Kidney Exchange with Distributed and Incremental Settings

Michael Romer, Jeffrey Nelson, Cheuk Ng, Utkarsh Vardan

Introduction

When a person's kidney is unable to properly filter blood, he or she may be diagnosed with kidney disease. Kidney disease can cause waste to increase in the body, as well as other health problems like heart attack, diabetes and high blood pressure. Chronic kidney disease is a condition in which kidney function reduces over a period of time. In the last stage of chronic kidney disease, the kidney fails to filter blood entirely, and a transplant is required for survival. At this point, an individual can survive on dialysis, a process in which blood is artificially filtered, for a short period of time. As per the Department of Health and Human Service, approximately 14 percent of the US population has chronic kidney disease, mainly as a result of high blood pressure and diabetes.[1] More than 20 million people have chronic kidney disease out of which 600,000 people are on dialysis and 100,000 people are on waiting lists for a kidney transplant.[1] Only 16,000 transplants are done per year and more than 50,000 Americans die as a result of kidney disease.[1]

It is usually very difficult for people in need of a kidney transplant to find a compatible donor, but paired kidney exchange can bridge the gap. Suppose someone requires a kidney transplant and a family member or friend is willing to donate. If the donor and receiver are incompatible because of blood type mismatch, the transplant cannot happen. Now consider two pairs, A and B, where donor A is compatible with receiver B and donor B is compatible with receiver A. These two pairs can swap donors so that a mutually beneficial payoff results. In general, there can be an n-way swap or trading cycle between n donor and receiver pairs. For incentive reasons, all surgeries in an exchange are conducted simultaneously. In other words, a 2-way exchange involves 4 simultaneous surgeries and a 10-way exchange requires 20 operating rooms and 20 surgical teams. This is logistically difficult for a hospital to manage, since the hospital only has enough resources to perform a certain number of surgeries at any given time, and one pair dropping out will disrupt the entire chain. For this reason, hospitals prefer, and sometimes enforce, a maximum exchange size.

Proposed Solution

For this project, we implemented a general solution to the kidney exchange problem and then extended the solution with distributed and incremental settings. We start by modeling a hospital as a list of exchange pairs, i.e. patient-donor pairs that wish to participate in the exchange, and a maximum number of surgeries that can be performed in a single time step. To begin, each hospital is aware of only the pairs belonging to this hospital. The number of rounds for which the exchange runs is configurable, and a single round of the kidney exchange consists of the following:

- Each hospital, from the pairs that it is aware of, determines a set of cycles for which it can feasibly perform surgeries in this round. (The algorithms used to determine this feasible matching will be explained in the next section.)
- Each hospital performs surgeries for the cycles which consist of only local pairs. Those pairs are removed from the exchange. For the cycles which contain pairs that belong to other hospitals, the hospital with the higher rank acquires these pairs so that surgery can be performed on this cycle in the next rank.
- Each hospital informs the hospital of immediately higher rank, where rank is determined by lower hospital identifier, of all the exchange pairs that it was unable to match in this round.

Notice that the hospital of highest rank will be aware of all pairs still in the exchange after a number of rounds equal to the number of participating hospitals. This sharing of information represents the distributed setting, and is implemented with the following constraints in mind:

- Initially, each hospital is likely to have more pairs to perform surgery on than surgery slots. Otherwise, the hospital would quickly exhaust its local matches and is incentivized to work with peers to make forward progress.
- Once a hospital has exhaust its local matches, the only method for forward progress is patient transfer. Remember, however, that it takes nonzero time to transfer a patient to a new hospital, and is disruptive to the patient, so a hospital is not incentivized to transfer unless the patient would otherwise be unable to be matched in his or her current hospital.

- Eventually, the hospital with highest rank will claim all pairs for which exchanges can still be performed, and all feasible surgeries will be executed.

From the above algorithm, it should be clear that each hospital is acting somewhat greedily. We do not begin by sharing all of our information and determining a globally optimal matching. Each hospital begins with a locally optimal matching and shares information only when it is unable to match all of its local patients. We chose this behavior because it is closest to how the real world works, where hospitals are not always incentivized to fully report their patient sets. More information on challenges to hospital reporting and kidney exchange matchings, which inspired our model formulation, can be found in work done by Roth, Sönmez, and Ünver.[2]

As for the incremental setting, we implemented this by adding or removing a pair from a hospital ⅖ of the time and forcing a valid matching to fall through ⅛ of the time. Adding a pair to a hospital may allow it to make immediate forward local progress, while removing a pair may prevent it from making forward progress that it previously could have made. A valid matching falling through simulates the case where a pair drops out of the exchange, likely due to death of the kidney recipient. In this case, the entire cycle containing the removed pair is broken and the affected pairs must rejoin the exchange and hope to be matched again in subsequent rounds. The incremental setting forced our implementation to be tolerant of these conditions that otherwise would be unable to arise.

Algorithms for Kidney Exchange

For each hospital to find a feasible set of cycles amongst a list of patients, we choose to implement solutions in two different programming domains and compare their results in terms of total matches and runtime.

Greedy Matching Implementation

The first implementation defines a greedy matching algorithm for determining the set of feasible cycles. The matching is a variant of the Top Trading Cycles (TTC) algorithm and is implemented as follows:

1. Construct a directed graph, where each node is a pair, and an edge between node i and node j means that the donor in pair j is compatible with the receiver in pair i.
2. Iterate across set of nodes in directed graph. Iterator must be delete safe.
3. Begin recursive DFS-walk from node pointed to by iterator. If iterator is NULL, go to #7.
4. At each step along DFS-walk, append node to the set of visiting nodes and store introducing node (NULL for the first node) and step in which this node was introduced in travel map.
5. For neighbor in node.neighbors:
    a. If neighbor is in visiting set, and the current step minus the step in which the neighbor was introduced is less than or equal to the remaining surgery slots:
        i. Remove cycle from graph, store cycle in matching, update remaining surgery slots, and go to #3.
    b. Else if neighbor is in finished set, continue to next neighbor in #5.
    c. Else, recurse on neighbor (go to #4)
6. Remove node from visiting set and add to finished, since it is guaranteed to not be part of a cycle (return back to #5 in caller).
7. Return matching, i.e. set of cycles.

Example

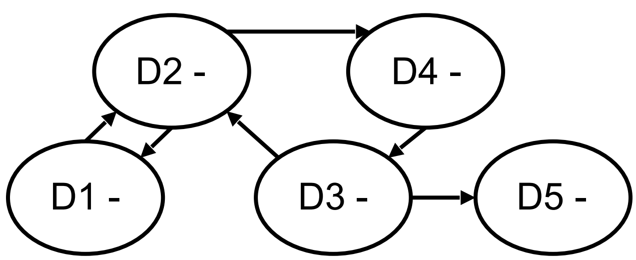

Figure 1

Consider *Figure 1* above for a hospital with 5 surgery slots.. The greedy implementation could return two possible matchings:

- $set([(D1,R1),\ (D2,R2)])$ or $set([(D2,R2),\ (D4,R4),\ (D3,R3)])$

The greedy matching is not guaranteed to return a **maximum matching**, unlike the algorithm we will see in the next section. Rather, it returns a **maximal matching**, so either solution is acceptable. If we choose the first solution, we are left with the following graph:

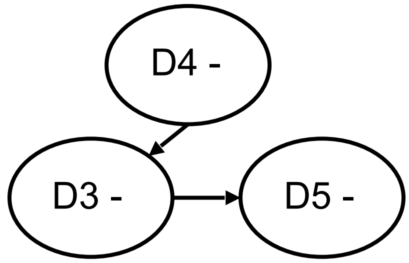

Figure 2

The only way for this hospital to make forward progress is to share these unmatched pairs with a peer hospital and hope that an inter-hospital matching can be formed.

Matching via Integer Linear Programming

As mentioned in the previous section, our greedy algorithm does not necessarily return a maximum matching. To illustrate, if in the previous example the greedy algorithm returned the matching comprising $set[(D2,R2),\ (D4,R4),\ (D3,R3)]$, a more globally optimal outcome could be realized. To address this, we formulate the kidney exchange problem as an integer linear program (ILP). For a hospital's directed graph of exchange pairs, the optimal solution to the ILP will yield a maximum matching.

Abraham et al. discuss two possible ILP formulations respectively based on edge formulation and cycle formulation [3]. Our second matching algorithm uses the cycle formulation approach to construct an ILP whose optimal solution computes a maximum weight cycle cover for a set of exchange pairs.

Our ILP consists of one binary variable per candidate cycle in the directed graph of exchange pairs. A variable x_i is set to one if cycle c_i is in the optimal cycle cover, and it is set to zero if it is not. A cycle is considered in the set of possible cycles C only if its length is less than or equal to the maximum number of surgeries S a hospital can perform. The weight w_i of a cycle c_i is equal to the weight of the edges in that cycle. The length of a cycle c_i is given by l_i.

Our model imposes the following constraints. First, the sum of the lengths of the cycles selected cannot exceed the maximum number of surgeries for a hospital. Second, each exchange pair can only contribute to at most one of the cycles selected. Together, this yields the following ILP:

$$
\begin{aligned}
\textit{maximize} \quad & \sum_{i\,:\,c_i \in C} w_i x_i \\
\textit{subject to} \quad & \sum_{i\,:\,c_i \in C} l_i x_i \le S \\
& \sum_{i\,:\,v_j \in c_i} x_i \le 1, \quad \forall v_j \in V \\
& x_i \in \{0, 1\}
\end{aligned}
$$

From the optimal solution of the above ILP, we can select the cycles of exchange pairs that together form the maximum matching returned by our implementation.

Performance Evaluation

In comparing the greedy matching algorithm, which has runtime complexity $O(V \cdot E)$to the ILP approach, which has runtime complexity $O(V^3)$, we collected the following data. We ran our kidney exchange for 5 rounds on random distributions of patient pairs across 5 hospitals. We varied the initial patient pairs per hospital from 10 to 25 and the max number of surgeries from 3 to 5. Below we graph both the execution time and the proportion of patient pairs who successfully were operated on before the end of the exchange simulation.

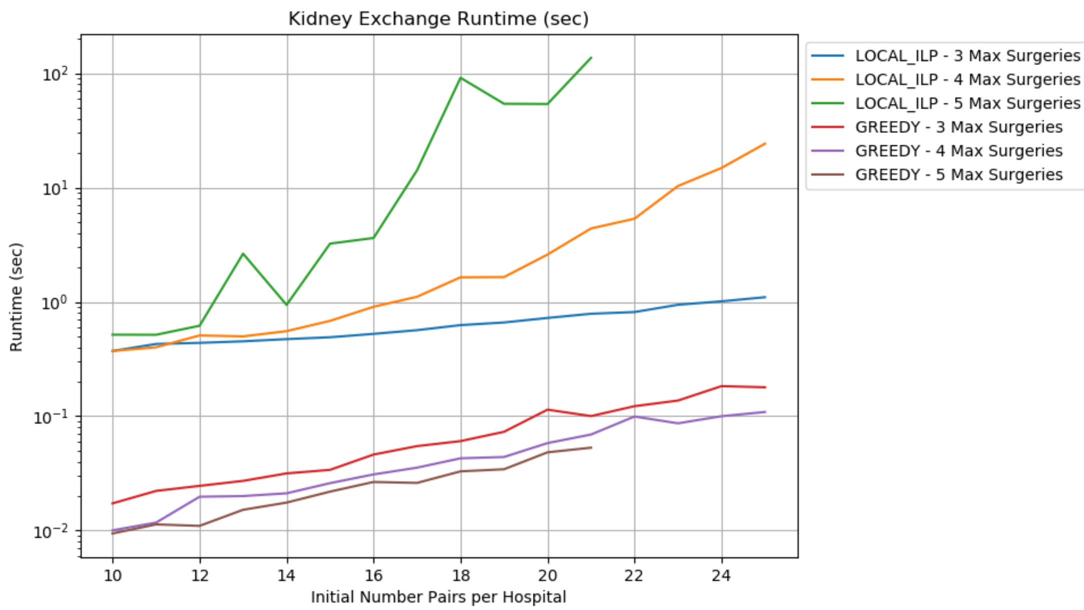

Figure 3

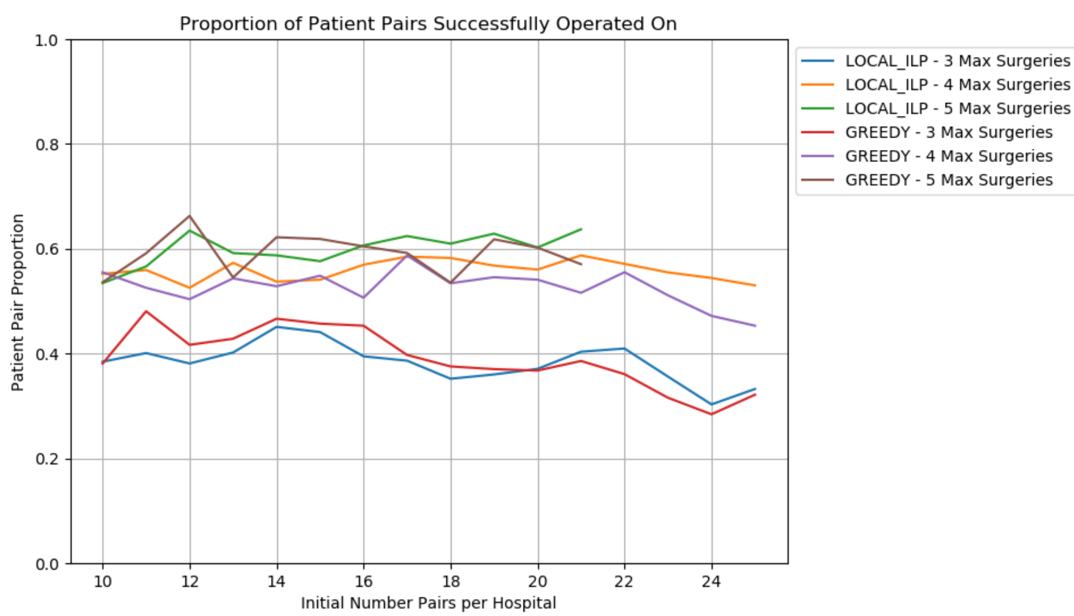

FIgure 4

Conclusion

From the data above, we can conclude that the Greedy algorithm scales better than the ILP algorithm with respect to runtime. The ILP algorithm is more locally optimal for each hospital than the Greedy algorithm, and matches a higher percentage of patients on average.

While our implementation very closely models a real-world kidney exchange and the associated challenges, we propose the following extensions to handle additional concerns that may arise in a real-world exchange as future work:

- Enhance the pair transfer logic to maximize for surgery slot usage and minimize patient disruption, defined as moving a patient when they could have stayed in their local hospital.
- Expand the number of constraints by modeling a patient with a “time to live” parameter, which means that the patient must be matched in X number of rounds or else the patient will die.
- Assign a per-patient cost function to move a patient from Hospital X to Hospital Y. Currently, the cost is implemented as one round.
- Include the following new types of exchange pairs:
    - A pair that is already matched, i.e. contains a self-cycle.
    - A pair with no donor (can model as a pair with a donor with incompatible type).
    - A pair with no receiver (can model as a pair with a receiver compatible with all types).
- Extend the compatibility function to account for different biological effects that improve the chances of matching.

References

[1] UNOS News Bureau *"Deceased organ donors in United States exceeded 10,000 for first time in 2017"* Jan. 2018 | News, OPO, Transplant center

[2] Alvin E. Roth, Tayfun Sönmez, M. Utku Ünver "KIDNEY EXCHANGE" NATIONAL BUREAU OF ECONOMIC RESEARCH 1050 Massachusetts Avenue Cambridge, MA 02138 September 2003

[3] Abraham, David J., et al. "Clearing Algorithms for Barter Exchange Markets." *Proceedings of the 8th ACM Conference on Electronic Commerce - EC 07*, June 2007, doi: 10.1145/1250910.1250954.

See the full source code at https://github.com/jdn5126/KidneyExchange.