1 Kidney Transplantation

In the US alone 640000 patients are in need of kidney transplants. Only 17000 patients receive a kidney each year. Therefore it is important to match the right kidney to the right patient. Not all kidneys are equal: Any implanted kidney will be rejected at some point. However, depending on factors like age and blood type, the expected time until a certain kidney is differs per patient. Moreover, depending on blood type, some kidneys can not be transplanted to certain patients, as they will be rejected immediately. There are 4 different blood types: AB,A,B or O.

– Donors of type O can donate to any patient.
– Donors of type A can donate to patients of type A or AB.
– Donors of type B can donate to patients of type B or AB.
– Donors of type AB can only donate to patients of type AB.

There are two ways for patients to receive a kidney:

– Waiting list: In case of deceased kidney donors, the highest compatible patient on the list will receive the kidney.
– Living donor: Living donors want to help a specific friend/family member in need of a kidney.

In case of a living donor, we denote such a donor and the friend/patient he wants to help a *patient-donor-pair*. In the next section, we will see that a patient does not always end up with the kidney of his own friend. When we talk about the *donor* of a patient, we simply mean the friend of that patient. The donor from whom the kidney is actually transplanted to the patient is called his *matched donor* People only need one of their two kidneys. This makes kidney transplantation a very interesting problem from a game theoretic perspective, as opposed to, say, blood transfer or heart transplantation.

2 Trading Donors

There are many patient-donor-pairs for which the patient would prefer a different kidney than the one of his donor. This could be either because another kidney yields a larger expected time until rejection, or because his donor's kidney is not compatible at all. When two patients prefer each other's donors, it is possible to trade: Patient a receives a kidney from patient b's donor, and vice versa. We have to take into account the following: Suppose patient a receives his kidney from patient b's donor first. Then patient a's donor could choose to withdraw and keep his kidney as his friend has been helped already. By law, it is not allowed to sign contracts or offer money to protect against this behaviour. The solution is that both operations take place simultaneously, therefore a trade requires 2 operating rooms. Trades involving more than two patient-donor pairs require even more operating rooms. It is also possible to trade with the waiting list. For example: Patient a receives priority on the waiting list, patient b receives a kidney from patient a's donor, patient c receives a kidney from patient b's donor, and patient c's donor donates his kidney to the highest compatible patient on the waiting list. In practice, the value of receiving priority on the waiting list depends in practice on the blood type of the patient, and the number of other patients receiving priority. However, to simplify the model, we use the following assumption: Receiving priority on the waiting list yield an expected time until rejection of 10 years, independent of blood type or number of patients who receive priority.

3 Goal

Our goal is to construct a matching of donors to patients which maximizes the total expected time until rejection (summed) over all patients. Mathematically, this problem is very similar to matching markets. However, a very important difference is the incentives of donors. While a product does not care which customer buys it, donors aim to obtain the best result for their patients. Therefore, we have to construct our matching, taking into account the following:

- Withdrawing: If we propose a certain matching, any donor will withdraw if the kidney his patient receives is less compatible than his own kidney. Moreover, this holds for any set of patient-resource pairs: If any subset of player-resource pairs can withdraw and trade among themselves, obtaining a better kidney for all patients involved, they will do so.
- Misreporting: Suppose given a matching, a certain patient would receive his third preferred kidney on his list. However, by falsely reporting that he prefers his second preferred kidney to his first preferred kidney, he obtains that one, then he will do so.

4 Procedure

Similar to matching markets, we create a preferred donor-graph. However, the process is not completely identical: We create a directed graph $G = (N, A)$ where initially, each node corresponds to one of the patient-donor pairs. For ease of notation, we denote by patient p and donor p the patient and donor corresponding to node a respectively. During the process, for some pair p, the donor is matched, but the patient is not matched yet. In that case the node corresponding to p represents only the patient. There is also one node w which corresponds to the waiting list. We keep track of a list of matched pairs of patient-donor pairs M, a list of patients who receive priority on the waiting list L, and finally, we maintain a list K of donors whose patient have been assigned a kidney.
At multiple stages in our procedure, we create arcs. Whenever we do so, arcs are created as follows:
Creating arcs
For each patient-donor pair $a \in N \setminus K$ for which the patient is not assigned yet, and each patient-donor pair $b \in N$ for which the kidney has not been assigned

yet, there is an arc from node a to node b patient a prefers the kidney of the donor of node b over all other kidneys. If patient a prefers priority on the waiting list over all donors, then there is an arc from a to w.

A *cycle* is an ordered set of nodes $(a, b, \dots, y, z)$ such that there is an arc from a to b, from b to $\dots$, from y to z and finally from z to a. We denote such a cycle by $(a \to b \to \cdots \to z \to a)$.
A *chain* is an ordered set of nodes $(a, b, \dots, y, z)$ such that there is an arc from a to b, from b to $\dots$, and finally from y to z. We denote such a chain by $(a \to b \to \cdots \to z)$. A chain $(a \to b \to \cdots \to z)$ is *maximal* if there is no arc from any node (either in or outside the chain) to a.

There is no arc starting from the waiting list, or from unassigned kidneys. We now use algorithm 1 to match pairs:

Create arcs **while there exists a node corresponding to an unmatched patient do**
- **if** *There exists a cycle* $C : (a \to b \to \cdots \to z \to a)$ **then**
  - match patient a to donor b, patient b to $\dots$, and patient z to donor a in M;
  - Remove all nodes in C from G
- **else**
  - select a chain $C' : \{a \to b \to \cdots \to z \to w'\}$
  - **if** w' *is the waiting list* w **then**
    - match patient a to donor b, patient b to $\dots$, and give patient z priority on the waiting list L;
    - Remove all nodes in C' except for a and z from G;
    - Add a to K;
  - **else**
    - w' is an unassigned donor d. Match patient a to donor b, patient b to $\dots$, and patient z to donor d;
    - Remove all nodes in C' except for a and z from G;
    - Add a to K;
    - Remove d from K;
  - **end**
- **end**
- ; remove all arcs and create new arcs

end

Note that we have not specified which cycle or path to choose, if there are several possible choices. In case of cycles, it does not actually matter which cycle we choose, as each cycle eventually gets chosen. In case of chains, choosing the right chain is crucial, since choosing a particular chain might destroy another chain. Which chains we pick not only influences the total expected time until rejection, but it could also incentivise misreporting. We will consider the following tie-breaking rules:

– Longest chain first
– Shortest maximal chain first
– Chains starting with a blood type O donor first

5 example

Consider the following example with 6 patient-donor pairs, and preferences as in table 1. The arcs created in each step of the procedure are shown in figure 5. There is one cycle ($c \to d \to e \to c$), so we match c to d, d to e and e to c. and remove c, d and e. Now the preferred donor of f is matched, so we create an arc to his second preferred donor b. Now there are two chains ($a \to b \to w$) and ($f \to b \to w$). Depending on our tie-breaking rule, we choose ($f \to b \to w$). We match f to b and b receives priority on the waiting list. We remove b, but we don't remove f, since only it patient is matched. Now only patients a is left. Since a is incompatible with donor f and he prefers his own donor to the waiting list, we create a self-loop for a. In the final step, we match a to its own donor, and we are done. In the end only donor f is unmatched, so he donates his kidney to the waiting list.

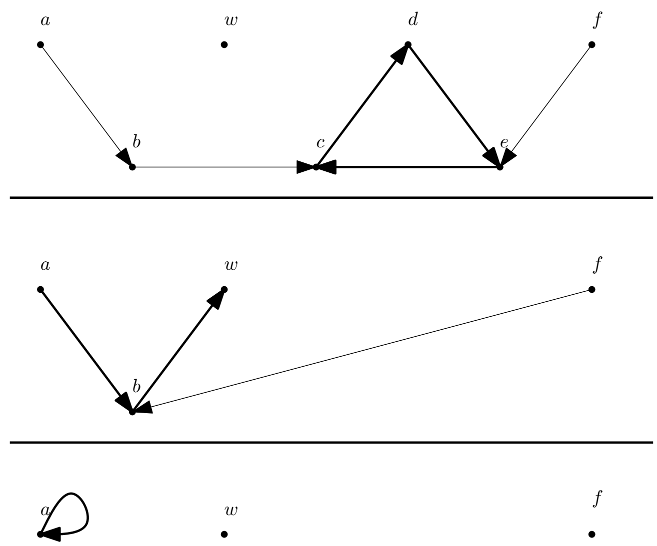
Table 1. Entries correspond to the expected time until rejection for each combination of patient (row) and donor (column)

	a	b	c	d	e	f
a	12	15	-	-	-	-
b	-	-	15	-	-	-
c	-	-	-	15	12	-
d	-	-	-	-	11	-
e	-	-	12	-	-	11
f	13	14	-	-	16	

6 questions

For the practitum, only do starred exercises. Other exercises are part of the project.

1*) What could be a reason to use the third tie-breaking rule: prioritize chains starting with a blood type O donor?



2) As mentioned in section 2, operations in cycles have to take place simultaneously. Suppose due to a small number of operating rooms, we would like to do the operations from chains sequentially. What order of operations would you recommend? start of chain to w, or w to start of chain? Give advantages and disadvantages for both orders.

3a*) Suppose we so not care about misreporting and the values in the example from section5 are the true values. What matching would maximize the total expected utility?

3b*) Which set of pairs has incentive to withdraw from the matching in 3a) and trade among themselves?.

4) In step 2 of the procedure in section 5, choosing $(a \to b \to w)$ would have yielded a higher total expected time until rejection $(15 + 10$ instead of $14 + 10)$. What could be a disadvantage of choosing the chain with the highest total expected time until rejection?

5*) Suppose some patient would prefer to wait for a better kidney, if he would be assigned a kidney with expected time until rejection of 12 or less. How would you model such a patient?

6) Construct an example where the 'longest chain first'-tie breaking rule gives an incentive to misreport. In other words, show that there is a player in that example who would benefit from lying about his preferences in order to obtain a higher expected time until rejection.

7 project

Write a program that, given input a matrix of patient-donor pairs such as table 1, and a list of blood types, computes the total expected time until rejection for the matching procedure using the following tie-breaking rules:

- Longest chain first
- Shortest maximal chain first
- Chains starting with a blood type O donor first

Also, let the program compute the optimal expected time until rejection, without taking into account any incentives, i.e. if no pair would withdraw or misreport. To find the matching which yields the optimal expected time until rejection, you are allowed to use an existing module, instead of implementing it yourself. However, dont forget to take the waiting list into account. Next to a discussion of the results, also include in your report answers to the following questions:

- What could be a disadvantage of using a tie-breaking rule which chooses the chain with the highest total expected time until rejection?
- Construct an example where the longest chain first-tie breaking rule gives an incentive to misreport. In other words, show that there is a player in that example who would benefit from lying about his preferences in order to obtain a higher expected time until rejection.
- Prove that in each step of the algorithm, there exists either a chain, or a cycle.
- Prove that regardless of the tie-breaking rule used in the algorithm, no group has incentive to withdraw (and form a better matching with each other).