

Kidney Exchange as Online Matching Problem with Stochastic Edges

Akbota Anuarbek

Abstract

Kidney exchange helps to solve one of the important problems in human health by allowing patients with willing but incompatible donors to find a compatible donor. Despite the importance of the problem and extensive number of literature written about the topic, some important issues in the area have not been fully addressed. This paper models the kidney exchange problem as an online matching problem with stochastic edges and takes into consideration such important elements of the kidney exchange as time dynamics of the patient-donor pool, stochastic features of transplantations, and 'health' costs incurred by a patient while waiting for a compatible donor.

I. Introduction

Kidney transplantation and especially kidney exchange was one of the most 'hot' topics of mechanism design over the last couple of decades. However, in addition to the blood type match, there is a long list of other factors that affect how successful the kidney transplantation will be. These factors among many others include the age of the donor, the number of matched antigens, the existence of a biological relationship between the donor and the patient, whether a kidney is from a deceased-donor or a living-donor. The greater the number of key variables that are met, the higher are the chances that the transplanted kidney will have better and longer functionality. Thus, it is intuitive to associate every patient-donor pair with some probability that the transplanted kidney will be successful, rather than merely categorizing pairs as compatible versus incompatible, which is the method adopted by almost all the available literature on kidney exchange. The National Kidney Registry [1] in fact says that donor that is matched better will increase the probability that the kidney will last longer and function better. What is more, they also write "a better HLA match is also correlated with lower patient mortality over time because a good HLA match will reduce the number of antibodies created from mismatched antigens, and make it easier for patients to be retransplanted in the future". All of these highlight the importance of carefully matching donors and patients. Thus, the kidney exchange problem can be in fact considered as an *online stochastic matching problem*. Since the pool of available patient-donor pairs is always changing as new pairs are arriving over time, sometimes it may be reasonable to hold some pairs unmatched until a better suitable pair (the one with higher probability of success) will arrive for them. If the probability of success is low for them, the transplant may be rejected, resulting in the waste of a kidney, as well as increasing the number of antigens in the patient's body and worsening the patient's state of health.

On the other hand, there is a risk that a pair with higher probability of success for both patients will not arrive in the near future. Waiting for the better matched pair is costly, because patient's health may deteriorate over time and sometimes make the surgery impossible for the patient as he/she becomes "too sick for transplantation". Here, the word *online* simply means that every time new vertices will arrive in the graph, and some of the information is not available at the time a decision is made about matching.

As mentioned earlier there is a multitude of papers that consider the kidney exchange problem as a static problem and mostly only focus on blood type compatibility between donor and patient, ignoring the fact that some pairs may be more suitable than others.

Papers with topics more relevant to this paper started appearing around 2010 and the number of those papers is still very few. Utku Unver [5] in his paper Dynamic Kidney Exchange considers efficient two-way and multi-way exchange mechanisms over a dynamically evolving pool of patient-donor pairs. However, his paper restricts the issue of compatibility to blood type compatibility without taking into consideration other factors that may have substantial effect. In contrast, Li et al.[6] associates every exchange with some type of utility and uncertainty. However, the model considered in the paper is static and does not account for time dynamics. Awasthi and Sandholm [4] provide a trajectory based algorithm for finding efficient allocation of kidneys. Dickerson, Procaccia, and Sandholm [3] further continues Unver's dynamic kidney exchange model. They assign each vertex some potential and try to optimize based on those potentials, though they consider only blood-type compatibility.

This paper models the pool of patient-donor pairs as an undirected graph similar to the papers mentioned above. Each person, whether he/she is a patient or a donor, is represented as a node in the graph. Each edge corresponds to a potential kidney transplantation. However, the problem in this paper is different from the above papers in several ways:

- (a) Kidney transplantation from any donor to any patient in the model is assumed to have some probability of success as opposed to having a utility that does not vary much across different pairs as some previous papers have assumed.
- (b) A patient's state of health is assumed to deteriorate over time and eventually, if exchange is not performed, he/she drops out of the pool together with the donor in the pair.

The rest of the paper is organized in the following way: *Section II* introduces online matching problem with stochastic edges and further develops the model. *Section III* describes different algorithms for kidney exchange, *Section IV* extends the model by including altruistic donors to the pool, *Section V* implements data simulation for comparing relative performances of these algorithms and discusses the results. Finally, *Section VI* concludes and proposes further extensions. Definitions and explanations for some graph and algorithm terms can be found in *Section VII*.

II. The Model

Paired kidney exchange problem can be modeled as an online matching problem in a graph with stochastic edges. For simplicity assume there are T periods, and let $G = (V, E)$ be an undirected

graph with the set of vertices (nodes) V and the set of edges E . Each node in V either corresponds to a patient or a donor, which come to the pool in pairs, so V has even number of elements. Thus, each patient-donor pair that arrives at the pool at some point in time during those T periods is included in V . Set E includes two types of edges: there is an edge between patient and donor that come to the pool together, as well as between any patient and donor, if they are blood type compatible. There are four blood types, compatibility of different types are illustrated in the figure 1 from Stanford Blood Center data [2].

		You can receive type			
If you are		O	A	B	AB
	O	+			
	A	+	+		
	B	+		+	
	AB	+	+	+	+

Figure 1: Blood Compatibility Chart, adopted from Stanford Blood Center [2]

Overall, there are $|V|/2$ patient-donor pairs in the graph G , however they will arrive at the pool during different periods and stay for only some number of periods. Some flexibility is allowed for the number of pairs arriving at each period $t \in T$. We assume that this number follows a Poisson distribution with parameter λ , where λ is the expected number of pairs arriving in a period. Let $n = \text{Poisson}(\lambda)$ be a random draw from the distribution. Once n pairs arrive (that is, $2n$ nodes), each node is assigned a blood type according to the distribution of blood types given in figure 2 from Stanford Blood Center [2].

Each newly arrived patient (donor) is connected to already existing donors (patients) via an edge if they have compatible blood types. Also, there is an edge connecting nodes corresponding to a patient and a donor, who arrive together. Figure 3 illustrates a randomly generated graph in this way given at some point in time. Here, red nodes correspond to patients and blue nodes correspond to donors.

Blood type	Frequency
O	44%
A	42%
B	10%
AB	4%

Figure 2: Blood Type Frequencies, adopted from Stanford Blood Center [2]

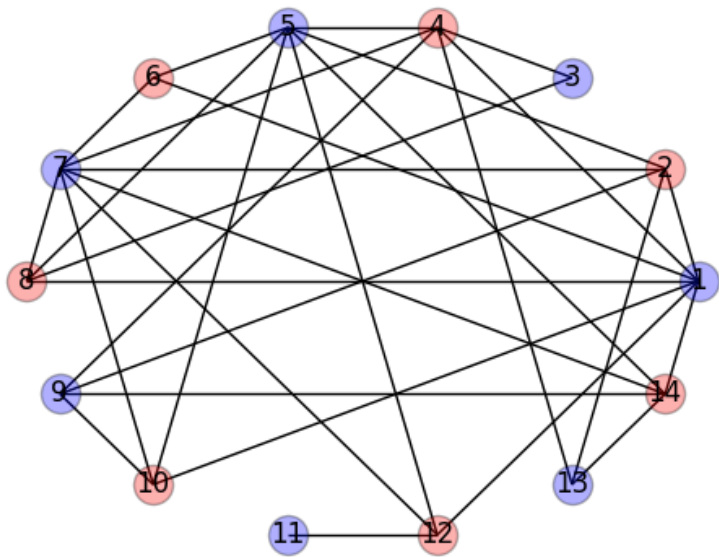


Figure 3: $G = (V, E)$

The previous paragraph introduced the online component of the problem. Let us now proceed with the stochastic component. Each edges in G has a probability p of success associated with it, which is simply the probability of success of the transplantation. If nodes in the edge are matched to each other, an independent random coin is tossed with probability p . If the outcome is 1, it means matching is successful, if it is 0 then transplantation fails. These probabilities p in fact vary across different edges and are assumed to be drawn from a $Beta(\alpha, \beta)$ distribution. Parameters α and β can be chosen based on medical research on transplant survival, and p can be more precisely defined as 5 year (or any appropriate number) survival probability of the transplant. Edge weights connecting patient and donor in the same pair are assumed to be 0 for simplicity. This is a reasonable assumption because pairs with high p do not need to join the exchange. Finally, the waiting cost should be taken into consideration. Every patient-donor pair is for simplicity assumed to drop out of the pool \bar{t} periods after its arrival. During this time interval the probability of matching the patient in the pair will decrease with time. j periods after arrival every transplantation probability for that vertex will be multiplied by c^j , where $0 \leq c \leq 1$ is the cost of waiting.

Assumptions about distributions are important and closely conform with reality. If no distributional assumptions were made, the graph would have been generated completely at random and there will be no way to consider the future while making a decision.

III. Kidney Exchange Algorithms

As is described in the previous section there is a graph that changes in each period according to some probability distribution. There is also some uncertainty associated with matching of donors to patients. Thus, when a new pair of patient-donor arrives, i.e. two nodes, the algorithm chooses either to match them with one of their neighbors or to hold them unmatched. The objective is then to compare different algorithms and find the most efficient one. The most efficient algorithm is the one which successfully matches highest number of pairs. In the following algorithms size of exchange cycles are not restricted.

Algorithm 1 (repeated static 0-1 probability matching): *At each period the algorithm matches maximum possible number of existing patient-donor pairs in the current period based only on blood type compatibility.*

The algorithm is essentially repeated application of the common static matching algorithm only accounting for blood type compatibility. At each period the maximum possible number of existing patient-donor pairs can be found using *Maximum Weighted Perfect Matching* assuming all pair edges have weight 0 and all other edges have probability of success 1 (or simply *Maximum Perfect Matching*). Here 0-1 weights correspond to 0-1 probabilities of success, the assumption used to consider only blood type compatibility. To be more precise, it is the algorithm that does not distinguish between edges of different p and finds matching only accounting for blood types, though in reality the success of the matched edge will be defined with probability p . The *reduction* to the *Maximum Weighted Perfect Matching* problem is straightforward and was introduced long ago. Maximum Weighted Perfect Matching is polynomial time solvable, thus the algorithm will run in polynomial time. Definitions and explanations of such graph and algorithm terms as maximum weighted matching, perfect matching, and reduction can be found in the Appendix.

Algorithm 2 (repeated static stochastic matching): At each period the algorithm matches existing patient-donor pairs by maximizing the total weight of matched pairs.

This algorithm is similar to the previous one and can also be reduced to *Maximum Weighted Perfect Matching*. The difference is that it takes into consideration success probabilities. Among two different matchings it will choose the one with higher total weights, whereas the previous algorithm does not distinguish between the two.

Algorithm 3 (simple deterministic algorithm): At each period the algorithm again matches existing patient-donor pairs by maximizing the total weight of matched pairs. However, the edge is not included in the matching if it has $p < \bar{p}$ for some low value of \bar{p} that can be chosen based on distribution of p .

The algorithm is also implemented by finding *Maximum Weighted Perfect Matching* at each period. However, edges with $p < \bar{p}$ are removed first and then matching is applied. This algorithm ensures that edges with very low probabilities of success are not chosen. Intuitively, if p is substantially lower than its expected value, it may be more reasonable to wait for another edge that most probably will have higher value of p .

IV. Including Altruistic Donors to the Model

The model can be further extended by including *altruistic donors* to the model. An *altruistic donor* is a donor with no relationship to pairs in the pool, who decides to donate a kidney. Including altruistic donors to the model allows exchanges not only in the form of cycles, but also in the form of chains. Figure 4 gives an example of a chain exchange. The green node (number 5) corresponding to an altruistic donor donates his kidney to node 2, and his pair node 1 donates his kidney to node 4. If node 3 is not matched in a current state, it may be matched in the following periods being considered as an altruistic donor.

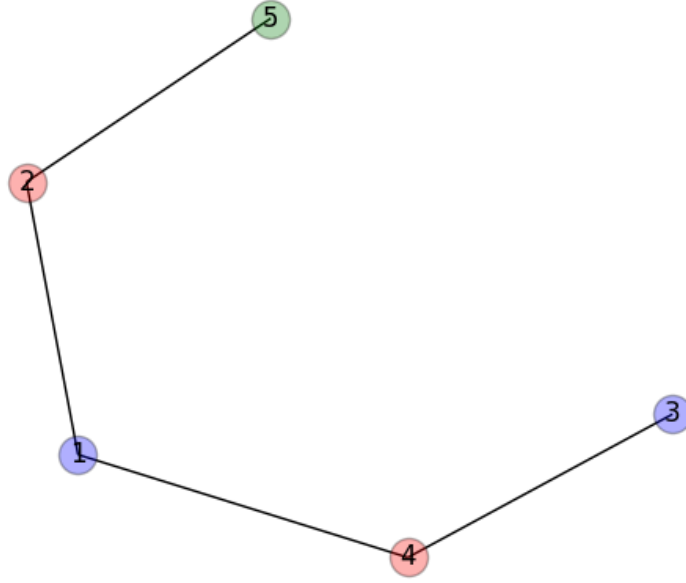


Figure 4: Chain exchange

Assume at each period there are n_a altruistic donors coming to the pool, where n_a is a random draw from the Poisson distribution with parameter λ_a . λ_a stands for the average number of altruistic donors coming to the pool in one period. Thus, at every period n_a altruistic donors join the pool, where every donor corresponds to a vertex in a graph. Blood types of donors are defined as a random draw from the distribution of blood types by Stanford Blood Center [2] given in the figure 2. Add an edge between every altruistic donor and a patient that are blood-type compatible, where weights of those edges given by p are obtained from $Beta(\alpha, \beta)$ distribution in the same way as in Section II.

Then the objective is to find an algorithm that will match greater number of patients across all T periods of time. In fact, all three algorithms described earlier can be applied to this extended model by introducing additional nodes and edges to the graph. Specifically, include one auxiliary node for every node corresponding to an altruistic donor, and add a zero weight edge between those two. Connect that auxiliary node with all donor nodes in the graph, including altruistic donors. Set success probabilities of those edges equal to zero. Then previously defined algorithms will do the following.

Algorithm 1 (repeated static 0-1 probability matching): At each period the algorithm finds cycles and chains in the graph such that the number of matched patients is maximized, where patients may be either matched to donors paired with other patients or to altruistic donors.

Algorithm 2 (repeated static stochastic matching): At each period the algorithm finds cycles

and chains in the graph by maximizing the total weight of edges participating in the matching, where patients may be either matched to donors paired with other patients or to altruistic donors.

Algorithm 3 (simple deterministic algorithm): At each period the algorithm again finds cycles and chains in the graph by maximizing the total weight of edges participating in the matching, where patients may be either matched to donors paired with other patients or to altruistic donors. However, the edge is not included in the matching if it has $p < \bar{p}$ for some low value of \bar{p} that can be chosen based on distribution of p .

All three algorithms are essentially implemented by finding perfect maximum weight matching in the graph, which is essentially the same as finding a set of disjoint cycles with the maximum weight. The above described modification to the graph allows us to obtain this. Let's consider a modified version of the simple graph given in figure 4. Add a yellow node numbered 6 as an auxiliary node to altruistic donor (node 5), and assume it to be compatible with all donors. Then the chain in the original graph corresponds to a cycle in the modified graph illustrated in figure 5.

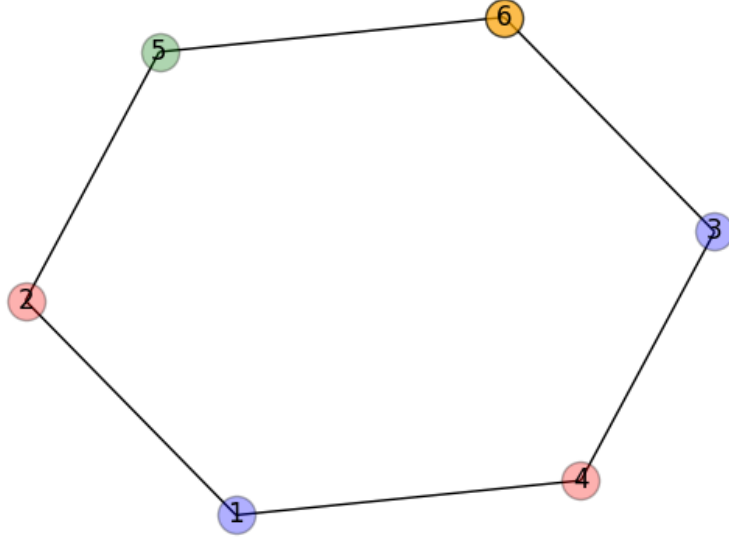


Figure 5: Chain exchange in the modified graph

V. Simulations and Results

First, simulation was performed excluding altruistic donors from the model. Random graph G with $T = 20$ periods and average pair arrival rate $\lambda = 50$ was simulated 40 times in Python using libraries

Networkx and *Numpy* for different values of α and β . Hence, in each simulation 2000 nodes were generated on average. After the graph with all the nodes, edges, different edge probabilities that decrease over time was created, the next step is to randomly toss a coin with probability $p(e, t)$ for each edge e to decide if that edge will succeed if matched in period t . Finally, three algorithms are implemented and their performances are compared with each other, knowing success information for each edge e at time t .



Figure 6: Number of successfully matched patient-donor pairs ($\alpha = 3, \beta = 2$)

Figure 6 shows number of successful matches for three algorithms with parameters $\alpha = 2$ and $\beta = 3$. The figure clearly illustrates that the performance of the *algorithms 1* is much worse relative to the performances of other two algorithms, which have roughly perform in a similar way from the graph.

Define efficiency of the algorithm as the ratio of successfully matched pairs to the total number of pairs in all periods. Figure 7 suggests similar results as figure 6, it shows efficiency rate of different algorithms for parameter values $\alpha = 3, \beta = 2$.

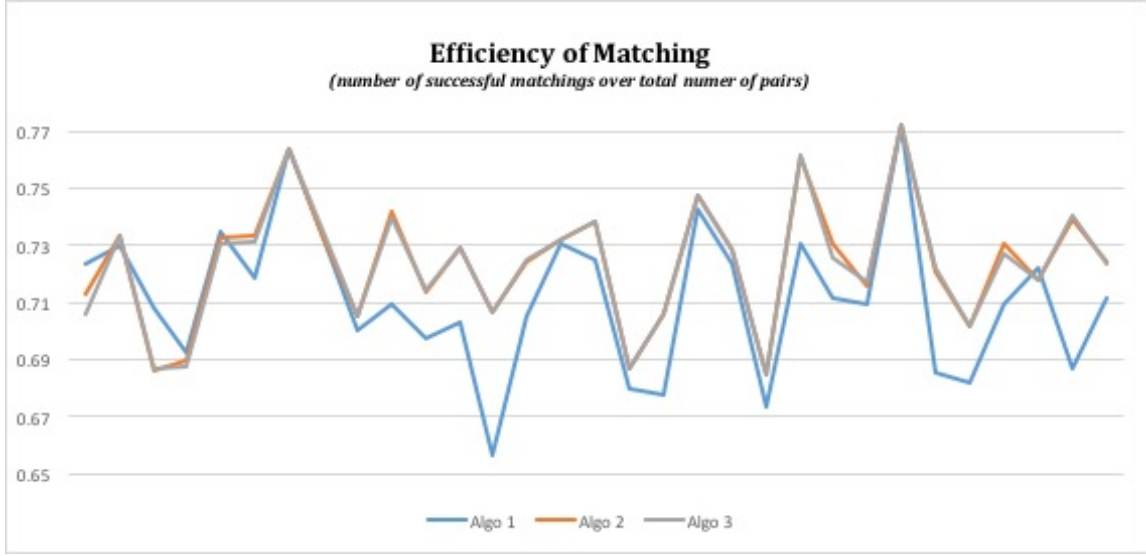


Figure 7: Efficiency of matching algorithms ($\alpha = 2$, $\beta = 3$)

Now let altruistic donors enter the pool. Implementation of simulation is very similar to the previous case. Random graph G with $T = 20$ periods and average pair arrival rate $\lambda = 50$ and average altruistic donor arrival rate $\lambda_a = 3$ was simulated 40 times. In this case on average even greater number of random nodes are generated. Also, in addition to accounting for such key features of the graph as stochastic edge probabilities and cost of waiting, the original graph has to be modified as described in the previous section in order to apply three matching algorithms. After implementing all of the above, all the results obtained are summarized in figure 8. Algorithms 2 and 3 clearly perform much better than algorithm 1. Nevertheless, it is hard to compare performances of algorithms 2 and 3. These two algorithms successfully match 84% to 90% of patients in the pool, which is a very high result.

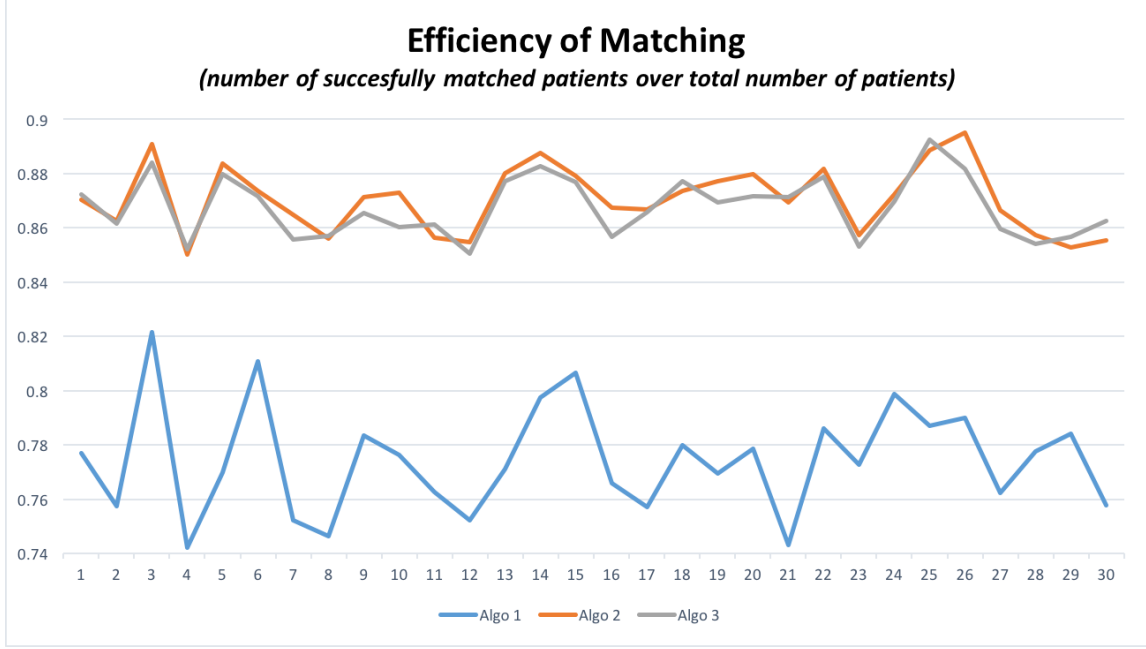


Figure 8: Efficiency of matching algorithms in a model with altruistic donors ($\alpha = 2$, $\beta = 3$)

VI. Conclusion and Extensions

The problem of kidney exchange among incompatible patient-donor pairs can be modeled as an *Online Stochastic Matching* problem. The simulation results obtained in this paper indeed suggest that defining the problem in such a way results in more efficient allocation of pairs. The distributional assumptions are crucial in defining the model. Otherwise, there is no place to start building expectations and predictions about the future. If there is no distributional assumption, then the future is completely uncertain.

The results obtained from simulations illustrate that modeling the kidney exchange problem as a stochastic problem results in a greater number of succesful matches. This is true in a basic model without altruistic donors, as well as in an extended model including altruistic donors. This result is especially evident in the model with altruistic donor. Algorithms 2 and 3 that implement matching considering the stochastic element of the problem perform much better than the algorithm 1 that matches only based on blood type compatibility.

One may think of extending the model further by including to the model patients without donors. While including altruistic donors to the model may still be a reasonable assumption, including patients without donors will make the already complex model even more complicated. The reason is that usually the number of patients without donors is very big, and those patients are also ranked based on different carachteristics. Including all of these will create unnecessary complications to

the model.

Another extension could be relaxing the assumption that an edge between a patient and a donor in a pair has weight of zero. Sometimes patients that are compatible with their donors may still choose to enter the exchange pool in order to find a better suitable pair. This assumption was made in the paper in order to avoid making the model too overwhelming, however it may lead to interesting insights if it is relaxed in some different model settings.

Finally, the model can be further extended by restricting the lengths of cycles to some amount. This is a common assumption made in many papers that consider static kidney exchange problem. Since it may be too complicated to organize exchanges of longer cycles due to capacity constraints in medical centers, this a reasonable assumption to be applied to the online stochastic kidney exchange problem. However, when number of cycles is restricted by amount greater than two, this problem is known to be NP-hard.

References

- [1] National kidney registry, April 2016.
- [2] Stanford blood center, April 2016.
- [3] Tuomas Sandholm John Dickerson, Ariel Procaccia. Dynamic matching via weighted myopia with application to kidney exchange. *Association for the Advancement of Artificial Intelligence*.
- [4] Tuomas Sandholm Pranjal Awasthi. Online stochastic optimization in the large: application to kidney exchange. *Int. Joint Conference on Artificial Intelligence (IJCAI)*.
- [5] Utku Unver. Dynamic kidney exchange. *Review of Economic Studies*, 77(1):372–414, 2010.
- [6] Yan Zhou Alan Leichtman Michael Rees John Kalbfleisch Yijiang Li, Peter Song. Optimal decisions for organ exchanges in a kidney paired donation program. *Stat Biosci*, 6(1):85–104, 2014.

VII. Appendix

1. Definitions

- *Undirected Graph* $G = (V, E)$ is collection of vertices in set S and collection of edges E connecting those vertices, such that each pair of nodes is connected with at most 1 undirected edge.
- *Matching* is a collection of edges of the original undirected graph G , such that no pair of edges have a common vertex.
- *Maximum Matching* is a matching with greatest number of edges.
- *Maximum Weighted Matching* is a matching with a greatest some of edge weights in an undirected weighted graph G .
- *Perfect Matching* is a matching such that every vertex is an endpoint for exactly one edge in a matching.
- *Reduction* is a method used in Algorithms to prove time complexities of different algorithms. If an algorithm can be reduced, that is, transformed in polynomial time, to some instance of another polynomially solvable problem, the original problem is also polynomially solvable. In the paper we reduced different algorithms to *Maximum Weighted Matching* problem to show that they run in polynomial time.
- *NP-hard* problem is a problem that has polynomial time verifier, but was not proven to be solvable in polynomial time.