Where t_i is the private (true) cost function (depending on the allocation) of the player i, which will be never be know by the mechanism for construction.
As a consequence, our algorithm does not know *a priori* which will be the the true cost t_i of the player i, so that condition should be valid for all the feasible cost function that may be the true one for that player. Thus, in order to provide a truthful mechanism the following condition must hold:

$$u_i(A(c)|c_i) \geq u_i(A(c')|c_i)$$

For all the players i and for all the couples of cost functions (true-reported) they can have, so that for every true cost function the player i has, he does not have any incentive to report a different (and false) one.
Let us first assume that the true cost of the player i is c_i. We can rewrite the inequality as follows, using the definition of utility

$$P_i(c|c_i) - c_i(A(c)) \geq P_i(c'|c_i) - c_i(A(c'))$$

But the payment that the algorithm outputs for the player i does not depend on whether his/her real cost, since it has been evaluated only on the reported cost of the player. Then, we can rewrite the previous inequality as follows:

$$P_i(A(c)) - c_i(A(c)) \geq P_i(c') - c_i(A(c'))$$
$$P_i(c') + c_i(A(c)) \leq P_i(c) + c_i(A(c'))$$

Symmetrically, we can repeat the same reasoning, this time assuming c'_i as the real cost function for the player i. In particular the following condition must hold

$$u_i(A(c')|c'_i) \geq u_i(A(c)|c'_i)$$
$$P_i(c'|c'_i) - c'_i(A(c')) \geq P_i(c|c'_i) - c'_i(A(c))$$

As we said before, the payment that the algorithm outputs for the player i does not depend on whether his/her real cost, then

$$P_i(c') - c'_i(A(c')) \geq P_i(c) - c'_i(A(c))$$
$$P_i(c) + c'_i(A(c')) \leq P_i(c') + c'_i(A(c))$$

Summing this inequality with the one obtained before (when we assumed c_i as true cost function for the player i), we obtain:

$$P_i(c) + c'_i(A(c')) + P_i(c') + c_i(A(c)) \leq P_i(c') + c'_i(A(c)) + P_i(c) + c_i(A(c'))$$

Thus, rearranging the terms we obtain

$$c_i(A(c)) + c'_i(A(c')) \leq c_i(A(c')) + c'_i(A(c))$$

This inequality is exactly what we wanted to prove.

Exercise 5

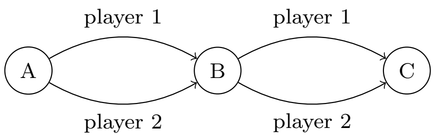
The exercise presents a multi-parameters mechanism design problem.
Our task is to prove that there is no monotone algorithm (in the sense of the exercise 4) that minimizes the maximum cost among the players.
If we are able to find a single example in which the choice of the algorithm which minimize the maximum is unique and the condition of monotony is not satisfied, then we can conclude that, in general, no monotone algorithm that can minimize the maximum cost among the players exists (otherwise it should have worked also in the case we have chosen).
First, let us recall the monotony condition given in the previous exercise.
An algorithm A is monotone if for all the players involved in the game, for any two inputs which differ only in the report of a single player (let us call i this player), that is,

$$c = (c''_1, ..., c''_{i-1}, c_i, c''_{i+1}, ..., c''_n)$$
$$c' = (c''_1, ..., c''_{i-1}, c'_i, c''_{i+1}, ..., c''_n)$$

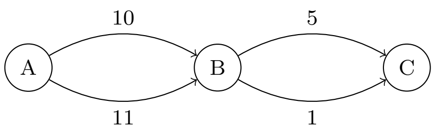
the algorithm satisfies the following:

$$c_i(A(c)) + c'_i(A(c')) \leq c_i(A(c')) + c'_i(A(c))$$

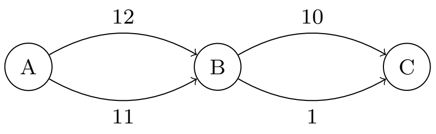
Let us consider the following example



In the graph above, the top edges belong to player 1 and the bottom ones to player 2.
Let us consider a first cost report given by the first player. Let us call this report c.



All the algorithms which minimize the maximum cost would choose the tree consisting of the edge AB of player 1 and the edge BC of player 2. In all the other feasible solutions, the maximum cost (based on the costs reported) of each player would be higher.
Let us consider another allocation, in which the second player reports the same costs as before and only the first one changes his reports. Let us call this allocation c'.



All the algorithms which minimize the maximum cost would choose the tree consisting of the edge AB of player 2 and the edge BC of player 1. As before, in all the other feasible solutions, the maximum cost (based on the costs reported) of each player would be higher.
Let us define the terms of the inequality of the previous exercise given the two inputs c and c'.

$$c_1(A(c)) = 10$$
$$c'_1(A(c')) = 10$$
$$c_1(A(c')) = 5$$
$$c'_1(A(c)) = 12$$

Rearranging all the terms we obtain that:

$$20 = c_1(A(c)) + c'_1(A(c')) > c_1(A(c')) + c'_1(A(c)) = 17$$

But this inequality contradicts

$$c_i(A(c)) + c'_i(A(c')) \leq c_i(A(c')) + c'_i(A(c))$$

We recall that, since the solution in these cases is unique, every algorithm which minimizes the maximum in these particular cases have the same value of $c_i(A(c))$, $c'_i(A(c'))$, $c_i(A(c'))$, $c'_i(A(c))$.
We have found a situation in which there is no algorithm which minimizes the maximum and satisfies the inequality $c_i(A(c)) + c'_i(A(c')) \leq c_i(A(c')) + c'_i(A(c))$. Thus, it is not true to say that the algorithm which minimizes the maximum satisfies the inequality for every pair of c and c' and we can conclude that no monotone algorithm (in the sense of the previous exercise) can minimize the maximum cost.