Research project

European Kidney Exchange Program

February 6, 2021

Théo Guyard

theo.guyard@insa-rennes.fr

5GM, Mathematics Department, INSA Rennes

Abstract

The Chronic Kidney Disease is the 11$^{\text{th}}$ most common cause of death globally, accounting for almost 1.2 million deaths worldwide per year. The most effective treatment is to receive a kidney transplant from another person. In many countries, kidney exchange schedules are done periodically. Pairs of donors and patients are pooled together, and the aim is to carry out as many transplants as possible. The problem is that in order to do a transplant, the donor and the recipient must be compatible. However, it is often not possible to test the compatibility between all the pairs in the pool. For this reason, strategies must be defined to choose which people to test in such a way that as many transplants as possible can be ultimately performed. These solutions result in long cycles of transplants involving several pairs, but a single failure due to incompatibility on one transplant causes the entire cycle in which it is included to fail. It is thus necessary to take into account the fact that all the transplants will not necessarily be able to be done but we do not know *a priori* which ones will be. In this report, stochastic programming models are used to tackle the problem. These models are designed to take into account the uncertainty of the data in the problem. In particular, two models corresponding to different practical cases are derived. The quality of these two models is evaluated along with the gains compared to other strategies, for instance the one where we only consider the average compatibility failure to decide about the strategy. The numerical results that are obtained will allow us to define strategies according to the number of tests that can be performed in the pool.

Keywords : Kidney Exchange Program, Stochastic programming with recourse, Edge formulation, Maximum matching, Failure-aware exchange model

Contents

1 Introduction

The 2016 Global Burden of Disease Study [Nag+17] identifies the Chronic Kidney Disease as the 11$^{\text{th}}$ most common cause of death globally, accounting for almost 1.2 million deaths worldwide per year or equivalently 2.17% of all yearly deaths [HE18]. Because of its practical importance, but also because of the technical challenges that this problem presents, the scientific community has taken up this problem for a few decades. No permanent cure exists at present for the Chronic Kidney Disease, but it is possible to receive a kidney from another person through a transplant. An alternative is dialysis, which is a costly treatment. For example, it is estimated that in UK dialysis costs between 15,000 and 35,000 pounds per patient per year [Bab+08]. Furthermore, dialysis leads to a worse life expectancy and a worst life quality. That is why kidney transplants are preferred.

The most common way to receive a kidney is from a deceased donor. People awaiting for a transplant are ordered as a waiting list using priority criteria such as the time spent waiting, the severity of the disease, etc... The first patient undergoes a compatibility test and if possible, the transplant is performed. Otherwise, one moves on to the next patient on the waiting list. The issue is that demand exceeds supply and waiting lists continue to grow. To meet this growing demand, most countries have set up a parallel donation system in which donors are living people. For instance, the donor and the receiver can be a relatives. However, it is difficult to find someone who is compatible withing one's relatives. In most countries, state organizations are in charge of *Kidney Exchange Programs* (KEPs). These programs aim to match kidney donors and patients to perform kidney transplants. Usually, patients are placed within a pool paired with a relative who is willing to donate a kidney but who is not compatible. When the pool is estimated to be large enough, a *KEP run* is organized. During this stage, doctors try to find out the best way to perform transplants between the pairs of the pool. The donor of a pair gives its kidney to the patient of another pair and so on until its paired patient also receives a kidney. The solution often results in long transplant cycles. The aim is to perform as many transplants as possible to get the most out of available kidneys.

Before undergoing a transplant, several compatibility criteria between the donor and the patient need to be checked. In general, it is not feasible to test all possible pair combinations. As a transplant can only be carried out if it passes all compatibility tests, doctors need to choose in advance who to test. Once compatibility outcomes are known, one performs as many transplants as possible, using only those which are compatible. Thus, doctors need a strategy to choose in advance who to test and which transplants will potentially be carried out. The issue is that if a tested transplant is finally not feasible, the entire transplant cycle in which it is included cannot be performed. Indeed, a pair giving a kidney must receive one to fit ethical considerations. It is very common for such situations to occur, and currently many transplant opportunities are lost. The strategy implemented to decide who to test is very important due to the impact of the decisions made. However, it is hard to find a good one because of the uncertainty of the compatibility test outcomes. Nowadays, it is possible to estimate the failure rate of each transplant, but it is hard to be really confident about it. Strategies implemented in practice are based on this data, but for the time being, no strategy is really effective and approved unanimously. In this report, we focus on exploring strategies to find out patients that should take compatibility tests in order to maximize the expected number of transplants ultimately carried out during a KEP run.

1.1 European Kidney Exchange Program

In this report, we focus on European KEPs, but such programs exist in most countries. They are much smaller that the United Network for Organ Sharing (UNOS) which is the US program for organ donation, but they are larger than KEPs of developing countries.

1.1.1 Context and stakes for KEPs

The most common way to receive a kidney in European KEPs is through a *deceased kidney donation* (DKD). For such donations, kidneys are transplanted from a deceased person to a living patient. Variations exist in the legislation and the logistic organization of European countries. For example, the Spanish DKD program had a transplantation rate of 57.6 per million people in 2016 which was the highest rate in Europe. In comparison, France had a rate of 52.6 per million people in 2017 [Ant17]. However, the demand for kidney transplantation exceeds the supply of kidneys retrieved from DKD. In most European countries, waiting lists for kidney donations are rising up to several thousand patients. In France, 9,089 patients were on the waiting list in 2012 and they were 14,291 in 2017 [Bio17]. This growing demand has led to the establishment of *living kidney donations* (LKD) alongside DKD in many countries. In a LKD, the patient receives a kidney from a living person who can be a relative, a friend or a non-related person. LKD have better long-term patient and transplant outcomes. However, the main issue is the lack of donors. For instance, the percentage of LKD over the total number of transplants (LKD and DKD) was 5% in Germany, 10% in Spain, 30% in the UK, and slightly over 50% in the Netherlands in 2017 [Eur17]. The advantage of LDK compared to DKD is that there is no need to carry out the transplant in a constrained period of time. Indeed, a kidney coming from a deceased person is either stored on ice or connected to a machine that provides oxygen and nutrients until the kidney is transplanted. In LKD, donors constitute a pool, and doctors organise periodically KEP runs. In order to maximize the number of transplants carried out, they are usually done between two or several donor-patient pairs which are compatible between each other but not within each donor-patient pair. This result in a complex problem where doctors have to decide how to match the donors and the recipients to get the most out of the KEP run without knowing which transplants can really be done because of the compatibility uncertainty.

1.1.2 Compatibility and logistics in KEPs

Before carrying out a transplant, one has to make sure that the donor and the patient, also called recipient, are compatible. The criteria for compatibility are :

- **ABO-compatibility** : Refers to blood type compatibility. Type O can donate to all types. Type A can donate to Type A and Type AB. Type B can donate to Type B and Type AB. Type AB can only donate to Type AB.

- **HLA-match** : Measures the extent to which the *Human Leukocyte Antigens* (HLA) of the recipient and the donor are alike. The more they are alike, the more the pair is considered to be compatible using this metric. When fully compatible, we speak of a *HLA-match*, or simply a *match*.

- **HLA-crossmatch** : Measures whether a recipient has antibodies to the HLA of the donor or not. If there are antibodies in high concentration, we speak of a *positive crossmatch* and the transplant is highly likely to be rejected by the immune system of the recipient. In the case of a *negative crossmatch*, the transplant can be carried out.

The ABO-compatibility and the HLA-match criteria can easily be assessed with blood samples. Usually, the entire pool can take these two first compatibility tests. However, the HLA-crossmatch test is more time consuming and more costly to process. First, the compatibility of the donor and the patient is guessed using a *virtual HLA-crossmatch test* based on ABO-compatibility and HLA-match results. However, this virtual test is not fully reliable. For pairs that are matched and ABO-compatible, a real HLA-crossmatch test must be undertaken by a laboratory before the transplant operation can be performed.

Managing these tests for a DKD transplant system is very straightforward. Each time that a kidney is available, patients are ordered in function of priority criteria that can be for instance the age, the time spent on the waiting list, etc... According to this priority ordering, the first patient takes an ABO-compatibility test, an HLA-match test and a virtual HLA-crossmatch test. If these tests succeed, an HLA-crossmatch test is performed. If this last test allows it, the transplant is performed. Otherwise doctors move on to the next patient.

For LKD transplants system, the situation is completely different. For most KEPs, LKD are carried out between donor-recipient pairs that are compatible between each other but not within each pair because it is hard to find a compatible relative. In LKD, each pair giving a kidney must receive a kidney. To maximize the number of transplants carried out, KEPs often constitute a pool of donor-recipient pairs and try to find the maximum number of transplants that can be performed within the pool. Solutions to such problems can involve long cycles of transplants between pairs. In a pool, it is often feasible to check ABO-compatibility and HLA-match criteria between each pair. However, it is usually too complicated and too expensive to perform all the pairwise HLA-crossmatch tests in the KEP pool. Furthermore, due to ethical considerations, KEPs require the transplants for all donors and pairs in a same cycle to occur simultaneously to avoid withdrawal of donors after their specified recipients have received kidneys but before donating themselves. Thus, doctors must choose in advance which pairs will take an HLA-crossmatch test, while being constrained by the capability of the laboratories and the budget allowed for the tests. After taking the HLA-crossmatch tests and once the results are known, they find the best solution which maximizes the number of transplants done, using only the ones with a negative HLA-crossmatch. The problem is that once the pairs taking an HLA-crossmatch test are chosen and that the outcomes of the tests are known, it is very likely that some transplants are finally not feasible because of a positive HLA-crossmatch. A transplant failure causes the entire transplant cycle in which it is included to fail. Thus, it is very important to choose carefully which HLA-crossmatch test to perform.

It is also possible that *altruistic donors* participate in LKD programs. Such donors are not associated with a recipient and want to give one of their kidneys to save lives. They do not need a kidney back when giving one of their own.

1.1.3 Differences between countries

The real challenge in LKD is that the number of HLA-crossmatch tests that can be undertaken is usually significantly smaller than the number of transplants that are ABO-compatible and HLA-matched in the KEP pool. Thus, doctors have to choose in advance which pairs will take an HLA-crossmatch test and hope that the outcome is negative so that the transplant can finally be carried out. For each recipient, a blood sample allows to establish the *percentage of reactive antibodies* (%pra) which informs about the probability of a negative HLA-crossmatch. This data is often used to choose the pairs that take an HLA-crossmatch test, but no single strategy is universally accepted across the different KEPs.

Most of the countries perform to a *KEP run* periodically or once there are enough pairs

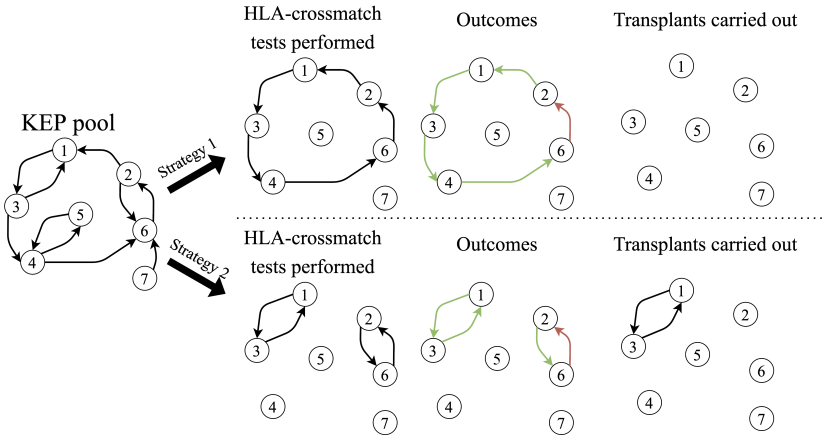

Figure 1: Each node represents a donor-recipient pair. Arrows represent potential transplants that are ABO-compatible and matched. Green and red arrows correspond respectively to negative and positive HLA-crossmatch outcomes. In this case, we consider that only 5 HLA-crossmatch tests can be carried out. The strategy 1 is more risky but can lead to a better solution because it involves more transplants. The strategy 2 is less risky as a positive HLA-crossmatch outcome only affects a part of all the transplants that can be carried out. With the outcomes in this example, no transplants are carried out in the first strategy because of the failure of 6-1. In the second strategy, transplants between 1 and 3 can still be carried out, regardless of the failure of 6-2.

in the pool. For instance, the UK KEP improves the likelihood that all transplants in a cycle will be carried out by allowing only pairwise and three-way exchanges in the pool to minimize the risk of breaking a transplant cycle because of a positive HLA-crossmatch in the pairs that are tested. After the HLA-crossmatch outcomes have been obtained, the KEP organization performs as many transplants as possible using the ones that were tested and have a negative HLA-crossmatch. France has the same strategy, but only allows two-way exchanges. In the Netherlands, if a positive HLA-crossmatch is found in the KEP run, tests are repeated to find the next-best solution, until all crossmatch tests are negative. In general, a limit on the maximum transplant cycle length is defined as a function of the hospital capability to perform several transplants simultaneously. A maximum number of HLA-crossmatch tests that can be taken is also established as a function of the capability of laboratories. Generally speaking, allowing larger cycles may lead to an increase in the number of potential transplants, but the proposed solution incurs an increased risk of positive HLA-crossmatch tests and entire cycle failures.

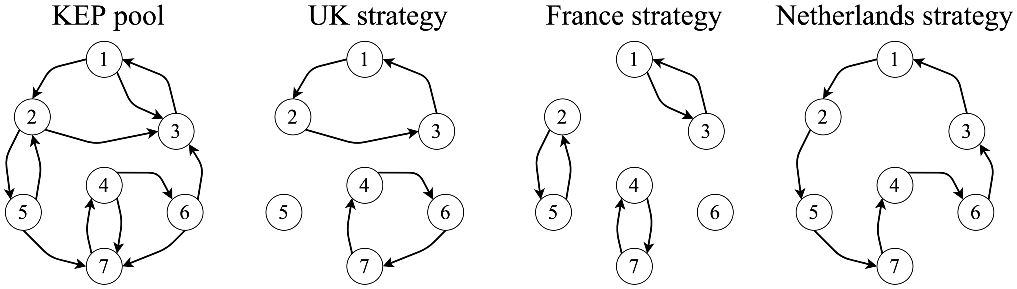

Figure 2: Example of three different strategies. UK allows only tree-way exchanges to be more robust to positive HLA-crossmatch outcomes. France has the same strategy but with two-way exchanges. Netherlands strategy is first to maximize the number of tests performed and if there are some positive HLA-crossmatch, new tests are taken to close broken transplant cycles.

1.2 Objective and constraints

In the sequel, we only focus on LKD. As HLA-crossmatch tests are made between the donor from a pair and the recipient from another pair, we also speak of HLA-crossmatch test for a transplant rather than for a couple of pairs. The aim is to find a strategy to choose a set of transplants that take an HLA-crossmatch test in order to maximize the number of transplants ultimately carried out during a KEP run.

Since HLA-crossmatch outcomes cannot be foreseen before the test is performed, they introduce an uncertainty into the problem. However, once the decision of which tests to perform is made and that HLA-crossmatch outcomes are known, all the uncertainty about compatibility is disclosed and the remaining problem becomes deterministic. To formalize the KEP problem, we consider the following constraints :

$\mathcal{C}_1$: Transplant chains and cycles are allowed in the KEP

$\mathcal{C}_2$: A pair can receive and give at most one kidney

$\mathcal{C}_3$: An altruistic donor can give at most one kidney and does not need to receive one

$\mathcal{C}_4$: A non-altruistic pair donating a kidney must receive a kidney

$\mathcal{C}_5$: Chains and cycles of length at most K are allowed in the KEP

$\mathcal{C}_6$: At most B HLA-crossmatch tests can be done in the KEP run

$\mathcal{C}_7$: The transplants ultimately carried out must only be tested transplants with a negative HLA-crossmatch

The value of K and B depends on the country, the KEP strategy, the hospital capabilities to perform several transplants simultaneously and the lab capabilities to do multiples HLA-crossmatch tests.

1.3 Available data

Information is shared by KEP institutions to help researchers improve KEPs using real data. As KEP is a very practical problem, improvement must be realistic. In the following, we present the data on which we rely.

1.3.1 PrefLib

PrefLib is an online library hosting more than 3000 datasets about organ donation [MW13]. In particular, its *Matching Data* (`MD-00001`) category contains 310 datasets of synthetic KEP pools generated similarly to real KEP pools [DPS13]. A `.wmd` file encodes pairs that constitute each pool and the transplants that are ABO-compatible and HLA-matched. No information is given about HLA-crossmatch of transplants. In addition, a `.dat` file provides meta-information about each pair, including :

- The donor and patient blood type
- Whether the person needing the kidney is the wife/husband of the donor or not
- The %pra of the patient
- Whether a given pair is an altruistic one or not

1.3.2 Failure rate and outcome generation

To build and evaluate strategies, we need to generate artificially HLA-crossmatch test outcomes. In the following, we refer to the probability of a positive HLA-crossmatch as the *failure rate* of a transplant. Thus, for each transplant, we can generate artificially an HLA-crossmatch outcome as a Bernoulli sample using its failure rate as the failure parameter of the Bernoulli law. We rely on [DPS13] in which four rules are defined to set the failure rate of each transplant :

- **Constant** : The failure rate of each transplant is set to 0.7.
- **Binomial** : For each transplant, two uniform random variables u and v in $[0, 1]$ are sampled. If $u < 0.25$, the failure rate is set to $0.2v$. Else, the failure rate is set to $0.8 + 0.2v$.
- **BinomialUNOS** : For each transplant, if the %pra of the recipient is less than 80%, the failure rate is set to 0.1. Else the failure rate is set to 0.9.
- **BinomialAPD** : For each transplant, if the %pra of the recipient is less than 75%, the failure rate is set to 0.28, else, the failure rate is set to 0.58.

We also define a fifth rule which has no practical meaning, but which will be useful for defining strategy quality criteria :

- **NoFailure** : The failure rate of each transplant is set to 0.

In the US KEP, about 7% of the total number of transplants succeed and about 16% fail because of a positive HLA-crossmatch. The other 77% of the total number of transplants are not done for various reasons : withdrawal of a donor-patient pair, the patient has received a kidney from a deceased donor, the patient has chosen to be treated by dialysis, etc ... Thus, on average, $\frac{16}{16+7} \simeq 70\%$ of the transplants among the one that takes an HLA-crossmatch test fail. In most KEPs, committees of professionals have to assign to each transplant a failure rate. They are almost all in agreement to rely on the %pra of the patient and to assign the failure rate as in the *BinomialUNOS* generation rule.

However, the medical knowledge of the %pra for each patient is often incomplete and the failure rate cannot be assigned as easily in general. In the US KEP, about 75% of the patients have a %pra larger than 80%. According to the *BinumialUNOS* rule, it is likely that their failure rate for transplants involving these patients is about 90%. For the remaining part of the pool, the failure rate is likely about 10%. In the *Binomial* generation rule, the variable u represents the random drawing of the %pra of the patient and the variable v is used to express the uncertainty about the exact knowledge of the %pra and thus the uncertainty about the failure rate. The *Constant* generation rule corresponds to the case where no information about the %pra is available. In this case, we only know that about 70% of the transplants cannot be carried out because of a positive HLA-crossmatch.

The *BinomialAPD* is different from the other rules and has an overall failure rate about 35%. This failure rate corresponds to the average failure rate observed in the US organ transplant system which includes kidney transplants but also heart transplants, etc ... This is a more optimistic rule than the *BinomialUNOS* one.

1.4 Literature background

The idea of receiving a kidney from a relative was introduced by [Rap86] in 1986. Since then, different disciplines, such as Medicine and Economy but also Mathematics and Computer Science, have been interested in the question of how to deal efficiently with kidney donations. After discussing ethical questions of this problem [Ros+97], first

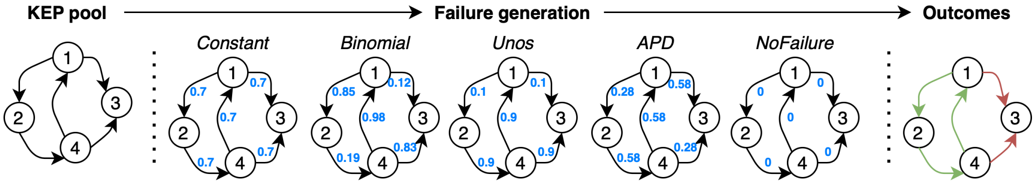

Figure 3: Data generation method. First, the KEP pool is extracted from the `.wmd` and `.dat` files of PerfLib. Then failure rates are set according to one of the 5 rules. Finally, outcomes are generated as Bernoulli samples using the failure rate.

strategies developed to match a pool of patient-donor pairs were using Edmond's Blossom algorithm [Kar71] which finds a maximum weighted matching on a graph. This algorithm is suited for the case of two-way exchanges. Later on, heuristics based on this algorithm were developed to extend strategies to tree-ways exchanges [Bof+17] or to cases where we also consider half-compatible pairs, that is pairs that are not matched but for which a treatment makes a transplant feasible, such as in the Scandinavian KEP for instance [AK16]. This past decade, new results comparing the hardness of different approaches of the problem have been provided. For example, authors in [BMR09] discuss about the complexity of the problem for the case of chains of bounded-length and with cycles of length 2 or 3. In [AJM13], these two problems are compared with respect to their robustness and waiting time in the waiting lists. Authors of [AR14] show the benefit of creating a large pool for KEP but this leads to more difficult problems to solve.

As KEP policies differ between countries, many different models were proposed for the KEP problem. Most commonly used are the edge formulation and the cycle formulation firstly derived in [ABS07a]. The edge formulation assigns one variable per transplant whereas the cycle formulation assigns one variable per transplant cycle. A column generation method and a cutting plane algorithm are also provided in this paper. A new model extending the edge formulation along with resolution algorithms suited for the UK KEP was proposed in [MO15] and is still used in practice in the UK KEP software. In [GKW14], a new Branch-and-Price framework with a polynomial pricing problem was introduced to handle the case where long cycles and chains are allowed. This framework allows to deal with larger pools within a reasonable solution time. Three new formulations, two of which are compact, were introduced in [Dic+16] and are based on graph duplication. These formulations also allow to tackle the problem on large-scale data, for example with the Branch-and-Price framework presented in [RBA20]. In [Bir+19] , most of the models which were introduced and that are used in practice are summarised. Most of the practical considerations and the differences between the different European KEPs are also discussed.

In addition to deterministic models designed for the KEP problem, many studies aim at taking into account the uncertainty about the pair compatibility while building a matching strategy. In such models, we seek to maximize the expected number of transplants ultimately carried out. In [Blu+15] and [AKL19], authors introduce an algorithm in which only a constant number of compatibility queries are required per vertex to retrieve the solution that could have been obtained with an infinite number of compatibility tests. Authors show that not all the transplants need to be tested in a near optimal strategy. Models considering random graphs are introduced by [TP15] and [Ünv10] along with heuristic resolution methods. The first stochastic models that are scalable were introduced by [And14]. A study about the classification of the different scenarios that can occur under compatibility uncertainty is performed by [Lee+18]. One of the most complete models taking into account the transplant failures is presented in [DPS13]. In this paper, a way of generating the data artificially in a realistic way is also provided. Recently, robust KEP models were proposed in [Glo+15] and [Car+20].

2 Two-stage stochastic programming with recourse

To tackle the KEP problem, we use a two-stage stochastic programming model with recourse. In this section, we review the theoretical background of these models.

2.1 Problem formulation

Stochastic programming refers to the optimization of some statistical functions. In the following, we consider the case of an expected value maximization, but stochastic programs can be extended to other statistical functions. In such problems, at least a part of the data is uncertain and is represented by random variables. In a two-stage stochastic program with recourse, *first-stage* decisions have to be taken before the uncertain data is disclosed. Then, at least a part of the realization of the random variables is revealed, and *recourse* actions can be taken in the *second-stage* of the problem as a function of the random variables realizations and of the first-stage decisions. In order to formulate a two-stage stochastic program with recourse, two types of variables are needed :

- First-stage *decision variables* denoted x, representing the decisions taken in the first-stage
- Second-stage *recourse variables* denoted $y(\omega)$, representing the recourse actions taken in the second-stage

Here $\omega \in \Omega$ is the random component of the problem. As $y(\omega)$ is chosen once the uncertainty is disclosed, it depends on the outcome of the uncertain data. The second stage can also depend on the first-stage decisions that were taken.

A stochastic program with recourse admits the following general formulation :

$$
(\mathbb{SP}):\quad z^{\star}_{\mathbb{SP}} = \left\{\begin{array}{lllr}
\displaystyle\max_{x,y(\omega)} & c^{\mathrm{T}}x + \mathbb{E}\Big[q(\omega)^{\mathrm{T}}y(\omega)\Big] & & \\
\text{s.t.} & Ax = b & & (2.1a) \\
 & T(\omega)x + W(\omega)y(\omega) = h(\omega) & \textit{ae. } \omega \in \Omega & (2.1b) \\
 & x \in \mathcal{X}, \quad y(\omega) \in \mathcal{Y} & \textit{ae. } \omega \in \Omega & (2.1c)
\end{array}\right.
$$

In this problem, $q(\omega)$, $T(\omega)$, $W(\omega)$ and $h(\omega)$ is the uncertain data of the problem. The objective is to maximize the profit of the first-stage decisions plus the expected profit of the recourse actions. Even though the problem is a maximization over x and $y(\omega)$, we are only interested in finding the optimal first-stage decisions. Indeed, once these decisions are taken, we do not need to fix the recourse variables immediately. Once the uncertainty is disclosed, it remains a deterministic problem with a fixed value for q, T, W, h but also for x as the first-stage decisions have already been taken. Thus, the remaining problem is just to find the optimal recourse actions for the case which has just arrived.

The constraint (2.1a) is the classical constraint encountered in linear programs that allows to control the domain of feasibility of first-stage variables. The constraint (2.1b) links the second-stage variables with both first-stage variables and realizations of the uncertain data. Finally, the constraint (2.1c) ensures that variables belong to the right set. For instance, one can have $\mathcal{X} = \big\{x, x \geq 0\big\}$ or $\mathcal{X} = \big\{x, x \in \{0,1\}\big\}$

The matrix $W(\omega)$ is called the *recourse matrix*. Most of the time, recourse strategies are fixed in advance so $W(\omega) \equiv W$. In this case, the problem is called a *fixed-recourse* problem. In the sequel, we only focus on this kind of stochastic problem in which the uncertain data of the problem can be packed into a single vector denoted $\xi = (q, T, h)$. The distribution of ξ is assumed to be known.

The *recourse function* corresponds to a second-stage problem where the first-stage decisions have already been fixed. It depends on the realization of the uncertainty. This recourse function, or *recourse problem*, can be defined as

$$Q(x, \xi(\omega)) = \begin{cases} \max\limits_{y(\omega)} & q^{\mathrm{T}}(\omega) y(\omega) \\ \text{s.t.} & W y(\omega) = h(\omega) - T(\omega) x \\ & y(\omega) \in \mathcal{Y} \end{cases}$$

This problem corresponds to the deterministic problem where decisions x have already been fixed, where a realization $\xi(\omega)$ of the uncertainty occurs and where we need to take recourse actions $y(\omega)$. Using this notation, a more convenient way to formulate $\mathbb{SP}$ is under its *implicit form* :

$$(\mathbb{SP}) : \quad z^{\star}_{\mathbb{SP}} = \max_{x \in X} \left\{ c^{\mathrm{T}} x + \mathbb{E}\left[Q(x, \xi) \right] \right\}$$

where $X = \{x \in \mathcal{X},\ Ax = b\}$. To lighten notations, the *expected value function*

$$\mathcal{Q}(x) = \mathbb{E}\left[Q(x, \xi) \right]$$

is also often used. For more insights and examples of stochastic programs with recourse, we refer to [Kin88].

2.2 Quality of a stochastic solution

There are different approaches to a problem with uncertainties. In the literature, several metrics were proposed to evaluate the relevance of using a stochastic programming model for this kind of problems.

2.2.1 Expected value problem

When dealing with a problem with uncertainty, a way to simplify it is to consider only the average case of the random realizations. In this approach, we lose a lot of the information about the uncertainty, but the problem is usually easier to solve because it becomes fully deterministic. This easier problem is called the *expected-value problem* and can be formulated as

$$(\mathbb{EV}) : \quad z^{\star}_{\mathbb{EV}} = \left\{ \max_{x \in X} c^{\mathrm{T}} x + Q(x, \mathbb{E}\left[\xi\right]) \right\}$$

When solving $\mathbb{EV}$, we get the best decisions $x_{\mathbb{EV}}$ that can be taken for the average realization of the uncertainty. Once these decisions have been fixed, the real realization of the uncertainty occurs and it remains to find the right recourse actions to take.

The *expected result of using the expected value solution* corresponds to the average value of the objective of $\mathbb{SP}$ if the decisions have been fixed to $x_{\mathbb{EV}}$. This value can be computed as

$$EEV = c^{\mathrm{T}} x_{\mathbb{EV}} + \mathbb{E}\left[Q(x_{\mathbb{EV}}, \xi) \right] \tag{2.3}$$

As $x_{\mathbb{EV}}$ is not necessarily the optimal solution of $\mathbb{SP}$, one has

$$z^{\star}_{\mathbb{SP}} \geq z^{\star}_{\mathbb{EV}} \tag{2.4}$$

Thus, to estimate the gain of solving $\mathbb{SP}$ instead of $\mathbb{EV}$, we can use the *value of the stochastic solution* which is defined as

$$VSS = z^\star_{\mathbb{SP}} - EEV \quad \text{or} \quad \%VSS = 100 \times \frac{z^\star_{\mathbb{SP}} - EEV}{z^\star_{\mathbb{SP}}} \tag{2.5}$$

This VSS value represents the gain obtained by considering a more complicated problem that does not neglect uncertainty rather than a simpler problem which considers the average case. The larger the VSS, the larger the gain and the more relevant to consider $\mathbb{SP}$ rather than $\mathbb{EV}$.

2.2.2 Wait-and-see value

An easy way to find an upper bound on the optimal value of $\mathbb{SP}$ is to consider that the realization of ξ that will occur is known. In this case, decision and recourse strategies can be chosen with all the data of the problem in hand. The mean objective value that can be obtained when all the uncertain data realization is known is called the *wait-and-see* value and is defined as

$$WS = \mathbb{E}\left[\max_{x \in X} c^{\mathrm{T}} x + Q(x, \xi)\right] \tag{2.6}$$

The WS case is the most optimistic case of the problem where all the realizations can be anticipated. Thus, it leads to a better solution than the one of $\mathbb{SP}$ as more information is known. One has

$$WS \geq z^\star_{\mathbb{SP}} \tag{2.7}$$

To quantify the loss of not having a total knowledge of the random variable realizations, one can use the *expected value of perfect information* which is defined as

$$EVPI = WS - z^\star_{\mathbb{SP}} \quad \text{or} \quad \%EVPI = 100 \times \frac{WS - z^\star_{\mathbb{SP}}}{z^\star_{\mathbb{SP}}} \tag{2.8}$$

If EVPI is small, then the solution of $\mathbb{SP}$ allows to get very close to an omniscient solution. If EVPI is large, then further investigation of the underlying uncertainty is warranted, as the availability of information has a significant impact on the optimal value.

2.2.3 Case of binary recourse variables

In $\mathbb{SP}$, the constraint (2.1c) ensures that variables belong to the right set. In many practical cases, decision or recourse variables need to be integers or binary. Imposing binarity constraints only for the decision variables x breaks the convexity of the problem, but Branch-and-Bound methods allow to deal with it. However, imposing binarity constraints on the recourse variable $y(\omega)$ is different. This breaks the problem structure that is usually exploited by solution methods. In such cases, second stage variables belong to the set

$$\mathcal{Y} = \{y, y \in \{0, 1\}^m\} \tag{2.9}$$

in the case where $|y(\omega)| = m$. Usually, this kind of problem is harder to solve. A way to quantify the quality of the solution of $\mathbb{SP}$ with (2.9) is to compare it with the solution of its convex relaxation where we only impose $y \in \mathcal{Y}^r = \{y, 0 \leq y \leq 1\}$. If $\mathbb{SP}$ has the constraint (2.9), we denote

$$(\mathbb{SP}\mathrm{r}): \quad z^\star_{\mathbb{SP}\mathrm{r}} = \begin{cases} \max\limits_{x, y(\omega)} & c^{\mathrm{T}} x + \mathbb{E}\left[q(\omega)^{\mathrm{T}} y(\omega)\right] \\ \text{s.t.} & Ax = b & & (2.10a) \\ & T(\omega)x + W(\omega)y(\omega) = h(\omega) & \textit{ae. } \omega \in \Omega & (2.10b) \\ & x \in \mathcal{X}, \quad y(\omega) \in \mathcal{Y}^r & \textit{ae. } \omega \in \Omega & (2.10c) \end{cases}$$

As $\mathbb{SP}\mathrm{r}$ is a relaxation of $\mathbb{SP}$, one has

$$z^\star_{\mathbb{SP}\mathrm{r}} \geq z^\star_{\mathbb{SP}} \tag{2.11}$$

Furthermore, solving $\mathbb{SP}\mathrm{r}$ still allows to get decision variables $x_{\mathbb{SP}\mathrm{r}}$ that are usable in $\mathbb{SP}$. However, $x_{\mathbb{SP}\mathrm{r}}$ are not the optimal decision variables of $\mathbb{SP}$ in general so the value of $\mathbb{SP}$ while fixing $x = x_{\mathbb{SP}\mathrm{r}}$ is usually suboptimal.

In the case of (2.9), the EEV usually has no practical meaning. For instance, we will see that in KEP models, a constraint of the following form is considered :

$$y(\omega) \leq h(\omega)$$

When solving the $\mathbb{EV}$, this constraint becomes

$$y \leq \mathbb{E}\left[h\right]$$

and if $\mathbb{E}\left[h\right] < 1$, we can deduce that the optimal solution is $y = 0$. Thus, the EEV breaks the structure of the solution and does not inform a lot about the quality of the stochastic solution. An alternative way in the case of (2.9) is to consider the expected value problem for $\mathbb{SP}\mathrm{r}$ rather than for $\mathbb{SP}$. In the above example, we rather consider a continuous y so the solution is not forced to be $y = 0$. We denote

- EEVr the EEV of $\mathbb{SP}\mathrm{r}$
- VSSr the VSS of $\mathbb{SP}\mathrm{r}$

As $\mathbb{SP}\mathrm{r}$ is a relaxation of $\mathbb{SP}$, the EEVr gives information about the quality of the relaxation of $\mathbb{SP}$. In the following, we always use EEVr instead of EEV as soon as recourse variables are integer.

2.3 Handling a stochastic program in practice

The problem $\mathbb{SP}$ is not tractable in practice because (2.1b) and (2.1c) represent an uncountable number of constraints when ξ has a continuous distribution. In the following, we introduce a method allowing to solve $\mathbb{SP}$ in practice.

2.3.1 Sample Average Approximation

When the random variable ξ has a discrete distribution, the value of $\mathcal{Q}(x)$ can be computed explicitly by splitting the expected value over each possible value of ξ weighted by its probability. The problem $\mathbb{SP}$ can be solved directly by explicitly creating a set of variables and constraints for each realization of the uncertain parameter. In such case, the problem has the following formulation:

$$(\mathbb{SP}) : \quad \max_{x \in X} c^{\mathrm{T}} x + \sum_{s=1}^{S} p^s Q(x, \xi^s) \tag{2.12}$$

where $(\xi^1, \ldots, \xi^S)$ are the values that ξ can take with probabilities $(p^1, \ldots, p^S)$. The problem (2.12) is fully deterministic and can be solved to find the optimal value of x. Many classical methods allow to tackle this kind of deterministic problem. The particular structure of (2.12) can also be exploited with decomposition methods such as the L-shaped method [LL93] which can even be used when binarity is required for first and second stage variables.

When the random data has a continuous distribution, there is an uncountable number of possible values for ξ. As constraints (2.1b) and (2.1c) must be satisfied *ae.* $\omega \in \Omega$, the number of constraints in the problem is also uncountable. This kind of problem is not tractable in practice. One way of tackling the problem in such cases is to use a Monte-Carlo simulation to approximate the continuous distribution of ξ by a discrete distribution. If we are able to generate a S-iid sample $(\xi^1, \dots, \xi^S)$ of ξ, it is possible to replace the expected value function $\mathcal{Q}(x)$ by the unbiased estimator $\hat{\mathcal{Q}}_S(x) = \frac{1}{S}\sum_{s=1}^{S} Q(x, \xi^s)$. This allows to retrieve a problem of the form (2.12) with a countable number of constraints and variables. The model created using this estimator is called the *Sample Average Approximation* (SAA) of $\mathbb{SP}$:

$$
(\mathbb{SP}_S) \quad \begin{cases} \max & c^{\mathrm{T}} x + \dfrac{1}{S}\displaystyle\sum_{s=1}^{S} (q^s)^{\mathrm{T}} y^s \\ \text{s.t.} & Ax = b & & \text{(2.13a)} \\ & T^s x + W^s y^s = h^s & \forall\, s \in \{1, \dots, S\} & \text{(2.13b)} \\ & x \in \mathcal{X}, \quad y^s \in \mathcal{Y} & \forall\, s \in \{1, \dots, S\} & \text{(2.13c)} \end{cases}
$$

In this formulation, constraints (2.1b)-(2.1c) are approximated by constraints (2.13b)-(2.13c). The quality of the approximation of $\mathbb{SP}$ by $\mathbb{SP}_S$ depends on the S-iid sample used. Usually, the larger the S, the better the approximation. In the continuous case, elements of the sample $(\xi^1, \dots, \xi^S)$ are called *scenarios*.

The main drawback of the SAA is that solving $\mathbb{SP}_S$ with a large S is hardly tractable in practice because of the dimension burden drawn by variables y^s that grows with the number of scenarios used. We do not want S too large in order to make $\mathbb{SP}_S$ tractable but we also want S large enough for the approximation to be correct. To calibrate the optimal number of scenarios S to use, it is possible to plot the objective value of $\mathbb{SP}_S$ as a function of S. The more the objective stabilises and its variance tends to 0, the better the quality of the approximation. The value of S is chosen as a trade-off between the quality of the approximation and the computational effort needed to solve the SAA of $\mathbb{SP}$.

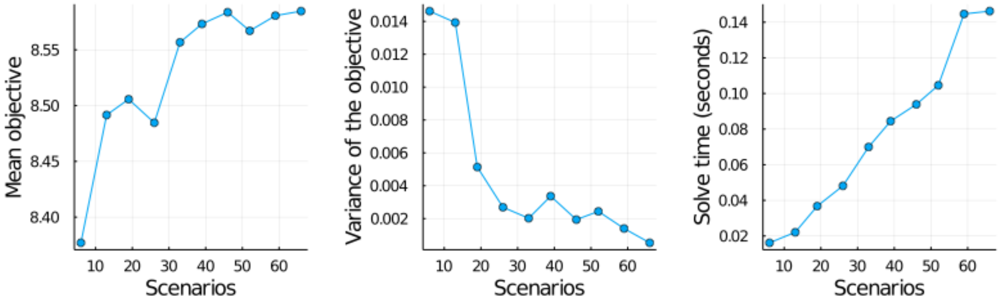

Figure 4: SAA of a KEP problem with different numbers of scenarios. For each number of scenarios, the SAA is solved 20 times for 20 different scenario samples, and we report the mean objective, the variance of the objective and the mean solution time. We can see that when the number of scenarios increases, the objective stabilizes but the solution time increases.

2.3.2 Real cost vs. perceived cost

When solving $\mathbb{SP}_S$ instead of $\mathbb{SP}$, optimal decisions obtained are not necessarily the ones that would have been obtained by solving $\mathbb{SP}$. This depends on the quality of the SAA and the number of scenarios used. Furthermore, the objective value obtained by solving $\mathbb{SP}_S$ is computed for a discretized approximation of the distribution of ξ and not for the real distribution of ξ. Thus, it is crucial to take into account that $\mathbb{SP}_S$ has a biased objective value and gives a biased optimal solution. For instance if $S = 1$, solving $\mathbb{SP}_S$

can lead to a decision with a very high objective value in $\mathbb{SP}_S$. If one uses this decision as the first stage action, one can expect to have such a good result after the revelation of the uncertainties. On the other hand, if the scenario that finally occurs is very different from the one used in the SAA, the real cost associated with the SAA decision will be very far from optimal. In the following, if $x^\star_{\mathbb{SP}_\mathrm{r}}$ are the optimal decisions of $\mathbb{SP}_S$, we denote

- z_S (*perceived cost* of $x^\star_{\mathbb{SP}_\mathrm{r}}$) : the objective of $\mathbb{SP}_S$ associated with the decision $x^\star_{\mathbb{SP}_\mathrm{r}}$
- $z^\star_S$ (*real cost* of $x^\star_{\mathbb{SP}_\mathrm{r}}$) : the expected cost that would have been obtained in practice if decisions $x^\star_{\mathbb{SP}_\mathrm{r}}$ are taken

Decisions $x^\star_{\mathbb{SP}_\mathrm{r}}$ are the only ones that can really be computed in practice when ξ has a continuous distribution. Once this solution is obtained by solving $\mathbb{SP}_S$, we only get a perceived cost corresponding to the sample of scenarios considered. However in practice, the scenario that finally occurs can be different from the ones considered in the SAA. The real cost of a decision $x^\star_{\mathbb{SP}_\mathrm{r}}$ is the one that we can expect to have in practice. The value $z^\star_S$ can be obtained in theory by solving $\mathbb{SP}$ with a fixed decision $x = x^\star_{\mathbb{SP}_\mathrm{r}}$ but this problem is still intractable in practice because of the uncountable number of constraints. Thus, we rather approximate the value $z^\star_S$ by solving a SAA of $\mathbb{SP}$ with the fixed decision $x = x^\star_{\mathbb{SP}_\mathrm{r}}$ and with a very large number of scenarios $S_e >> S$.

The solution of the SAA is an optimistic one. Indeed, the SAA can be seen as a relaxation of $\mathbb{SP}$ as it neglects a part of the uncertainty by approximating the distribution of ξ. The more scenarios, the better the approximation and the smaller the difference. One has,

$$z^\star_{\mathbb{SP}} \leq \cdots \leq \mathbb{E}\left[z_S\right] \leq \mathbb{E}\left[z_{S-1}\right] \leq \cdots \leq \mathbb{E}\left[z_1\right] \tag{2.14}$$

In addition, if $\mathcal{Y}$ is a convex set, the estimator $\hat{\mathcal{Q}}_S$ of $\mathcal{Q}$ is consistent :

$$\lim_{S\to+\infty} z_S = z^\star_{\mathbb{SP}} \quad \text{wp.1} \tag{2.15}$$

Issues about the practical evaluation of $\mathbb{SP}$ also occur when computing the EEV and the WS value. Thus, we will approximate $\mathbb{E}\left[\xi\right]$ using a Monte-Carlo approximation

$$\mathbb{E}\left[\xi\right] \simeq \frac{1}{S_e}\sum_{s=1}^{S_e} \xi^s \tag{2.16}$$

when computing the expected-value solution and the EEV. Furthermore, the WS value is approximated by computing a discrete expected value over a set of S_e scenarios as

$$WS \simeq \frac{1}{S_e}\sum_{s=1}^{S_e}\left[\max_{x\in X}\{c^\mathrm{T}x + Q(x,\xi^s)\}\right] \tag{2.17}$$

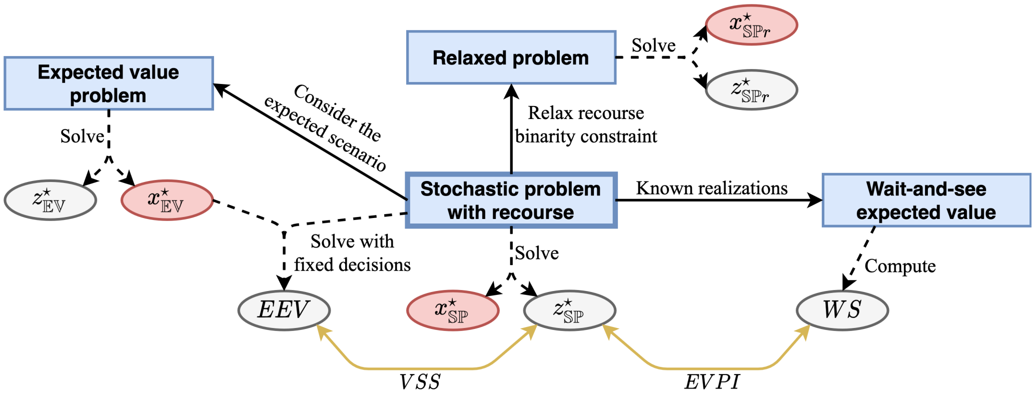
Relaxed problem
Solve
$x^{\star}_{\mathbb{SP}r}$
$z^{\star}_{\mathbb{SP}r}$
Expected value problem
Consider the expected scenario
Relax recourse binarity constraint
Solve
$z^{\star}_{\mathbb{EV}}$
$x^{\star}_{\mathbb{EV}}$
Stochastic problem with recourse
Known realizations
Wait-and-see expected value
Solve with fixed decisions
Solve
Compute
EEV
$x^{\star}_{\mathbb{SP}}$
$z^{\star}_{\mathbb{SP}}$
WS
VSS
$EVPI$

Figure 5: Summary of the different problems and values defined in this section. Dashed arrows are computations that are not tractable in practice an that are usually approximated : SAA to solve a problem or Monte-Carlo simulation to evaluate an expected value.

3 KEP stochastic models

The following aims to define some two-stages stochastic programming formulations of the KEP problem. In a KEP, the uncertain data is the outcome of the HLA-crossmatch tests. In a first stage, we want to find out which transplant is interesting to take an HLA-crossmatch test. We only consider transplants that are ABO-compatible and matched, which are those having a chance to succeed. Then, uncertainties about the HLA-crossmatch are disclosed for tested transplants and we can finally find out which transplants to carry out in the KEP run using only the ones with a negative HLA-crossmatch.

3.1 Modelling a KEP

Before deriving models, we introduce a way to model a KEP run using graphs. There are other ways of modelling a KEP it but the graph model has the advantage of having variables with a practical meaning.

3.1.1 Graph model

A KEP pool can be represented by a directed graph $G(V, A)$ where each vertex $v \in V$ corresponds to a donor-patient pair. Each directed edge $a \in A$ represents a potential feasible transplant from a donor to a recipient, that is to say ABO-compatible and matched but without the knowledge of the HLA-crossmatch. The subset $V_a \subset V$ denotes the set of altruistic donors in G if there are any. Altruistic donors do not need to receive a kidney, however we add a dummy return edge between each non-altruistic pair and each altruistic pair. With these dummy edges, we only have to consider cycles and not both cycles and chains. In some cases, it can be interesting to weight each transplant to denote its priority. For dummy return edges, this weight is set to 0.

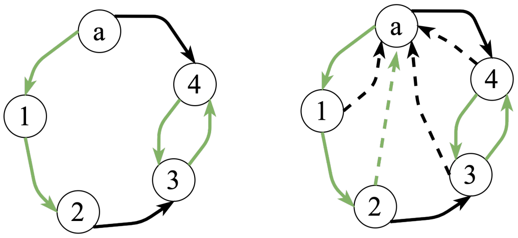

Figure 6: Left : Graph representation of the pool with and altruistic donor a before adding the dummy return edges. The donor of 2 does not necessarily need to give its kidney according to $\mathcal{C}_4$. Transplants carried out are in green. Solution is formed of cycles and chains. Right : Graph representation with dummy return edges, represented as dashed lines. The transplant $2 \rightarrow a$ is not really carried out but this dummy edge allows to consider only solutions formed of cycles.

As HLA-crossmatches are unknown *a priori*, it is possible that some elements of A can not be used in the final KEP run. Thus, the real objective is to select a set of edges in A which will take HLA-crossmatch tests. Once that we have selected the edges and that the HLA-crossmatch outcomes are known, the problem is just to perform as many transplants as possible while satisfying $\mathcal{C}_1$-$\mathcal{C}_5$ and using only tested transplants with a negative HLA-crossmatch. This problem is easier because it is fully deterministic.

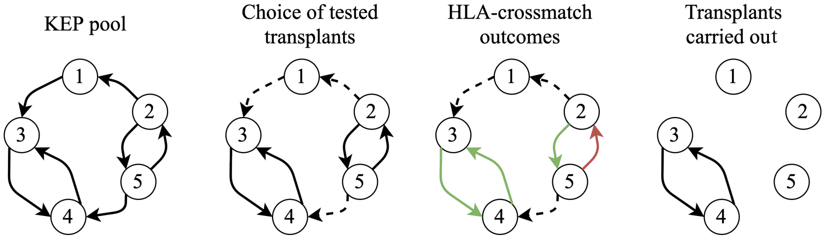

Figure 7: Example of KEP run. Donor-recipient pairs are the vertices, feasible transplants are the black arrows. Arrows with a dashed line do not take a HLA-crossmatch test. A red (resp. green) arrow corresponds to a positive (resp. negative) HLA-crossmatch.

3.1.2 Notations

The following notations are used in the models derived further.

Graph-related notations

- $\mathrm{C}/\ \mathrm{C}_k$: Cycles in G/ Cycles in G of length at most k
- δ_S^- : Edges arriving in $S \subset V$ in a directed graph
- δ_S^+ : Edges leaving $S \subset V$ in a directed graph
- δ_S : Edges arriving or leaving $S \subset V$ in an undirected graph

Data-related notations

- $w_a \geq 0$: The priority of the transplant $a \in A$
- $R_a(\omega) \in \{0, 1\}$: The binary r.v. coding the outcome of the HLA-crossmatch test for the transplant $a \in A$ where $\omega \in \Omega$ is the realization of the uncertain data. If $R_a = 1$, the transplant can be carried out (negative HLA-crossmatch).

The priority of the transplant is a positive number that weights the transplant a. For instance, it can be fixed by the doctors and it meaning is that the larger w_a, the faster the transplant a must be done. The HLA-crossmatch outcomes are supposed to be independent. We also use the vector notation $\mathbf{w} = \{w_a\}_{a \in A}$ and $\mathbf{R} = \{R_a\}_{a \in A}$.

3.2 Variables, sets of constraints and objective

In the following, we define variables that are used and some constraint sets. These are defined for a given graph $G(V, A)$ corresponding to the instance that is treated, and they will be combined later on to write models in a simple way.

3.2.1 Variables

In the graph formulation, there is one first stage and one second stage variable per transplant.

- x_a : The binary first-stage decision variable indicating whether $a \in A$ is chosen to take a HLA-crossmatch test or not.
- $y_a(\omega),\ \omega \in \Omega$: The binary second-stage recourse variable indicating whether $a \in A$ is used in the final KEP run or not, depending on a realization of the HLA-crossmatch test outcome $R_a(\omega)$.

In the following, we also denote $\mathbf{x} = \{x_a\}_{a\in A}$ and $\mathbf{y} = \{y_a\}_{a\in A}$.

3.2.2 Sets of constraints

It is useful to introduce sets of constraints to write several models in a simple way. The following sets of constraints will be combined along with an objective function and the variables defined above to define models.

• $\mathcal{X}_B$: The first set of constraints corresponds to the constraint $\mathcal{C}_6$. It expresses that the total number of HLA-crossmatch tests that can be taken is B and that either we test a transplant, or we do not.

$$\mathbf{x} \in \mathcal{X}_B \iff \begin{cases} \sum_{a\in A} x_a \leq B & \text{(3.1a)} \\ \mathbf{x} \in \{0,1\}^{|A|} & \text{(3.1b)} \end{cases}$$

• $\mathcal{F}$: This second set of constraints expresses $\mathcal{C}_2$ and $\mathcal{C}_3$ and can be seen as flow constraints for $G(V, A)$, ensuring that every pair giving a kidney receives one, except altruistic donors.

$$\forall\, \omega \in \Omega, \quad \mathbf{y}(\omega) \in \mathcal{F} \iff \begin{cases} \sum_{a\in\delta_v^+} y_a(\omega) \leq \sum_{a\in\delta_v^-} y_a(\omega) \leq 1 \quad \forall\, v \in V \setminus V_a & \text{(3.2a)} \\ \sum_{a\in\delta_v^+} y_a(\omega) \leq 1 \quad \forall\, v \in V_a & \text{(3.2b)} \end{cases}$$

• $\mathcal{T}_{\mathbf{x}}$: This set corresponds to $\mathcal{C}_7$ and expresses that transplants carried out must have been tested and must have a negative HLA-crossmatch. These constraints depend on the first-stage variables as only tested transplants can be carried out.

$$\forall\, \omega \in \Omega, \quad \mathbf{y}(\omega) \in \mathcal{T}_{\mathbf{x}} \iff \begin{cases} \mathbf{y}(\omega) \leq \mathbf{R}(\omega) & \text{(3.3a)} \\ \mathbf{y}(\omega) \leq \mathbf{x} & \text{(3.3b)} \end{cases}$$

• $\mathcal{C}_K$: This set expresses the constraint $\mathcal{C}_5$ which is treated separately because this is a complicating constraint. The set is indexed by the maximum length K of the cycles.

$$\forall\, \omega \in \Omega, \quad \mathbf{y}(\omega) \in \mathcal{C}_K \iff \sum_{a\in c} y_a(\omega) \leq K - 1 \quad \forall\, c \in \mathrm{C} \setminus \mathrm{C}_K \tag{3.4}$$

The number of constraints contained in the set is not polynomial as a function of the graph data, contrary to the other sets defined above as it enumerates all the cycles of length at most K.

• $\mathcal{Y}$ and $\mathcal{Y}^r$: Finally, we introduce a set coding that a transplant can either be carried out or not. In addition, we also define its relaxed version corresponding to the unrealistic case where only portions of transplants can be carried out. This last set will be useful to define the EEVr for the KEP problem as explained in 2.2.3.

$$\forall\, \omega \in \Omega, \quad \mathbf{y}(\omega) \in \mathcal{Y} \iff \mathbf{y}(\omega) \in \{0,1\}^{|A|} \tag{3.5a}$$

$$\forall\, \omega \in \Omega, \quad \mathbf{y}(\omega) \in \mathcal{Y}^r \iff \mathbf{y}(\omega) \in [0,1]^{|A|} \tag{3.5b}$$

3.2.3 Objective

In a KEP, we seek to chose the right transplants to test to maximize the total utility of the transplants ultimately carried out. As we do not know HLA-crossmatch outcomes, we maximize the expected total utility of transplants carried out. The quantity we want to maximize is

$$\mathbb{E}[\mathbf{w}^{\mathrm{T}}\mathbf{y}(\omega)] \tag{3.6}$$

We can notice that in this case, first-stage variables are not involved in the objective value, they only appear in the constraints.

3.3 Basic arc formulation

We first focus on the basic arc formulation introduced in [And14]. This formulation aims to maximize (3.6) while satisfying all the constraints given in 1.2. For this formulation, the recourse function can be defined as

$$\forall \omega \in \Omega, \quad Q^{K}_{\mathbb{BA}}(\mathbf{x}, \mathbf{R}(\omega)) = \left\{ \begin{array}{rl} \max_{\mathbf{y}(\omega)} & \mathbf{w}^{\mathrm{T}}\mathbf{y}(\omega) \\ \text{s.t.} & \mathbf{y}(\omega) \in \mathcal{F} \cap \mathcal{T}_{\mathbf{x}} \cap \mathcal{C}_K \cap \mathcal{Y} \end{array} \right. \tag{3.7}$$

The recourse function is indexed by K which is a problem parameter denoting the maximum length of cycles allowed. The value of K depends mostly on the capability of hospitals to perform several transplants simultaneously. Using this recourse function, the basic arc formulation is defined as

$$(\mathbb{BA}) \; : \; z^{\star}_{\mathbb{BA}} = \max_{\mathbf{x} \in \mathcal{X}_B} \; \mathbb{E}\left[Q^{K}_{\mathbb{BA}}(\mathbf{x}, \mathbf{R}(\omega))\right] \tag{3.8}$$

The value of B is the maximum number of HLA-crossmatch tests that can be carried out and also depends on the KEP considered. The $\mathbb{BA}$ formulation expresses that given a maximum number B of HLA-crossmatch test allowed, we want to find which transplants to test in order to maximize the expected total utility of the transplants ultimately carried out, while ensuring that all the constraints defined in 1.2 are satisfied.

We can notice that the number of variables is polynomial as a function of the input data. However, there is an exponential number of constraints because of the constraint set $\mathcal{C}_K$, even when considering a SAA of the problem. Indeed, we need to enumerate all the cycles of lengths at most K in $G(V, A)$. This can become a serious issue when solving a KEP problem on pools with a large number of pairs. To deal with this issue, two choices are considered.

- The first option is to relax the complicating constraint (3.4) by setting $K = +\infty$. Thus, we retrieve a polynomial number of constraints. In this case, cycles of length greater than K can be formed so the solution will be supra-optimal and will not necessarily satisfy $\mathcal{C}_5$.

- The second option is to fix $K = 2$ in $\mathbb{BA}$. Indeed, $\mathbb{BA}$ is $\mathcal{NP}$-hard when $K \geq 3$ but becomes $\mathcal{P}$ if $K = 2$ [ABS07b]. In this way, the problem can be reformulated into a maximum-weighted matching problem with some extra constraints. As only cycles of length two can be formed, the solution will be sub-optimal for KEPs in which cycles of length $K \geq 3$ are allowed.

In the following, we give the two models corresponding to these two options. In our numerical experiments, we never solve $\mathbb{BA}$ directly. We rather solve one of the two formulations derived in 3.4 (case with $K = +\infty$) and in 3.5 (case with $K = 2$).

3.4 Relaxed arc formulation

In a first instance, one considers the first option in which the complicating constraint (3.4) of maximal cycle length in $\mathbb{BA}$ is relaxed. In this way, we retrieve a problem which has a polynomial number of variables and constraints. However, it is possible that long cycles are formed in the solution.

This relaxed formulation is very simple to write as fixing $K = +\infty$ is equivalent to removing the constraint (3.4) from the formulation. Thus, the recourse function of this formulation can be written as

$$\forall \omega \in \Omega, \quad Q_{\mathbb{RA}}(\mathbf{x}, \mathbf{R}(\omega)) = \begin{cases} \max_{\mathbf{y}(\omega)} & \mathbf{w}^{\mathrm{T}}\mathbf{y}(\omega) \\ \quad \text{s.t.} & \mathbf{y}(\omega) \in \mathcal{F} \cap \mathcal{T}_{\mathbf{x}} \cap \mathcal{Y} \end{cases} \tag{3.9}$$

Notice that here $\mathbf{y}(\omega)$ does not necessarily belong to $\mathcal{C}_K$. The relaxed arc formulation can be defined as

$$(\mathbb{RA}) \ : \max_{\mathbf{x} \in \mathcal{X}_B} \ \mathbb{E}\left[Q_{\mathbb{RA}}(\mathbf{x}, \mathbf{R}(\omega))\right] \tag{3.10}$$

The second stage problem can be seen as a maximum-cycle cover problem using only edges with a negative crossmatch for the uncertainty realization considered. In the numerical application section, we study how many cycles of each length are created in the solution to see how much the constraint $\mathcal{C}_5$ is still respected when it is removed from the problem.

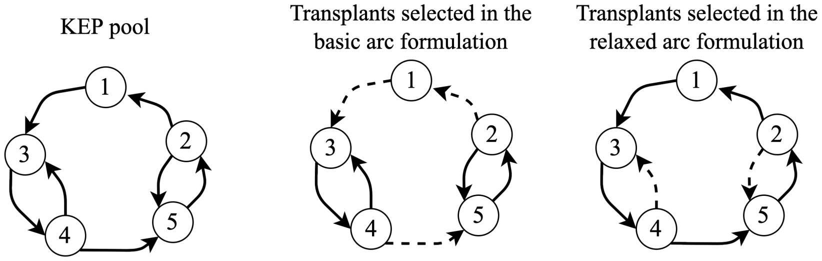
KEP pool
Transplants selected in the basic arc formulation
Transplants selected in the relaxed arc formulation
1
2
3
4
5
1
2
3
4
5
1
2
3
4
5

Figure 8: A solution of $\mathbb{BA}$ and a solution of $\mathbb{RA}$ with B=5 and K=3. Even if K=3, the relaxed arc formulation allows to form cycles of larger length. This is not possible in the basic arc formulation.

3.5 Matching formulation

As a second option, one can set $K = 2$ in $\mathbb{BA}$. The advantage is that it is possible to transform the initial KEP graph into a *matching graph* and to solve a matching problem for this transformed graph, which is a polynomial solvable problem. In this case, we build a solution that matches pairs two by two. This can be relevant for some KEPs, such as the French one as explained in 1.1.3.

3.5.1 Graph transformation

It is possible to transform the initial directed graph G into an undirected graph G_m where each pair of directed edges (i, j) and (j, i) in G is transformed into a single undirected edge $\{i, j\}$ in G_m. Vertices in G and in G_m remain identical. The graph transformation is

$$G = (V, A) \rightsquigarrow G_m = (V, \tilde{A})$$

where $\tilde{A} = \big\{\{i, j\} : (i, j) \in A, (j, i) \in A\big\}$ is now a set of undirected edges. Thus, each edge in G_m represents a possible pairwise exchange. Furthermore, the data of the problem must also be modified. We denote :

- $\tilde{w}_{\{i,j\}} = w_{(i,j)} + w_{(j,i)}$: the merged weight in G_m for $\{i,j\} \in \tilde{A}$.
- $\tilde{R}_{\{i,j\}}(\omega) = R_{(i,j)}(\omega) \times R_{(j,i)}(\omega),\ \forall \omega \in \Omega$: the merged outcome of $\{i,j\} \in \tilde{A}$.
- $\tilde{B} = \lfloor B/2 \rfloor$: the modified maximal number of HLA-crossmatch tests that can be carried out.

In order to carry out the transplant $\{i,j\} \in \tilde{A}$, we need $R_{\{i,j\}} = 1$, that is to say $R_{(i,j)} \times R_{(j,i)} = 1$. Furthermore, the gain of carrying out the transplant $\{i,j\} \in \tilde{A}$ is the gain of carrying out both the transplant (i,j) and the transplant (j,i). Finally, to ensure that $\{i,j\} \in \tilde{A}$ can be done, both (i,j) and (j,i) need to be tested, so the maximum number of HLA-crossmatch tests for G_m must be divided by 2 and still needs to be integer.

In the matching formulation, choosing to test $\{i,j\} \in \tilde{A}$ is equivalent to test the two edges (i,j) and (j,i) in G. Likewise, carrying out the transplant $\{i,j\} \in \tilde{A}$ is equivalent to carrying out the two transplants (i,j) and (j,i) in G. Thus, in the matching model variables denotes pairwise transplants rather than a unique transplant. Furthermore, the objective (3.6) in this model rather represents the expected total utility of pairwise transplants.

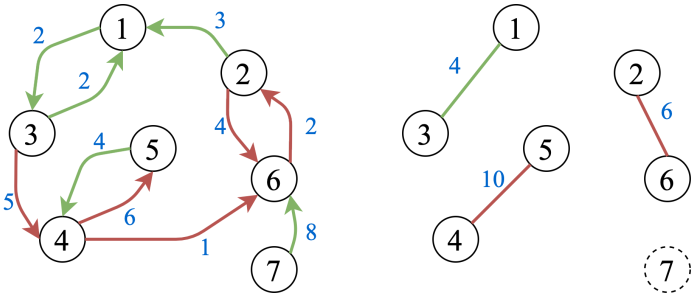

Figure 9: Left : Original graph with weights and HLA-crossmatch outcomes. Right : Matching graph with modified weights and modified HLA-crossmatch outcomes.

3.5.2 Problem formulation

In the following, sets of constraints defined in 3.2.2 are applied for the graph G_m. However, the set $\mathcal{F}$ is defined for directed graphs and cannot be applied to G_m. In the case were $K = 2$ the set $\mathcal{F} \cap \mathcal{C}_K$ applied to G is equivalent to the set $\tilde{\mathcal{F}}$ applied to G_m where

$$\forall\, \omega \in \Omega, \quad \mathbf{y}(\omega) \in \tilde{\mathcal{F}} \iff \sum_{a \in \delta_v} y_a(\omega) \leq 1 \quad \forall\, v \in V \tag{3.11}$$

Using sets defined in 3.2.2 applied to G_m and the new set $\tilde{\mathcal{F}}$, the recourse function of the matching model can be expressed as

$$\forall \omega \in \Omega, \quad Q_{\mathbb{MM}}(\mathbf{x}, \tilde{\mathbf{R}}(\omega)) = \left\{ \begin{array}{rl} \max_{\mathbf{y}(\omega)} & \tilde{\mathbf{w}}^{\mathrm{T}} \mathbf{y}(\omega) \\ \text{s.t.} & \mathbf{y}(\omega) \in \tilde{\mathcal{F}} \cap \mathcal{T}_{\mathbf{x}} \cap \mathcal{Y} \end{array} \right. \tag{3.12}$$

Notice that here, $\mathbf{y}(\omega)$ necessarily belongs to $\mathcal{C}_K$ with $K = 2$ because of the definition of $\tilde{\mathcal{F}}$. The matching formulation can be defined as

$$(\mathbb{MM}) \ : \ z^{\star}_{\mathbb{MM}} = \max_{\mathbf{x} \in \mathcal{X}_{\tilde{B}}} \ \mathbb{E}\left[Q_{\mathbb{MM}}(\mathbf{x}, \tilde{\mathbf{R}}(\omega)) \right] \tag{3.13}$$

Here, B is replaced by $\tilde{B}$ in the definition domain of $\mathbf{x}$. The second stage of this problem is a maximum-weighted matching problem with only the edges $a \in \tilde{A}$ of G_m such that

$\tilde{R}_a = 1$ for the considered realization ω of HLA-crossmatch outcomes. Such problem is easily tractable. The number of variables and constraints is polynomial as a function of the data.

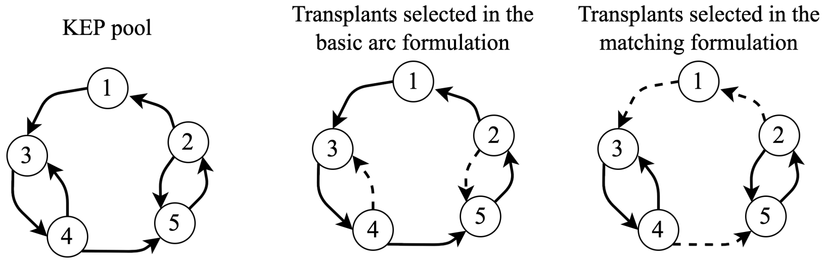
KEP pool
Transplants selected in the basic arc formulation
Transplants selected in the matching formulation
1
2
3
4
5
1
2
3
4
5
1
2
3
4
5

Figure 10: A solution of $\mathbb{BA}$ and a solution of $\mathbb{MM}$ (transformed back from the matching graph to the original graph) with $B = 5$ and $K = 5$. Even though cycles of length 5 are allowed in $\mathbb{BA}$, only cycles of length 2 can be formed in the $\mathbb{MM}$ solution.

3.6 Quality of the stochastic solution for KEP models

In the following, we give definitions and a practical meaning for the values defined in 2.2 in the case of KEP.

3.6.1 Wait-and-see value

The WS value in a KEP corresponds to the expected objective in the case where HLA-crossmatch outcomes are known. For all possible realizations of the uncertainty, the problem to solve is just to find the best KEP run among the transplants that are feasible and have a negative crossmatch. To solve the problem, it is sufficient to remove edges with a positive crossmatch from G and it remains a fully deterministic problem. What is tricky is that as defined by (2.6), at most B transplants can be done as the constraint is that transplants need to be tested in order to be ultimately done. Thus, even if it is possible to obtain a solution with more than B transplants, the WS solution is capped to at most B transplant selections among all the feasible ones in G. The WS value for $\mathbb{RA}$ and for $\mathbb{MM}$ can be computed as

$$\mathrm{WS}_{\mathbb{RA}} = \mathbb{E}\left[\max_{\mathbf{x}\in\mathcal{X}_B} Q_{\mathbb{RA}}(\mathbf{x}, \mathbf{R}(\omega))\right] \quad \text{and} \quad \mathrm{WS}_{\mathbb{MM}} = \mathbb{E}\left[\max_{\mathbf{x}\in\mathcal{X}_{\tilde{B}}} Q_{\mathbb{MM}}(\mathbf{x}, \tilde{\mathbf{R}}(\omega))\right]$$

3.6.2 EEV

The KEP problem has integer recourse variables $\mathbf{y}$. As explained in 2.2, the EEV can be meaningless in such cases. The constraint (3.3a) in the $\mathbb{EV}$ becomes $y_a(\omega) \leq \mathbb{E}\left[R_a\right]$ and as in general $\mathbb{E}\left[R_a\right] \in]0,1[$, the solution of the $\mathbb{EV}$ is just to fix $\mathbf{y}(\omega) = \mathbf{0}$, $\forall\omega$. Thus, this value does not inform much about the real gain comparing to the deterministic problem considering only the average outcome scenario. That is why we rather compute the EEVr for the problem where integrity constraints on the second stage variables are relaxed. Thus, the solution of $\mathbb{EV}$ still produces integer first-stage variables but the EEVr computed in this way is an upper bound on the real EEV as it is a relaxation. However, there is not a practical sense for the decision variables outputted by the expected-value problem. For $\mathbb{RA}$ and $\mathbb{MM}$, expected-value problems can be defined as

$$(\mathbb{EV}_{\mathbb{RA}}) : \left\{ \begin{array}{rl} \max_{\mathbf{x},\mathbf{y}} & \mathbf{w}^{\mathrm{T}}\mathbf{y} \\ \text{s.t.} & \mathbf{x} \in \mathcal{X}_B \\ & \mathbf{y} \in \mathcal{F} \cap \mathcal{T}_{\mathbf{x}} \cap \mathcal{Y}^r \end{array} \right. \quad \text{and} \quad (\mathbb{EV}_{\mathbb{MM}}) : \left\{ \begin{array}{rl} \max_{\mathbf{x},\mathbf{y}} & \tilde{\mathbf{w}}^{\mathrm{T}}\mathbf{y} \\ \text{s.t.} & \mathbf{x} \in \mathcal{X}_{\tilde{B}} \\ & \mathbf{y} \in \tilde{\mathcal{F}} \cap \mathcal{T}_{\mathbf{x}} \cap \mathcal{Y}^r \end{array} \right.$$

with respectively $\mathbb{E}[\mathbf{R}]$ and $\mathbb{E}[\tilde{\mathbf{R}}]$ as unique possible HLA-outcome in $\mathcal{T}_{\mathbf{x}}$. Notice that here, the set of constraints $\mathcal{Y}$ is replaced by its relaxed version $\mathcal{Y}^r$. As $\mathbf{R}$ is a vector of Bernoulli variables, $\mathbb{E}[R_a] = \mathbb{P}(R_a = 1) = 1 - p, \ \forall a \in A$ where p is the failure rate defined in 1.3.2. Furthermore, elements of $\tilde{\mathbf{R}}$ are the product of two independent Bernoulli variables so similarly, $\mathbb{E}[\tilde{R}_a] = (1 - p_1)(1 - p_2), \ \forall a \in \tilde{A}$ where p_1 and p_2 are the failure rates of the two directed edges in the original graphs that form the undirected edge a.

3.6.3 Omniscient budget and budget factor

In the models, the budget B is a parameter and is defined by the labs capability to analyse HLA-crossmatch tests and the money invested in the KEP. To set a reference for this value, we define the *omniscient budget* B_{om} after which it is not possible to get a better solution. This value is defined as

- **Omniscient budget** B_{om}: Maximum number of transplants that can be carried out in the case where we know that all the transplants will have a negative HLA-crossmatch.

This indicates that whatever the HLA-crossmatch outcomes, it will not be possible to carry out more than B_{om} transplants. Equivalently, this is the WS value for the *NoFailure* generation rule defined in 1.3.2 in the case of unit transplant utilities and with an infinite budget. Using this omniscient budget, we introduce a budget factor $b_f \geq 0$ which corresponds to the proportion of B_{om} that is allowed in the KEP. This value is defined as :

$$B = b_f \times B_{om}$$

In the case of unit transplant utilities, it is possible that WS meets B_{om} for a given $b_f \leq 1$. The WS value is equal to B_{om} for budget factors greater than 1 because this corresponds to the case that enough HLA-crossmatch tests are allowed to test all the transplants that could have been involved in the optimal solution. However, it can be interesting to have $b_f \geq 1$ in the practical case because it allows to test more people so as to be more robust in case of positives crossmatch outcomes. When b_f increases, the problem is less constrained which leads to a higher optimal value.

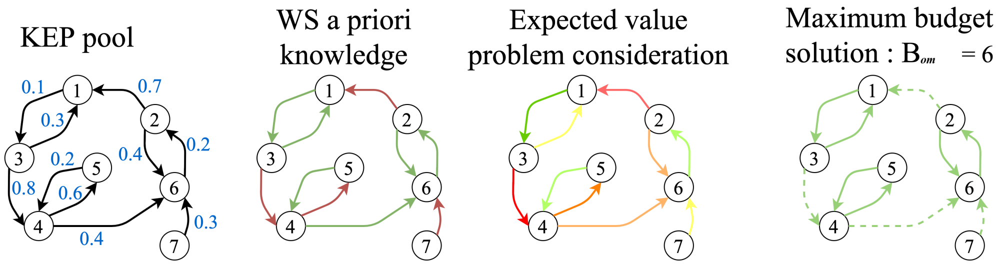
KEP pool
WS a priori knowledge
Expected value problem consideration
Maximum budget solution : B_om = 6
0.1
0.7
0.3
0.2
0.4
0.2
0.8
0.6
0.4
0.3
1
2
3
4
5
6
7

Figure 11: From left to right : Original KEP pool with failure rates. Wait-and-see problem a priori knowledge for one of the realization of HLA-crossmatch outcomes. Graph and HLA-crossmatch outcomes considered in the $\mathbb{EV}$ of $\mathbb{SP}$r (transplants have a "proportion" of failure as we take the expected outcome). Graph considered when computing the value of B_{om} which is 6 in this case (at most 6 transplants can be carried out in this graph, while satisfy the constraints in 1.2).

4 Numerical results

In this section, we assess $\mathbb{MM}$ and $\mathbb{RA}$ on PerfLib instances. The uncertain data does not have a continuous distribution but enumerating all the possible outcomes for the HLA-crossmatch leads to $2^{|A|}$ possibilities and as many constraints. Thus, we approximate the problem by SAA where we only consider a smaller sample of HLA-crossmatch outcome scenarios. We first calibrate the number of scenarios to use in order to have a good SAA and then we evaluate the relevance of the models using quality criteria defined in 2.2. In the following, all the transplants have a unit priority weight except dummy edges returning to altruist donors for which the weight is zero.

4.1 Instances

To evaluate the models described above, we use PrefLib instances. Outcomes of HLA-crossmatch tests are generated according to the rules defined in [DPS13] and derived in 1.3.2. As explained in 3.5, the matching model relies on a modified matching graph which has fewer edges than the original graph. Furthermore, the transformation may leave vertices unconnected from the rest of the graph, which are removed. Ultimately, G_m may have fewer vertices than G. The following table shows the number of vertices, altruistic vertices and edges in the instances for the original graph and for its matching version.

PrefLib instances	Original graph			Matching graph		
MD-00001-x	$\lvert V\rvert$	$\lvert V_a\rvert$	$\lvert A\rvert$	$\lvert V\rvert$	$\lvert V_a\rvert$	$\lvert\tilde{A}\rvert$
00000010	16	0	47	7	0	7
00000020	17	1	89	10	1	8
00000030	18	2	124	15	2	21
00000040	32	0	168	6	0	4
00000050	33	1	278	30	1	46
00000060	35	3	345	28	3	54
00000070	36	4	372	32	4	78
00000075	64	0	961	39	0	84
00000080	64	0	888	52	0	104
00000085	67	3	1282	66	3	226
00000090	67	3	988	50	3	101
00000095	70	6	1510	66	6	296
00000100	70	6	1597	63	6	268
00000105	73	9	1921	71	9	388

Table 1: PrefLib instances considered with the number of vertices, altruistic vertices and edges in their corresponding graph and for its matching version.

Here, we can notice that transforming a graph into a matching graph reduces considerably the number of edges in the graph and consequently the number of variables in the $\mathbb{MM}$ model comparing to the $\mathbb{RA}$ model.

4.2 Scenario calibration

To evaluate models, we solve a SAA of $\mathbb{MM}$ and of $\mathbb{RA}$, so we need to calibrate the number of scenarios to use. The following figures represent the objective value evolution of $\mathbb{MM}_S$ for different numbers of scenarios with 50 repeats each. For this model, the number of recourse variables grows as $|\tilde{A}| + |\tilde{A}| \times S$. Thus, the solution time is very sensitive to the number of scenarios used. For this calibration, the budget is fixed to $B = |V|$. When the SAA is solved over S scenarios, the real cost of the solution is approximated by evaluating the decisions over $S_e = 5000$ scenarios. The scenario factor s_f represented on the x-axis is computed as $S = s_f \times |A|$. Notice that here, we use the value $|A|$ even if the $\mathbb{MM}$ is applied on the graph $G_m = (V, \tilde{A})$.

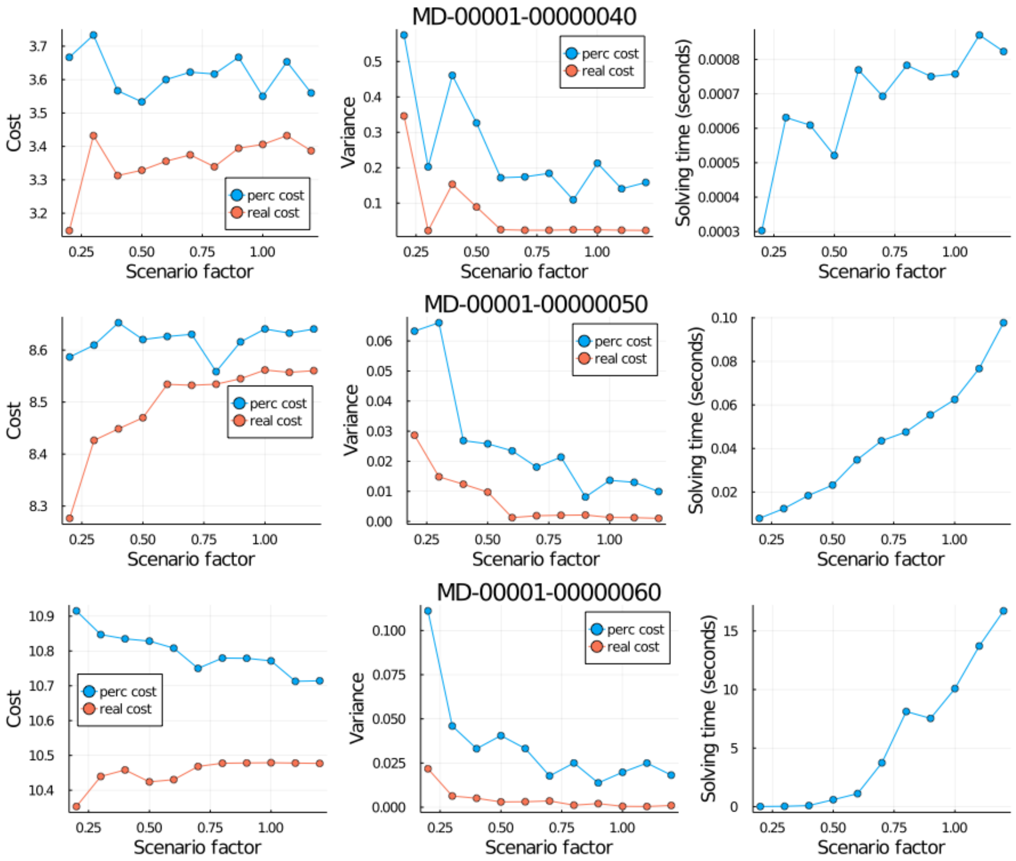

Figure 12: SAA of the matching formulation for 3 different KEP instances and for different scenario factors. For each scenario factor, 50 SAA are solved and the mean objective cost, the variance of the objective cost and the mean solution time is reported.

According to these figures, setting $S = |A|$ allows to well approximate $\mathbb{MM}$ by $\mathbb{MM}_S$ because the objective value does not oscillate much and the variance of the cost is close to 0. However, it is usually more convenient to fix S and S_e once for all to be able to compare the results for different instances. Thus, we use $S = 100$ and $S_e = 5000$ in the sequel. We can still see the calibration correctness by comparing the perceived and the real cost.

4.3 Matching model

We first assess the matching model $\mathbb{MM}$ for a budget $B = |V|$ and $B = 2|V|$ using a *BinomialUNOS* failure generation rule. In the following, we report the perceived and the real cost of the solution of $\mathbb{MM}_S$. We also evaluate the real cost if the decision used is the solution of $\mathbb{MM}_S$ where recourse binarity constraints are relaxed. We also report the EEVr, the WS value, the solution time t for the original problem and the solution time t_r for the relaxed problem in seconds. The value C_i represents the mean number of cycles of length i in the scenario solutions. We also give the percentage of cycles of each length i in the solution. For $\mathbb{MM}$, only cycles of length 2 are allowed in the solution. For each instance, the set of scenarios used for $B = |V|$ and $B = 2|V|$ is not necessary the same.

	MD-00001-x	perc cost	real cost	real cost (relax)	EEVr (%VSSr)	WS (%EVPI)	C_2	t	t_r
1	00000010	4.00	3.98	3.98	3.98 (0%)	3.98 (0%)	2.00 (100%)	0.00	0.00
2	00000020	2.94	2.93	2.93	2.93 (0%)	2.93 (0%)	1.87 (100%)	0.00	0.00
3	00000030	9.00	8.78	8.78	8.73 (0.47%)	8.87 (0.97%)	5.50 (100%)	0.10	0.01
4	00000040	3.62	3.61	3.61	3.61 (0%)	3.61 (0%)	1.81 (100%)	0.00	0.00
5	00000050	8.58	8.60	8.59	8.08 (6.02%)	8.62 (0.35%)	4.74 (100%)	0.04	0.01
6	00000060	10.92	10.70	10.71	10.62 (0.77%)	10.83 (1.14%)	6.96 (100%)	0.26	0.15
7	00000070	11.94	11.82	11.81	11.59 (1.99%)	12.04 (1.85%)	7.94 (100%)	1777.71	443.98
8	00000075	20.44	20.08	20.04	19.88 (0.98%)	20.57 (2.41%)	10.22 (100%)	5.23	4.07
9	00000080	17.84	17.78	17.73	17.35 (2.41%)	17.82 (0.19%)	8.92 (100%)	0.35	0.19
10	00000085	23.86	23.22	23.25	23.22 (0.02%)	23.61 (1.64%)	13.43 (100%)	37.80	12.02
11	00000090	16.84	16.59	16.59	16.21 (2.25%)	16.61 (0.14%)	9.92 (100%)	0.33	0.18

Table 2: Matching model results for $B = |V|$. Other instances were not solved in less than 3600 seconds.

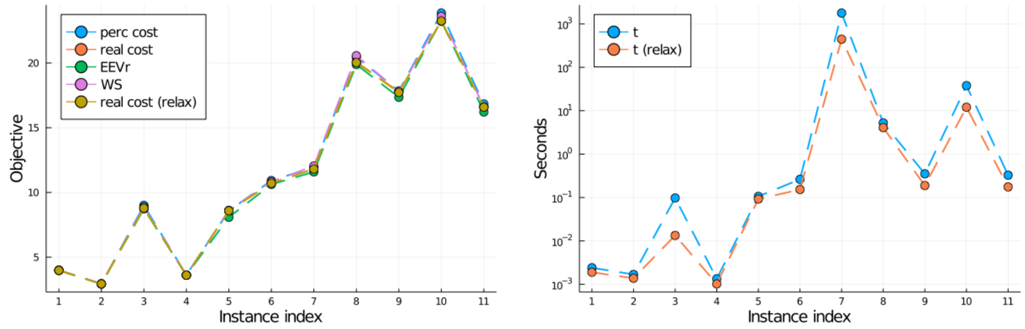

Figure 13: Matching model results and solution time for $B = |V|$.

	MD-00001-x	perc cost	real cost	real cost (relax)	EEVr (%VSSr)	WS (%EVPI)	C_2	t	t_r
1	00000010	3.94	3.97	3.97	3.97 (0%)	3.97 (0.00%)	1.97 (100%)	0.00	0.00
2	00000020	2.96	2.92	2.92	2.92 (0%)	2.92 (0.00%)	1.98 (100%)	0.00	0.00
3	00000030	8.98	8.83	8.83	8.82 (0.19%)	8.83 (0.01%)	5.49 (100%)	0.02	0.02
4	00000040	3.62	3.61	3.61	3.61 (0%)	3.61 (0.00%)	1.81 (100%)	0.00	0.00
5	00000050	8.56	8.61	8.62	8.14 (5.49%)	8.62 (0.11%)	4.78 (100%)	0.09	0.09
6	00000060	10.80	10.83	10.82	10.68 (1.42%)	10.84 (0.03%)	6.90 (100%)	0.15	0.12
7	00000070	12.08	12.05	12.06	11.94 (0.95%)	12.08 (0.22%)	8.04 (100%)	0.24	0.18
8	00000075	20.34	20.50	20.48	19.35 (5.60%)	20.59 (0.44%)	10.17 (100%)	0.34	0.17
9	00000080	17.66	17.80	17.80	17.02 (4.40%)	17.81 (0.02%)	8.83 (100%)	0.21	0.19
10	00000085	23.32	23.59	23.58	23.24 (1.46%)	23.61 (0.08%)	13.16 (100%)	2.43	1.42
11	00000090	16.42	16.56	16.57	16.27 (1.76%)	16.58 (0.13%)	9.71 (100%)	0.22	0.09
12	00000095	30.00	30.07	29.94	29.07 (3.32%)	30.14 (0.24%)	18.00 (100%)	12.48	9.69
13	00000100	29.98	29.61	29.64	28.46 (3.87%)	29.72 (0.38%)	17.99 (100%)	5.30	2.53
14	00000105	32.84	32.47	32.38	31.48 (3.05%)	32.60 (0.39%)	20.92 (100%)	56.73	35.22

Table 3: Matching model results for $B = 2|V|$.

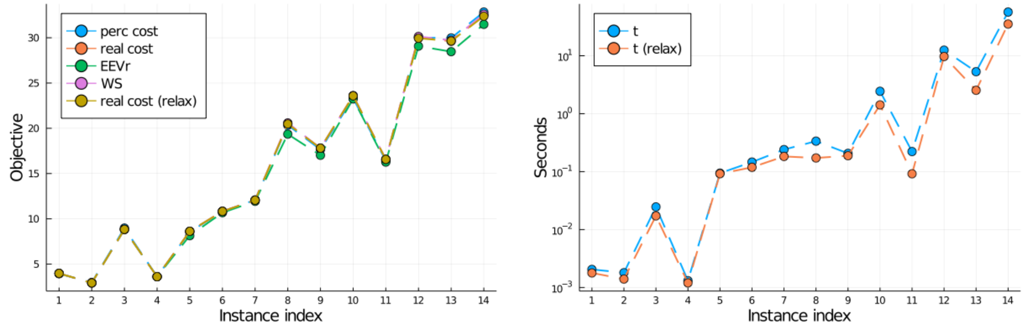

Figure 14: Matching model results and solution time for $B = 2|V|$.

We can notice that for these instances, the SAA well approximates the original stochastic problem because the perceived cost is close to the real cost. Furthermore, it seems that the objective of the relaxed problem is very close to the one of the non-relaxed problem. This relaxed problem can give a very good starting point or a very good initial upper bound. Setting a budget $B = |V|$ allows to retrieve enough information about the HLA-crossmatch outcomes to nearly meet the WS value corresponding to this budget. However, increasing the budget to $B = 2|V|$ allows to carry out many more transplants and improves the solution time. It seems that the VSSr gain is larger than the EVPI gain. The %VSSr rises up to about 6% with $B = |V|$ or with $B = 2|V|$. This means that by considering a stochastic problem rather than considering the mean scenario allows to perform 6% more transplants. On the other hand, involving more budget to get more confident about the HLA-crossmatch failure outcomes is not very interesting as we can only hope to increase by at most 2.5% the number of transplants that will be performed.

4.3.1 Budget sensibility

Above, the budget was chosen arbitrarily, but we can see that it has a significant impact on the solution. In the following, we study the sensibility of the model to the budget by varying the budget factor b_f defined in 3.6. We show the results for the instances `MD-00001-00000080` and `MD-00001-00000090`.

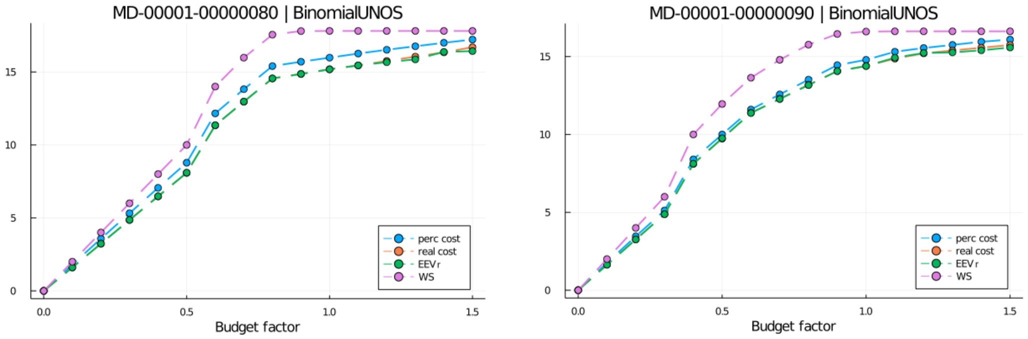

Figure 15: Budget sensibility for a *BinomialUNOS* failure generation rule.

We can clearly notice the breaking point around $b_f = 0.8$ for the left instance and around $b_f = 0.9$ for the right instance. After this breaking point, the WS value remains constants. Before this breaking point, the perceived and the real cost are close to the EEVr but after, they continue to increase to finally meet the WS value. In a practical case, it should be interesting to pay more to perform more HLA-crossmatch tests in order to get closer to the WS value. However, if the budget allowed is very low, the decisions that are given by the EEVr model are still very interesting and it may be unnecessary to consider a stochastic model.

4.3.2 Failure generation sensibility

The above graphics are obtained with the *BinomialUNOS* failure generation rule defined in 1.3.2. However, the results are very different when considering a different rule. In the following, we report the objective value for different budget factors and for the same instances as in 4.3.1 but for the *Constant*, the *Binomial* and the *BinomialAPD* failure generation rules.

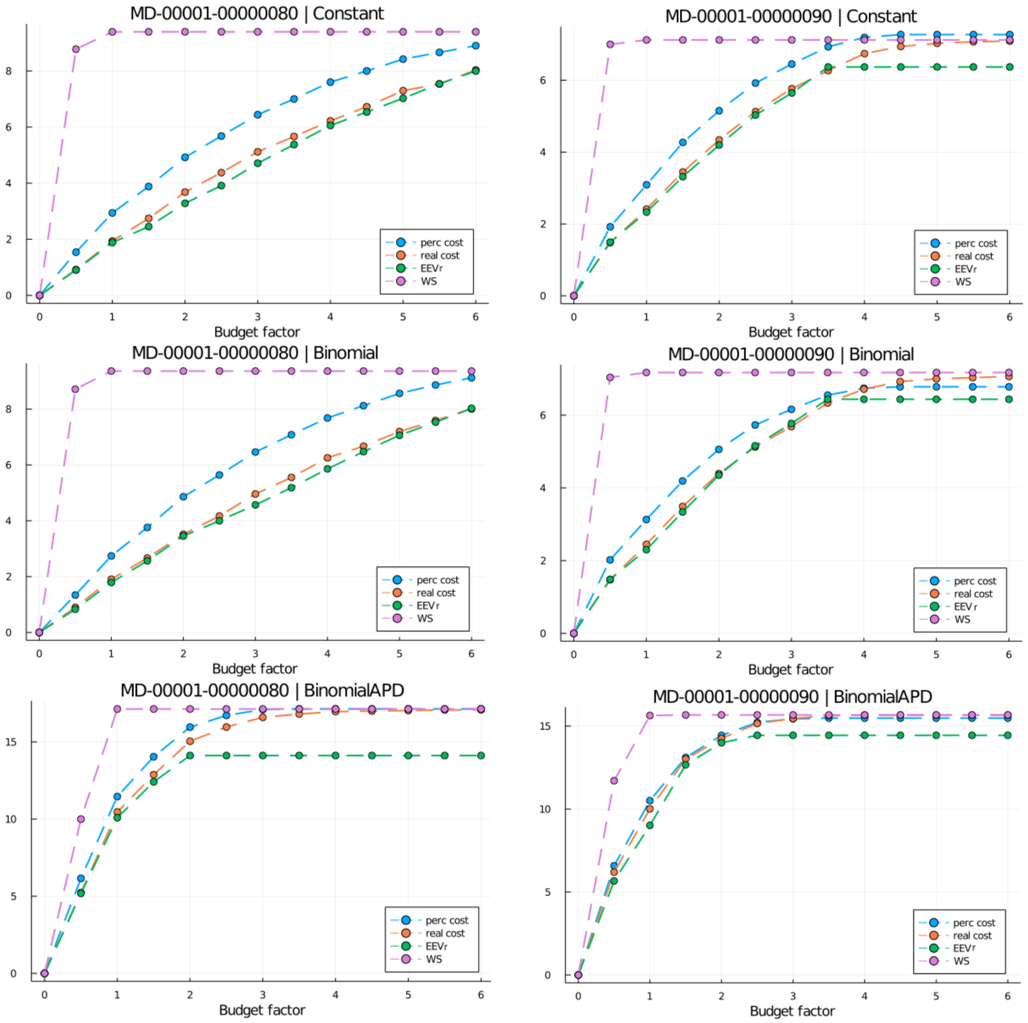

Figure 16: Budget sensibility for the *Constant*, the *Binomial* and the *BinomialAPD* failure generation rule.

On these figures, we can see that generation rules for which failure rates are very different lead to a smaller gap between the EEVr and the WS value. This is quite intuitive because if transplants have very different failure rates, there will be a tendency to select only those with a low failure rate. On the contrary, if all transplants have a similar failure rate, one will necessarily choose to test transplants with a modest failure rate. In practice, if we are not very confident about the probabilities of failure of the transplants in the pool, we can hope to have a better solution with the stochastic problem than with the expected value problem, but we will be very far from the optimal solution anyway, unless we provide a very large budget. On the other hand, if the failure rate is roughly known, the WS value will not be reached either, but we will have an objective close to the WS for any budget considered.

4.4 Relaxed-arc model

For the relaxed-arc model $\mathbb{RA}$, we also first study the results for the budget $B = |V|$ and $B = 2|V|$ and for the *BinomialUNOS* generation. We report the same values as for the matching model. C_{4+} denotes the mean number of cycles of length 4 and more in the scenario solutions. For $\mathbb{RA}$, cycles of length greater than 4 are allowed. For each instance, the set of scenarios used for $B = |V|$ and $B = 2|V|$ is not necessary the same.

	MD-00001-x	perc cost	real cost	real cost (relax)	EEVr (%VSSr)	WS (%EVPI)	C_2	C_3	C_{4+}	t	t_r
1	00000010	3.98	3.98	3.99	3.83 (3.68%)	3.99 (0.18%)	0.85 (59.86%)	0.00 (0%)	0.57 (40.14%)	0.13	0.07
2	00000020	6.59	6.55	6.55	6.50 (0.83%)	6.75 (3.06%)	0.37 (22.29%)	0.31 (18.67%)	0.98 (59.04%)	0.69	0.55
3	00000030	10.58	10.30	10.30	8.37 (18.79%)	11.88 (15.28%)	1.68 (42.21%)	0.94 (23.62%)	1.36 (34.17%)	1253.24	813.50
4	00000040	3.52	3.61	3.61	3.54 (1.80%)	3.61 (0.00%)	1.76 (100%)	0.00 (0%)	0.00 (0%)	0.15	0.11

Table 4: Relaxed-arc model results for $B = |V|$. Other instances were not solved in less than 3600 seconds.

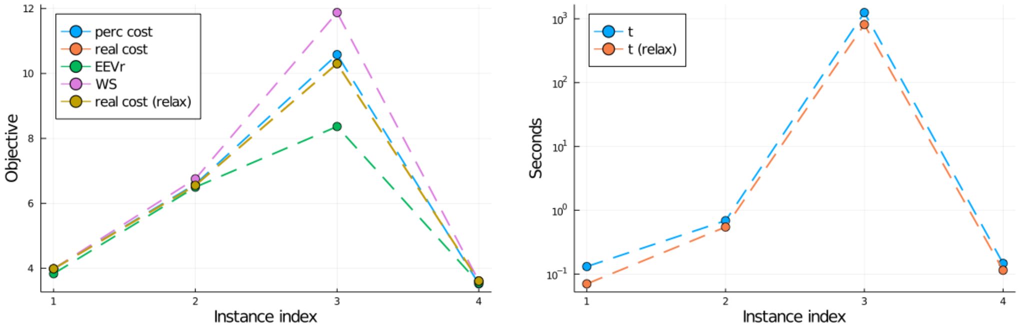

Figure 17: Relaxed-arc model results and solution time for $B = |V|$.

	MD-00001-x	perc cost	real cost	real cost (relax)	EEVr (%VSSr)	WS (%EVPI)	C_2	C_3	C_{4+}	t	t_r
1	00000010	3.98	3.99	3.99	3.84 (3.64%)	3.99 (0.01%)	1.03 (68.21%)	0.00 (0%)	0.48 (31.79%)	0.12	0.05
2	00000020	6.78	6.72	6.73	6.50 (3.35%)	6.75 (0.44%)	0.46 (29.11%)	0.12 (7.59%)	1.00 (63.29%)	0.19	0.17
3	00000030	11.91	11.66	11.75	11.46 (1.67%)	11.87 (1.86%)	0.53 (19.56%)	0.67 (24.72%)	1.51 (55.72%)	6.56	4.60
4	00000040	3.64	3.61	3.61	3.61 (0.00%)	3.61 (0.00%)	1.82 (100%)	0.00 (0%)	0.00 (0%)	0.14	0.12
5	00000050	13.99	13.99	13.94	13.39 (4.30%)	14.00 (0.04%)	0.28 (11.81%)	0.27 (11.39%)	1.65 (76.79%)	16.78	10.11
6	00000060	16.85	16.70	16.60	15.04 (9.99%)	16.82 (0.70%)	0.64 (23.70%)	0.24 (8.89%)	1.82 (67.41%)	215.07	168.28
7	00000070	15.08	15.03	14.85	13.80 (8.20%)	15.13 (0.65%)	0.60 (23.17%)	0.18 (6.95%)	1.81 (69.88%)	369.77	241.43

Table 5: Relaxed-arc model results for $B = |V|$. Other instances were not solved in less than 3600 seconds.

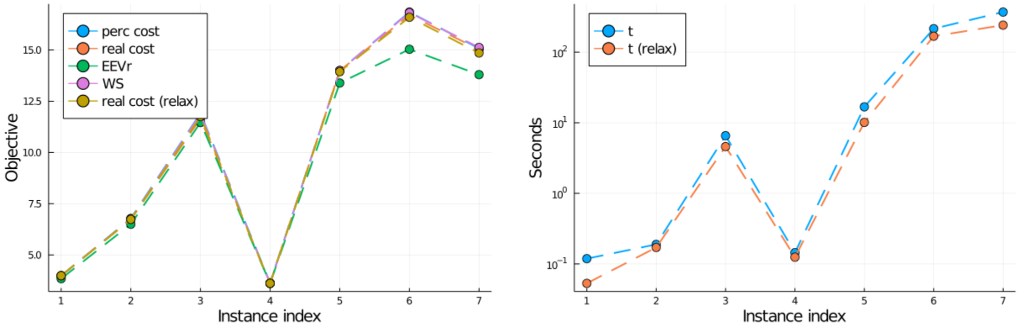

Figure 18: Relaxed-arc model results and solution time for $B = 2|V|$.

The first notable remark is that the $\mathbb{RA}$ model takes longer to be solved than the $\mathbb{MM}$ model. This can be explained by the larger number of variables and constraints as we can see in 4.1. Nevertheless, we can notice that the perceived cost, the real cost, the WS and the real cost of the relaxed problem are really close. Thus, we can draw the same conclusions as for the $\mathbb{MM}$ model : the relaxed solution seems to be a very tight upper bound and a budget $B = |V|$ or $B = 2|V|$ allows to retrieve enough information to nearly meet the WS value. We can notice that as for $\mathbb{MM}$, increasing the budget leads to a better solution and a smaller solution time. Finally, we can see that most of the cycles formed in the solution are of length at least 4. Usually, it is very hard to perform more than 5 transplants simultaneously so in general, $K \leq 4$. Thus, because we have relaxed the cycles length constraint, we may form infeasible solutions.

4.4.1 Budget sensibility

When fixing $B = |V|$, the perceived cost, the real cost, the WS value and the real cost of the relaxed problem seemed less close than when $B = 2|V|$. In the following, we study the sensibility of the solution to the budget with the budget factor b_f introduced in 3.6 for a *BinomialUNOS* failure generation rule for two instances.

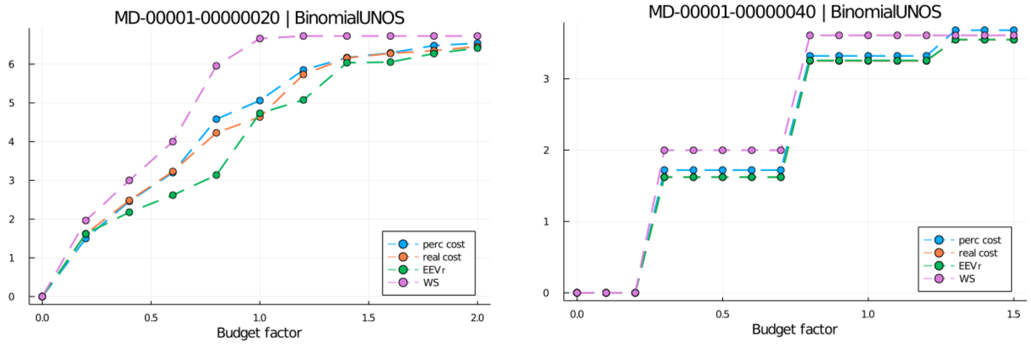

Figure 19: Budget sensibility for a *BinomialUNOS* failure generation rule.

The threshold b_f which allows the WS to reach it maximum value is about 1.2 in the left instance and about 0.8 in the right instance. Contrary to the $\mathbb{MM}$ model, we observe here that the real cost and the perceived cost do not stick to the EEV first. It seems that the VSS is larger than the one of $\mathbb{MM}$ before the WS reaches its maximum value. These costs grow with b_f and meet the WS value for $b_f = 1$ in the left instance and for $b_f = 0.8$ in the right instance.

The cost of the instance `MD-00001-00000040` evolves step-wise because of the particular structure of this instance. If we look at the table in 4.1, we can see that only 2 cycles of lengths 2 can be formed. When looking at the value of the cost of the solution (for $b_f \geq 0.8$), it seems that the solution contains between 3 and 4 transplants as in our instances, $w_a = 1, \forall\, a \in A$. In fact, the solution is only composed of the 2 cycles of lengths 2 and depending on the scenario, it is possible to finally perform transplants for 0, 1 or for the 2 cycles of length 2.

4.4.2 Failure generation sensibility

We now study the budget sensibility for the other generation rules with the same instances as above.

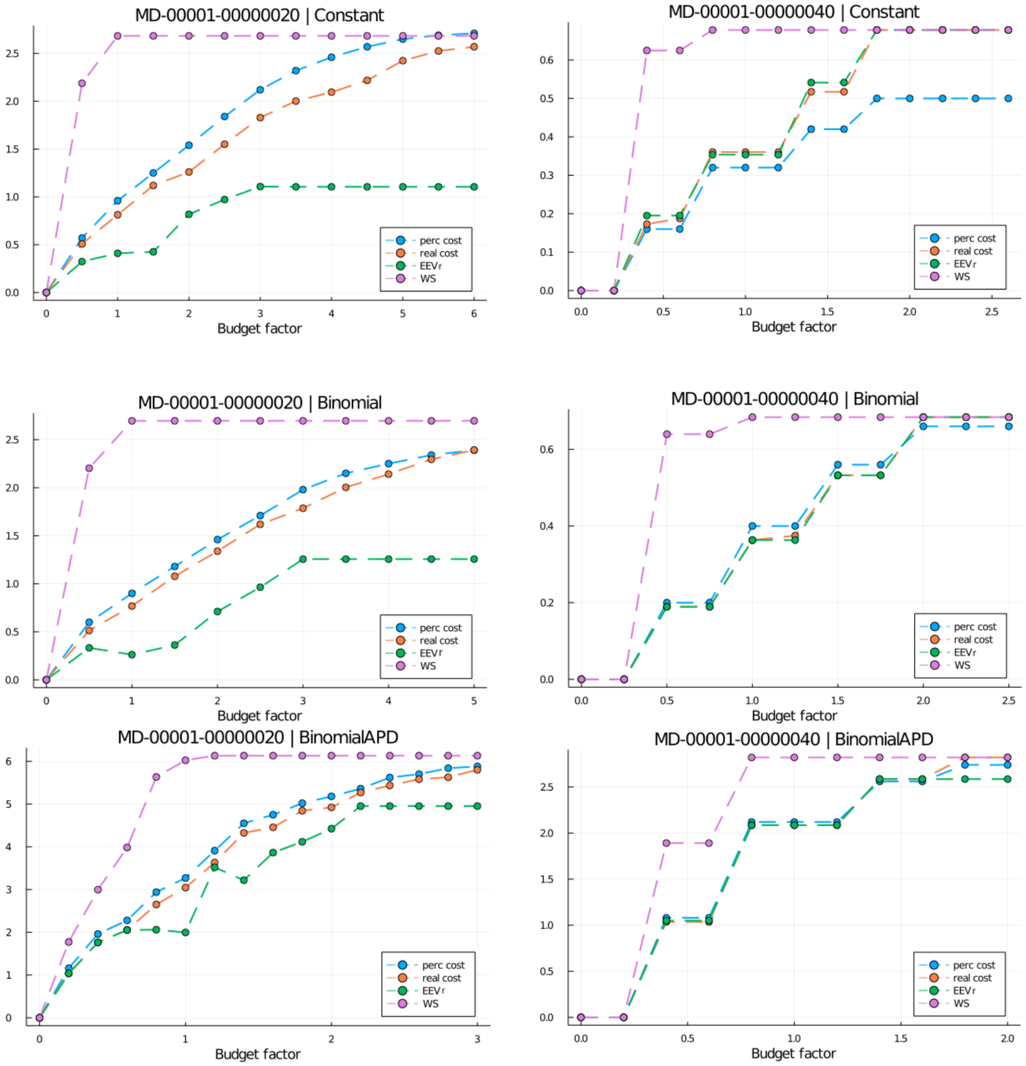

Figure 20: Budget sensibility for the *Constant*, the *Binomial* and the *BinomialAPD* failure generation rule.

Here again, we notice that the results are very different depending on our confidence in the transplant failure rate. If we know how to evaluate the failure rate of each transplant well enough, solving the stochastic problem will not bring a great gain. On the other hand, if we are not very good at estimating the transplant failure rate, we can expect to do many more transplants with a stochastic model rather than with an expected value model. Indeed, the stochastic problem will allow to create solutions that are more robust to transplant failures.

4.5 Recommended KEP policies

With the different results obtained with the two models, we can recommend particular strategies according to the cases considered.

Low budget and low cycle length

For the case where the KEP policy is to form cycles of length at most 2, we recommend to use a matching formulation. This allows to retrieve a model that is faster to solve and which can only give solutions involving cycles of length at most 2. However, if the available budget for HLA-crossmatch test is very constrained, it will be hard to create a solution better than the one of the expected-value problem. If we are very confident about the failure rate of each transplant, then we can hope to have a small gain. However, when the failure rate is not well known, the gain will be insignificant.

Large budget and low cycle length

When we still need to form cycles of length 2 but when the budget allowed for HLA-crossmatch tests is larger, we can hope to test smartly in order to be robust to transplant failures. We still need to use the matching formulation but by increasing the budget, it is possible that the solution meets the WS value. Furthermore, if the failure rate is well known, we can expect to have a gain when compared to the expected-value-problem. As the matching formulation can be solved in a reasonable time, it is recommended to use it anyway because it gives a better solution than the expected-value problem.

Low budget and large cycle length

In the case where cycles of length $K \geq 2$ are allowed but where the budget is still constrained, it can be interesting to use the relaxed-arc model. Indeed, its VSSr is larger than the one of the matching model in most cases. Furthermore, if the failure rate is known without much uncertainty, the gain can be very significant. However, this formulation usually creates solutions with very long cycles as their length is not constrained. Thus, it is important to make sure to be able to perform many transplants simultaneously in the hospitals.

Large budget and large cycle length

When large cycles are allowed and the HLA-crossmatch test budget is large, we can really hope to have a significant gain with the relaxed-arc model. Indeed, we can clearly see the large VSSr, especially when the failure rate is well known. Seeing the gains that can be made, we can say that it is really worthwhile to increase the funds allocated to KEPs and hospitals to allow for greater freedom in the solutions considered. However, we must take into account that solution are likely to exceed the maximum cycle length fixed. Thus, it is important to make sure to be able to perform many transplants simultaneously in the hospitals.

5 Conclusion

In this project, we were interested in the kidney exchange problem. This problem is particularly difficult because we have to maximize the number of transplants ultimately carried out in a KEP run, but we do not know which ones will be possible when taking the decision. To do this, we have the possibility to perform a certain number of HLA-crossmatch compatibility tests, but this number is often much lower than the total number of possible transplants. It is therefore necessary to define strategies to create transplant cycles that are robust to compatibility failures. We have studied a first model which is not usable in practice because of its non-compact formulation. We chose to eliminate the complicated constraint of maximum cycle length either by relaxing it or by considering that only cycles of length two were possible. These two options give rise to simpler problems to solve. In the matching model where only size two cycles are allowed, we manage to have a small gain on the expected value problem but it is quite hard to reach WS for a small budget. On the other hand, the gain is significantly higher for the arc relaxation model. This second model is particularly interesting whatever the test budget allowed. It creates a much more robust solution to compatibility failures. However, this model is much longer to solve. It could be interesting to develop structure-exploiting methods such as the L-shaped method in order to efficiently solve this problem. As this formulation can produce transplant cycles that are too long to be handled in practice, it could also be interesting to work on a post-treatment heuristic allowing to reduce the cycle length to fit practical constraints.

References

[Kar71] Richard M Karp. "A simple derivation of Edmonds' algorithm for optimum branchings". In: *Networks* 1.3 (1971), pp. 265–272.

[Rap86] Rapaport. *The case for a living emotionally related international kidney donor exchange registry*. 1986.

[Kin88] AJ King. "Stochastic programming problems: Examples from the literature". In: *Numerical techniques for stochastic optimization* (1988), pp. 543–567.

[LL93] Gilbert Laporte and Franccois V Louveaux. "The integer L-shaped method for stochastic integer programs with complete recourse". In: *Operations research letters* 13.3 (1993), pp. 133–142.

[Ros+97] Lainie Friedman Ross et al. *Ethics of a paired-kidney-exchange program*. 1997.

[ABS07a] David J Abraham, Avrim Blum, and Tuomas Sandholm. "Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges". In: *Proceedings of the 8th ACM conference on Electronic commerce*. 2007, pp. 295–304.

[ABS07b] David J. Abraham, Avrim Blum, and Tuomas Sandholm. "Clearing Algorithms for Barter Exchange Markets: Enabling Nationwide Kidney Exchanges". In: *Proceedings of the 8th ACM Conference on Electronic Commerce*. San Diego, California, USA: Association for Computing Machinery, 2007, pp. 295–304.

[Bab+08] Keshwar Baboolal et al. "The cost of renal dialysis in a UK setting—a multicentre study". In: *Nephrology Dialysis Transplantation* 23.6 (Jan. 2008), pp. 1982–1989.

[BMR09] Péter Biro, David F Manlove, and Romeo Rizzi. "Maximum weight cycle packing in directed graphs, with application to kidney exchange programs". In: *Discrete Mathematics, Algorithms and Applications* 1.04 (2009), pp. 499–517.

[Ünv10] M Utku Ünver. "Dynamic kidney exchange". In: *The Review of Economic Studies* 77.1 (2010), pp. 372–414.

[AJM13] Itai Ashlagi, Patrick Jaillet, and Vahideh H Manshadi. "Kidney exchange in dynamic sparse heterogenous pools". In: *arXiv preprint arXiv:1301.3509* (2013).

[DPS13] John P Dickerson, Ariel D Procaccia, and Tuomas Sandholm. "Failure-aware kidney exchange". In: *Proceedings of the fourteenth ACM conference on Electronic commerce*. 2013, pp. 323–340.

[MW13] Nicholas Mattei and Toby Walsh. "PrefLib: A Library of Preference Data". In: *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT 2013)*. Lecture Notes in Artificial Intelligence. Springer, 2013.

[And14] Ross Michael Anderson. "Stochastic models and data driven simulations for healthcare operations". PhD thesis. Massachusetts Institute of Technology, 2014.

[AR14] Itai Ashlagi and Alvin E Roth. "Free riding and participation in large scale, multi-hospital kidney exchange". In: *Theoretical Economics* 9.3 (2014), pp. 817–863.

[GKW14] Kristiaan M Glorie, J Joris van de Klundert, and Albert PM Wagelmans. "Kidney exchange with long chains: An efficient pricing algorithm for clearing barter exchanges with branch-and-price". In: *Manufacturing & Service Operations Management* 16.4 (2014), pp. 498–512.

[Blu+15] Avrim Blum et al. "Ignorance is almost bliss: Near-optimal stochastic matching with few queries". In: *Proceedings of the Sixteenth ACM Conference on Economics and Computation*. 2015, pp. 325–342.

[Glo+15] Kristiaan Glorie et al. *Robust models for the kidney exchange problem*. Tech. rep. Working paper, 2015.

[MO15] David F Manlove and Gregg O'malley. "Paired and altruistic kidney donation in the UK: Algorithms and experimentation". In: *Journal of Experimental Algorithmics (JEA)* 19 (2015), pp. 1–21.

[TP15] Panos Toulis and David C. Parkes. "Design and analysis of multi-hospital kidney exchange mechanisms using random graphs". In: *Games and Economic Behavior* 91 (2015), pp. 360–382. ISSN: 0899-8256. DOI: https://doi.org/10.1016/j.geb.2015.01.001. URL: http://www.sciencedirect.com/science/article/pii/S0899825615000020.

[AK16] Tommy Andersson and Jörgen Kratz. "Pairwise kidney exchange over the blood group barrier". In: *Lund University Department of Economics Working Paper* 2016 (2016), p. 11.

[Dic+16] John P Dickerson et al. "Position-indexed formulations for kidney exchange". In: *Proceedings of the 2016 ACM Conference on Economics and Computation*. 2016, pp. 25–42.

[Ant17] Christophe Antoine Corinne; Legendre. "Organ Transplantation in France". In: *Transplantation* 101.3 (2017), pp. 445–448.

[Bio17] Agence de la Biomedecine. *Le rapport médical et scientifique du prélèvement et de la greffe en France*. https://www.agence-biomedecine.fr/annexes/bilan2017/donnees/organes/06-rein/synthese.htm#tR8. 2017.

[Bof+17] Miquel Bofill et al. "The Spanish Kidney Exchange Model: Study of Computation-Based Alternatives to the Current Procedure". In: *Artificial Intelligence in Medicine*. Ed. by Annette ten Teije et al. Cham: Springer International Publishing, 2017, pp. 272–277. ISBN: 978-3-319-59758-4.

[Eur17] Council of Europe. *European Directorate for the Quality of Medicines*. 2017.

[Nag+17] Mohsen Naghavi et al. "Global, regional, and national age-sex specific mortality for 264 causes of death, 1980–2016: a systematic analysis for the Global Burden of Disease Study 2016". In: *The Lancet* 390.10100 (2017), pp. 1151–1210.

[HE18] Institute for Health Metrics and Evaluation. *GBD compare: Global burden of disease study*. http://www.healthdata.org/. 2018.

[Lee+18] Hyunwoo Lee et al. "Accounting for fairness in a two-stage stochastic programming model for kidney exchange programs". In: *International journal of environmental research and public health* 15.7 (2018), p. 1491.

[AKL19] Sepehr Assadi, Sanjeev Khanna, and Yang Li. "The stochastic matching problem with (very) few queries". In: *ACM Transactions on Economics and Computation (TEAC)* 7.3 (2019), pp. 1–19.

[Bir+19] Péter Biró et al. "Modelling and optimisation in european kidney exchange programmes". In: *European Journal of Operational Research* (2019).

[Car+20] Margarida Carvalho et al. "Robust models for the kidney exchange problem". In: *INFORMS Journal on Computing* (2020).

[RBA20] Lizeth Carolina Riascos-Álvarez, Merve Bodur, and Dionne M Aleman. "A Branch-and-Price Algorithm Enhanced by Decision Diagrams for the Kidney Exchange Problem". In: *arXiv preprint arXiv:2009.13715* (2020).