Effects on Waiting Times when Pairs are Prioritized in Kidney Paired Donation

Author:
Else Wolfswinkel

Supervisors:
Dr. Kristiaan Glorie
Dr. René Bekker

VU UNIVERSITY AMSTERDAM

Faculty of Sciences
De Boelelaan 1081
1081 HV Amsterdam

January 2016

Preface

Writing a research paper is one of the compulsory parts of the Master's program Business Analytics at the VU University Amsterdam. The aim of this research paper is to perform a small research in the field of Business Analytics after which you describe this in a clear manner for the benefit of a manager who has some general expertise in the subject area. The challenge is to write the paper concise and 'to the point' because the manager has only limited time to read it.

In this research paper I investigated, using simulation techniques, what the effects are on waiting times of patients who are waiting for a kidney transplant, when patients (and their living but incompatible donors) with certain characteristics are given priority in a matching round. In order to be able to do this, I first had to implement an integer linear program (ILP) in an ILP solver (CPLEX). Because I had no experience at all with implementing an ILP nor with CPLEX, this was quite a challenging task. But eventually, it was also instructive and sometimes even fun to do.

I would like to thank Kristiaan Glorie for providing me with this challenging, instructive topic, for his support up close and at distance and for guiding me through the statistical stuff. Second, I would also like to thank Annemieke van Goor for involving René Bekker into this process. Without her and René this paper probably would have remained unfinished. So also, Rene, thank you for bringing this paper to a successful conclusion.

Else Wolfswinkel
Hilversum, January 2016

Management summary

To reduce the number of patients who are waiting for a kidney transplant from a deceased donor, programs have been developed in which patients can receive a kidney from a biologically compatible living donor. But what if the donor is biological incompatible with his or her intended recipient? That is why the Kidney Paired Donation (KPD) programs exists. The goal in KPD is, given a set of incompatible patient-donor pairs and donors who are willing to donate their kidney to any patient, to find a set of transplants that maximizes the total number of transplants performed. This problem is generally referred to as the Kidney Exchange Problem (KEP). A disadvantage of maximizing the total number of transplants is that it does not take the waiting times of patients into account. Previous research showed that highly sensitized patients, patients with blood type O and patients having a donor with blood type AB are difficult to match. As a consequence, these patients are hardly or never matched in KPD and finally, they might become too weak to undergo a transplant or they might even die. The purpose of this research paper is to investigate what the effects are on waiting times when patients with certain characteristics are prioritized above others.

In order to do this, first an algorithm to solve a KEP instance was implemented. The algorithm that Anderson et al. (2014) proposed in their research was used for this purpose. Then the effects on the average waiting times are investigated using simulation techniques. In the simulation every three months (one period) a KEP instance is solved. This is repeated twelve times, such that the simulation horizon is three years. In total seven simulation runs are performed. The effects over the seven simulation runs are investigated on the total population and on the subgroup to which priority is given.

When looking at the effects on the total population, there is no significant difference in the average waiting time if one of the subgroups was prioritized. However, there is a small increase from 61.8% to around 63% of the percentage matched. Looking at the results of the effects on the prioritized subgroups, there is no significant difference in the average waiting time for patients with blood type O and there is no effect on the percentage matched. Both highly sensitized patients and patients having donors with blood type AB have a significant difference in the average waiting time. For both subgroups, the average waiting time increases. This is also observed for the effects on percentage matched. The percentage matched increased from 74% to 92% for highly sensitized patients and for patients having donors with blood type AB this percentage increased from 37% to 65%.

To conclude, when highly sensitized patients are given priority above others, there is an (increasing) effect on the average waiting time of this subgroup, but more patients of this subgroup are matched. When patients having donors with blood type AB are given priority above others, there is also an (increasing) effect on the average waiting time of this subgroup, but also in this case more patients are matched. These increasing effects on the average waiting time can be explained due to the fact that in the first couple of periods of the simulation there could be some highly sensitized patients or patients having donors with blood type AB who are not compatible with other donors and hence, could not be matched at all. If one or more compatible donors appear in one of the next periods, the patients are matched with priority, but their waiting time is already incurred.

Contents

1. Introduction

1.1. Background information

In December 2015 there are more than 550 patients waiting for a kidney transplant in the Netherlands [6]. To reduce this waiting list, in addition to deceased donors, living donors can also donate a kidney. However, not every living donor is biologically compatible with his or her intended recipient due to blood type and crossmatch incompatibility.

Blood type (in)compatibility [4]

There are four different kind of blood types: A, B, AB and O. Every patient is compatible with their own blood type and possibly with others, see also Figure 1:

Figure 1: Blood type compatibility.

- Patients with blood type AB can receive a kidney of any blood type.
- Patients with blood type A can receive a kidney from someone with blood type O or A.
- Patients with blood type B can receive a kidney from someone with blood type O or B.
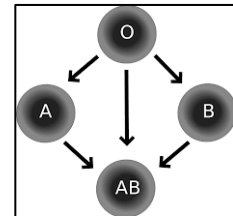- Patients with blood type O can only receive a kidney from someone with blood type O.

A patient-donor pair is incompatible when the combination of blood type differs from the ones described above. For example, a patient with blood type O and a donor with blood type AB are an incompatible patient-donor pair.

Crossmatch (in)compatibility [5]

When a crossmatch is performed, blood from the patient is mixed with blood from the donor. A positive crossmatch means the patient probably will reject the kidney after the transplant due to antibodies of the patient reacting against the donor's cells. In this case the pair is considered as crossmatch incompatible. A negative crossmatch means that the pair is crossmatch compatible. The panel reactive antibody (PRA), which is expressed as a percentage between 0% and 99%, represents the proportion of the population to which the patient will react via pre-existing antibodies. A patient with a PRA higher than 80% is called highly sensitized [8].

Kidney Paired Donation (KPD)

For incompatible patient-donor pairs so called kidney paired donation (KPD) exists. KPD allows patients to obtain transplants from other living donors. These living donors can belong to other incompatible patient-donor pairs. Two or more incompatible patient-donor pairs can exchange kidneys in cycles (Figure 2a). In order to ensure that every patient receives a kidney before his incompatible donor donates his kidney, these transplants are conducted simultaneously. Additionally, living donors can also be non-directed donors (NDDs), also called altruistic donors. This are donors who are willing to donate a kidney to any patient such that they can initiate a chain, followed by one or more incompatible pairs (Figure 2b).
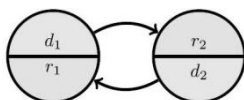

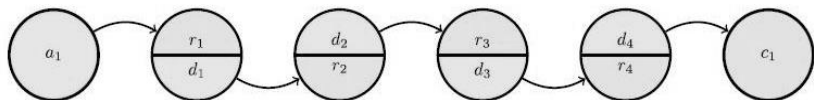**Figure 2a: Cycle of two incompatible patient (r)-donor (d) pairs.**

Figure 2b: Chain, initiated by an altruistic donor (a_1), followed by four incompatible pairs, ended with a patient (c_1) e.g. from the waiting list.

Source Figure 2a/2b: Anderson et al. (2014)

Kidney Exchange Problem (KEP)

Given a set of altruistic donors and a set of incompatible patient-donor pairs the goal in KPD is to find a set of transplants, organised in cycles and chains, that maximizes the total number of transplants performed. Anderson et al. (2014) refer to this problem as the Kidney Exchange Problem (KEP). They propose two new algorithms for solving the KEP. One of them is described in Chapter 2. This algorithm is implemented in order to be able to address the purpose of this research, which is stated below.

1.2. Problem description

When patients (or donors) do not receive a timely transplant they might become too weak to undergo a transplant or they might die [7]. Therefore, it would be interesting to take the waiting time into account when maximizing the total number of transplants. Segev et al. (2005) simulated incompatible patient-donor pairs using a decision tree model to predict the median waiting times for each blood type combination and two PRA subgroups, highly sensitized and non-highly sensitized. They showed that patients with PRA less than 80% match within a few months unless donors have blood type AB or patients have blood type O (except those with O donors). The longest waiting time has been seen by patients with PRA higher than 80% and blood type O in combination with AB donors. Only 2% of the patients with this combination is matched within three years. The result that this combination is the worst of all investigated combinations is not surprising. This is due to the fact that a patient with blood type O can only receive a kidney from donors with blood type O and a donor with blood type AB can only give a kidney to a patient with blood type AB, see Figure 1. In Appendix A all results of the research of Segev et al. (2005) of the median waiting times and likelihood of matching after 1, 2 and 3 years are displayed.

As mentioned before, a long waiting time is undesirable. But what happens to the waiting times if some sub-groups of patients and/or donors are given priority over others? The purpose of this research paper is to investigate what the effects are on the average waiting times when some priority is given to patients and/or donors who are difficult to match (blood type O patients, blood type AB donors, highly sensitized patients). Opposite to Segev et al. (2005) incompatible patient-donor pairs are simulated using the Saidman-generator [8]. Besides this, not the median, but the average waiting times are taken into account. The reason for this is that every human life counts. A small group of extreme high values, in this case patients with a very long waiting time, influences the average more than the median.

1.3. Outline

The outline of this research paper is as follows. It starts with the description of one of the algorithms proposed by Anderson et al. (2014) in Chapter 2. Methods and materials used to investigate the effects on the average waiting times when certain pairs are given priority are discussed in Chapter 3. Chapter 4 describes the results and the interpretations. Finally, this research paper ends with some conclusions in Chapter 5 and some limitations and recommendations in Chapter 6.

2. Algorithm for the Kidney Exchange Problem

As stated in the introduction, Anderson et al. (2014) developed two new algorithms for solving the KEP. One of them is implemented in this paper and is described in this chapter. How this algorithm is implemented can be found in Appendix B.

2.1. Background information

The algorithm of Anderson et al. is inspired by the Prize-Collecting Traveling Salesman Problem (PC-TSP), which is a generalization of the Traveling Salesman Problem (TSP). In the TSP the goal is, given a list of cities and costs between each pair of cities, to find a cycle that visits each city exactly once, while minimizing the costs of the overall tour. In the PC-TSP, there is an additional option to skip cities and paying a penalty for doing this. The goal is to find a cycle visiting each city at most once, while minimizing the costs of the tour plus the penalties of the cities not in the tour. Anderson et al. state: "Qualitatively, the PC-TSP is similar to the KEP in that one wants to find long paths in a graph (which the PC-TSP then closes off as a cycle), without the need to visit every node."

2.2. Outline of the sets, parameters and variables.

Before describing the variables, which are used in the algorithm, some common terms are explained.

An instance of KEP can be represented as a graph existing of nodes and arcs. Figure 3 shows a graph with seven nodes and eight arcs. Node q represents an altruistic donor and the other nodes represent an incompatible patient-donor pair. Arc a from node u to node v means that the donor of node u is compatible with the patient of node v. Node u is called the origin of arc a, node v is called the destination of arc a. If, for instance, node v contains a difficult to match patient, a weight can be assigned to arc a. Furthermore, this graph contains one cycle. The cycle has length 2, referring to the number of nodes in this cycle (nodes r and s).

Figure 3: A KEP instance represented as a graph.

The algorithm of Anderson et al. (2014) uses an integer linear programming formulation (ILP) of the KEP with binary variables for every edge and every cycle of length at most k.
It has constraints such that each node is used at most once (patient as well as donor). In order to block cycles longer than length k, the algorithm solves the KEP using delayed constrained generation, which is explained in the next section.

Table 1 represents an overview of the sets, parameters and variables described with the terms explained above.

Table 1: Overview of the sets, parameters and variables used in the algorithm of Anderson et al. (2014)

Sets	Description
N	Set of altruistic donors
P	Set of incompatible patient-donor pairs
V	Set of all nodes (altruistic donors and pairs)
S	Subset of V $(S \subset V)$
E	Set of all arcs
C_k	All possible cycles in a graph that use k or fewer nodes, $k \in \mathbb{N}^0$
$C_k(v)$	Cycles from C_k containing an arc incident to node v
$\delta^-(v)$	Set of arcs with destination node v, $v \in V$
$\delta^+(v)$	Set of arcs with origin node v, $v \in V$
$\delta^-(S)$	Set of arcs into S
$\delta^+(S)$	Set of arcs out of S

Parameters	Description
w_e	Weight of arc e, $e \in E$
w_C	Weight of cycle C, $w_C = \sum_{e \in C} w_e$, $C \in C_k$

Decision variables	Description
y_e	1 if arc e is selected, 0 otherwise
z_C	1 if cycle C is selected, 0 otherwise

2.3. Algorithm of Anderson et al. (2014)

Below, the algorithm of Anderson et al. is presented.

$$(1) \quad \max \quad \sum_{e \in E} w_e y_e + \sum_{C \in C_k} w_C z_C$$

$$(2) \quad \text{s.t.} \quad \sum_{e \in \delta^+(v)} y_e + \sum_{C \in C_k(v)} z_C \leq \sum_{e \in \delta^-(v)} y_e + \sum_{C \in C_k(v)} z_C \leq 1 \quad v \in P$$

$$(3) \quad \sum_{e \in \delta^+(v)} y_e \leq 1 \quad v \in N$$

$$(4) \quad \sum_{e \in \delta^-(S)} y_e \geq \sum_{e \in \delta^-(v)} y_e \quad S \subseteq P, v \in S$$

$$(5) \quad y_e \in \{0,1\} \quad e \in E$$

$$(6) \quad z_C \in \{0,1\} \quad C \in C_k$$

Objective function

The objective function (1) represents the goal of the algorithm. The goal is to find a set of cycles and arcs (from which the chains can be derived) that maximizes the sum of the weights of all transplants performed.

Constraints

Constraint (2) ensures that from all the outgoing arcs of a pair v (also the ones which are part of a cycle) at most one is selected. If an outgoing arc is selected, also an ingoing arc must be selected. It is not possible that a donor of a pair donates his kidney to another patient while the patient belonging to this donor, does not receive a kidney. The other way around is possible though. A patient can receive a kidney, but it is not necessary that the donor who belongs to the patient donates his kidney. This holds for all pairs. Constraint (3) says that from all the outgoing arcs of an altruistic donor, at most 1 can be selected. This must hold for all altruistic donors.

The decision variables can only be set to 1 or 0, meaning an arc or cycle is selected or not. This is ensured by constraints (5) and (6). When the algorithm is solved without adding constraint (4), the selected arcs can form cycles. However, this is not desirable, because they can form cycles longer than length k. For instance, if k is set to three and the arcs selected are $\{c, d, e, f\}$ as displayed in Figure 4, this is not a feasible solution. Constraint (4) ensures that if an incoming arc of pair $v \in S$ is selected, where $S \subseteq P$ (so S does not contain any altruistic donor), then for every way that pair v can be cut off from the altruistic donors, there is at least one arc selected going over this cut. In Figure 4, the graph contains one altruistic donor, labelled n. If node v is to be involved in any chain, then at least one of the arcs a or b that go across the cut separating S from the remaining pairs and altruistic donor, must be selected.

Figure 4: Example of constraint (4).
Adjusted from Anderson et al. (2014)

Delayed constraint generation

As a consequence of constraint (4), this formulation has exponentially many constraints. Therefore, the KEP is first solved without constraint (4). Then a check is done if the solution contains cycles longer than k. If so, the violated constraint is added and a new solution is obtained. This procedure is repeated until a feasible solution is found and is called delayed constraint generation. In this research this is referred to as the lazy constraints. The outline of the implementation can be found in Appendix B.

3. Method and materials

This chapter describes the method and materials used to investigate the effects on the average waiting time when certain pairs are given priority. The main technique used in this research is simulation. First, an outline of the simulation is given. Second, the parameters used for the simulation are explained after which the simulation is described in more detail. Next, the way the average waiting time is determined is described and is explained how to prioritise difficult to match patients and/or donors. Finally, an overview of the statistical analyses performed is given.

3.1. Outline of the simulation

The idea of the simulation is as follows. Starting with a waiting list containing incompatible patient-donor pairs and altruistic donors, new arrivals occur according to a Poisson process. After a certain period the KEP instance (consisting of the waiting list and the new arrivals) is solved by the algorithm of Anderson et al. (2014). Pairs who are matched are stored with their waiting time and the unmatched pairs form the new waiting list and the process starts again. This is repeated twelve times. After twelve times one simulation run is completed. The simulation runs are repeated seven times with several seeds[1].

3.2. Choosing the parameters

Generating incompatible patient-donor pairs and altruistic donors

Incompatible patient-donor pairs and altruistic donors are generated using the Saidman-generator [8]. The distributions of patient and donor data used in the Saidman-generator can be found in Appendix C.

Length of exchanging round (period)

Glorie et al. (2013) showed that when exchanges are run once per three months instead of once per month there are approximately 7% more transplantations. Therefore, the length of an exchanging round, called period in this research paper, is set to three months.

Arrival rate

At this moment in the Netherlands there are around 100 new arrivals of incompatible patient-donor pairs and altruistic donors per year. Therefore, the arrival rate for one period (three months) is set to 25 arrivals. These arrivals can either be altruistic donors or incompatible patient-donor pairs.

Initial waiting list

Suppose the simulation is started with an empty waiting list. The patients arriving in the first couple of periods have a disadvantage, because there is no donor pool yet and they depend on what kind of donors will arrive whether they will be matched or not, whereas patients who arrive later on, can benefit from a larger pool of donors. As a consequence, the waiting time of the patients arriving within the first couple of periods will be larger and the average waiting time will increase. In order to avoid this problem an initial waiting list is created, such that the percentage matched is more or less the same over time. The difference in the percentage matched between no waiting list at all and a waiting list of 200 altruistic donors and incompatible patient-donor pairs is depicted in Figure 5 and Figure 6. When there is no waiting list in the first period, only 41% of the patients is matched, this is a difference of around 20% compared to period 12. With 200 altruistic donors and incompatible patient-donor pairs as a starting point this difference is around 5%. Therefore an initial waiting list of 200 altruistic donors and incompatible patient-donor pairs is created.

Weights

The weights that are used in this research to prioritize difficult to match patients and/or donors are {2,3,4,5,10,20,50,100,200,500,1000}.

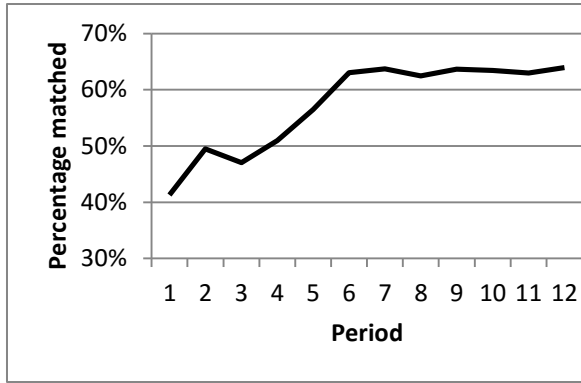[1] A seed is a number used to initialize a pseudorandom number generator.

Figure 5: Percentage matched over time, starting without a waiting list.
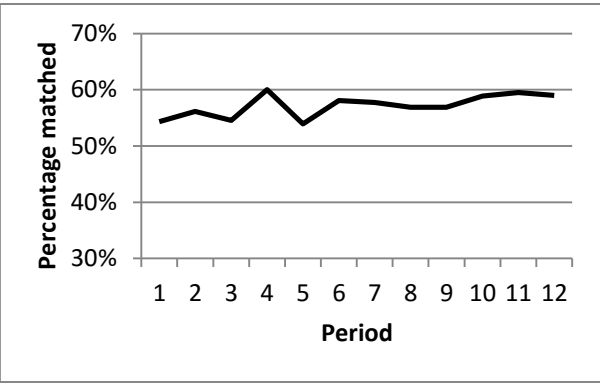
Figure 6: Percentage matched over time, starting with a waiting list of 200 altruistic donors and incompatible patient-donor pairs.

Cycle length

As explained in the introduction, transplants from cycles are conducted simultaneously. Because it is very complex to organize surgeries simultaneously from a logistical point of view, the maximum cycle length used in the algorithm of Anderson et al. (2014) is set to three.

Simulation horizon and number of seeds

Due to the algorithm of Anderson et al. (2014), referring to the lazy constraints method and the way the lazy constraints are implemented in this research, some running time problems occurred during the simulation. Every time a KEP instance is solved, the probability exists that exponential many constraints need to be added to the model and as a consequence, exponentially many models need to be solved. As a result, the running time of solving a KEP instance can take an exponential amount of time[2]. Taking a long simulation horizon or taking a large number of simulation runs increases the likelihood that solving one or more KEP instances takes an exponential amount of time. By experimenting with different lengths of the simulation horizons and the number of simulation runs a simulation horizon of three years (corresponding to twelve periods) and seven simulation runs appeared to perform best. The problem described in this section will be discussed in Chapter 6.

3.3. Simulation

By simulating a Poisson process with parameter $25 * 12$ (arrival rate $*$ periods) the arrival dates of the total instance for one simulation run are determined. Now we know the number of the nodes to be generated. The total instance for one simulation is, as mentioned before, generated using the Saidman-generator [8]. Next, the instance for the first period is derived from the total instance according to the arrival dates (arrivals within three months). Also the initial waiting list is added to this instance. After splitting the total instance, the compatibility between each patient and donor of this instance is determined. Then, this partial instance is solved by the algorithm of Anderson et al. (2014), according to the specified weights (see Section 3.5). Nodes that are matched are stored with their waiting time and the unmatched nodes are added to the next partial instance. This process is repeated for all of the 12 periods. After completing one simulation run, the average waiting time of this run is determined. A simulation run is done seven times (with different seeds, as explained in the previous section) with the same weights. Finally, the average waiting time belonging to the specified weights are computed. How this is done, is explained in the next section.

---

[2] After a running time of 24 hours, trying to solve an instance consisting of 1 altruistic donor and 84 incompatible patient-donor pairs, there were more than 5600 lazy constraints added to the model and no feasible solution was still found. The simulation was stopped by hand.

3.4. Determining the average waiting time

Inclusion criteria
Altruistic donors and patients from the initial waiting list are not taken into account when the average waiting time is determined. The waiting time of an altruistic donor is not important for this research and the waiting list is just created to prevent disturbances in the average waiting time due to so called start-up problems as explained in Section 3.2. Also patients who are not matched are not included for determining the average waiting time, because these patients are still waiting for a kidney transplant, but they are included in the calculation of the percentage matched patients.

Average waiting time over the seven runs
After each run the average waiting time and the number of nodes matched are determined. Of the seven average waiting times per run a weighted average is determined with respect to number of nodes matched.

3.5. Prioritizing difficult to match patients and/or donors

In the algorithm of Anderson et al. (2014) it is possible to assign weights to the arcs. If all weights are equal to 1, this means that no priority is given to any patient and/or donor. Recall from Chapter 2 that an arc a from node u to node v, means that the donor of node u is compatible with the patient of node v. If, for instance, node v represents a difficult to match patient, the weight of the arc a can be increased to give this patient/node priority.

In this research three subgroups are distinguished having characteristics of patients that are difficult to match:
1. Highly sensitized patients (PRA $\geq 80\%$);
2. Patients with blood type O;
3. Patients having donors with blood type AB.

For each subgroup the weights are increased and the effects on the average waiting time and the percentage matched compared to the case when no priority is given, are investigated. This is done for the effects on the total population, meaning the total instances of the seven simulation runs each consisting of twelve periods, but without the initial waiting lists, as well as the effects on the subgroups.

3.6. Statistical analysis

In order to be able to say something about the reliability of the average waiting times for subgroups the 95% confidence intervals for each average waiting time is reported. Furthermore, to test the hypothesis for the total population (H_0: Higher weights do not influence the average waiting times of the total population against H_1: Higher weights do influence the average waiting times of the total population) and the hypothesis per subgroup (H_0: Higher weights do not influence the average waiting times of the subgroup given priority against H_1: Higher weights do influence the average waiting times of the subgroup given priority), the sign test is performed, with the number of trials equal to the number of simulations runs (seven).

4. Results

In this chapter the results of this research are displayed and interpreted. First, some general statistics of the total population is given. As explained in Section 3.5 the total population refers to the instance over seven simulation runs each consisting of twelve periods, but without the initial waiting lists. Second, the effects of increasing the weights on the number of percentage matched and on the average waiting times of the total population is given. Finally, these results are also given for the effects on the subgroups.

4.1. General statistics of the total population

The total population consists of 83 altruistic donors and 2001 incompatible patient-donor pairs. Hence, in total there are 2001 patients, of which 18% (= 358 patients) is highly sensitized, and 2084 donors. Table 2 and Figure 7 show the blood type distributions of this population. As can be seen more than 50% of the patients have blood type O. As explained in the introduction, based on blood type compatibility, these patients can only receive a kidney from donors with blood type O. In this population there are in total 540 donors (altruistic donors and donors from pairs) with blood type O. So, in advance it is known that at least $1132 - 540 = 592$ patients with blood type O will not be matched at all. This is about 30% of the total population.

Table 2: Number of patients, donors and altruistic donors per blood type.

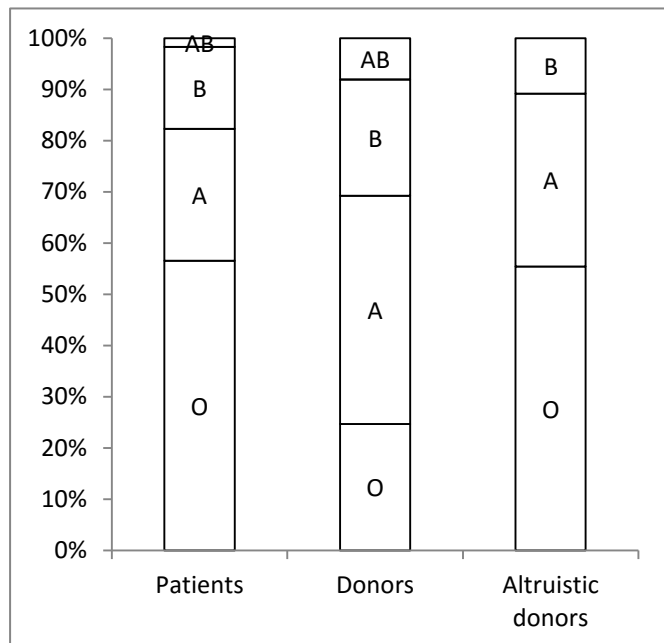| Blood type	Patients	Donors	Altruistic donors
O	1132	494	46
A	515	891	28
B	320	454	9
AB	34	162	0
Total	2001	2001	83

Figure 7: Blood type distributions of patients with incompatible donors, donors with incompatible patients and altruistic donors.

For more detailed information about the statistics based on patient/donor blood types and patient sensitization, see Appendix D. In the next section the results of percentage matched and the average waiting time for the total population are given. Recall from Chapter 3 that the percentage matched is a percentage of the total number of pairs (not the altruistic donors) and the average waiting time is only calculated of those pairs who indeed are matched.

4.2. Effects on the total population

If all patients are equally weighed then in total 61.82% of the 2001 patients are matched. Figure 8 shows the effects on the number of percentage matched of the total population when subgroups are given priority. If all highly sensitized patients are given priority, starting with a weight of 2, then this percentage decreases slightly to 61.72%, when a weight of 3 is given this percentage is increased to more than 63%, and so on. The effects are also displayed when patients with blood type O are given priority and when patients having a donor with blood type AB are given priority.
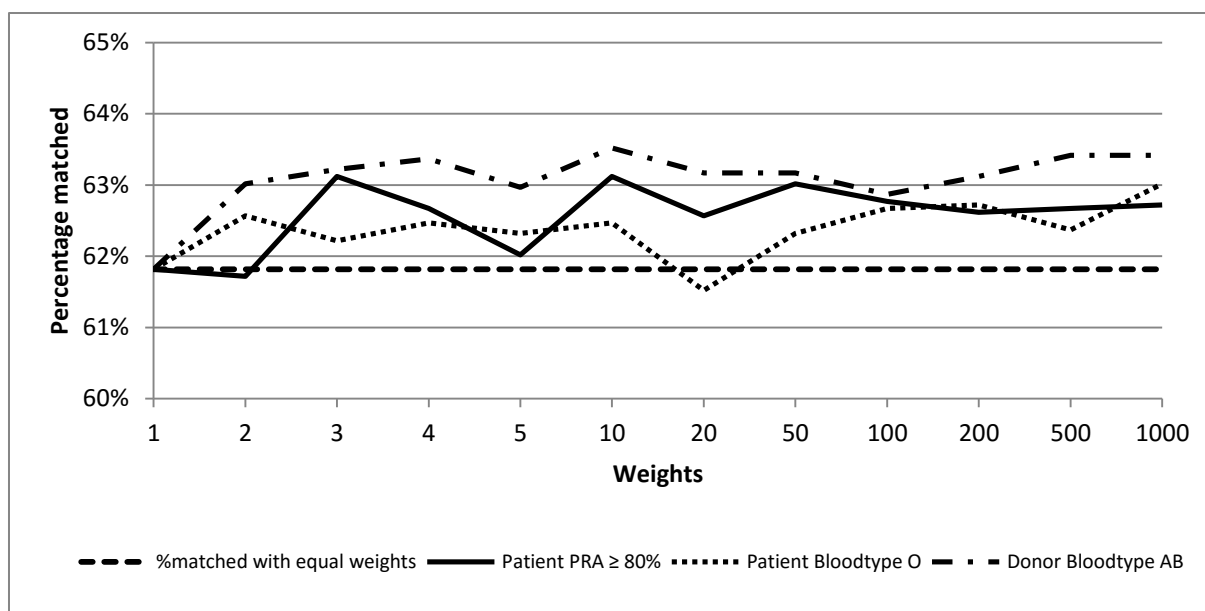
Figure 8: Percentage patients matched when changing the weights of subgroups.

From Figure 8 it can be seen that increasing the weight within a subgroup sometimes has an increasing effect and sometimes a decreasing effect on the total percentage matched patients. However, in general it holds that for each prioritized subgroup the percentage matched increased about 1% when looking at the higher weights (≥ 100) compared to all patients having equal weights. Changing the weight within the subgroup "Donor blood type AB" not only has the highest effect but the line is more stable compared to the other two subgroups. As said in the previous section at most 70% of the patients can be matched due to the shortage of blood type O donors.

The average waiting time of the total population, if all weights are equal, is 102 days.
Figure 9 shows the effects on the average waiting time of the total population when subgroups are given priority. This figure can be read similar as Figure 8.
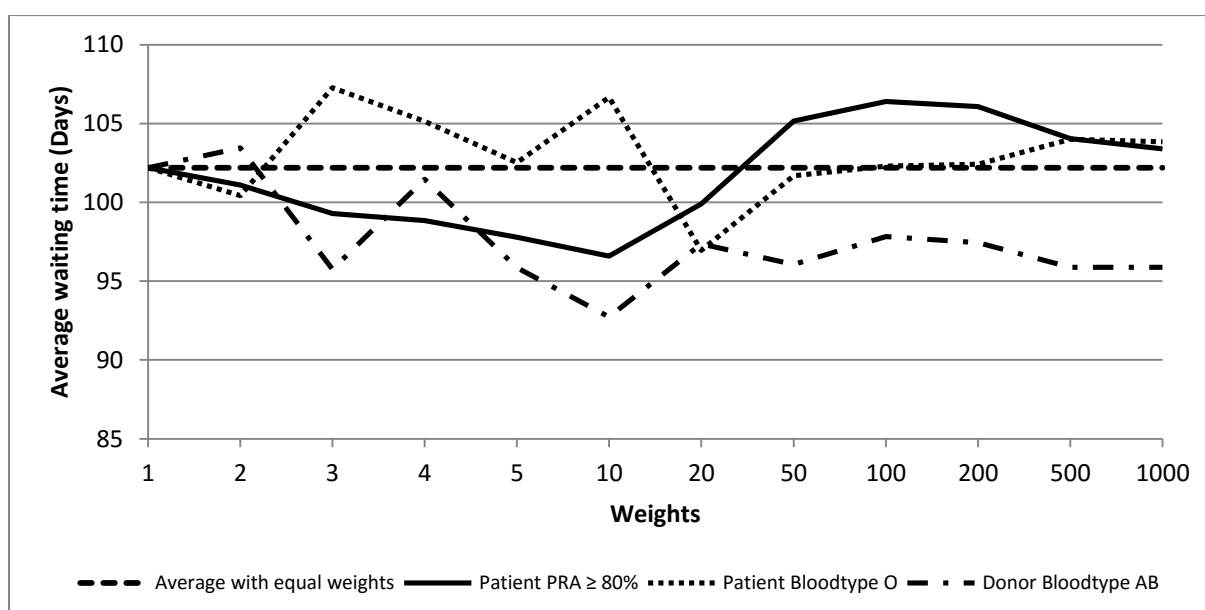

Figure 9: Average waiting time when changing the weights of subgroups.

Increasing the weight shows for the subgroups "Patient PRA $\geq 80\%$" and "Patient blood type O" that sometimes the average waiting time is higher compared to all patients having equal weights and sometimes it is lower, so no clear pattern is seen here. None of the differences are significant

11

according to the sign test ($P \geq 0.13$). Only the subgroup "Donor blood type AB" shows in general a lower average waiting time. Looking again at the higher weights (≥ 100) the average waiting time of the total population is decreased to 96 days. However, also for this group, none of the differences in waiting times are significant ($P \geq 0.45$). An overview of the results (p-values) of the sign tests is given in Appendix E.1. Therefore, the hypothesis that higher weights do not influence the average waiting times of the total population, is not rejected.

Because no significant difference was found in the average waiting time of the total population, also the effects per subgroup are studied. The results are presented in the next section.

4.3. Effects per subgroup
Figure 10, Figure 11 and Figure 12 show the effects on percentage matched (left vertical axis) and the average waiting time (right vertical axis) of the subgroup to which priority is given. Also, the 95% confidence intervals for the average waiting time is displayed.

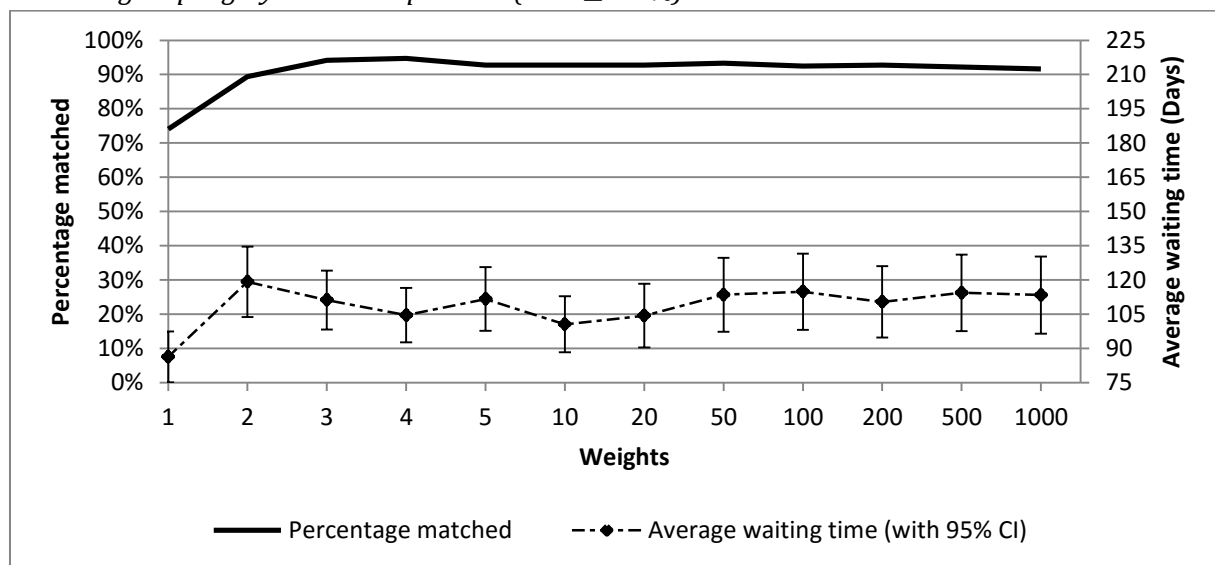*4.3.1 Subgroup highly sensitized patients (PRA $\geq$ 80%)*

Figure 10: Percentage matched and average waiting time with the 95% confidence interval of highly sensitized patients.

If all patients are equally weighed, a total of 74% of the 358 highly sensitized patients is matched and they have an average waiting time of 86 days. If all highly sensitized patients are given priority, then the percentage matched increases and when the weight is greater than or equal to 3, this percentage is hovering around the 93%. However, the average waiting time is increasing as well up to an average waiting time of 113 days for higher weights (≥ 50). The difference in the average waiting time is significant when the weight is greater than or equal to 50 ($P = 0.016$), see Appendix E.2. Hence, the hypothesis: higher weights of highly sensitized patients do not influence the average waiting times of these subgroup, is rejected. At first sight, an increasing average waiting time might seem like a surprise. However, this can be explained due to the fact that in the first couple of periods of the simulation there could be some highly sensitized patients who are not compatible with other donors and hence, could not be matched at all. If one or more compatible donors appear in one of the next periods, the patients are matched with priority, but their waiting time is already incurred.

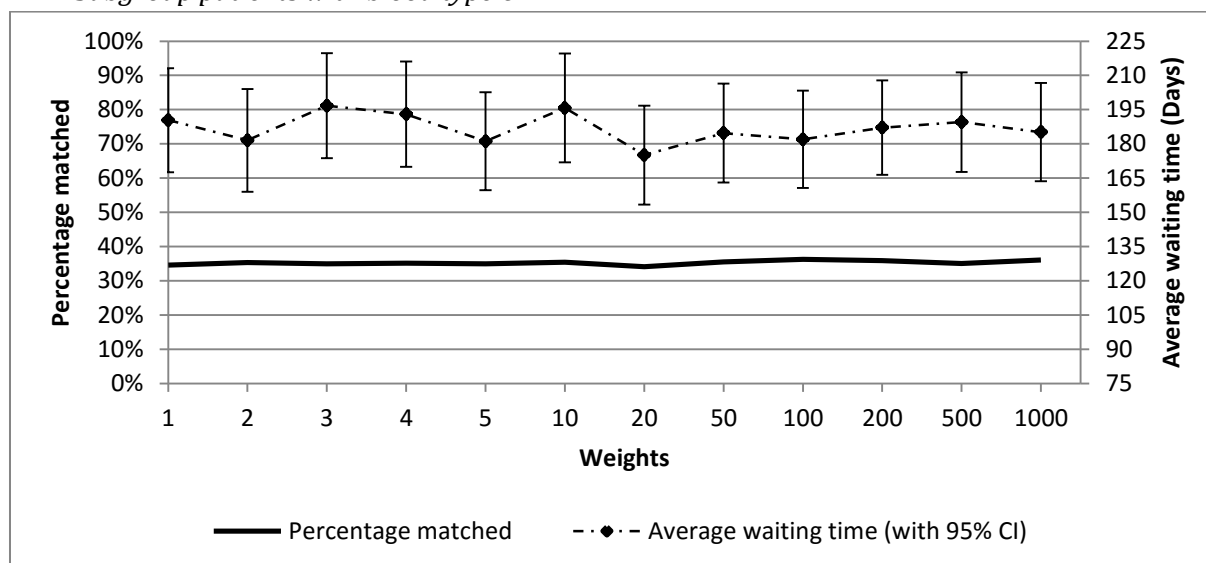## 4.3.2 Subgroup patients with blood type O

Figure 11: Percentage matched and average waiting time with the 95% confidence interval of patients with blood type O.

If all patients are equally weighed, a total of 34.54% of the 1132 patients with blood type O is matched and they have an average waiting time of 190 days. If all patients with blood type O are given priority, then the percentage matched is hovering between 35% and 36%. A reason that this percentage is rather low (and increasing the weights has slightly influences on the percentage matched), is explained in Section 4.1. At most 48% of the patients with blood type O can be matched due to the shortage of blood type O donors. Likewise, the average waiting time does not differ too much when the weight of patient with blood type O is increased. None of the differences are significant ($P \geq 0.13$), see Appendix E.2. Therefore, the hypothesis: higher weights of patients with blood type O do not influence the average waiting times of these subgroup, is not rejected.

4.3.3 Subgroup patients having a donor blood with type AB

Figure 12: Percentage matched and average waiting time with the 95% confidence interval of donors with blood type AB.

If all patients are equally weighed, a total of 37% of the 162 patients having a donor with blood type AB is matched and they have an average waiting time of 134 days. If all of the patients having a donor with blood type AB are given priority then the percentage matched increases and when the weight is greater than or equal to 20, this percentage is hovering around the 65%. As explained in the introduction, donors with blood type AB can only give a kidney to patients with blood type AB. Of the

total population there are only 34 patients with blood type AB. The other possibility is that the patient of these donors is the end of the chain. This chain must be initiated by an altruistic donor of which there are 83 in this population. So, at most $\frac{34+83}{162} * 100\% = 72\%$ patients having a donor with blood type AB could be matched. In addition to the percentage matched, the average waiting time is increasing as well. The difference of the average waiting time is significant when the weight is greater than or equal to 20 ($P = 0.016$), see Appendix E.2. Hence, the hypothesis: higher weights of patients having a donor with blood type AB do not influence the average waiting times of these subgroup, is rejected. Just as for the highly sensitized patient, the increasing average waiting time can be explained with the same kind of argumentation. It might be the case that in de first couple of periods of the simulation there are patients having a donor with blood type AB, whereas there are no patients with blood type AB and/or no altruistic donors. As a consequence, these patients could not be matched at all. If one or more patients with blood type AB and/or altruistic donors appear in one of the next periods, these patients are matched with priority, but their waiting time is already incurred.

5. Conclusion

The purpose of this research paper was to investigate what the effects are on the average waiting times when some priority is given to patients and/or donors who are difficult to match. Three subgroups were distinguished having characteristics of patients and/or donors that are difficult to match:

1. Highly sensitized patients (PRA $\geq 80\%$);
2. Patients with blood type O;
3. Patients having donors with blood type AB.

Because patients who are not matched at the end of the simulation run are not included for determining the average waiting time, the percentage of matched patients is calculated in order to say something about the proportion of this population. When looking at the effects on the total population, there is no significant difference in the average waiting time if one of the subgroups was given priority. However, a small increase from 61.8% to around 63% of the percentage matched is observed. Looking at the results of the effects on the prioritized subgroup, there is no significant difference in the average waiting time for patients with blood type O, nor an effect was observed on the percentage matched.

Both prioritizing highly sensitized patients and patients having donors with blood type AB have a significant difference in the average waiting time compared to patients who are not prioritized. For the highly sensitized patients this difference is significant for weights greater than or equal to 50, for patients having donors with blood type AB this difference is significant for weights greater than or equal to 20. For both subgroups the average waiting time seems to increase. This is also observed for the effects on percentage matched. The percentage matched is increased from 74% to 92% for highly sensitized patients. For patients having donors with blood type AB this percentage is increased from 37% to 65%.

To conclude, when highly sensitized patients are given priority above others, there is an (increasing) effect on the average waiting time of this subgroup, but also more patients of this subgroup are matched. When patients having donors with blood type AB are given priority above others, there is also an (increasing) effect on the average waiting time of this subgroup, and also more patients are matched.

6. Discussion

6.1. Limitations
In this research, in the simulation process, is assumed that patients and donors do not die or become too weak when for instance their waiting time becomes too long. Which is in practice not a realistic assumption. On the other hand, after a matching round donors at the end of a chain disappear, whereas it would be an opportunity to use these donors in the next matching round as an altruistic donor. As such they can initiate a new chain. In general, these kind of donors are so called bridge donors [3]. Using bridge donors to initiate a new chain is also hardly applied in practice.

However, the greatest limitations of this research were the lazy constraints of the algorithm of Anderson et al. (2014) (1) and that the optimization process in CPLEX is some kind of black box (2).

(1)
With the algorithm of Anderson et al. (2014), as explained in Section 3.2, due to the lazy constraints method and the way the lazy constraints are implemented in this research, some running time problems occurred. For this reason, the simulation horizon was set to a period of three years and only seven simulation runs were performed. Increasing the number of simulation runs is preferable. If there are more runs performed, the outcomes of the sign test become more reliable.
For the simulation horizon it would be preferred to have a simulation period larger than three, but not too large. Increasing the simulation horizon will probably give a higher average waiting time, because in the worst case scenario a patient now has the possibility to wait more than three years. On the other hand, increasing the simulation horizon will give a better approximation of the percentage matched.

(2)
In CPLEX it is not clear, which choice is made if several optimal solutions for a KEP instance exist. It might be expected that if all weights are multiplied by factor of ten, that for the same instance the same solution is given. Unfortunately, this was not always the case. In this research the situation of multiplying all weights is not considered, but the consequences not giving the same solution might have influenced the results.

6.2. Recommendations
In this research only the effects on the average waiting time when a certain subgroup is given priority are investigated. But it might be interesting to look at a combination of subgroups. For instance, giving all highly sensitized patients and all patients with blood type O a weight of two, but all highly sensitized patients with blood type O a weight of four. Another interesting research direction would be to double the weight every time a patient is not matched in a matching round.

7. References

[1]   Anderson R, Ashlagi I, Gamarnik D, et al. Finding long chains in kidney exchange using the traveling salesman problem. *www.pnas.org/cgi/doi/10.1073/pnas.1421853112;* 2014.

[2]   IBM (2015), IBM ILOG CPLEX Optimization Studio: CPLEX User's Manual. Version 12 Release 6

[3]   Glorie K, de Klerk M, Wagelmans A, et al. Coordinating unspecified living kidney donation and transplantation across blood-type barrier in kidney exchange. *Transplantation* 2013; 96: 814-820.

[4]   Johns Hopkins Medicine 2015, *Blood type incompatibility.* Available from: <http://www.hopkinsmedicine.org/transplant/programs/kidney/incompatible/blood.html>. [11 December 2015].

[5]   Johns Hopkins Medicine 2015, *Crossmatch incompatibility.* Available from: <http://www.hopkinsmedicine.org/transplant/programs/kidney/incompatible/positive_crossmatch.html>. [11 December 2015].

[6]   Nederlandse Transplantatie Stichting 2015, *Actuele cijfers organen.* Available from: <http://www.transplantatiestichting.nl/cijfers/actuele-cijfers-organen>. [11 December 2015].

[7]   Nederlandse Transplantatie Stichting 2015, *Wachtlijst.* Available from: <http://www.transplantatiestichting.nl/nieuws/wachtlijst-neemt-af-goed-slecht-nieuws>. [11 December 2015].

[8]   Saidman S, Roth A, Sonmez T, et al. Increasing the opportunity of live kidney donation by matching for two- and three-way exchanges. *Transplantation* 2006; 81: 773-782.

[9]   Segev D, Gentry S, Melancon J, et al. Characterization of waiting times in a simulation of kidney paired donation. *American journal of transplantation* 2005; 5: 2448-2455.

Appendices

Appendix A: Median waiting times and likelihood of matching by Segev et al. (2005)

Median waiting time (in months) and likelihood of matching after 1,2,3, years through KPD.

| Blood type | | Waiting time | | % Matched After 1 year | | After 2 year | | After 3 Year | |
Donor	Recipient	PRA < 80	PRA ≥ 80	PRA < 80	PRA ≥ 80	PRA < 80	PRA ≥ 80	PRA < 80	PRA ≥ 80
O	O	0	11	100%	55%	100%	74%	100%	83%
A	O	3	*	100%	19%	100%	35%	100%	48%
B	O	3	*	100%	7%	100%	14%	100%	24%
AB	O	10	*	75%	0%	99%	0%	100%	2%
O	A	0	2	100%	87%	100%	95%	100%	97%
A	A	0	14	100%	48%	100%	67%	100%	77%
B	A	0	25	100%	32%	100%	51%	100%	61%
AB	A	3	*	100%	3%	100%	9%	100%	15%
O	B	0	5	100%	73%	100%	90%	100%	95%
A	B	0	19	100%	39%	100%	60%	100%	73%
B	B	0	*	100%	16%	100%	29%	100%	41%
AB	B	4	*	99%	3%	100%	5%	100%	10%
O	AB	0	3	100%	85%	100%	93%	100%	97%
A	AB	0	9	100%	60%	100%	79%	100%	87%
B	AB	0	12	100%	53%	100%	71%	100%	84%
AB	AB	0	*	100%	9%	100%	18%	100%	22%

Waiting time of 0 = matched on first round; * = fewer than half matched after 36 months; PRA = panel reactive antibody.

Source: Segev et al.(2005)

Appendix B: Implementation of the algorithm

The algorithm, which is formulated as an ILP model, is implemented in the commercial software CPLEX. CPLEX offers C, C++, Java, .NET, and Python[3] libraries that solve linear programming and related problems. Among other things, it solves linearly constrained optimization problems where the objective to be optimized can be expressed as a linear function. For more information, see the user manual of CPLEX [2].

Building the model
The model is built into CPLEX by columns according to the following steps.
1. *Generating the decision variables*
   In this step, all possible arcs are generated and all possible cycles with a maximum of length 3[4] are generated.
2. *Add empty objective function and constraints*
   a) The objective is initialised as a maximization problem
   b) Altruistic constraints: For each altruistic donor an empty constraint is created (constraint 3 of the model), if there are no altruistic donors in a certain instance then no altruistic constraints are created.
   c) Pair constraints:
      For implementation, constraint 2 of the model is divided into two parts:
      i) $\sum_{e\in\delta^+(v)} y_e + \sum_{C\in C_k(v)} z_C - \sum_{e\in\delta^-(v)} y_e - \sum_{C\in C_k(v)} z_C \leq 0 \qquad v \in P$, which can be rewritten as: $\sum_{e\in\delta^+(v)} y_e - \sum_{e\in\delta^-(v)} y_e \leq 0 \qquad v \in P$
      ii) $\sum_{e\in\delta^-(v)} y_e + \sum_{C\in C_k(v)} z_C \leq 1 \qquad v \in P$
      Hence, for each pair two empty pair constraints are created.
3. *Add created variables to the objective function and the constraints*
   In this step the generated variables are added to the objective function with their specified weight. Next, the generated variables are added to the constraints they belong. For example, if there is an arc from node 4 to node 9 and node 4 is an altruistic donor then this arc is added to the altruistic constraint belonging to node 4. The arc is also added to the first pair constraint belonging to node 4 and to the two constraints belonging to node 9, in which it has a negative sign in the first pair constraint.
4. *Tell CPLEX that the generated decision variables must be binary variables*

Up to this part everything went well, but so far nothing has been done with constraints 4 of the model (the lazy constraints). CPLEX has a library for adding lazy constraint to a model. However, despite of the reading the user manual [2] and searching the internet thoroughly, it was not possible to implement this via the library. So, the lazy constraints are added manually to the model. This procedure is described in the following section.

Adding lazy constraints to the model
5. *Let CPLEX solve the model, without adding any lazy constraint*
6. *Check if there are cycles created which length is longer than 3*
   In order to be able to do this, it is needed to connect the selected arcs to each other.
   a) Create three lists:
      i)   altruistic arc list in which selected arcs are stored if the origin of the selected arc come from an altruistic donor
      ii) pair arc list in which all other selected arcs are stored
      iii) cycle and chain list, an empty list for storing arrays

---

[3] C, C++, Java, .NET, and Python are all programming languages
[4] The reason why cycles are bounded by length three, is explained in Section 3.2

b) Create chains from altruistic arc list
    i) select arc A from altruistic arc list
    ii) nodeFrom = A.origin
    iii) nodeTo = A.destination
    iv) add nodeFrom en nodeTo to array Q
    v) find in the pair arc list the arc B where nodeTo equals B.origin
    vi) add B.destination to Q
    vii) nodeTo = B.destination
    viii) remove B from pair arc list
    ix) repeat from step 6.b.v until no matches can be found
    x) add Q in cycle and chain list
    xi) do this for every selected arc in the altruistic list

c) Create cycles from the remaining selected arcs in the pair arc list
    i) select arc A from pair arc list
    ii) nodeFrom = A.origin
    iii) nodeTo = A.destination
    iv) add nodeFrom en nodeTo to array Q
    v) remove A from pair arc list
    vi) find in the remaining pair arc list the arc B where nodeTo equals B.origin
    vii) add B.destination to Q
    viii) nodeTo = B.destination
    ix) remove B from pair arc list
    x) repeat from step 6.c.vi until no matches can be found
    xi) add Q to the cycle and chain list
    xii) if the pair arc list is not empty repeat from step 6.c.i

d) Get the largest cycle stored in the chain and cycle list and check if his length is greater than 3, if not the solution is found, if so, go to step 7

7. *Add lazy constraint to model belonging to the largest cycle.*
Recall the lazy constraint of the model: $\sum_{e\in\delta^-(S)} y_e \geq \sum_{e\in\delta^-(v)} y_e$ $S \subseteq P, v \in S$, rewrite this for implementation: $\sum_{e\in\delta^-(v)} y_e - \sum_{e\in\delta^-(S)} y_e \leq 0$ $S \subseteq P, v \in S$
Note that all arcs from which the origin is not in S and from which the destination is v, cancels out in this equation.

a) The set S is formed by the nodes in the cycle, let v be the first node of S
b) Select arc A from the generated arc list (step 1)
c) If A.destination equals v and and A.origin is in S: add A to the lazy constraint with a positive sign.
d) If A.origin is not in S and A.destination is in S, but is not equal to v: add A to the lazy constraint with a negative sign.
e) Repeat from b on, until all arcs are checked
f) Add the total lazy constraint to the model

8. Solve the model again and go to step 6, repeat until a solution is found.

Appendix C: Patient and donor distributions by Saidman et al. (2006)

Patient and living donor distributions in simulations using Organ Procurement and Transplantation Network and Scientific Registry of Transplant Recipients (OPTN/SRTR) data.

Characteristic	Frequency (%)
Patient blood type	
O	48.14
A	33.73
B	14.28
AB	3.85
Patient sex	
Female	40.90
Male	59.10
Unrelated living donors	
Spouse	48.97
Other	51.03
PRA distribution	
Low PRA	70.19
Medium PRA	20.00
High PRA	9.81

Source: Saidman et al. (2006)

Additionally:
- Each low PRA patient (PRA<10%) has a positive crossmatch probability of 5% with a random donor,
- Each medium PRA patient (10-80%) has a positive crossmatch probability of 45% with a random donor,
- Each high PRA patient (PRA>80%) has a positive crossmatch probability of 90% with a random donor.

Appendix D: Detailed statistics of the total population

This appendix provides more insight in the statistics of the total population based on patient/donor blood types and patient sensitization. The total population includes 2001 incompatible patient-donor pairs. From these 2001 pairs more than 25 percent consist of the combination of a non-highly sensitized patient with blood type O and a donor with blood type A.

Blood type patient	Blood type donor	Patient PRA < 80%	Patient PRA >= 80%
O	O	7,95%	3,90%
A	O	5,55%	2,65%
B	O	2,15%	1,70%
AB	O	0,55%	0,25%
O	A	25,64%	2,45%
A	A	3,10%	2,70%
B	A	9,15%	0,95%
AB	A	0,40%	0,15%
O	B	11,09%	1,25%
A	B	8,50%	0,95%
B	B	0,50%	0,15%
AB	B	0,20%	0,05%
O	AB	3,85%	0,45%
A	AB	2,15%	0,15%
B	AB	1,25%	0,15%
AB	AB	0,10%	0,00%
Total		82,11%	17,89%

Appendix E: Results of the sign test

In this appendix the results of the sign test are given.

E.1 Results for the total population

For the total population the hypothesis tested is:

H_0: Higher weights do not influence the average waiting times of the subgroup given priority.

H_1: Higher weights do influence the average waiting times of the subgroup given priority.

Weight	Patient PRA$\geq$ **80**%	Patient blood type O	Donor blood type AB
2	1	1	1
3	1	1	0,45
4	0,45	1	0,45
5	0,45	1	0,45
10	0,45	0,13	0,45
20	1	0,45	0,45
50	0,45	1	0,45
100	0,45	1	0,45
200	0,45	1	0,45
500	0,45	1	0,45
1000	1	0,45	0,45

E.2 Results per subgroup

For each subgroup the hypothesis tested is:

H_0: Higher weights do not influence the average waiting times of the subgroup given priority.

H_1: Higher weights do influence the average waiting times of the subgroup given priority.

Weight	Patient PRA$\geq$ **80**%	Patient blood type O	Donor blood type AB
2	0,13	0,45	0,13
3	0,45	1	0,13
4	1	1	0,13
5	0,13	1	0,45
10	1	0,45	0,45
20	0,13	0,13	0,016
50	0,016	1	0,016
100	0,016	1	0,016
200	0,016	0,45	0,016
500	0,016	1	0,016
1000	0,016	1	0,016