Testing re-optimisation strategies in international kidney exchange programmes by the ENCKEP simulator

Lilla Matyasi[1,2] · Péter Biró[2,3]

Abstract

We tested re-optimisation strategies for international kidney exchange programmes using the simulator developed by the ENCKEP COST Action. Kidney exchange programmes (KEPs) are operating in most of the European countries to facilitate the exchange of kidney donors for the recipients with incompatible living donors. The optimal solutions for national and international KEPs are typically selected in every three months based on the compatibilities estimated on the individual immunological data. However, these estimations are not always accurate, and if a positive crossmatch is found in the laboratory crossmatch tests then the corresponding exchange cycle (or chain) must be cancelled. Depending on the matching process, the coordinators may use different re-optimisation strategies to repair the failed solutions. We examine the effects of using multiple rounds of re-optimisation with different optimisation strategies, such as fixing good cycles in the intermediate solutions or prioritising transplants with negative crossmatch tests in previous rounds. In the context of international KEPs we also consider the possibility of testing and prioritising national transplants in the solutions. We measure the performance of these policies regarding the number of transplants and the number of compatibility tests conducted in a time period. By extending our results presented in [16], we performed simulations for a large number of instances to measure the effects of various re-optimisation policies. Our main findings show a clear trade-off in the number of transplants versus the number of tests and re-optimisation rounds.

Keywords Kidney exchange programmes · Integer programming · Computer simulation

✉ Péter Biró
  peter.biro@krtk.hu

  Lilla Matyasi
  lilla.matyasi@krtk.hu

[1] Budapest University of Technology and Economics, Budapest, Hungary

[2] Institute of Economics, KRTK, Budapest, Hungary

[3] Corvinus University of Budapest, Budapest, Hungary

🖄 Springer

1 Introduction

Kidney exchange programmes (KEPs) help patients with willing donors, who would donate their kidneys to their loved ones, but the transplantation is not possible due to immunological incompatibility. The goal of a KEP is to perform exchanges (or chains triggered by altruistic donors) between incompatible patient-donor pairs so that each patient can receive a compatible kidney. For example, if the recipient of the first pair can receive a kidney from the donor of the second pair, and the recipient of the second pair can receive a kidney from the donor of the first pair, a pairwise kidney exchange can be performed between them. National KEPs have started to operate in all the large countries in Europe, except Germany, the first was the Dutch programme in 2004, and the largest is the UK programme (D.F. Manlove D, G. 2014), see details on the European programmes in Biró et al. (2019).

The immunological compatibility of patients and donors in a KEP is decided by crossmatch tests. At the beginning of the process *virtual crossmatch tests* are performed based on the immunological data of the donors and recipients. These are quick and inexpensive to perform, however, these tests are not completely accurate, especially if they are conducted at low resolution, and the recipient is sensitised, i.e., has high PRA value. Based on the results of the virtual crossmatch tests, a *virtual compatibility graph* is created, where nodes denote the patient-donor pairs and arcs denote the possible transplants. Then, an optimal set of exchange cycles and altruistic chains are selected by an optimisation algorithm, typically using Integer Programming (IP) technique, as already suggested in Abraham et al. (2007). The European applications use a variety of optimisation criteria, but the main objective is to maximise the number of transplants (Modelling and optimisation in European Kidney Exchange Programmes et al. 2020).

However, a laboratory crossmatch test must also be performed prior to the actual implantation, which is much more costly, time consuming, and if it gives a positive result, the exchange cycle in which this recipient-donor pair was involved cannot be accomplished. (Failure can also come from other medical reasons, such as unsuccessful desensitisation of a recipient, or simply due to the illness of some patients.) One of the main challenges in the mechanism design of the KEPs is how to deal with such failures. The related strategies are called recourse, failure-aware or re-optimisation policies, and they have been extensively studied in the literature (see the details below in the Literature review).

The European practices of re-optimisation varies across countries. The Dutch application uses a single HLA-lab in Leiden to collect all the blood samples, and therefore they can conduct the laboratory crossmatch tests quickly (within 2–3 days) for a selected solution, and thus they have time to conduct even 5–6 re-optimisation rounds, if needed. Because of this flexibility they consider 4-cycles as well in the solutions, even though they still use low-resolution HLA-typing (that is less accurate than high-resolution, but cheaper). When a failure happens in a selected cycle then they completely re-optimise their solution (so potentially cancel also some good cycles). In the two largest European programmes, the UK

and Spanish programmes, the pools are much larger and multiple HLA-labs are involved in the laboratory testing phase, so the samples are shipped across the country. This makes it impossible to conduct multiple re-optimisation rounds. Partly because of such risks of failures they only allow 2- and 3-cycles in their proposed solutions. In the UK, they only consider re-optimisation within the cancelled 3-cycles for the so-called embedded 2-cycles. In fact, they first maximise the number of 2-cycles and 3-cycles with embedded 2-cycles in their optimisation (D.F. Manlove D, G. 2014). The Spanish programme also has one re-optimisation round, and they also prioritise the 3-cycles with embedded 2-cycles in their solution, but their approach is more flexible, the new cycles may involve new donor-recipient pairs as well.

Multi-country collaborations have also been established in several regions in Europe, in between Vienna and Prague since 2016 (Böhmig et al. 2017), extended with Israeli hospitals recently, in Spain, Portugal and Italy since 2018 (Valentin et al. 2019), and for Sweden and Denmark by Scandiatransplant from 2019. International cooperation offers many new opportunities, but also new challenges to solve.

Literature review Kidney exchange programmes have been studied extensively in the operations research literature in the last 15 years, as the solution technique used for solving the underlying matching problems are typically based on integer programming models (Abraham et al. 2007; Modelling and optimisation in European Kidney Exchange Programmes et al. 2020). To study the effect of different optimisation constraints and objectives, the most realistic way is to conduct long term computer simulations. A survey on such simulation results can be found in Santos et al. (2017). In our research, we used and further developed the kidney exchange simulator of the ENCKEP (European Network for Collaboration on Kidney Exchange Programmes) COST Action (Druzsin et al. 2022; Klimentova et al. 2022; Website of the CA15210 COST Action: European Network for Collaboration on Kidney Exchange Programmes (ENCKEP) xxxx;). This simulator can take real historical or generated instances of national or international KEPs as input and conduct the match runs for a given time period under various optimisation criteria and collaboration policies.

The recourse, failure-aware or re-optimisation policies or robust optimisation have been studied in several papers in the last decade. When the probability of failure can be estimated e.g., from the PRA-level of the recipient, then maximising the expected number of successful transplants is a reasonable strategy (Alvelos et al. 2019a; Dickerson et al. 2013; Klimentova et al. 2016; Smeulders et al. 2022). A priori probing means a pre-selection of some transplants to be tested in laboratory in order to improve the expected number of successful transplants in the (final) match run (Alvelos and Viana 2017; Alvelos et al. 2019b). Robust optimisation for KEPs with failure means that the set of recipients selected for transplants after re-optimisation should not change much assuming a worst case scenario in a so-called defender-attacker-defender model (Blom et al. 2022; Carvalho et al. 2021).

International KEPs, motivated by the European practice, have been studied recently with regard to the mutual benefits of the countries (Klimentova et al. 2021), and the special restrictions in international exchanges (Mincu et al. 2021). Analysing re-optimisation strategies in the context of an international KEP is a novelty of this paper.

Contributions In this paper we present our results on testing different re-optimisation strategies with the ENCKEP-simulator for national and international KEPs by simulations conducted on 1k generated instances for each parameter and policy setting. The configuration files for the test datasets, the used policy and objective files are available at the https://github.com/lilla06/Testing-re-optimisation-strategies-in-international-kidney-exchange-programmes-by-ENCKEP link. Our main simulation results show that re-optimisation strategies can indeed improve the performance of both national and international KEPs in the long run. In particular, we facilitated the re-optimisation in any given number of rounds, the possibility of fixing the good cycles, and also to track and prioritise those transplants where the compatibility was already proved in the laboratory tests. Regarding the international features, we enabled the prioritisation of internal (i.e., national) transplants, and the separate laboratory testing of the internal transplants within the cycles selected. All the new features that we implemented resulted in some improvements either by increasing the number of transplants or by decreasing the number of crossmatch tests.

Organisation of the paper In Sect. 2 we describe the new features that we implemented in the ENCKEP simulator to study various re-optimisation strategies for national and international KEPs and measure their performance in terms of the number of transplants and the number of crossmatch tests conducted. In Sect. 3 we describe our main findings from the national simulations and the international simulations are described in Sect. 4.

2 New features implemented in the ENCKEP-simulator

We used the ENCKEP simulator for our study, however, we wanted to experiment with further settings and test scenarios related to re-optimisation, and therefore we implemented some new functions that we describe in this Section. The ENCKEP-simulator already had the option of having failed edges and pairs in the input, that can be generated by setting the edge and pair failure probabilities. These failures are pre-set, but the simulator will detect them only when conducting the match runs. When some edge or pair fails in a selected cycle or chain, the simulator denotes them in the solution, by cancelling the effected cycles and chains. The failed edges and nodes are then removed from the pool for future match runs. However, the simulator did not allow re-optimisation within a match run, except the so-called recourse option. If the user selects this option, then a failed cycle or chain is repaired by the best sub-solution within the cycle. Our most important new development in the ENCKEP-simulator was the implementation of the re-optimisation option which is a frequently used protocol to reduce the effect of a positive laboratory cross match test in European KEPs.

Implementing re-optimisation has also allowed us to develop a number of other features based on re-optimisation which includes fixing of good cycles and a new optimisation criterion, the maximisation of the tested edges in the solution. Furthermore, we added another new optimisation criteria to maximise the number of internal transplants selected in the exchange cycles, or in other words to minimise the number of international transplants.

In order to evaluate the results obtained, besides the number of successful transplants we also needed to count the number of crossmatches and positive crossmatches that were carried out in a simulation. The number of crossmatches was tracked already in the original simulator, but due to re-optimisation this had to be changed and improved significantly. Note that if one would not care about the number of laboratory crossmatch tests, then the optimal policy would be to test all the possible transplants in the laboratory before the match run. But this is not practical in large programmes, so we shall always aim to a good balance to maximise the number of successful transplants and to minimise the number of crossmatch tests and/or the number of re-optimisation rounds.

2.1 Re-optimisation

Re-optimisation in the basic setting means that when any of the transplants in the proposed solution fails then the solution will be completely re-optimised, without caring about the cycles that could have been successfully implemented. This is the practice in the Netherlands, where the usage of a central HLA-laboratory enables the quick laboratory testing.

When implementing the re-optimisation feature, we aimed to create a solution that is as general as possible, so that many different cases can be easily simulated. Therefore, we decided to implement the re-optimisation separately for the different types of collaborations. This was necessary because, with the exception of the borderless cooperation, the pools interact with each other.

2.1.1 Re-optimisation without international cooperation

In this case the national KEPs do not cooperate, therefore their results are not affecting each other. If there is no cooperation between pools, each pool will be re-optimised for the number of rounds specified in one of the input files, in the so-called policy file. Since there is no effect of optimising one pool on the other pools, we can perform these rounds of re-optimisation in all of them at once in a parallel way.

After our modifications, the main algorithm of the simulator chooses the highest re-optimisation number specified for some of the pools, and it performs a matching

```
{
"general":{
      …
      "optimization_runs": 1
},
"NL": {
      …
      "optimization_runs": 7
      }
}
```

Fig. 1 Setting the number of re-optimisation rounds separately for each pool when the countries do not collaborate with each other

round in each pool which can have that many re-optimisation rounds. Since the pools are separate, the results can be collected and stored separately (Fig. 1).

The modified main algorithm of the simulator first checks what the largest "optimization_runs" value is in the policy file at the beginning of each matching round and runs that many optimisation rounds. In each optimisation round it searches for the optimal solution (as in the original algorithm), but it does not save the result obtained automatically, since that may not be the final one.

At the beginning of each optimisation round, it collects the pools that should be optimised in that round: these are the pools that have active pairs and are allowed to run as many optimisation rounds as the index of the current optimisation round.

Since there is no cooperation between pools, each selected cycle or chain is within a single pool. For all the selected cycles and chains in the optimal solution, it checks whether they are in a pool that should be optimised in that round. If a cycle is not in such an active pool, then it does not get tested again, since the final solution has already been fixed for that pool in some previous optimisation round. If the cycle is in one of the active pools, it checks whether the given cycle or chain passes all the laboratory crossmatch tests and either it saves the implementable (sub) solution for later processing or deletes the failed edges and pairs, and so the simulator can search for a new solution in the next optimisation round with this new knowledge.

The simulator saves the solution for each pool separately in a dictionary. When all the optimisation rounds have been performed, we have the optimal solution for each pool in the dictionary. However, it is possible to reach the optimal solution in an earlier optimisation round. It occurs when all selected edges pass the laboratory crossmatch tests for an optimisation round. In this case it would be unnecessary to run more optimisation rounds, so we keep track of each optimisation round to see if there was a failed cycle or chain and we finish the re-optimisation in that pool if there was no failure.

2.1.2 Re-optimisation with international cooperation

In this case, the re-optimisation performed on each pool can affect the other pools, so it does matter in which order we test and re-optimise the pools.

To facilitate the study of the most general re-optimisation policies, we made it possible to specify which pools should be (re-)optimised in the re-optimisation rounds (Fig. 2).

```
      {
"general":{
      …
      "optimization_rounds":[["NL"],["UK","NL"],      ["UK"],
["NL"], ["general"]]
      }
}
```

Fig. 2 Specifying re-optimisation rounds for international KEPs. In this example the first round is restricted to the Netherlands, the second round for the UK and Netherlands, etc., finally a testing of all the selected (international) cycles are performed in the general pool

Since a chain or cycle that spans multiple pools will only be fully tested if all of its edges are tested, it is necessary to test all edges (i.e., the selected edges in the general pool) in the last re-optimisation round in order to check the final result. So, to avoid possible inconsistency, if the last element of the "optimization_rounds" specified in the policy file (i.e., the last optimisation round) does not contain "general", the software adds an extra optimisation round to the rounds, in which all edges are tested.

In this case, as many re-optimisation rounds should be run in each match round as the number of the elements in the optimisation rounds array in the policy file. In each optimisation round, the pools to be optimised are listed in the next element of the array.

To collect the results, we used a dictionary as in the previous case. After the algorithm gets the optimal solution, it checks the correctness of each cycle and chain, but it cannot save the errors for all pairs and edges, only for those in the pools re-optimised in that round, so it filters out the edges that it could not test in that re-optimisation round from the pre-set array of failed edges and pairs.

If the simulator tests all edges, or if the tested cycle or chain is entirely in a pool for which it is re-optimising, then everything works as before, except that now there may be international cycles and chains that are in the "general" pool. However, if none of the above cases apply, then we cannot test all the edges of the selected cycle or chain, only those that are in the pool we are optimising for in the current round. Since just some of the edges can get tested, we cannot confirm the correctness of the whole cycle or chain, we can only detect a failure if some of the edges or pairs would fail.

The algorithm continues from here in the same way as in the previous case with no collaboration, except that here we cannot stop earlier for a country even if all selected and tested rounds and chains have been successful in a re-optimisation round, since a failure in another country may affect the global solution.

2.2 Fixing the good cycles

Fixing the good cycles means that if a cycle or chain is successful in a re-optimisation round, we can fix this cycle, and then it will be surely part of the final solution and only the remaining pairs will be used in the next rounds to find the next optimal solutions. In this case, we may not get the globally optimal solution as the final solution, but this can be still a beneficial strategy, because it can lead to a relatively good solution much faster, with fewer laboratory tests conducted.

In our new implementation, it is possible to specify in the policy file whether the user wishes to fix cycles and chains in the pool. If an empty string is specified then no, in all other cases it is allowed (Fig. 3).

We have modified the algorithm so that if a cycle or chain is fully tested and passed the laboratory tests, and for the respective pool the cycle-fixing option is enabled, then the pairs in the cycle or chain get immediately removed from the active pairs and the algorithm saves this cycle as a fixed part of the final solution.

```
{
"general":{
      …
      "fix_cycles": ""           # not allowed
      },
"UK": {
      …
      "fix_cycles": "true"      # fixing the good cycles
      }
}
```

Fig. 3 Setting the option of fixing the cycles in the policy file. In the example the cycles are not fixed in the merged pool, but they are fixed, when the cycle is selected in the UK pool

2.3 Maximising the number of confirmed edges

If an edge has already been tested and the laboratory crossmatch test has shown that the donor and the patient is indeed compatible, then, assuming that the recipient is not affected by an immunologically sensitive event, we may assume that this donor-recipient pair remains compatible, so this edge does not need to be tested again in the lab. Therefore, if this transplant does not take place in match run, for example, because the corresponding cycle is cancelled due to the failure of another arc in that cycle, then this knowledge can be used in subsequent rounds to reduce the number of crossmatch tests, and to improve the probability of the selected cycles getting performed.

To implement the goal of maximisation of confirmed edges, we introduced a new optimisation criterion that computes a weight for each cycle and chain based on the number of edges in the cycle that had been previously tested by the simulator. Each of these edges adds a pre-set value to the total weight of the cycle, that can be set in the objective.json input file. This new optimisation criterion can be used either as a new level in the hierarchic optimisation policy, or at the last level of the hierarchic optimisation, where different objectives are considered with different weights, see (Klimentova et al. 2022).

The function returns the total weight for each cycle and chain, so the solver can find a solution that maximises this value, among other optimisation criteria.

2.4 Maximising the number of internal edges

International KEPs need to test international and national (internal) edges in the lab. The latter may be simpler, cheaper, or quicker, especially if the blood samples are stored in a common location (e.g., in Leiden for the Netherlands). Testing international edges, however, can be more complicated and time-consuming.

As we have already presented, we facilitated the possibility of testing the national and international edges separately in a re-optimisation strategy. However, it can also be reasonable to prioritise the internal edges in the solution, by taking this objective into account in the optimisation. The implementation of this criterion does not differ much from the previous criterion, just in this case the

criterion provides weights on the cycles and chains based on the number of internal edges.

Note that internal edges can also mean edges belonging to the same HLA-centre in a large country, such as the UK or Spain. So, our simulations can also give insight when the consideration is to conduct quick laboratory tests for possible transplants, where the donor and recipient both belong to the same HLA-centre.

2.5 An illustrative example

We provide a simple example to illustrate the re-optimisation processes considered in our international simulations. all other cases it is allowed.
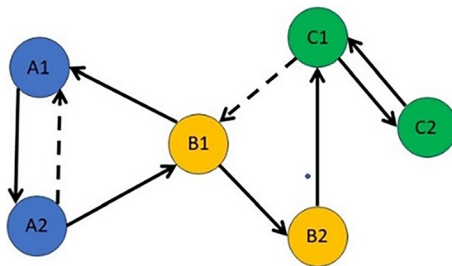
Suppose that we have three countries, and each has two recipient-donor pairs, A1, A2 in country A, B1, B2 in country B, and C1, C2 in country C. The above graph represents the (virtual) compatibilities, as follows. Every arc corresponds to a possible transplant that was expected to be compatible in the virtual crossmatch test. A dashed line means that there is a positive crossmatch test in the laboratory, so the compatibility was incorrectly estimated. A solid line means that the estimated compatibility is confirmed in the laboratory test. Suppose that we consider an international matching run, and therefore we select a solution with 2-cycles and 3-cycles such that the number of transplants in maximised in the virtual compatibility graph. There are two optimal solutions, with five expected transplants in each, namely solution 1: [(A1,A2,B1), (C1,C2)], and solution 2: [(A1,A2), (B1,B2,C1)]. After one of the solutions is chosen by the optimisation policy, the selected transplants are tested in the laboratory. In case of solution 1, there is no failure. However, in case of solution 2, the transplants from A2 to A1, and from C1 to B1 fail, and depending on the re-optimisation policy the process might continue. In some practical applications, the testing of internal (national) arcs can be easier, since the blood samples are stored at the same central HLA-laboratory (such as the central lab in Leiden in the Netherlands). This can mean that in the re-optimisation strategy national testing rounds are conducted first, and in case of a failure a quick re-optimisation may take place. Furthermore, the policy makers can also decide to maximise the number of internal arcs selected in the international solution, and so the chance of detecting a failure early can be improved.

3 Results for national KEPs

In this paper we wanted to examine the long-term effect of the different re-optimisation strategies, therefore we run all the simulations for a five-year interval on 1000 generated instances.

To make the simulation realistic, we used the generator function of the ENCKEP-simulator, where the parameters were set according to a historical data from the Spanish KEP. The most important parameter is the PRA distribution of the pool, that can be very special for KEP patients. The main reason for this difference is that patients with low PRA values are more likely to get a compatible kidney outside of a KEP, from a direct willing donor or from the deceased waiting list. Furthermore, patients with high PRA values have less chance to get matched in the KEP pools as well, therefore they spent more time in the programmes, and so they can accumulate in the pools. We measured the PRA distribution for the patients in the Spanish KEP in a historical dataset for 2009–2022, and we used that distribution to generate the 1000 instances.

For the generator function of the ENCKEP-simulator we used the parameters described in Fig. 4. Besides the PRA-distribution of the recipients, one can set the blood-type distributions for both the donors and recipients, the distribution for the number of registered donors for the recipients (one recipient can have multiple donors registered). Regarding the age, the arrival time and duration of stay, it is possible to set the distributions separately for (normal) incompatible donor-recipient pairs, for compatible pairs and for altruistic donors.

For the simulations with only one pool, we used general settings. In the policy file we set the interval between matches to 90 days, which is a typical value in European KEPs. Initially we did not enable any type of error correction, such as recourse, re-optimisation or fixing the good cycles (Fig. 5).

In the objective input file, we specified the general optimisation criteria. In the first level of the hierarchical optimisation process we maximise the number of transplants (which criterion is used in the first place in every European KEP, except in the UK), then after fixing the number of transplants to be maximal, we maximised

```
{
    "name": "general",
    "dist_blood_type": {"O": 0.44, "A": 0.42, "B": 0.1, "AB": 0.04},
    "dist_num_donors": {"1": 0.9, "2": 0.1, "3": 0.0},
    "dist_pra":    {"[0,    9]":    0.17836981566820276,   "[10,    69]":
0.15529953917050693,  "[70,  89]":  0.12206221198156683,  "[90,  94]":
0.05904377880184332, "[95, 99]": 0.48522465437788015},
    "dist_age": {"patient": {"(18, 73)": 1.0}, "donor": {"(18, 73)":
1.0}, "altruistic": {"(18, 73)": 1.0}},
    "arrival_time": {"inc": {"avg": 7.5}, "com": {"avg": 37.5}, "alt":
{"avg": 45.0}},
    "duration_stay": {"inc": {"avg": 365.0, "min": 60, "max": 2190},
"com": {"avg": 365.0, "min": 60, "max": 2190}, "alt": {"avg": 180.0,
"min": 60, "max": 240}},
    "prob_fail": {"inc": {"max": 0.0}, "com": {"max": 0.0}, "alt":
{"max": 0.0}},
    "prob_pass_hla": {"*-*": 1.0}
}
```

Fig. 4 The config file for the general pool

```
{
"general":{
        "matching_interval": 90,
        "collaboration_approach": "none",
        "max_cycle_length": 3,
        "max_chain_length": 3,
        "internal_recourse": "",
        "continue_chain": "true",
        "optimization_rounds": [],
        "fix_cycles": "",
        "optimization_runs": 1
    }
}
```

Fig. 5 The general policy file without error correction

the number of cycles, since shorter cycles or chains are more likely to be successful, and finally, as the third objective, we prioritised patients with high PRA value. See more details about the these and other objectives available in the ENCKEP-simulator in Klimentova et al. (2022) and (Druzsin et al. 2022) (Fig. 6).

3.1 Performance without any error correcting methods

We conducted the baseline simulations without re-optimisation using the above general setting on all 1000 instances to provide comparison for the further simulations with novel settings. The two most important evaluation metrics are the number of performed transplants and the number of laboratory crossmatch tests conducted in the matching process, however, we also measured the number of positive crossmatch tests in some of the cases. Besides the these performance measures, we also included

```
{
        "general": {
                "lexicographic_order": [
                        "max_num_transplants",
                        "max_num_cycles",
                        "prior_highly_sensitised_recipients"
                ],
                "cycle_level_weighted": [],
                "arc_level_weighted": [],
                "criterion_parameters": {
                        "max_num_transplants": {
                                "score": 1
                        },
                        "max_num_cycles": {
                                "count_chains": "true",
                                "score": 1
                        },
                        "prior_highly_sensitised_recipients": {
                                "function": "reciprocal",
                                "score": 30
                        }
                }
        }
}
```

Fig. 6 The general objective file
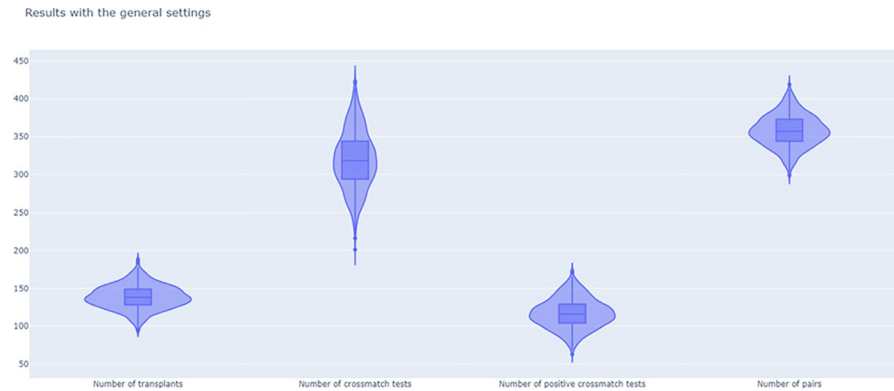
Fig. 7 Results with the general settings: number of transplants, crossmatch tests, positive crossmatch tests, and number of registrations over the five-year period
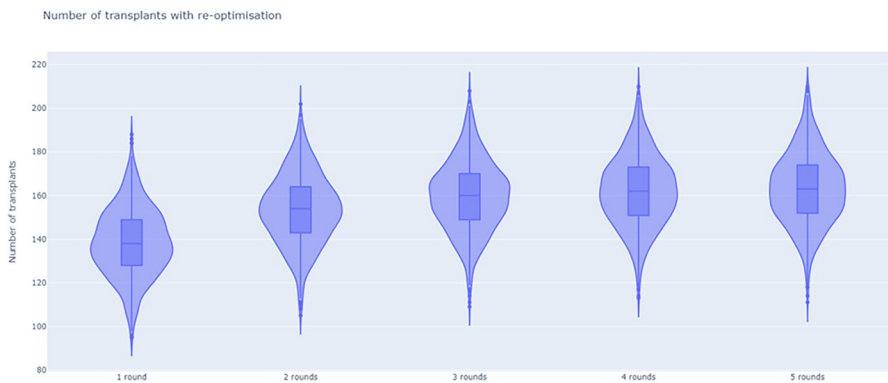

Fig. 8 Number of transplants according to the number of re-optimisation rounds

the number of registrations over the five-year period in Fig. 7, and so the success rate (number of transplants per number of registrations) can also be estimated. Every time we display the average result on the 1000 instances, and for Figs. 7 and 8 we also included the distributions of these values using violin plots.

3.2 Re-optimisation

For all 1000 instances we repeated the simulations with a maximum of 2, 3, 4 and 5 re-optimisation rounds. The number of transplants is shown on Fig. 8 and the number of crossmatch tests in Fig. 9.

The second re-optimisation round made the largest difference: it increased the number of transplants by 11%. Further re-optimisation rounds produced slightly smaller improvements, with an average improvement of 17.9 percent by the end of all five rounds.
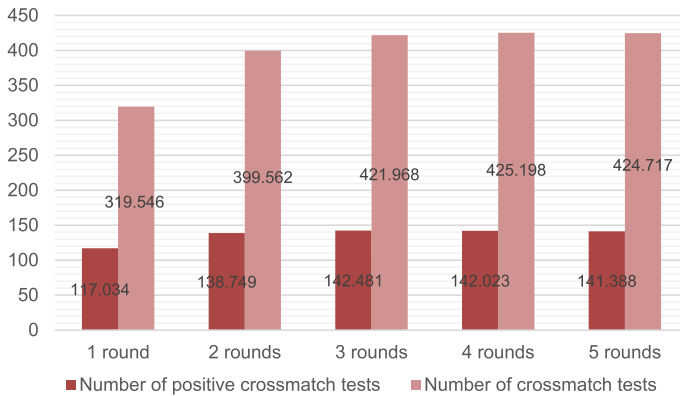
Fig. 9 Number of cross match and positive crossmatch tests with re-optimisation

Naturally the re-optimisation also increases the number of crossmatch and positive crossmatch tests.

With all 5 re-optimisation rounds the simulator made 20% more crossmatch tests than without re-optimisation, and the number of positive crossmatch tests increased by 30% compared to the baseline scenario.

3.3 Recourse

Recourse is an option that was already implemented in the ENCKEP-simulator. If the user selects this option, then a simple re-optimisation takes place, whenever a cycle or chain fails, the best sub-cycle/chain gets implemented for the remaining nodes and arcs spanning the subgraph or the original cycle/chain. Therefore, when using only the Recourse strategy, our ability to correct errors is much more limited than in the case of a full re-optimisation. As it was expected, the usage of this strategy would result in a more modest improvement in the number of transplantations than re-optimisation.

However, the usage of recourse in combination with re-optimisation can further improve the results. In this scenario, the recourse takes places only after the last re-optimisation round, in case there is a failure. For one round of optimisation, the use of recourse resulted in a 3.2% improvement in the number of transplants, while in the case of two rounds this value is less than 1%. For multiple rounds this value is even lower.

On the other hand, the number of crossmatch tests increased by 3.5, 5.2, 5.7, 6 and 6.1%. Therefore, recourse is a good strategy for countries, where only one optimisation round can be performed (such as the UK), but it is not efficient in the case of more optimisation rounds since it increases the number of crossmatch test significantly while affects the number of transplantations minimally (Figs. 10, 11).
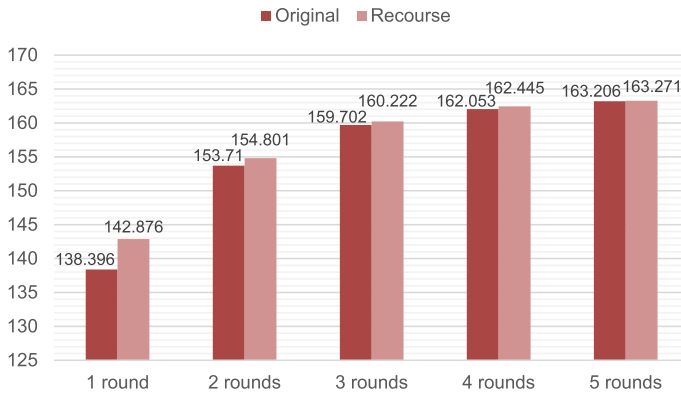
Fig. 10 Number of transplants with recourse
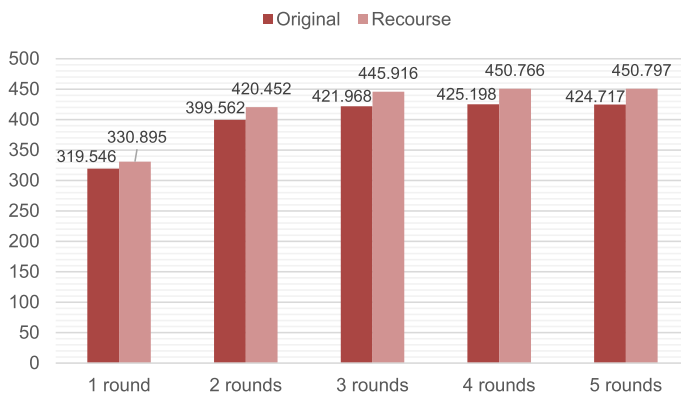
Fig. 11 Number of crossmatch tests with recourse

3.4 Fixing the good cycles

Fixing the good cycles and chains can be a useful feature in case of a re-optimisation, but for one round of optimisation, it does not change anything in the results.

The goal of re-optimisation is to get the best solution (or the best that can be obtained for a given number of optimisation rounds) in each matching round, while here we do not necessarily get the globally optimal solution, only a near optimal one, but in return we can get results faster, with fewer tests.

For two and three optimisation rounds this strategy even increased the number of transplants by 1,5% and 1%, and in the other cases the loss was less than 1% meanwhile the number of crossmatch tests decreased significantly with 13.5, 17.5, 18.1 and 18%, respectively. Therefore, we believe that fixing the good cycles can be a useful strategy in KEPs with more than one optimisation round (Figs. 12, 13).
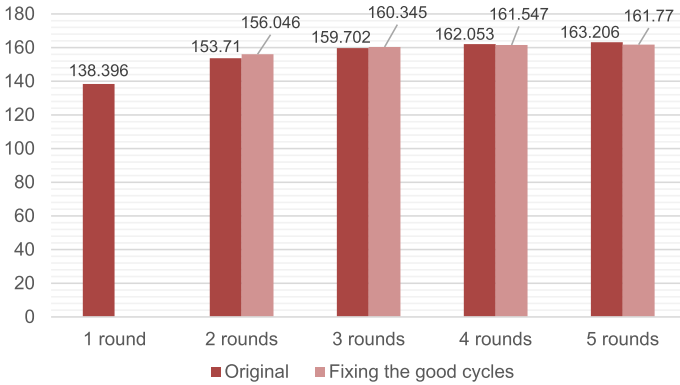
Fig. 12 Number of transplants with fixing the good cycles
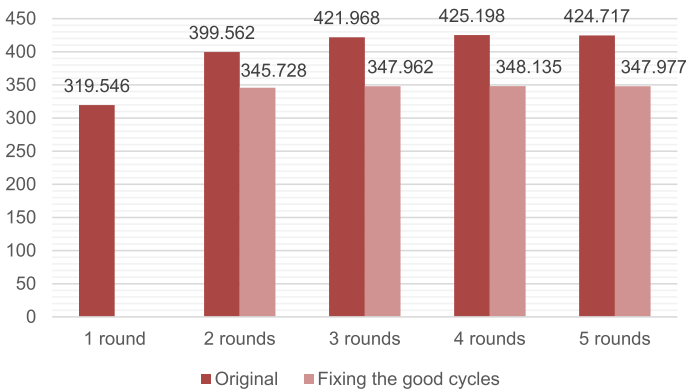
Fig. 13 Number of crossmatch tests with fixing the good cycles

3.5 Maximising the number of the good edges

Selecting good edges for the solution that have already been confirmed in a laboratory test should reduce the number of crossmatch tests needed and decrease the risk of failures and cancellations. To implement this strategy, we had to modify the optimisation criteria used as follows: after the number of transplants, we chose the maximisation of the number of good edges as the second criterion (Figs. 14, 15).

Using this new optimisation criterion increased the number of transplants by around 1–2% while it decreased the number of the crossmatch tests significantly by 2.9%, 13.4%, 16.2%, 16.5% and 16.5%, respectively.

In conclusion, the usage of this new optimisation criterion, the maximisation of the number of good edges could be beneficial for KEPs, since it reduces the number of crossmatch tests and increases the number of transplants at the same time.
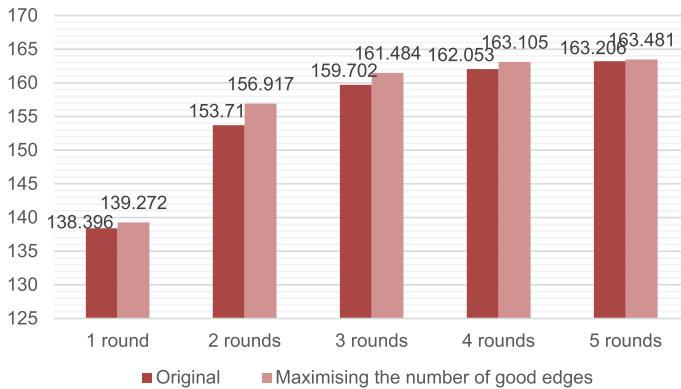
Fig. 14 Number of transplants with the maximisation of the good edges
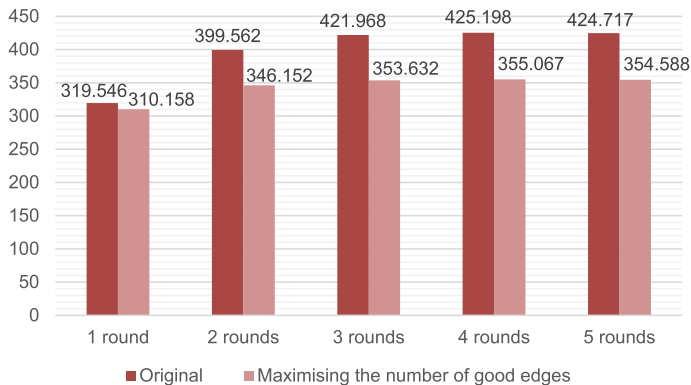
Fig. 15 Number of crossmatch tests with the maximisation of good edges

3.6 Testing cycle length limitations

Lastly for a single pool, we tested the effect of the maximum allowed cycle length limits on the results. Most of the European KEPs allow a maximum of 2, 3 or 4 long cycles and chains (Biró et al. 2019), therefore we decided to test these values.

While allowing 3-cycles increased the number of transplantations with 4.9% compared to 2-cycles, the 4-cycles did not make a significant difference, less than an additional 0.2% transplants compared to the case of 3-cycles. However, it increased the number of crossmatch tests with 23.9% and 33.4%, respectively.

In conclusion, according to our simulations, allowing 4-cycles does not increase the number of transplants significantly, only the number of crossmatch tests, therefore it seems unnecessary to use 4-cycles and chains (Fig. 16).
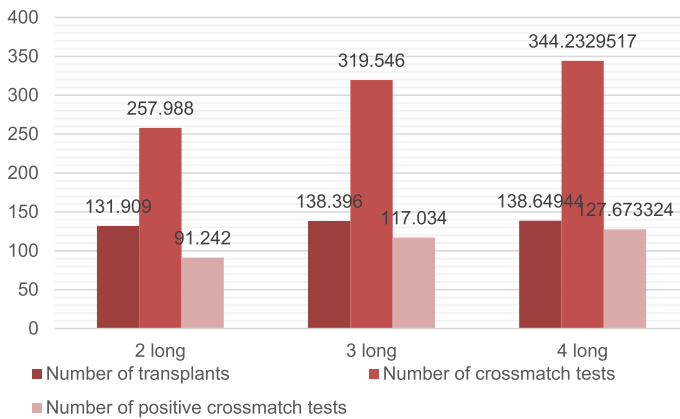
Fig. 16 The effect of allowing different maximum cycle length

4 Results for international KEPs

In this section we present our findings on simulations for international KEPs.

4.1 Type of collaboration

We tested all the three types of collaboration policies between countries that are implemented in the ENCKEP simulator. In the first, baseline case there is no cooperation, every country conducts their match runs separately.

In case of the so-called consecutive strategy the countries try to match their patients first in their own KEPs, and only for the remaining patient-donor pairs they seek international exchanges. This collaboration policy is used in between Spain, Italy and Portugal in their international collaboration, called KEPSAT.

When the so-called borderless policy is used, the KEPs work together completely, like one, larger program. This is the case in the Czech-Austrian-Israeli collaboration, and also in the international KEP coordinated by Scandiatransplant, called STEP.

The consecutive and the borderless policies give more opportunity to find a match for every patient, therefore they increased the total number of transplants by 11.5% and 22.3%, respectively.

It should be noted that in this case, the number of transplants has increased for all countries, but it is possible that although the total number of transplantations is increasing, the number of transplants for a country is decreasing (Fig. 17).

Based on the number of transplantations the usage of the borderless cooperation policy sounds really promising, however it has a huge negative effect as well: it increases the number of crossmatch tests, especially the number of international crossmatch tests. This was not considered in earlier studies, such as (Druzsin et al. 2022) and (Mincu et al. 2021).

The number of crossmatch tests increased by 17.9% and 47.5% in the consecutive and the borderless cases, while 15.8% and 38.2% of the crossmatch test were international, respectively. These international tests are time consuming and expensive,
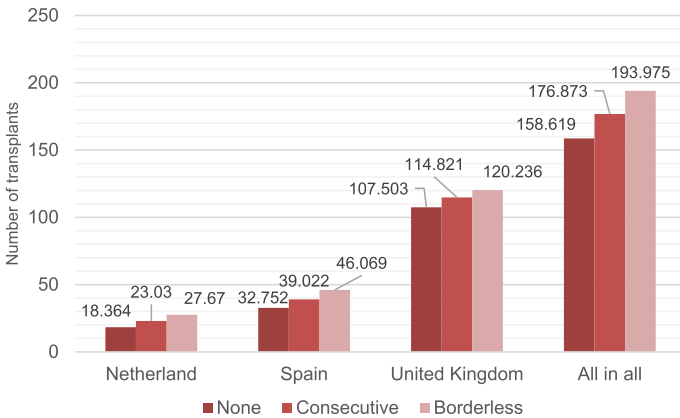
Fig. 17 Number of transplants with different types of cooperation

therefore we implemented a new optimisation criterion to try to reduce their numbers (Fig. 18).

4.2 The effect of different re-optimisation strategies

In our implementation for re-optimisation, it is possible to specify the order of testing and re-optimisations for different pools. With this function one can organise more optimisation rounds for some of the pools or simulate any order. The two combined strategies we tested were the testing of all the internal edges (i.e., the edges that run within a pool) and the testing of the internal edges for each pool in separate rounds one by one. Intuitively the advantage of conducting the testing and re-optimisation sequentially is that if a failure happens in one country, then the globally optimal solution can be changed immediately before testing the rest of the solution in other countries.
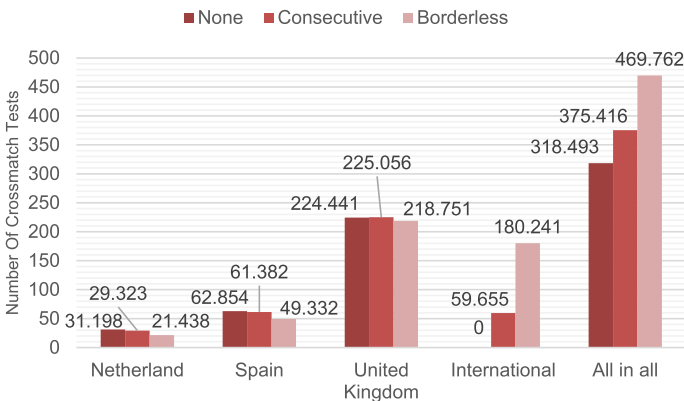
Fig. 18 Number of crossmatch tests with different types of cooperation
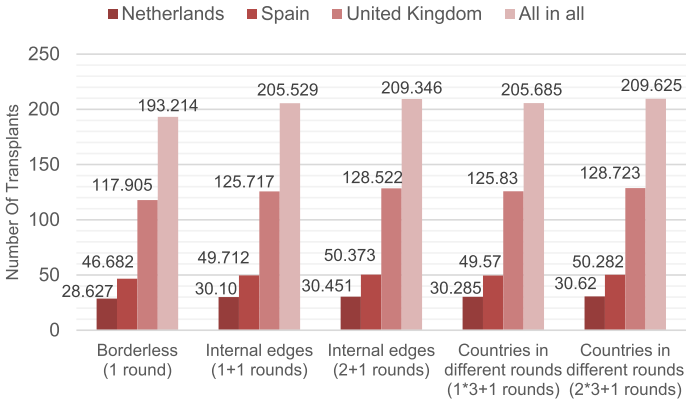
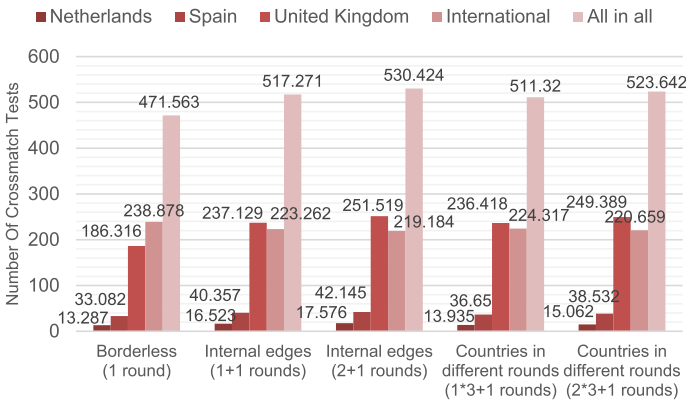**Fig. 19** Number of transplants with different borderless re-optimisation strategies

Fig. 20 Number of crossmatch tests with different borderless re-optimisation strategies

Beside the original borderless settings, we conducted simulations for both strategies with one and two rounds of testing, supplemented by a final full round of testing (general round). The separate testing rounds per pool strategy proved to be the most successful, as it resulted in slightly more transplants with slightly fewer crossmatch tests. Understandably, more optimisation rounds lead to more transplants and crossmatch tests also in these cases (Figs. 19, 20).

4.3 Maximising the number of internal edges

As mentioned above, it can be a serious problem for international programmes if many international crossmatch tests are required. By maximising the number of internal edges after the number of transplants, the number of crossmatch tests between pools can be reduced.

For the simulations, we used the same settings and data sets as before, with the difference that we chose to maximise the number of internal edges as the second optimalisation criterion (Fig. 21).

With this new optimisation criterion, we could reduce the number of international crossmatch tests by 24.5% without effecting the number of transplants. This is an expected effect of the policy. Therefore, maximising the number of internal edges seems a very promising approach for international KEPs.

4.4 Maximising the number of internal edges with different re-optimisation strategies

With our last simulation we combined the different re-optimisation orders with the maximising of the internal edges, since by conducting quicker tests internally further improvements are possible to achieve with sequential testing and re-optimisation strategies.

The number of transplantations increased by 0.4%, 0.9%, 1.4%, 0.9% and 1.4%, respectively, whilst the number of international crossmatch tests decreased by 24% in every case compared to the results without the new optimisation criterion. Therefore, the maximisation of the number of internal edges is a useful optimisation criterion with every case of re-optimisation (Figs. 22, 23).

5 Conclusions

In this paper we showed by simulations on realistically generated instances that new re-optimisation policies can improve the long-term performance of both national and international KEPs either by increasing the number of transplants implemented or by decreasing the number of crossmatches. The main novelty of our approach is that we also studied the international aspects, by allowing separate testing rounds for internal transplants combined with the prioritisation of internal transplants in the
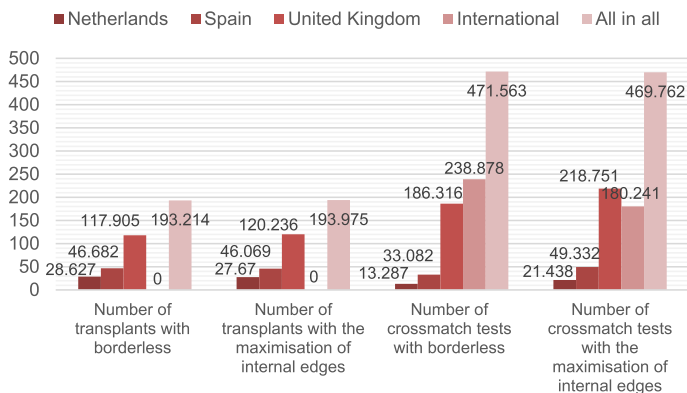
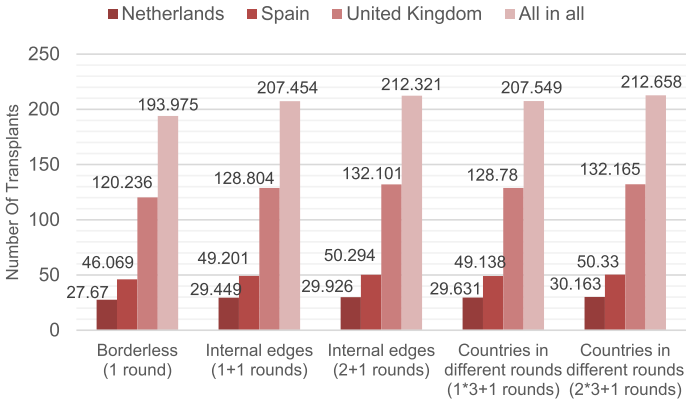

Fig. 21 The effect of the maximisation of internal edges

Fig. 22 Number of transplants with the maximisation of internal edges with different borderless re-optimisation strategies
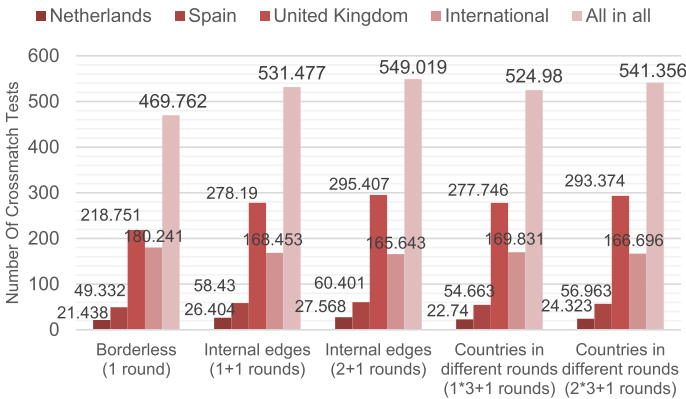

Fig. 23 Number of crossmatch tests with the maximisation of internal edges with different borderless re-optimisation strategies

solution. We also studied the effects of fixing the good cycles in the intermediate solutions, and to prioritise those transplants where the laboratory crossmatch tests have already been conducted. All the new strategies that we studied resulted in significant improvements regarding the increasement in the number of successful transplants and/or in the reduction of the number of crossmatch tests.

We plan to conduct additional studies in this framework by considering further optimisation criteria, such as maximising the expected number of transplants (Alvelos et al. 2019a). We will also incorporate some of the novel features into the KEP-SOFT software, that is a new software developed in the follow up COST Innovators Grant project of the ENCKEP COST Action, called KEP-SOFT. This software is going to be used by several European national and international KEPs, so hopefully our novel features will also be available for these application to use.

Declarations

Conflict of interest The authors declared that they have no conflict of interest.

References

Abraham D, Blum A, Sandholm T (2007) Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges. Proc EC 295–304:2007

Alvelos FP, Klimentova X, Viana A (2019a) Maximizing the expected number of transplants in kidney exchange programs with branch-and-price. Ann Oper Res 272:429–444

Alvelos FP, Viana A (2017) Kidney exchange programs with a priori crossmatch probing. In: Operations Research Proceedings 2017: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Freie Universiät Berlin, Germany, 2017, 363–368. Cham: Springer International Publishing (2018)

Alvelos FP, Viana A, Klimentova X (2019b) Probing for maximizing the expected number of transplants. Working Paper

Biró P, Haase-Kromwijk B, van der Klundert J (2019) Building kidney exchange programmes in Europe—an overview of exchange practice and activities. Transplantation 103(7):1514–1522

Biró P, van der Klunder J, Manlove DF et al (2020) Modelling and optimisation in european kidney exchange programmes. European J Operat Res 291:447–456

Blom D, Hojny C, Smeulders B (2022) Cutting plane approaches for the robust kidney exchange problem. Available at SSRN 4329425

Böhmig GA, Fronek J, Slavcev A, Fischer GF, Berlakovich G, Viklicky O (2017) Czech-Austrian kidney paired donation: first European cross-border living donor kidney exchange. Transplant Int 30(6):638–639

Carvalho M, Klimentova X, Glorie K, Viana A, Constantino M (2021) Robust models for the kidney exchange problem. INFORMS J Comput 33(3):861–881

Dickerson JP, Procaccia AD, Sandholm T (2013) Failure-aware kidney exchange. In: Proceedings of EC-2013, 323–340

Druzsin K, Biró P, Fleiner R, Klimentova X (2022) Large scale performance analysis of international kidney exchange programmes by the ENCKEP simulator, VOCAL Optimization Conference: Advaced Algorithms

Klimentova X, Pedroso JP, Viana A (2016) Maximising expectation of the number of transplants in kidney exchange programmes. Comput Oper Res 73:1–11

Klimentova X, Viana A, Pedroso JP, Santos N (2021) Fairness models for multi-agent kidney exchange programmes. Omega 102:102333

Klimentova X et al. (2022) International kidney exchange programmes in Europe: practice, solution models, simulation and evaluation tools, Handbook of Working Group 3 and 4 of the ENCKEP COST Action

Manlove DF, O'malley G (2014) Paired and altruistic kidney donation in the UK: algorithmss and experimentation. ACM J Experiment Algor 19:1–2

L Matyasi, P Biró (2022) Testing re-optimisation strategies in international kidney exchange programmes by the ENCKEP simulator, VOCAL Optimization Conference: Adv Algor

Mincu RS, Biró P, Gyetvai M, Popa A, Verma U (2021) IP solutions for international kidney exchange programmes. CEJOR 29(2):403–423

Monteiro T, Pedroso JP, Viana A, Klimentova X (2021) Reinforcement learning for robust optimization: an application in kidney exchange programs. IO 2019 Tomar, Portugal, Springer International Publishing, In Operational Research

Santos N, Tubertini P, Viana A, Pedroso JP (2017) Kidney exchange simulation and optimization. J Operat Res Soc 68:1521–1532

Smeulders B, Bartier V, Crama Y, Spieksma FC (2022) Recourse in kidney exchange programs. INFORMS J Comput 34(2):1191–1206

Valentin MO, Garcia M, Costa AN et al (2019) International cooperation for kidney exchange success. Transplantation 103(6):e180–e181

Website of the CA15210 COST Action: European Network for Collaboration on Kidney Exchange Programmes (ENCKEP). https://www.cost.eu/actions/CA15210/