Longevity-Aware and Failure-Robust Prioritization Schemes for Kidney Matching

Luca D'Amico-Wong, Jennifer Liang, Phyllis Zhang

December 6, 2021

Please visit our website to see some interactive visualizations and plots and a synthesis of this paper!

Abstract

In this paper, we explore the effectiveness of different kidney matching algorithms under practical constraints. While maximum-cardinality approaches provide clear theoretical advantages in the case of perfect compatibility, algorithmic matches are by no means guaranteed to work in practice. In fact, Dickerson et al. (2018) find that 93% of all algorithmic matches never result in a transplant. To address this problem, they propose a failure-aware algorithm that maximizes the expected number of transplants while taking into consideration the risks of failed matches. We expand upon the results of Dickerson et al., creating a dynamic, multi-round simulator to test various different algorithms under these considerations of failed matches.

We also propose a new failure-robust longevity-aware (FRLA) matching algorithm meant to minimize the expected number of deaths in the transplant network rather than maximizing the expected number of transplants. Simulations suggest that FRLA outperforms failure-aware matching and other traditional matching algorithms in two key metrics – the number of total deaths in the exchange and the total number of lives saved by our transplants. These advantages are particularly evident when there is a high mean and variance among the death probabilities of the patients. Future directions for analyzing FRLA are suggested.

§1 Introduction

Matching algorithms play an important role in the setting of kidney-paired donation, in which participants are patient-donor pairs who look to participate in chains and cycles to donate and receive kidneys. In the course, we've discussed the maximum cardinality matching approach, which provides a Pareto dominant matching. While in theory, this would maximize the possible number of transplants at a given time of the model (under the assumption that all matches will result in transplants), there arise a number of complications with this design in practice. In particular, this simple model does not consider the dynamic nature of the system; patients and donors are continuously joining and leaving the system. It also fails to take into consideration the probability that matches may fail to result in transplants; with close to 93% of all proposed algorithmic matches failing [3], this is an extremely real and extremely important concern.

The reasons for these failed transplants are numerous and include positive crossmatches, a type of incompatibility beyond simple ABO incompatibility. Additionally, agents may decide to leave the exchange, be "sniped" by a different kidney exchange, die, or exit for some other reason. While efforts have been made to address this issue, they typically do so in the context of maximizing the expected number of transplants that occur. This paper modifies the traditional objective function of kidney matching, viewing the ultimate goal of a kidney exchange to save the lives of patients. Assuming that doctors have a good idea of a patient's condition, this information should be made readily available to an exchange. We propose a new algorithm, failure-robust longevity-aware (FRLA) matching, that takes this into consideration and attempts to minimize the number of patients in the exchange who die.

The rest of this paper is devoted to explaining the theoretical underpinnings of FRLA and evaluating its efficacy through dynamic, multi-round simulations. Section 2 provides some theoretical background on FRLA, section 3 discusses the results of our simulations, and section 4 showcases our visualizations and website. Section 5 offers some final conclusions while section 6 considers possible directions for future work with the FRLA algorithm.

§2 Theoretical Background

In this section, we present our failure-robust longevity-aware (FRLA) algorithm to minimize the expected number of deaths in our network.

§2.1 Failure-Aware Matching

Dickerson et al. propose a failure-aware matching algorithm designed to address the problem of algorithmic matches failing to result in real-life transplants. Given that each transplant has a probability of failing, the failure-aware matching algorithm maximizes the expected number of transplants that will occur. For more on the underlying details of this algorithm, see the original paper. [3]

§2.2 Failure-Robust Longevity-Aware (FRLA) Matching

While failure-aware matching maximizes the expected number of matches conditional on the risk of a transplant failing, this might not be the most important goal for a kidney exchange. If we consider the purpose of a kidney exchange to be to save lives through kidney transplants, our objective function would attempt to minimize the expected number of deaths in the network, or in other words, to maximize the expected number of lives saved by the exchange. This is the motivation behind our FRLA matching algorithm.

Formally, let d_e denote the probability that the patient at the terminal node of edge e dies in the current round and let f_e denote the probability that edge e fails to result in a transplant. We can now model the utility of the kidney exchange as a function of a given matching.

Specifically, the kidney exchange's utility is given by the sum of the current round death probabilities of all patients who successfully receive a kidney. In any given static round, the exchange wishes to maximize the expected number of deaths prevented due to the kidney

matching, yielding utility that depends on the sum of current round death probabilities (or the number of expected deaths saved due to the given matching).

Assumption 2.1 (Kidney Exchange Utility Function)
We assume that we have the following utility function for kidney exchanges for a given matching $\mathcal{M}$.

$$U(\mathcal{M}) = \sum_{c \in \mathcal{M}} u(c)$$

where the utility of a cycle or chain c is given by

$$\sum_{e \in c} (d_e \cdot \mathbb{1}_e)$$

where $\mathbb{1}_e$ is the indicator function for whether or not the transplant represented by edge e actually occurs.

Under this assumption, we can calculate the expected utility of adding a given cycle to our matching. Note that if a single transplant in a cycle fails, then the whole chain breaks down as we cannot leave any stranded patients in a patient-donor pair.

Claim 2.2 (Expected Utility of Cycles) — The expected utility – measured by the expected number of deaths prevented – of any given cycle c is given by

$$\mathbb{E}[u(c)] = \left[\sum_{e \in c} d_e\right] \cdot \prod_{e \in c} (1 - f_e)$$

Similarly, we can calculate the expected utility of adding a given k-chain (a chain with k edges) to our matching. Note that unlike cycles, a failed transplant in a chain only affects those patients who follow in the cycle; all previous transplants will have already occurred.

Claim 2.3 (Expected Utility of Chains) — Let ℓ_n denote the n^{th} edge of our given k-chain ℓ. Then the expected utility – measured by the expected number of deaths prevented – of ℓ is given by

$$\mathbb{E}[u(\ell)] = \sum_{j=1}^{k} \left[d_{\ell_i} \cdot \prod_{i=1}^{j} (1 - f_{\ell_i}) \right]$$

Having calculated the expected utility of including any given cycle or chain, we thus define the FRLA algorithm to be the algorithm that returns the matching with the highest expected utility.

Claim 2.4 (The FRLA Algorithm) — Below, we provide the IP formulation for the FRLA algorithm where $\mathcal{C}$ is the set of all possible cycles and chains in our network V.

$$\begin{aligned} &\max_{y_c} \mathbb{E}[u(c)] \cdot y_c \\ \text{s.t.} \sum_{c \in \mathcal{C} \text{ s.t. } v \in c} &y_c \leq 1, \forall v \in V \\ &y_c \in [0,1], \forall c \in \mathcal{C} \end{aligned}$$

Note that we can cap the length of our cycles and chains in $\mathcal{C}$ to obtain an approximation to the optimal matching under our given utility function. When simulating, we will do so to ensure that the algorithm is not too costly.

However, it is important to note that the FRLA algorithm already does this for us – longer chains and longer cycles are implicitly penalized by the algorithm since a single failed transplant threatens to destroy an entire cycle or much of a chain. The FRLA algorithm will tend to favor shorter chains and cycles to longer ones, even at the cost of the total cardinality of our matching. To illustrate this attribute of the FRLA algorithm, consider the following analysis.

Optimal Cycle Length: Assume that death probabilities and failure probabilities are uniform across all patients and edges and equal to d and f respectively. What is the optimal length of a given cycle under the FRLA algorithm, ignoring all other existing or potential chains/cycles in the graph?

For a cycle c of length k, the expected utility is given by Claim 2.2, so maximizing this, we have

$$\begin{aligned} \operatorname*{argmax}_{k \geq 2} \mathbb{E}[u(c)] &= \operatorname*{argmax}_{k \geq 2} k \cdot d \cdot (1-f)^k \\ &= \operatorname*{argmax}_{k \geq 2} k \cdot (1-f)^k \end{aligned}$$

For sufficiently large f (roughly above 0.3), this is maximized at $k = 2$, meaning that we prefer 2-cycles, the shortest possible cycle, to all other cycles. Thus, FRLA (and Failure-Aware Matching) will prioritize shorter cycles and chains compared to the typical maximum cardinality algorithm.

Having established the theoretical properties and motivation behind the FRLA algorithm, we proceed to Section 3, where we analyze the FRLA algorithm through a variety of simulations and compare it to its counterpart algorithms.

§3 Simulated Results

Here, we present the results of the simulations run using a variety of different algorithms under different conditions.

§3.1 Setup

§3.1.1 Github Repository

All our code is located in the following Github, modified from James Trimble's original repository. [5] We modified the original repository to accommodate dynamic, multi-round simulations with agents entering and exiting the exchange over time. Our contributions are located primarily in the main.py file, which is entirely our creation.

§3.1.2 Patient/Donor Generation

We generated our own patient donor pairs with the following blood type distribution. Note that we have AB patients and O donors despite the fact that these will always be blood-type compatible with their partner in the pair. This is due to the possibility of positive crossmatches and other compatibilities, and the probability distribution has been adjusted to reflect this concern. Thus, the below represents a slight modification of the general American population's blood type distribution, reflecting the lower probability that we see AB patients or O donors.

	Patient	Donor
AB	1%	5%
A	41%	51%
B	12%	14%
O	46%	30%

Table 1: Distribution of donor and patient blood type

Seven patient-donor pairs were generated per round with an expected 0.7 altruistic donors generated per round as well; these figures were selected to accurately reflect the small proportion of altruistic donors in the population while still allowing for the presence of altruistic donors to meaningfully affect the simulation.

§3.1.3 Death Probability Generation

For each patient-donor node in our graph, the patient is associated with a probability of dying in the current round. The initial probabilities of death are generated according to a modified Normal distribution. To ensure that the death probabilities are in between 0 and 1, we randomly sample from our normal distribution until the probability is valid, meaning that the true distribution of death probabilities is right-skewed and no longer normal. This initial probability evolves according to the following assumption.

Assumption 3.1 (Evolution of Death Probabilities)

We choose to model the probability that a given kidney patient dies as an exponential function of time where

$$d_{n+t} = d_n(1.005)^t$$

where d_n represents the probability of death in period n and t is the number of periods in the future from n.

We assume that altruistic donors will not die over the course of the simulation.

§3.1.4 Failed Match Probabilities

Each edge in our compatibility graph is associated with some probability of failure. For our simulations, this probability is fixed to some constant value, though our code allows for non-constant failure probabilities.

§3.1.5 Compatibility Graph Generation and Matching

The simulation proceeds for a specified number of rounds. New agents are generated each round as described in 3.1.2. Agents are removed from the pool either when they die or when they receive a successful transplant.

At the beginning of each round, any operations from the previous round are checked to see if they are successful. From the pool of patients awaiting a kidney, deaths are simulated according to each patient's probability of death. Any dead patients are removed from the pool. A compatibility graph is generated given the current pool of patients and optimal cycles/chains are selected according to the algorithm in question. These cycles and chains are noted for operation in the next round. This process repeats for the given number of rounds, tracking successful transplants and deaths along with other relevant statistics.

§3.1.6 Default Parameters

Throughout the experiments we run, we vary three parameters – the number of rounds in the kidney exchange, the universal probability of a failed match, and the probability distribution of death rates facing patients. Each of these parameters is varied in isolation while the other two are kept at the default values. The default values are reported below.

- Total Number of Rounds: 36
- Initial Death Probability Distribution: $d_e \sim \mathcal{N}(0.02, 0.02)$
- Failed Match Probability: 0.7

where the failed match probability of 0.7 is taken from Dickerson et al. [3] The initial death probabilities might seem low, but under these defaults, the probability of any patient surviving for 36 consecutive rounds without a transplant is roughly approximated by $(1 - 0.02)^{36} \approx 0.5$.

§3.2 Experimental Parameters

§3.2.1 Algorithms Under Consideration

We run each experiment with four different algorithms.

- **UEF:** This is the unextended edge formulation for the kidney matching problem, placing no restrictions on chain or cycle length.
- **PICEF:** This is the position-indexed chain-edge formulation, a formulation developed to allow for more compact representations of chains in the model. [1] We cap cycle length at 3 and chain length at 20.
- **FR-PICEF:** This is a failure-robust/failure-aware implementation of PICEF a-la Dickerson et al. and explained in Section 2.

- **FRLA:** Finally, we have FRLA which we run by modifying FR-PICEF and weighting each edge according to the probability that the patient on the terminal node dies in the current round.

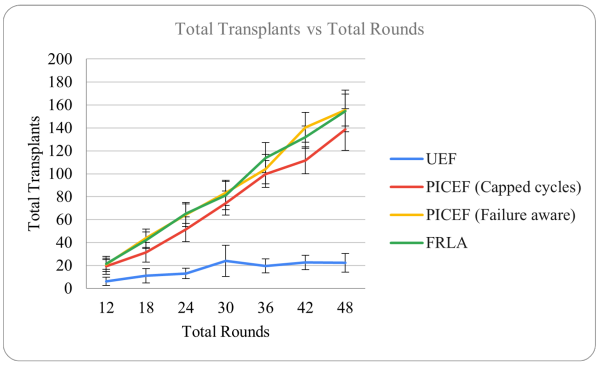
§3.2.2 Relevant Metrics

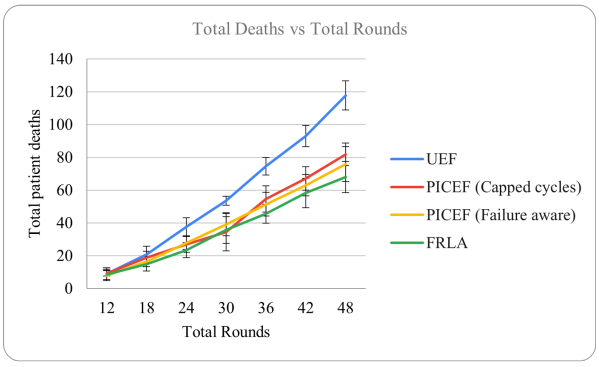
We evaluate each algorithm on three key metrics.

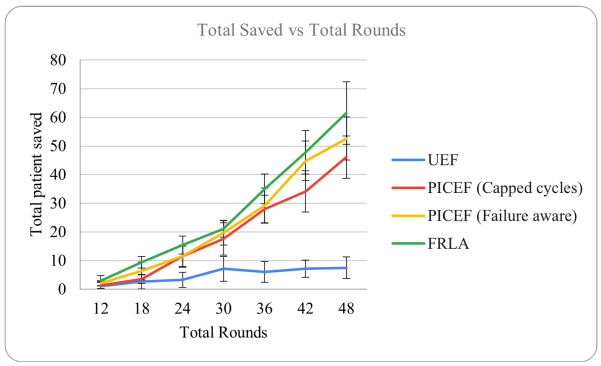
- **Successful Transplants:** This is the typical benchmark for kidney matching algorithms.
- **Deaths:** As people die in our exchange, these deaths are tracked and reported for each simulation.
- **Saved Lives:** This is a counterfactual implemented into our simulation – we continue to simulate the probability of death for those who have received successful transplants and keep track of whether they would eventually have died throughout all rounds of the simulation. We then count how many agents who would have otherwise died received a successful transplant before death, representing our metric for the number of lives saved by an algorithm.

§3.3 Experiment 1: Number of Rounds

In this experiment, we vary the total number of rounds we run the simulation for. We test seven different values for the total number of rounds – 12, 18, 24, 30, 36, 42, 48. Ten simulations are performed for each algorithm at each value, and averages and standard deviations are collected. Below are the graphs displaying the results from the simulations for each relevant metric.







As the number of rounds increases, the differences in performance between the four algorithms become more clear. Most clear is UEF's very poor performance, completing only 20 transplants and saving only 10 lives over 48 rounds compared to the 160 and

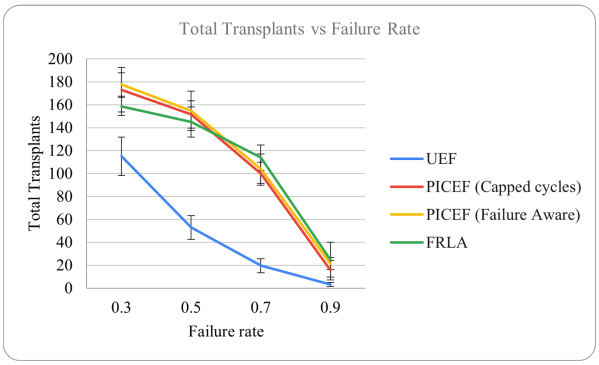
60 of FRLA. UEF struggles in this failure-aware environment precisely because of the unrestricted nature of its formulation – it favors long cycles and chains to shorter ones, and these longer dependencies are much more likely to fail than the typical 2-cycles formed in failure-aware PICEF or FRLA.

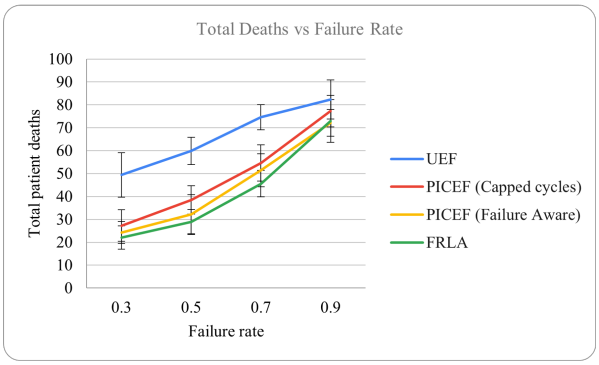
Both failure-aware algorithms perform better than their failure-ignorant counterparts, resulting in more transplants, fewer deaths, and more lives saved. Note that while FRLA clearly outperforms failure-aware PICEF in terms of deaths and lives saved, it does not result in significantly more transplants. This is related to the different objective functions of the two algorithms – unlike failure-aware PICEF, FRLA does not attempt to maximize the number of transplants; instead, it attempts to minimize deaths and maximize lives saved.

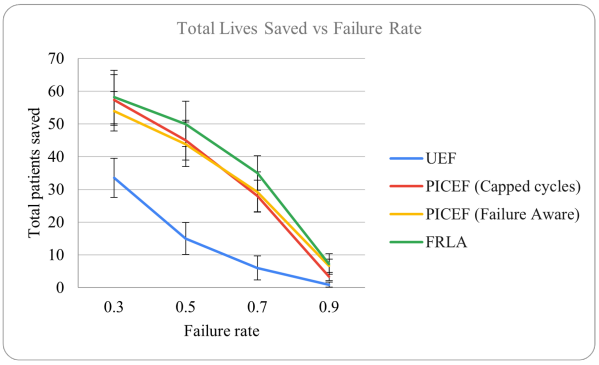
The trends in this graph suggest that with an increasing number of rounds, differences between the two algorithms will become increasingly more apparent. The error bars for this graph are rather large, indicating high variance in our results. Unfortunately, we were only able to run 10 simulations at each value due to computational constraints, but we would expect that taking a larger sample would result in more precise estimates. The same will hold true for the following two experiments.

§3.4 Experiment 2: Universal Failure Probabilities

In this experiment, we vary the uniform failure probability associated with each edge. That is, we change how likely any given transplant is to fail. We test four different values of this universal failure probability – 0.3, 0.5, 0.7, 0.9. Ten simulations are performed for each algorithm at each value, and averages and standard deviations are collected. Below are the graphs displaying the results from the simulations for each relevant metric.







Intuitively, we expect that larger failure rates will result in larger differences between the failure-aware algorithms and their failure-ignorant counterparts. The above graphs

demonstrate that this is at least partially true; failure rates of 0.5 and 0.7 result in substantially more difference between UEF and the other algorithms than a failure rate of 0.3. However, for a sufficiently high failure rate of 0.9, the performance of all four algorithms deteriorates dramatically due to the extreme difficulty of having a successful transplant.

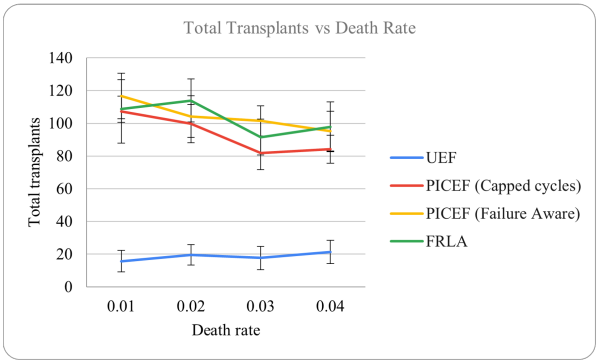
Perhaps most interesting to note here is that PICEF and failure-aware PICEF do not seem to be significantly different, even at high failure rates. This is likely due to the fact that we capped the length of cycles at 3 for PICEF, meaning that it very rarely falls into the same trap as UEF of creating very long cycles. As any 2-cycles have the same probability of failing, so long as PICEF primarily selects 2-cycles, it is not substantively different from failure-aware PICEF. Introducing non-uniform failure probabilities would likely make these differences more apparent.

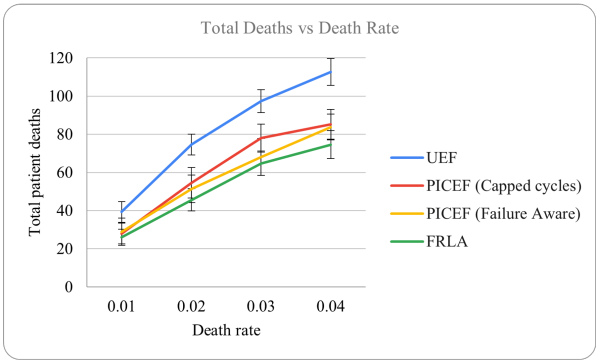
Corroborating our results from experiment 1, FRLA continues to outperform PICEF and failure-aware PICEF in terms of the total number of lives saved and the total number of deaths among patients in the exchange. However, these advantages are observed to be most drastic when the failure rate is in between 0.5 and 0.7. A threshold of 0.9 results in almost all proposed transplants failing, leading to very little differentiation between any of the four algorithms. Seeing as Dickerson et al. estimate the probability of a positive crossmatch to be around 0.7 from the UNOS data, it is unlikely that such a high failure rate holds in practice.

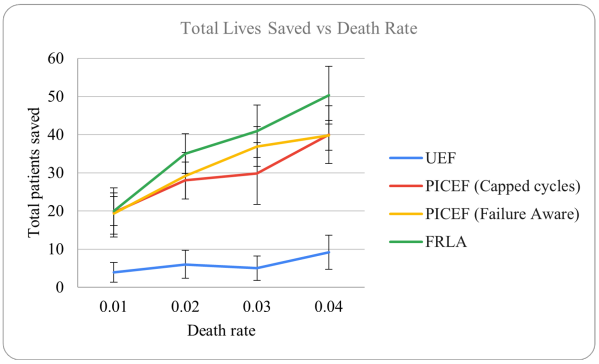
Another interesting observation to note is FRLA's behavior with low failure rates of 0.3 or 0.5. Here, it results in fewer total transplants than both other versions of PICEF, but it nonetheless continues to result in fewer deaths and more lives saved. As mentioned briefly in 3.3, this represents the implicit trade-off FRLA makes when determining optimal cycles/chains; FRLA will prefer having fewer total transplants so long as it picks high-value transplants expected to save a significant number of lives.

§3.5 Experiment 3: Initial Death Probability Distributions

In our final experiment, we vary the initial distribution of death probabilities among patients. We test four different initial normal distributions – $\mathcal{N}(0.01, 0.01), \mathcal{N}(0.02, 0.02)$, $\mathcal{N}(0.03, 0.03), \mathcal{N}(0.04, 0.04)$. Recall that these do not reflect the true initial distributions due to the requirement that generated probabilities lie in the interval $[0, 1]$, but they serve as a relative proxy for the true distribution's mean and variance. Ten simulations are performed for each algorithm at each value, and averages and standard deviations are collected. Below are the graphs displaying the results from the simulations for each relevant metric.







Intuitively, we expect that higher initial probabilities of death will lead to FRLA more clearly outperforming the other algorithms in terms of the number of lives saved and the total number of deaths in the exchange. This is in fact confirmed by the results of the experiment. With an initial $\mathcal{N}(0.01, 0.01)$ distribution, differences between all three non-UEF algorithms in saved lives and deaths are negligible – few people die and few lives are saved.

However, at an initial $\mathcal{N}(0.04, 0.04)$ distribution, FRLA dramatically outperforms its two PICEF counterparts, saving 50 lives compared to the 40 of failure-aware and regular PICEF. Fewer deaths are also recorded. This represents a 25% increase in saved lives; if we consider this on a larger scale, such an increase could save hundreds and perhaps thousands of lives.

The reason for these increases can largely be explained by the increasing variance of death probabilities among the distribution. When death probabilities do not vary significantly, FRLA's outperformance is only minor – any transplant is roughly the same in the expected number of deaths it prevents, so FRLA operates much like failure-aware PICEF. However, with a larger variance in death probabilities, FRLA is able to more clearly pick out those patients who are in severe need of a kidney – prioritizing them for transplants and resulting in significantly more saved lives.

If we expect large variance among the severity of patients' conditions in the real world, this experiment seems to suggest that FRLA provides meaningful advantages over other algorithms that are not longevity-aware. In one sense, FRLA is allocatively efficient in this way, allocating transplants to those who are most in need of them or those who desire them most.

§4 Website and Visualizations

In addition to our empirical findings, we also created visualizations of the evolution of kidney matching networks in JavaScript using D3. Along with a more visual presentation of our results through interactive scatterplots and bargraphs, these example network visualizations can be found on our website.

§5 Conclusions

FRLA is designed to approach kidney matching with the goal of minimizing the number of deaths in the network, in contrast to the typical objective of maximizing the total number of transplants. It does so by maximizing the number of expected deaths prevented through transplants, allowing patients more likely to die to receive kidneys sooner. The theoretical properties of FRLA are examined, and FRLA is designed precisely to achieve this goal.

We find that simulations support the theory behind FRLA, suggesting that there are meaningful benefits to implementing FRLA, with more lives saved and fewer deaths reported in the exchange. The benefits of FRLA increase as the distribution of death probabilities among patients in the exchange becomes wider, representing more variance in the severity of the patients' conditions. Additionally, longer simulations result in larger benefits to implementing FRLA. Higher failure probabilities have little effect on FRLA's relative performance, though we do see deterioration among all four tested algorithms at sufficiently high rates of failure

As one would expect from its design, FRLA does not result in significantly more transplants than failure-aware matching or capped PICEF matching, and there are times when fewer total transplants are recorded – dependent on the priorities of the exchange, this could be a concern associated with implementing FRLA.

§6 Future Work

While the initial experimental results of FRLA are promising, there are several different directions for further research. Most clearly, there is a need for further simulation, likely with larger sample sizes to reduce the variances in our results. More experiments would be welcome as well. What happens when you introduce non-uniform edge failure probabilities – is FRLA robust to such modifications, and is the algorithm still computationally feasible? What would happen if failure and death probabilities are correlated, as one might expect with especially weak patients who might struggle to receive a kidney?

Questions of ethics and fairness are also at the forefront of the FRLA algorithm. In prioritizing patients with high probabilities of dying, we risk forcing patients with less severe conditions to wait in the exchange for a much longer time compared to their counterparts with more severe need for a kidney. For these less at-risk patients, waiting time is especially high, and if we feel that each person should have an equal chance to receive a kidney, important ethical questions are raised. Additionally, a more precise investigation into the waiting time of patients under the FRLA algorithm would be welcome; the question of waiting time has already been studied in the literature [4], so it would be interesting to see how it applies here.

Finally, we are interested in how myopic and static matching compare under the FRLA algorithm. Is it possible that matching intermittently rather than every round yields fewer deaths and more saved lives? Assigning potentials to particularly useful edges/nodes has been experimented within the literature [2], and it would be interesting to see how those potentials would interact with the FRLA algorithm – could one perhaps incorporate these longevity concerns into these potentials?

References

[1] John P Dickerson, David F Manlove, Benjamin Plaut, Tuomas Sandholm, and James Trimble. Position-indexed formulations for kidney exchange. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 25–42, 2016.

[2] John P Dickerson, Ariel D Procaccia, and Tuomas Sandholm. Dynamic matching via weighted myopia with application to kidney exchange. In *Twenty-sixth AAAI conference on artificial intelligence*, 2012.

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

[4] Tiago Monteiro. A comparison of matching algorithms for kidney exchange programs addressing waiting time. *Central European Journal of Operations Research*, 29(2):539–552, 2021.

[5] James Trimble. Kidney solver. https://github.com/jamestrimble/kidney_solver, 2020.

Group Member Contributions

All three of Phyllis, Jennifer, and Luca helped modify the existing Github repository to allow for dynamic multi-round simulations. Phyllis also designed the entire website and all visualizations/plots that can be found there. Jennifer helped to run simulations and created all plots seen in this write-up. Luca also helped run simulations and was responsible for the write-up, developing the theory behind FRLA. All group members can attest to a fair and equal distribution of work.