ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Faculty of informatics

NEUMANN JÁNOS
FACULTY OF INFORMATICS

DEGREE PROJECT
THESIS

OE-NIK

2022

Fruzsina Eszter Kulcsár

T/008110/FI12904/N

Óbuda University
John von Neumann Faculty of Informatics
Institute for Cyber-physical Systems

DIPLOMA THESIS
TOPIC DESCRIPTION

Student name: **Fruzsina Eszter Kulcsár**
Registration number: T/008110/FI12904/N
Neptun's code: [redacted]

Title of diploma thesis:

Statistical analysis module for realistic kidney exchange simulations
Statisztikai elemző modul valósághű vesecsere szimulációkhoz

Internal supervisor: Dr. Rita Fleiner
External supervisor: Dr. Péter Biró

Deadline: 15th of May, 2022.

Subjects of the final exam: Engineering computational methods
Optimization
System and control theory
Stochastics

Topic description

Kidney Exchange Programs (KEPs) are organized in several countries to help patients who are waiting for kidney transplantation. Each KEP has a software, which is searching for exchange options in the set of registered incompatible donor-patient pairs. The collaboration of the individual KEPs would give much better chances for the patients to find matches, as exchanges between the countries give more options. This motivated a European project, called ENCKEP, where software is being developed to simulate the cooperation of different KEPs with the purpose to show the countries the potential, and convincing them to collaborate.
The subject of this work is a new module of this software, a tool, which facilitates realistic data generation with the help of statistical analysis. Realistic data generation is fundamental in the project, as the used data determines the correctness of the simulations. The aim of this thesis, to develop a tool that uses statistical analysis to examine the generated datasets and decide whether they are close to reality or not by comparing the statistics of the generated datasets to the statistics of real datasets. The data generation process should be improved until the process is perfected and realistic datasets are produced.
The tool seems extremely useful, as it does not require real datasets, only statistics about them, which are less protected.

The diploma thesis should cover:

- Introduction of Kidney Exchange Programs and the needed medical and technical definitions of kidney exchange
- Study of statistical analysis methods
- Plan of the tool which improves realistic data generation with the help of statistical analysis
- Implementation of the tool
- Description of the used statistical methods

Fleiner R.
.....
Dr. Rita Feliner
head of institute

Term of limitation: **15th of May, 2024.**

The diploma thesis is adequate and can be submitted:

.....
external supervisor

Fleiner Rita
.....
internal supervisor

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Faculty of Informatics

STUDENT'S DECLARATION

I the undersigned student hereby declare that this degree project / thesis is the result of my own work; I have disclosed the references and tools used in an identifiable manner. The results indicated in my degree project / thesis completed may be used by the university and the institution announcing the task for their own purposes free of charge.

Dated: Budapest, 2022.05.11.

.....

student's signature

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar
9. sz. melléklet

KONZULTÁCIÓS NAPLÓ

Hallgató neve:	Neptun kód:	Tagozat:
Kulcsár Fruzsina Eszter	[redacted]	*OE NIK*

Telefon: Levelezési cím (pl: lakcím):

[redacted]

Szakdolgozat / Diplomamunka[1] címe magyarul:

Statisztikai elemző modul valósághű vesecsere szimulációkhoz

Szakdolgozat / Diplomamunka[2] címe angolul:

Statistical analysis module for realistic kidney exchange simulations

Intézményi konzulens:	Külső konzulens:
Fleiner Rita	*Biró Péter*

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	**2021.09.26.**	**Szakdolgozat témájának megbeszélése**	Fleiner Rita
2.	**2021.10.05.**	**Feladatlap megbeszélése**	Fleiner Rita
3.	**2021.11.29.**	**Irodalomkutatás megbeszélése**	Fleiner Rita
4.	**2021.12.06.**	**Dokumentumok beadásának megbeszélése**	Fleiner Rita

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4[3] tantárgy követelményét teljesítette, beszámolóra / védésre[4] bocsátható.

Budapest, 2021.12.10.

Fleiner Rita
intézményi konzulens

[1] Megfelelő aláhúzandó!
[2] Megfelelő aláhúzandó!
[3] Megfelelő aláhúzandó!
[4] Megfelelő aláhúzandó!

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

9. sz. melléklet

KONZULTÁCIÓS NAPLÓ

Hallgató neve:	Neptun kód:	Tagozat:
Kulcsár Fruzsina Eszter	[redacted]	**NIK**

Telefon: Levelezési cím (pl: lakcím):

[redacted]

Szakdolgozat / Diplomamunka[1] címe magyarul:

Statisztikai elemző modul valósághű vesecsere szimulációkhoz

Szakdolgozat / Diplomamunka[2] címe angolul:

Statistical analysis module for realistic kidney exchange simulations

Intézményi konzulens:	Külső konzulens:
Fleiner Rita	**Dr. Biró Péter**

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	**2022.02.26.**	**Irodalomkutatás véglegesítése**	Fleiner Rita
2.	**2022.03.05.**	**Követelmény specifikáció, rendszerterv**	Fleiner Rita
3.	**2022.04.29.**	**Kódolás, eredmények összegzése**	Fleiner Rita
4.	**2022.05.06.**	**Dokumentumok beadásának megbeszélése**	Fleiner Rita

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4[3] tantárgy követelményét teljesítette, beszámolóra / védésre[4] bocsátható.

Budapest, 2022.05.10.

Fleiner Rita
intézményi konzulens

[1] Megfelelő aláhúzandó!
[2] Megfelelő aláhúzandó!
[3] Megfelelő aláhúzandó!
[4] Megfelelő aláhúzandó!

Contents

Abstract

The aim of this work was to develop a new module for an existing kidney exchange software called ENCKEP. The ENCKEP software has a big role in motivating national KEPS to collaborate in order to improve the quality and quantity of kidney transplants. With the software, the cooperation of different KEPs can be simulated using various optimization criteria. Due to strict data regulations, for the simulations, it is difficult to access real datasets, so there was a generator module developed to be able to generate datasets. Data generation also makes robust investigations possible (e.g. different policies could be compared by generating 1000 or 10000 datasets and running simulations on them with different settings).

To make the simulations reliable, the generated datasets have to be realistic, and it would be very useful if we could verify the datasets' authenticity. This motivated the new module which is the subject of this thesis: to create a tool which is able to generate datasets close to real ones, and the closeness can also be verified.

The developed tool makes it possible to generate datasets close to existing real datasets, and it also investigates the similarities and differences giving the possibility of improving those features of the generated data which differ.

For this, the tool uses statistical analysis. The tool collects aggregate statistics of a real dataset, and as the statistics do not contain protected information, they can be provided to us. We generate a dataset with our existing generator, and after calculating the same statistics on it, the two datasets can be compared. The tool compares the statistics pairwise with the help of statistical tests, and knowing which statistics differ can guide us on how to modify the generation process.

In the thesis, the existing generation process is analysed, the new statistical analysis module is implemented and described, and the possible ways for improving the generator are foreshadowed. The tool is ready for application and it is planned to be used with collaborating countries very soon.

Chapter 1

Introduction

1.1 Motivation

About 10% of the European population suffers from chronic kidney disease [1]. The most effective known treatment for patients affected by end-stage renal disease is kidney transplantation.

One of the biggest challenges of kidney transplantation is finding compatible donors for the patients. If a patient has a relative or friend, who is willing to donate a kidney, still there is a big chance that the transplantation cannot be performed due to medical issues, like blood type mismatch. We call such donor-recipient pairs *incompatible pairs*.

In several countries, there are Kidney Exchange Programs organized, to help these pairs [2]. Incompatible pairs can register to these programs in the hope that they get matched.

But unfortunately, most of these KEPs are developed individually by the different countries, and neither the methods nor the pools are shared, meaning that exchanges are performed only between patients in the same program. The collaboration of the programs would give much better chances, especially for recipients that are difficult to match.

This motivated the European initiative for creating a framework for the collaboration of Kidney Exchange Programs. The project is called ENCKEP (European Network for Collaboration on Kidney Exchange Programmes), and it has already developed a software which is being improved continuously. The software aims to simulate the cooperation of different KEPs using various optimization criteria and

to find the best set of criteria for each case.

The subject of this work is a new module of this software, a tool, which facilitates realistic data generation with the help of statistical analysis.

1.2 The problem

The purpose of the ENCKEP software is to simulate the cooperation of nations and to show the individual KEPs how the number of successful transplants would be affected by their collaboration. The software also gives an opportunity of finding the best setting of the optimisation criteria, which can also give a big improvement in the number of transplanted patients.

Thus we can see that the project aims to draw the attention of the nations to the importance of collaboration. But the authenticity of the results of the software is crucial.

In order to convince the nations about the correctness of our results, the simulations have to be close to reality. Thus, it is fundamental to use realistic datasets.

Due to the strict data regulations, in most cases, we cannot gain access to the real datasets of the collaborating parties. Thus, generating datasets similar to their real data is an important task.

Also, being able to generate realistic datasets would make it possible to perform much more robust investigations. Different policies could be compared by generating 1000 or 10000 datasets and running simulations on them with different settings.

The ENCKEP software already has a module for data generation, but the similarity of the generated data to real datasets is not guaranteed. Thus, the generation process has to be perfected.

This motivates the subject of this work.

1.3 Solution proposal

The aim is to create a module, which helps improve the generation process without the need of accessing real datasets directly. The idea is that aggregate statistics of real datasets contain lots of information, but do not contain any sensitive data. Thus, if our module could collect such statistics, we could use that information to improve the generator.

Collaborating countries do not have to give access to their datasets, it would be enough if they run our module on the data. The module would collect the aggregate statistics, which we can use for our work without violating data regulations.

The module would work the following way: First, it would collect the statistics of the *historical data* (we call historical data, the collaborating countries' real datasets). It would generate a dataset taking some of these statistics into account. Then, the module would calculate the same aggregate statistics of the *generated data*. The aggregate statistics of the two datasets are compared pairwise, and if they are not similar enough, the generation process is modified, and the results are examined again. The process is repeated until an acceptable result is received, meaning that the generated dataset is realistic enough and the way of generation is perfected.

Chapter 2

Literature review

2.1 Medical background

In this section, we detail the medical background of kidney exchange, for which we use the second handbook of Working Group 1 in the COST Action [3].

We are trying to find matches of donors and recipients such that the success rate of their transplants is high. The key to the success is mainly the compatibility of patients. When examining compatibility, we have to consider the blood types of the patients, and certain antigens, namely *Human Leukocyte Antigens* (HLA).

We say that donor and recipient are *ABO-Compatible*, if they satisfy the following rules. Every patient has blood Type A, B, AB or O. Type A donors can donate to Type A and Type AB recipients. Similarly, Type B donors to Type B or Type AB recipients. Type AB donors can donate only to Type AB recipients. And finally, Type O donors can donate to everyone. The Rhesus factor is irrelevant in kidney transplants.

The amount of Human Leukocyte Antigens two patients share determines their *HLA-Matching*. The more antigens they share, the more closely their HLA is matched. Patients whose HLA is closely matched are more likely to have a successful transplant. When the HLA of the patients are fully alike, they have a *perfect HLA-Match*. Note that a perfect HLA-Match is very rare, it occurs e.g. in the case of identical twins. Classically only the most important antigen alleles were compared in the HLA-Matching, but nowadays the examination gets more detailed.

When the HLA-Match of the patients is determined, a second test is done, the

HLA-Crossmatch. This time, it is examined whether the recipient has antibodies to the HLA of the donor. If the answer is yes, then the patients are said to have a *positive crossmatch.* This means that it is very likely that the immune system of the recipient will reject the transplanted kidney. Hence, the desired result for this test is a negative crossmatch. Note that in the case of a perfect HLA-Match, the crossmatch is always negative.

Lately, it is possible to present successful transplants not only among ABO-compatible patients with perfect HLA-match. Desensitization techniques have been developed, to make the immune system less defensive and make it accept the implanted kidney [4]. Even a new definition was created for ABO-incompatible patients with negative crossmatch who need desensitization to overcome ABO-incompatibility. These patients are called *half-compatible.* The outcome of a transplant between half-compatible patients does not differ significantly from the ABO-compatible results.

The comparison of the ABO types and HLA data of the donor and recipient are called *virtual crossmatch* tests. After two patients are said to be compatible due to virtual crossmatch, a *laboratory crossmatch* must be done before transplantation is approved meaning that the blood- and HLA-compatibility must be checked in laboratory as well.

But unfortunately, HLA data of the patients is not always available. In these cases, a parameter, called *PRA value* can give information about the patient's sensitivity. PRA is the short form of Panel-reactive antibody. Each recipient has a PRA value, which is an integer number between 0 and 100, representing the recipient's HLA-sensitivity in percentage. It is an estimate of the number of donors that will be crossmatch-incompatible for the recipient [5].

If the HLA data of a recipient is not available, it cannot be decided for sure whether she is compatible with a given donor. In this case, the PRA value of the recipient is used for giving a realistic estimate of the compatibility probability.

Originally PRA is tested in laboratory from the recipient's blood, but nowadays it is common that it is simply calculated by computer programs from the HLA data of the recipient. The calculated PRA is usually called *cPRA*. In this work, we only work with this calculated value, thus using any of the expressions PRA and cPRA, we mean the calculated value.

2.2 Kidney Exchange Programs

To a Kidney Exchange Program, mostly incompatible pairs register, where they are added to the *pool*, to the set of patients.

Usually incompatible pairs register, but half-compatible or even compatible pairs can also occur, as they can try to find a more compatible option.

It is also possible for donors to register alone. We call such donors *altruistic donors*. As in every other case, altruistic donors are also registered to the pool as a pair. When giving the data of the pair, the recipient's data parameters are set to -1 and it is indicated in the pair type parameter that it is an altruistic pair.

Furthermore, there can be recipients with more than 1 donor. In such cases, there are more pairs in which the same recipient is present.

Graph model

To be able to solve a problem, it is important to formulate it mathematically. For the formulation of the matching problem of Kidney Exchange Programs, usually, a graph model is defined. We describe the model based on the 'Modelling and Optimisation in European Kidney Exchange Programmes' Handbook [3].

For each KEP a graph is defined, where the vertices are donor-recipient pairs, so the related patients (original pairs) are in one block. E.g. vertex v_i actually contains the d_i–r_i pair, where d_i is the donor and r_i is the recipient of the same pair. The graph is directed, and there is an arc from vertex v_i to v_j, if d_i is compatible or half-compatible with r_j. It is also possible that an original pair is compatible or half-compatible, as they can still hope to get a better transplant option. In such cases, there is a self-loop at the vertex of the pair.

Figure 2.1 shows an example of a compatibility graph of three donor–recipient pairs. The solid line denotes compatibility, while the dashed lines mean half-compatibility. The self-loop denotes the compatibility of the original pair.

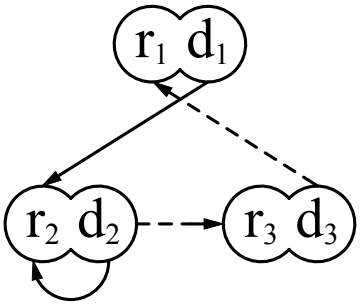

Figure 2.1: The compatibility graph of recipients r_1, r_2, r_3 with related donors d_1, d_2, d_3. Recipient r_2 is compatible with donor d_1, $r_2 - d_2$ are also compatible, and $r_3 - d_2$, $r_1 - d_3$ are half-compatible. Any other two patients are incompatible.

Match runs

In the pool, the task is to find matches. When a new patient joins a pool, her medical data are registered to the system and she will already take part in the next match run. Match runs are executed periodically (e.g. every three months).

A match run gives exchange possibilities as a result. These possibilities have to be checked in laboratory before the exchanges are performed. The operated pairs leave the pool.

Several criteria and policy can be used for the match runs, which is chosen by the country/user. E.g. if in a country the maximum length of cycles is 3, then in the match run, longer cycles will not be considered.

2.3 ENCKEP

ENCKEP is a network supported by the European Cooperation in Science and Technology (COST), which is a funding organization for the creation of research networks, called COST Actions. These networks offer an open space for collaboration among scientists across Europe and give support to research advancements and innovation [6].

The aim of ENCKEP is to exchange the best KEP practices, develop prototype software for achieving transnational KEPs and to stimulate European policy dialogue [6, 7].

2.4 Statistical tests

Comparing the historical and generated datasets is a crucial task in development. For this, the same statistics are created about the two datasets and these statistics have to be compared pairwise. For this comparison, we call for the help of statistical tests.

But which tests are feasible for us? Knowing our data, a statistical test is good for us if it satisfies the following criteria:

- It can be used for two different samples from different populations.
- Suitable for samples of the same size.
- Suitable for small sample size.

Moreover, we will need different methods for comparing

1. columns containing numbers
2. discrete/categorical distributions
3. continuous distributions.

We looked for commonly used methods which satisfy our conditions. For the 3 cases listed above, we found the following methods.

1. T-test

For comparing columns containing numbers, the *Independent two-sample t-test* seems to be the most suitable [8]. This test compares the means of two groups and helps to decide whether there is a statistically significant difference between them.

This test is based on hypothesis testing, which means that a claim (the null hypothesis) is set, and it is examined whether it is true or not. We use the two-sided (two-tailed) type of the test because we do not know the direction of the possible deviation in advance. Thus, the null hypothesis will be that the two means (the means of the two samples) are the same, and the alternative hypothesis will be that they are different.

The test is based on the t-statistic, which is calculated by the following formula:

$$t = \frac{\overline{X_1} - \overline{X_2}}{s_p \cdot \sqrt{\frac{2}{n}}},$$

where

$$s_p = \sqrt{\frac{s_{X_1}^2 + s_{X_2}^2}{2}}.$$

In the formulas, X_1 and X_2 denote the two samples, n is the size of both of them, and $s^2_{X_1}$ and $s^2_{X_2}$ are the unbiased estimators of the variances of the two samples. We call s_p the pooled standard deviation.

The test works the following way. We calculate the t-statistic with the above-defined formula. Then we choose the significance level. After this, we use the t-table, the table which contains the quartiles of the Student's t-distribution for different significance levels. In the t-table, we will find the critical t-value, which will help in the decision. If the calculated t-statistic is smaller than the critical t-value, then we accept the null hypothesis and reject it otherwise.

The test has the following conditions. It is only applicable when:

- there are two samples
- the samples have Normal distribution
- the two sample sizes are equal
- it can be assumed that the two distributions have the same variance.

These conditions are met in our case.

2. Chi-Square test

For comparing discrete/categorical distributions, we decided to use the *Chi-Square two-sample test* [9]. This test is usually used for testing the relationships between categorical variables (on binned data).

In the case of this test, we have two samples and the binning has to be the same on both of them. The question is whether the two data samples come from the same distribution. The test decides this question by checking whether the observed number of points in each bin is similar.

Thus, the null hypothesis is the following: The two samples come from a common distribution. And the alternative hypothesis is the denial of this.

For the test we use the Chi-Square statistic, for which we have the following formula:

$$\chi^2 = \sum_{i=1}^{k} \frac{(K_1 R_i - K_2 S_i)^2}{R_i + S_i},$$

where k is the number of bins, R_i is the observed frequency for bin i for sample 1, and S_i is the observed frequency for bin i for sample 2. K_1 and K_2 are scaling constants to compensate for unequal sample sizes.

Similarly to the t-test, here we also work with a p-value. We count the number of points in the bins. If the difference between expected and observed counts is large, the p-value will be small, and the null hypothesis will be rejected.

3. Kolmogorov-Smirnov test

A commonly used method for comparing continuous distributions is the Kolmogorov-Smirnov test, thus we will use the *Kolmogorov-Smirnov two-sample test* for our two-sample case [10], [11].

With this test, we can check whether the two data samples come from the same distribution. For the test we use the following statistic:

$$D = max_{1 \leq i \leq N} |E_1(i) - E_2(i)|,$$

where E_1 and E_2 are the empirical distribution functions for the two samples.

The null hypothesis is that the two samples come from a common distribution. From the Kolmogorov-Smirnov table, we can gain the critical value, which will help us decide. The null hypothesis is rejected if statistic D is greater than this value.

2.5 The software

In this section, the modules of the ENCKEP software will be detailed based on the relating Handbook [12].

2.5.1 Input reader module

In lucky cases, the ENCKEP software can receive historical datasets from collaborating parties. This means that countries with kidney exchange activity make available their full or partial datasets from the past, which can be used in the simulator module for simulation.

The Input reader module was created to handle the received historical datasets. After reading the data, it can do data generation in case of missing values (similarly to the Generator module), and it creates the compatibility graph by calculating the arcs between the patients.

2.5.2 Generator module

In case there is no historical dataset available, the Generator module can be used to generate datasets for the simulations.

This module has similar output files, as the Input Reader module, as they both provide input files for the Simulator.

The generation happens based on distributions, which can be specified before each generation and can be provided in a configuration file. Figure 2.2 shows an example of the configuration file. The elements of the configuration file will be described in Section 2.5.4.

```
{
    "name"            : "pool_1",
    "dist_blood_type" : { "0":0.45, "A":0.43, "B":0.09, "AB":0.03 },
    "dist_num_donors" : { "1":0.95, "2":0.05 },
    "dist_pra"        : { "[0, 9]":0.64, "[10, 79]":0.27, "[80, 99]":0.09 },
    "dist_age"        : { "patient"    : { "[18, 73]":1.0 },
                          "donor"      : { "[18, 73]":1.0 },
                          "altruistic" : { "[18, 73]":1.0 } },
    "arrival_time"    : { "inc": { "avg":4.98 },
                          "com": { "avg":-1.0 },
                          "alt": { "avg":60.0 } },
    "duration_stay"   : { "inc": { "avg":365.0, "min":60, "max":2190 },
                          "com": { "avg":365.0, "min":60, "max":2190 },
                          "alt": { "avg":180.0, "min":60, "max":240  } },
    "prob_fail"       : { "inc": { "max":0.05 },
                          "com": { "max":0.05 },
                          "alt": { "max":0.05 } }
}
```

Figure 2.2: An example for the configuration file. It contains distributions of the relevant parameters of patients and pairs. E.g. PRA is a medical parameter of recipients (its meaning was described in Section 2.1). In this example, 64% of the recipients have a PRA value uniformly distributed between 0 and 9, 27% falls between 10 and 79, and the remaining are in the [80,90] interval.

2.5.3 Simulator module

The Simulator module is responsible for simulating the operation of the Kidney Exchange Program. The implementation and testing of the optimisation criteria are done here.

First, the module reads the input files containing the pool and patient information. Also, a policy file can be provided, and the criteria can be chosen, which will be considered in the run.

When running the module, it gives an optimal solution as a result.

2.5.4 Data description

As we mentioned above, the input of the Generator module is the configuration file. This file contains the following data:

- **name**: The name of the pool which we generate.
- **dist_blood_type**: Blood type distribution. The blood type generation of donors and recipients will happen based on this distribution.
- **dist_num_donors**: Distribution of the number of donors per recipient. A recipient can have more donors, and the number of donors for each recipient will be generated based on this distribution.
- **dist_pra**: PRA distribution of the recipients. The PRA value of the recipients will be generated based on this distribution.
- **dist_age**: Age distribution. The age of the donors and recipients will be generated based on this distribution.
- **arrival_time**: Arrival time distribution. It is important to know the day of a pair's arrival because it helps us to determine in which match runs they take part. The arrival time of the pairs will be generated based on this distribution.
- **duration_stay**: Distribution of the duration of stay in the pool. Similarly to the arrival time, it is important to know how long the pair stays in the pool. The number of days the pair stays in the pool will be generated based on this distribution.
- **prob_fail**: Distribution of the probability of failure. The probability of the failure of a pair is an important parameter because it gives the probability that the pair fails and the transplantation cannot be performed due to the pair's withdrawal or other issues. The probability will be generated based on this distribution.

The input of the Input reader is very similar to its output, thus we describe only its output files which contain additional parameters.

Either the Input reader or the Generator module provides data for the simulation the same 4 CSV files are created. These serve as the input of the Simulator module.

The files are the following:

- pairs.csv
- arcs.csv
- failed_pairs.csv
- failed_arcs.csv
- hla.csv

Pairs

This file contains information of the pairs in the pool. The data parameters in each row of the file are as follows:

0 *Pair ID*
1 *Donor ID*
2 *Donor blood type*
3 *Donor age*
4 *Patient ID*
5 *Patient blood type*
6 *Patient PRA*
7 *Patient crossmatch probability*
8 *Patient age*
9 *Pair arrival day*
10 *Pair departure day*
11 *Pair type*
12 *Reason for incompatibility*
13 *Pair probability of failure*
14 *Pair from pool*
15 *Pair original ID*
16 *Donor original ID*
17 *Patient original ID*

As it was described in Section 2.1, the *PRA value* can give information about the patient's sensitivity. This parameter is an integer number between 0 and 100, representing the recipient's HLA-sensitivity in percentage. It is an estimate of the number of donors that will be crossmatch-incompatible for the recipient [5].

The *Patient crossmatch probability* is a parameter that gives the probability that a pair whose virtual crossmatch result is negative (said to be compatible), will have a positive laboratory crossmatch (said to be incompatible). This parameter is

calculated from the PRA value of the recipient. This parameter helps to calculate the failed arcs in the compatibility graph, that is the arcs which were created due to compatibility, but later the transplant failed due to withdrawal or other issues.

In the *Pair from pool* parameter the name of the pool can be given, e.g. country to which the pair belongs.

Pair arrival day and *Pair departure day* are integer numbers indicating the duration of the pair's presence in the pool such that $0 \leq arrival < departure \leq last\_day$, where $last\_day$ means the day of the last match run.

The *Pair type* can take two values: incompatible or compatible.

And the reason for the incompatibility can be given in the next parameter. The *Reason for incompatibility* parameter has four possible values: *, ABO, HLA, ABO-HLA, where * means compatibility.

The *Pair probability of failure* parameter gives the probability of transplant failure due to the pair's withdrawal or other issues.

Arcs

This file contains the virtual compatibility graph of the pairs of the pool. The data parameters in each row of the file are as follows:

0 *Pair ID of Donor Pair*
1 *Pair ID of Recipient Pair*
2 *Arc weight*
3 *ABOi value*

A row in this file means that the donor of the first pair is virtually compatible with the recipient of the second pair.

The *Arc weight* is always set to 1.0, because no optimisation objective is known at the stage of generating historical data. The *ABOi value* shows whether the pair is ABO-incompatible or not. It can take value 0.5 in case the exchange is believed to be half-compatible or value 1.0 in case of full compatibility.

Failed pairs

This file contains the IDs of the pairs which fail due to the pair's withdrawal or other issues. Each row contains a pair ID of a failed pair.

Failed arcs

Recall how the virtual compatibility graph was built: a Donor Pair and a Patient Pair created an arc if they were said to be compatible due to the virtual crossmatch test, and in such a case, they created an arc. This file contains those arcs which fail in the real crossmatch test, which is done if an arc was chosen for transplantation. The data parameters in each row of the file are as follows:

0 *Pair ID of Donor Pair*
1 *Pair ID of Recipient Pair*

HLA data

In case HLA data is not provided by the user, this file is empty. Otherwise, when the HLA data of the pairs is available, the data parameters in each row of the file are as follows:

0 *Pair ID*
1 *Donor HLA antigens*
2 *Patient HLA antigens*
3 *Patient HLA antibodies*

The HLA antigens and antibodies are expected to have format

$$type * number|type * number|\ldots|type * number,$$

where type can be A, B, C, DRB1, DQB1, DQA1, DPB1, DPA1 or DRB3/4/5 and after a star, a number is present, which specifies the allele.

Remember that the format of the containing file is CSV, that is the parameters are separated by commas. As an example, a row looks the following:

$$0, A*50|A*02|B*30|C*50|DRB1*15|DQB1*01, A*15|B*12|C*10|DQB1*51, A*11$$

Chapter 3

Solution

3.1 Parameter checker objective

To the problem of generating perfectly realistic datasets, we offer a solution by developing a tool, called the *Parameter checker*.

The basic idea of the tool is that it needs only aggregate statistics of a historical dataset. Most of the partner countries are not allowed to provide their real datasets, but they may well be allowed to give us these aggregate statistics.

The tool uses some of these statistics for creating the configuration file with the required distributions for the data generation.

The remaining statistics, which cannot be considered in the generation process, help in the investigation of the generated dataset. The same statistics are calculated for the generated dataset, and the two results are compared.

If the statistics about the historical and the generated dataset do not differ significantly, the way of generation is proper, and the generated data reflects the reality. Otherwise, the way of the generation has to be modified. The parameter with the bad results has to be investigated: we can try to modify the corresponding distribution in the configuration file, or do modifications in the generator. Afterwards, the statistics have to be calculated again, and the comparison also has to be done again. The process is repeated until we gain an appropriate result.

3.2 The statistics

The investigated parameters can be separated into two groups.

We call Pre-Statistics those statistical indicators that can be determined without the Simulator module and can be considered in the generation process.

While Post-Statistics are those statistical indicators that cannot be considered in the generation process and most of them cannot be determined without the simulator module.

Pre-statistics

As it was described in Section 2.5.2, the data generation happens with the help of the configuration file, which contains distributions regarding the data parameters.

Pre-Statistics contain statistics from which these distributions can be calculated, and the configuration file can be created for the generation.

Thus, the following statistics are considered:

- blood type distribution
- distribution on the number of donors for recipients
- recipients' PRA distribution
- age distribution
- arrival time distribution
- probability of failure distribution
- duration of stay distribution

Post-statistics

Other statistics, which give useful information about the datasets, but cannot be considered in the generation process, are included in the Post-Statistics.

In the Post-Statistics the following parameters are examined.

For each match run:

- Number of cycles
- Weights of cycles
- Number of transplanted pairs
- Pool size
- Number of 2-cycles, 3-cycles, 4-cycles, 5+
- Number of compatible pairs
- Number of altruistic donors
- Number of possible transplants (edges)

For pairs:

- Waiting time of the patients
- Donors' compatibility distribution (Out-degree of each pair(node))
- PRA distribution of the informed pairs
- Bloodtype distribution of the compatible pairs
- Bloodtype distribution of the incompatible pairs
- Distribution of the number of 2-cycles and 3-cycles for the pairs(nodes)

3.3 Flow diagram

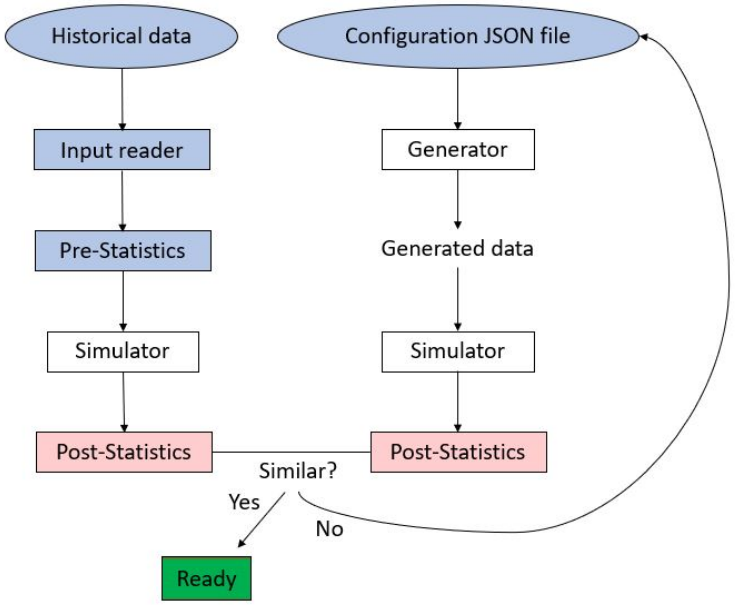

Figure 3.1: Process of Parameter checker

Figure 3.1 captures the process of the Parameter checker.

The process can be separated into two parts: the historical part on the left-hand side of the diagram (LHS), and the generation part on the right-hand side (RHS).

For the LHS process, a historical dataset is needed. If the dataset is protected, the Parameter checker tool can be run by the owner of the dataset so that the data will not be seen by anyone unauthorized.

The Input Reader module is used to read the historical dataset and it also takes care of missing data. It uses the default configuration or the country's own con-

figuration provided alongside the historical data to generate a replacement for the missing values.

After reading and completing the data, the Pre-Statistics (see Section 3.2) are calculated, which actually contain all the information needed for the configuration file for the generation part (RHS). We named this connection between the two parts (LHS and RHS) the Configuration creation segment, about which you can read in Section 3.4.

As the Input reader has converted the data to the correct format, the Simulator module can be run next. This gives the optimal solution (best matching of pairs) and several new information about the match run so that we can create more statistics: Post-Statistics. Post-Statistics are detailed in Section 3.2.

For the RHS process, the configuration file is created with the help of the Pre-Statistics from the LHS. Read about the creation of the configuration file in Section 3.4.

The Generator module needs the created configuration file as an input to generate data. This generated data has the same format as the data on the LHS after reading it with the Input reader.

There is no need of calculating the Pre-Statistics here because we used the LHS Pre-statistics for the generation which means that we would get the same result.

The generated data are in the correct format for being the input of the Simulator, so the module can be run and the Post-Statistics can be calculated.

In Section 3.5, the Comparison segment deals with the comparison of the Post-Statistics of the LHS and RHS, which is the key point of this thesis.

This is followed by the Generation modification segment in Section 3.6, where we investigate the possible steps for improving the generation process.

3.4 Configuration creation segment

In this segment, there is a connection between the LHS and the RHS. For the data generation on the RHS, the configuration file is created from the Pre-Statistics of the LHS. The Pre-Statistics intentionally include only those distributions that are needed in the configuration file (see the configuration file in Figure 2.2 and the elements of the Pre-Statistics in Section 3.2), all the other statistics are included

by the Post-Statistics. That is why we only need to transform the Pre-Statistics to the appropriate format and write them to a JSON file to which the proper name is given so that the generator can be run automatically with this input file.

But why do/can we use the LHS for our RHS process? This step saves us an unnecessary round: In the first step, we could generate a dataset with a random configuration setting. However, after this step, we would act similarly as in the case of the Post-Statistics, calculate the statistics and compare them. If they differ, we modify the generation process until they become similar. Common sense tells us how to modify the configuration file in the next step: use the same distributions which was calculated in the Pre-Statistics of the LHS. That is why we started with these distributions originally.

3.5 Comparison segment

The purpose of the Parameter checker tool is to perfect the data generation process, and we measure the goodness of the process by comparing the Post-Statistics of the LHS and RHS. Figure 3.2 shows which part of the Flow diagram is the comparison segment.

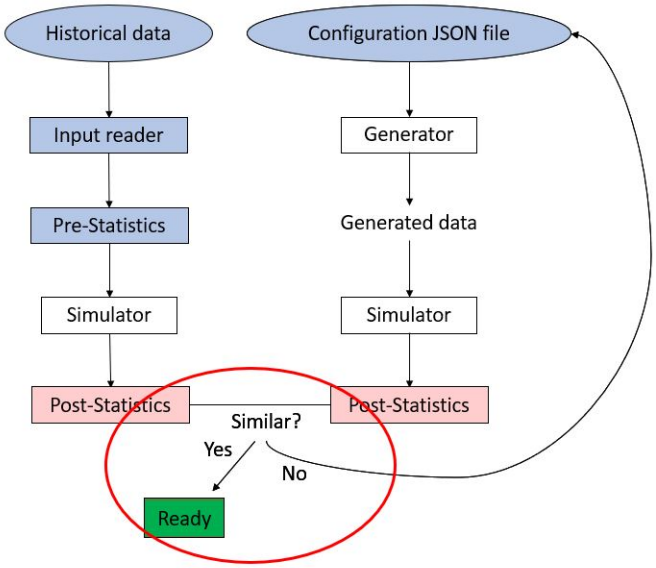

Figure 3.2: Comparison segment

Our comparison technique is based on our research about statistical tests (see the literature review in Section 3.2). We examined the way of comparing statistics and we found that due to the different kinds of statistics we have in the Post-Statistics, three different statistical tests are needed for the comparison: T-test, Chi-square test, and Kolmogorov-Smirnov test. These tests are suitable for the comparison of the Post-Statistics, as

- we have 2 different samples from different populations (Post-Statistics of the LHS and RHS)
- the samples are of the same size (generation on the RHS happens for the same time interval as the historical data has on the LHS)
- we have a small sample size.

The mathematical background of the tests is explained in Section 3.2.

If we check the content of the Post-Statistics (see in Section 3.2), we find that we have 3 kind of statistics for which we use the 3 statistical tests. The type of statistics paired with the used tests are as follows:

- Columns containing numbers (e.g. the number of compatible pairs): **T-test**
- Discrete/categorical distributions (e.g. the blood type distribution of the compatible pairs): **Chi-Square test**
- Continuous distributions (e.g. the PRA distribution): **Kolmogorov-Smirnov test**

Figure 3.3 shows which test belongs to which statistic.

3.6 Generation modification segment

After comparing the statistics of the LHS's and RHS's Post-Statistics, we might need to modify the generation process. Figure 3.4 shows the result of the Comparison segment, which helps us to find out which part of the generation needs modification.

The idea is to modify the generation until we get similar results for all the statistics, that is the Result column of Figure 3.4 contains value 'similar' only.

We would like to ask some countries to run the parameter checker and give us the output statistics. We can compare their statistics with the statistics of our generated dataset, and do modifications to the generator to improve its authenticity.

Unfortunately, the collaboration with countries is a slow procedure, and we did

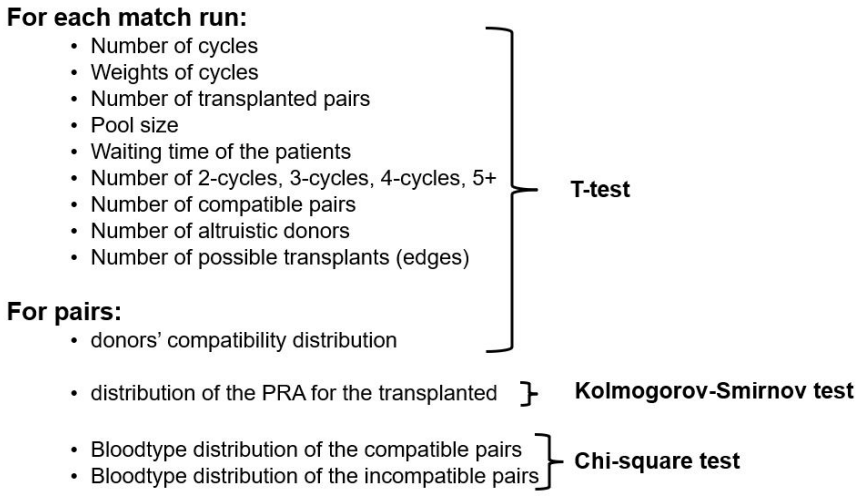

Figure 3.3: Statistics paired with the used tests

	historical	generated	p-value	Result
cycles_runs	1,25	3,25	0,03	DIFFERENT
cycle_weight_runs	3,38	8,62	0,04	DIFFERENT
transplanted_runs	2,50	4,75	0,23	similar
pool_size_runs	14,33	14,00	0,92	similar
waiting_time	286,00	216,17	0,05	similar
2_cycles	0,38	1,12	0,05	similar
3_cycles	0,88	2,12	0,11	similar
4_cycles	0,00	0,00	nan	similar
more_cycles	0,00	0,00	nan	similar
out_degree_dist	14,77	19,72	0,02	DIFFERENT
2_cycle_pair	0,25	1,66	0,02	DIFFERENT
3_cycle_pair	0,75	10,20	0,05	similar
4_cycle_pair	0,00	0,00	nan	similar
arc_pair	4,21	10,76	0,01	DIFFERENT
compatible_pairs	1,88	3,75	0,11	similar
altruist_pairs	0,00	0,00	nan	similar
pra_distribution	24,78	21,76	0,57	similar
BT_compatible			1,00	similar
BT_incompatible			0,97	similar

Figure 3.4: Example of the output of the tool: result of comparison

not receive any output statistics yet. Without these, we cannot determine the appropriate generation modification steps. Therefore, we will show only some examples of the process.

Before that, we introduce some literature on the ENCKEP generator and also some other generators to see alternative generation processes and show some ideas on how can one modify the process of a generation.

3.6.1 Literature review: Generators

For the generation modification, we need to get to know our generator better and do some research about other generation principles. Thus, we analyse the ENCKEP generator along with two well-known generators.

Generally, the aim of these generators is to generate instances of a Kidney Exchange pool. These instances are donor-recipient pairs. As compatible pairs are very rare in Kidney Exchange Programmes, some of the generators create only incompatible pairs, but some of them allow compatible pairs too.

Saidman generator

First, we would like to describe the widely used Saidman generator. For this, we use the article about Matching for Two- and Three-Way Exchanges [13] and the article about the Delorme generator which gives a nice introduction about Saidman in its literature review [14].

The Saidman generator takes the following input parameters:

- number of donors and recipients (**n**)
- distribution of donor and recipient blood-group (**BT**)
- distribution of the number of donors per recipient (**dist_num_don**)
- distribution of cPRA (**cPRA**)

Pairs are generated until there are n incompatible pairs. The generation happens as follows.

1. **Generating a recipient.** This is done by randomly drawing BT, dist_num_don and cPRA parameters.
2. **Generating a donor.** This is done by randomly drawing a BT parameter.
3. **Compatibility check.** If the generated donor and recipient are found to be compatible, they are discarded. Both the recipient and the donor are discarded because keeping the recipient would result in an unrealistic pool.

There are two important factors about this generation method which should be highlighted:

1. The donor BT, recipient BT and recipient cPRA are assumed to be independent.

2. Discarding compatible pairs causes the distribution of the generated data to be not equal to the input distributions. E.g. as recipients with low cPRA are more likely to be compatible with their donor, they are discarded more often. Thus, low-cPRA recipients are under-represented in the generated data.

For this second problem, the solution could be

- tuning the input parameters after seeing the results. E.g. seeing that the low-cPRA recipients are under-represented, the probability of their occurrence in the input distribution should be increased.
- checking several input distribution settings and choosing the one which generated the most realistic dataset.

This solution proposal takes us to the next analysed generator.

Delorme generator

The Delorme generator tunes all the input parameters of the Saidman generator and tries to generate data with a distribution closer to the input distribution. The article about Delorme generator [14] also gave us a good understanding of the Saidman generator, and the way it was modified to maintain this new type of generator.

The following assumptions of the Saidman generator are investigated:

1. The independence of donor BT, recipient BT and cPRA.
2. Direct linear relationship between cPRA and donor compatibility.
3. Equivalence of all recipients with given cPRA value.

The Delorme article argues against these assumptions:

1. **The independence of donor BT, recipient BT and cPRA.**
    - It draws attention to the fact that the cPRA value of BT-compatible pairs is higher than in the case of BT-incompatibility. Thus, it would be important to use different cPRA distributions in the case of BT-compatible and BT-incompatible pairs.
    - It proposes to calculate the BT-compatibility as follows. First, calculate the recipient BT distribution. Afterwards, calculate 5 pieces of donor BT distribution, different distributions for the 4 possible recipient blood types (O,A,B,AB) and 1 for altruistic donors.

2. **Direct linear relationship between cPRA and donor compatibility.**
    - Transplant compatibility is not only determined by cPRA, but by several external factors like the size of the kidney or the age of donor. Also, paediatric recipients may have stricter compatibility requirements. Thus, instead of calculating the probability of compatibility the original way, they found a formula which gives more realistic results. The formula can be obtained by applying a linear least-squares model weighted by the distribution of cPRA amongst the recipients.
    - The original way: $\mathbb{P}(\text{compatibility}) = 1 - \frac{cPRA}{100}$
    - The formula of Delorme article: $\mathbb{P}(\text{compatibility}) = 0.58 - 0.55 \times \frac{cPRA}{100}$
    - There were experiments for finding alternatives for this formula, they compare a few formulas in the article.
3. **Equivalence of all recipients with given cPRA value.**
    - Recipients with cPRA value 0 should be compatible with all of their BT-compatible donors by definition. But the observations show that it is not true in reality. The following was experienced:
    - Approximately 18% of the cPRA 0 recipients were found to be compatible with **all** of their BT-compatible donors.
    - Approximately 40% of the cPRA 0 recipients were found to be compatible with **at least 95%** of their BT-compatible donors.
    - And almost 26% of the cPRA 0 recipients were found to be compatible with **less than 25%** of their BT-compatible donors.
    - To modify the generation, the determined distribution from which we picked the above-listed facts, can be used for calculating the compatibility of cPRA 0 recipients. This is what the Delorme generator does.

ENCKEP generator

We can find similarities between the above-described generators and the ENCKEP generator, but it also does several things differently. The way the Saidman generator was modified to obtain the Delorme generator is a good example for us, as we also try to do modifications to a generator, just this time, on the ENCKEP generator.

Compared to Saidman, the ENCKEP generator takes more input parameters. These are the following:

- name of pool
- blood type distribution

- distribution of the number of donors per recipient
- PRA distribution
- age distribution
- arrival time distribution
- distribution of the duration of stay in the pool
- distribution of the probability of failure

These parameters are collected in the configuration file which is the input of the generator. An example of the configuration file is in Figure 2.2.

The ENCKEP generator also allows compatible pairs. In the generated dataset the type of the pair will be stored, it can take value "incompatible", "compatible" and "altruistic". The distribution of these types is defined by the user in the arrival_time configuration element. If there is no need for compatible pairs, their arrival time parameter is simply set to -1 meaning that they never arrive. See an example below.

```
"arrival_time"    : { "inc": { "avg":4.98 },
                      "com": { "avg":-1.0 },
                      "alt": { "avg":60.0 } }
```

To generate the Pair arrival time of pairs, the generator uses Poisson distribution with these average arrival frequencies of the arrival_time parameter. This means that we do not know the exact number of generated pairs in advance. This is a difference again compared to the Saidman generator, as they generate a fixed number of pairs and do not use Poisson distribution.

To generate the Pair departure day for the pairs, a duration of stay value is added to the Pair arrival time. That is an exponential distribution is applied using the average duration given in the duration_stay parameter of the configuration file. The minimum and maximum values provided in the duration_stay parameter are used to truncate the exponential distribution.

All the other parameters of a pair (e.g. blood type of recipient and donor) are generated simply from the corresponding distribution of the configuration file.

More specifically, the generation happens the following way:

- **Generate arrival times.**
  The arrival_time parameter of the configuration file contains the average number of days passing between arrivals separately for each pair type. For each pair type, the arrivals follow Poisson distribution. We know that Poisson dis-

tribution deals with the number of occurrences in a fixed period of time, and the time intervals between those occurrences follow Exponential distribution. We generate these Exponential distributions and put them one after the other. This way we get Poisson distribution, where the fixed period of time is the duration of the pool, and the occurrences are the arrivals.

- **Generate the number of donors for incompatible pairs.**
  The dist_num_donors parameter of the configuration file contains the distribution of the number of donors for incompatible pairs. In the case of the other pair types, this parameter does not make sense.
- **Generate pairs and assign arrival times (previously generated) to them.**
  In this step, we examine the generated arrival times. We count how many incompatible, compatible and altruistic arrival times were generated. These counts will determine how many pairs to generate of each type.
  - **When generating altruistic pairs**, we simply create a pair with no recipient. This time only those parameters are generated which correspond to the donor.
  - **When generating compatible pairs**, we start generating pairs (generate all the needed parameters: donor BT, recipient BT, recipient PRA, donor age, recipient age, pair_prob_fail). After generating a pair, we check if it is compatible. If not, we discard the pair. We stop generating when the desired number of compatible pairs are created.
  - **When generating incompatible pairs**, we start by generating a recipient by generating all of its parameters. It also has a parameter called num_don, which determines the number of donors belonging to this recipient. We generate the proper number of donors and check if all the generated donors are incompatible with the recipient. If not, then we discard all of the donors and the recipient. Otherwise, they are kept. We repeat this process until the desired number of incompatible pairs is not created.

3.6.2 Generation modification

After reviewing the generators, and getting a better understanding of our generator, we show some examples for modifying the generation process.

The result of the Comparison segment is a table containing information about

the Post-Statistics. It is determined for all of its elements whether it is 'different' or 'similar'.

To be able to rethink the generation process, we need to deeply understand the output statistics. Most of the statistics are extracted from the result of the simulator. The simulator determines the optimal solution in every match run. There can be cases where there is more than one optimal solution. But even then, one will be chosen, which is the final result. The pairs in the optimal solution are called *informed pairs*. These pairs are chosen for transplantation, so if the laboratory tests also succeed, and no other problem occurs, they can be transplanted. As we are working with historical datasets, we also know if the transplant is realised or not. Thus, in the output statistics, we distinguish the informed and transplanted pairs, where the transplanted pairs are the subset of the informed pairs.

In the following, we define the output statistics. There are statistics which examine one match run at a time, and they are calculated for each match run. And there are statistics where we concentrate on the pairs, and there the match runs are not separated, a statistic considers all the results of all the match runs as a whole. This means that there can be edges considered that never existed, as the pairs were never in the same pool (at the same time). But it is possible to consider them, as we are working with historical datasets. Why can it be useful to consider such edges? Because with this method we can increase the sample size significantly, and there are statistics which gave a very small sample size without it, thus, statistical tests would not work properly on them.

The definitions are as follows.

For each match run:

- **Number of cycles:** The number of cycles (regardless their length) in the optimal solution of a match run.
- **Weights of cycles:** The sum of the weights of the cycles in the optimal solution of a match run. The weight of a cycle is the sum of its edge weights, which are 1 by default.
- **Number of transplanted pairs:** The number of successfully transplanted pairs, who were selected for transplantation in one of the match runs.
- **Pool size:** The size of the pool for which the match run is executed.
- **Number of 2-cycles, 3-cycles, 4-cycles, 5+:** The number of cycles of the given length in the optimal solution of a match run.
- **Number of compatible pairs:** The number of compatible original pairs

(nodes) who are selected for the optimal solution in a match run.

- **Number of altruistic donors:** The number of altruistic donors who are selected for the optimal solution in a match run.
- **Number of possible transplants (edges):** As every edge means a possible transplant, this statistic is the number of edges in a match run.

For pairs:

- **Waiting time of the pairs:** The number of days a pair spent in the pool. A pair leaves the pool when she is transplanted, or for other reasons without a transplant.
- **Donors' compatibility distribution (Out-degree of each pair(node)):** For each pair (who participated in any of the match runs), we count the number of recipients the donor is compatible with.
- **PRA distribution of the informed pairs.**
- **Bloodtype distribution of the compatible pairs:** The distribution of the compatible original pairs among the informed pairs.
- **Bloodtype distribution of the incompatible pairs:** The distribution of the incompatible original pairs among the informed pairs.
- **Distribution of the number of 2-cycles and 3-cycles for the pairs(nodes):** The average number of 2-cycles/3-cycles pairs participated in any of the match runs.

As an example, let us suppose that the **pool size** comes out to be different. Then, it is very likely that increasing or decreasing the average arrival frequency of pairs, which can be set in the arrival time distribution in the configuration file, would solve the problem.

If the **number of possible transplants** also differs, the probability of the compatibilities should be modified. If the difference is because there are fewer possible transplants in the generated dataset, then we should achieve to have more edges (more compatibilities). For this, a solution can be the tuning of the PRA distribution in the configuration file: it is expected that increasing the number of low-PRA patients (and decreasing the number of high-PRA patients) increases the possibility that a patient is compatible with a donor, which increases the expected number of edges.

Although, it is not that easy that for each differing statistic we just modify one

of the input distributions and that solves the problem. Modifying a distribution can affect several other statistics possibly spoiling them. Thus, it is recommended to examine the whole generation process (possibly modify the generator's functions), aiming to get fewer statistics to come out to be different so that only a few input distributions have to be tuned. In this, seeing comparison results on several countries' statistics can help a lot. One example of this generation process modification is the PRA's connection to compatibility. Delorme [14] points out that the supposed linear relationship between PRA and compatibility is probably wrong. We can also consider separating blood type distribution for donors and recipients.

3.7 Technical review

The Parameter checker tool is implemented in Python. As it can be seen in the Flow diagram, it uses some earlier implemented modules of the ENCKEP software to apply statistical analysis on their results. For the statistical analysis the *scipy* was the most important Python library we used, because its sub-package the *stats* contains the functions for the statistical tests: *stats.ttest_ind* for the T-test, *stats.kstest* for the Kolmogorov-Smirnov test, and *stats.chisquare* for the Chi-square test.

Summary

The aim of this work was to develop a new module for an existing kidney exchange software, called ENCKEP. The role of the new module is to improve the data generation process (the existing generator module).

For this, the tool uses statistical analysis. As in most cases, collaborating countries are not allowed to provide their real datasets due to strict data regulations, the tool calculates statistics on their datasets which can be provided without violating the rules.

The tool would collect aggregate statistics of the real dataset (historical data), and the same statistics of the generated dataset (created by the generator module). The statistics of the two datasets are compared pairwise, and if they are not similar enough, the generation process is modified, and the results are examined again. The process is repeated until an acceptable result is received, meaning that the generated dataset is realistic enough and the way of generation is perfected.

The module is separated into segments, and their connection and the pipeline are captured in the flow diagram. The statistical analysis happens in the comparison segment. For comparing statistics, statistical tests were used, for which the mathematical background is described in the literature review. In the next segment, the possibilities for the generation modification are investigated, for which also a thorough literature review had to be done.

The module can be used as soon as a collaborating country is willing to run the tool on their dataset and provide the results. If this happens, we will do the necessary steps to modify the ENCKEP generator so that it will be able to generate similar datasets. This scenario seems to be realised very soon, as Spain and some other countries show great willingness to collaborate and run the tool.

Bibliography

[1] Dr. Bettina Albers. Chronic Kidney Disease (CKD): A Challenge for European Healthcare Systems. 2019.

[2] Péter Biró, Bernadette Haase-Kromwijk, Tommy Andersson, Eyjólfur Ingi Ásgeirsson, Tatiana Baltesová, Ioannis Boletis, Catarina Bolotinha, Gregor Bond, Georg Böhmig, Lisa Burnapp, et al. Building kidney exchange programmes in European overview of exchange practice and activities. *Transplantation*, 103(7):1514, 2019.

[3] Péter Biró, Joris van de Klundert, David Manlove, William Pettersson, Tommy Andersson, Lisa Burnapp, Pavel Chromy, Pablo Delgado, Piotr Dworczak, Bernadette Haase, et al. Modelling and optimisation in European Kidney Exchange Programmes. *European Journal of Operational Research*, 2019.

[4] Tommy Andersson and Jörgen Kratz. Pairwise kidney exchange over the blood group barrier. *The Review of Economic Studies*, 87(3):1091–1133, 2020.

[5] Nicolau Santos, Paolo Tubertini, Ana Viana, and João Pedro Pedroso. Kidney exchange simulation and optimization. *Journal of the Operational Research Society*, 68(12):1521–1532, 2017.

[6] About COST | Funding research networking. `https://www.cost.eu/who-we-are/about-cost/`.

[7] Introduction - Kidney Exchange Program. `https://www.enckep-cost.eu/page/introduction`.

[8] Yadolah Dodge. *The concise encyclopedia of statistics*. Springer Science & Business Media, 2008.

[9] Chi - Square two sample test. https://www.itl.nist.gov/div898/software/dataplot/refman1/auxillar/chi2samp.htm.

[10] Kolmogorov - Smirnov test. https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test.

[11] Kolmogorov - Smirnov two sample test. https://www.itl.nist.gov/div898/software/dataplot/refman1/auxillar/ks2samp.htm.

[12] Xenia Klimentova and et al. Modelling and optimisation in European Kidney Exchange Programmes. *International Kidney Exchange Programmes in Europe: Practice, Solution Models, Simulation and Evaluation tools: Handbook of Working Group 3 and 4 of the ENCKEP COST Action: European Network for Collaboration on Kidney Exchange Programmes (ENCKEP)*, 2021.

[13] Susan L Saidman, Alvin E Roth, Tayfun Sönmez, M Utku Ünver, and Francis L Delmonico. Increasing the opportunity of live kidney donation by matching for two-and three-way exchanges. *Transplantation*, 81(5):773–782, 2006.

[14] M Delorme, S Garcia, J Gondzio, J Kalcsics, DF Manlove, W Pettersson, and J Trimble. A new matheuristic and improved instance generation for kidney exchange programmes. *Manuscript*, 2021.