

Matching mechanisms for kidney transplantations

Tristan François

January 21, 2021

1 Two Simple Approaches

Question 1

Algorithm 1: Direct Donation

Input: A number n of patient and, for all patients $i \in \mathbb{N}$, their set of compatible donors $K_i \subseteq \llbracket 1, n \rrbracket$.

Output: A set of pairs donor-patient pairs c and a waiting list w .

```
1 initialization
2    $c = \emptyset$ ;
3    $w = []$ ;
4 begin
5   for  $i = 1$  to  $n$  do
6     if  $i \in K_i$  then
7        $c = c \cup \{(i, i)\}$ ;
8     else
9        $w.add(i)$ 
10  return  $c, w$ 
```

The implementation in python is pretty straightforward. The python function takes as parameter an integer n and an integer set array of size n . The use of python set allows to test the if condition in constant times. The only difference with pseudo-code is the use of a list rather than a set for the variable c . This choice was made to avoid the unnecessary heaviness of the set data structure, but does not change the complexity.

Question 2

Our matching algorithm must prioritize the patients with the highest priority. It must also ensure that all patients will leave with a donor that suited them better than their paired donor and the waiting list, otherwise some patients may not participate in the exchange program.

Our greedy algorithm therefore consists of browsing the patients in descending order of priority and matching them to their preferred donor whose paired patient does not prefer the waiting list or his donor to the donor of the priority patient.

Algorithm 2: Greedy Matching

Input: A number n of patient, a strict priority list of the patients U and, for all patients $i \in \mathbb{N}$, their set of compatible donors $K_i \subseteq [1, n]$ and their strict preference relation P_i over $K_i \cup \{k_i, w\}$. The priority list is an integer list that starts with the index of the highest priority patient and goes to the index of the lowest priority patient.

Output: A matching M .

```

1 initialization
2    $M = \emptyset$ ;
3    $is\_matched = Boolean[1..n]$ ;
4    $sorted\_K = IntegerList[1..n]$ ;
5   for  $i = 1$  to  $n$  do
6      $is\_matched[i] = False$ ;
7      $sorted\_K[i] = sort(K_i, P_i)$  ;                               //  $\mathcal{O}(n \log(n))$ 
8 begin
9    $/*$  Iterate over all patient by decreasing priority           $*/$ 
10  for not  $U.is\_empty()$  do
11     $u = U.pop()$ ;
12    if not  $is\_matched[u]$  then
13      while not  $sorted\_K[u].is\_empty()$  do
14         $v = sorted\_K[u].pop()$ ;
15        if not  $is\_matched[v]$  then
16           $/*$  If  $t_v$  prefer  $k_u$  over  $k_v$  and  $w$                    $*/$ 
17          if  $t_v P_v t_u$  and  $t_v P_v w$  then
18             $M = M \cup \{(k_u, t_u), (k_v, t_v)\}$ ;
19             $is\_matched[u] = True$ ;
20             $is\_matched[v] = True$ ;
21  return  $M$ 

```

We start by initializing an *is_matched* array to keep track of which pair has already been matched and an integer list array *sorted_K* that contains, for each patient of index i , the indexes of the preferred donors in descending order of preference. The sort function returns a list of the elements of K_i sorted in descending order according to the preference relation P_i .

2 Efficient strategy-proof exchange mechanism

Question 3

Let (k_1, t_1) be a pair of donor-patient and assume there is no cycle. Consider the chain starting from (k_1, t_1) . Since there is a finite number n of donor-patient pairs and there is no cycle, the chain cannot be infinite. Finally, since the only way for a chain to stop is that w belong to the chain, (k_1, t_1) is indeed a tail of a w -chain.

Thus either there exists a cycle, or each pair is the tail of some w -chain.

Question 4

Question 5

Let's run the Cycles and Chains Matching Algorithm with rule A on this example. The initial state of the graph, before the first round, is shown in figure 1.

The initial graph contains a cycle $C_1 = (k_3, t_3, k_2, t_2, k_{11}, t_{11})$. The first round of the algorithm therefore consists in detecting this cycle, allocating the kidneys to the patients and then eliminating the cycle.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

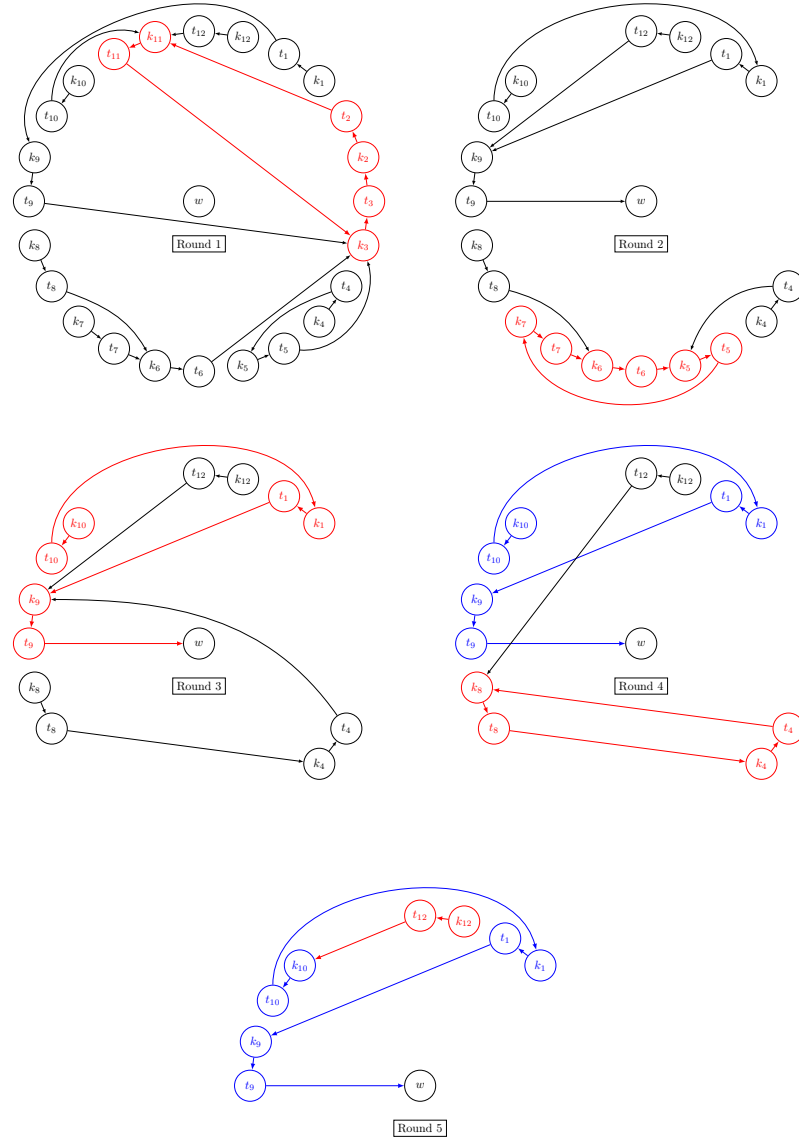


Figure 1: Test