Harvard CS 1360, Spring 2025
Instructor: Ariel Procaccia
Scribe: Jessica Chen

Matching 2: Kidney Exchange

—

Lecture 14

Motivation: Kidney transplant is a preferred method of treatment for kidney failure. A kidney may be transplanted from either a deceased or live donor, as long as the donor is blood-type and tissue-type compatible. On March 25, 2025, there were 90,489 patients waiting for kidney transplant in the United States, and the number of patients in the queue continues to grow. (Live transplant database)

In some cases, a patient may have a willing but incompatible donor, such as a sibling or spouse with a different blood type. However, if there are two such patient-donor pairs, in which the patients and donors are cross-compatible, then the two transplants can still take place in a pairwise exchange.

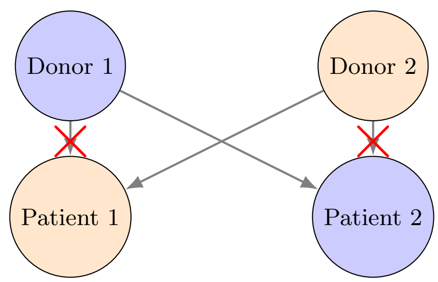


Figure 1: Pairwise exchange between patient-donor pairs. Purple vertices are medically compatible with each other, as are orange vertices. Donor 1 and Patient 1 are a pair (such as siblings), but they are not medically compatible. As such, they exchange with another patient-donor pair.

Note: While the figure above shows a pairwise exchange, we could more generally have exchanges along longer cycles. However, because the operations must take place simultaneously (can't simply promise to donate a kidney later), there is a practical limit to the length of the cycles. Typically, cycles of length 3 is the limit.

Definition 1 (Kidney Exchange Graphs)**.** The model is a directed graph $G = (V, E)$ where V is a set of donor-patient pairs and there is an edge from u to v if the donor of u is compatible with the patient of v.

We formulate the kidney exchange problem as a Cycle-Cover problem.

Definition 2 (Cycle-Cover Problem)**.** Given a directed graph G and $L \in \mathbb{N}$, find a collection of disjoint cycles of length $\leq L$ in G that maximizes the number of covered vertices.

In this problem, each vertex represents a patient-donor pair and each edge represents a transplant. The goal is to find the collection of disjoint cycles of some length L that maximizes the number of covered vertices. The cycles must be disjoint, as one donor cannot give a kidney twice, so each vertex can only be part of one exchange cycle.

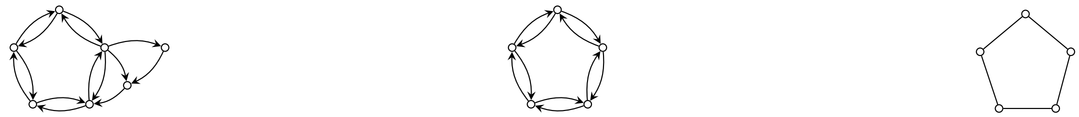
Theorem 1. *For any constant $L \geq 3$,* Cycle-Cover *is NP-complete.*

In Assignment 4, we look at the case where there's no upper-bound limit on the cycle length, so the problem is in P.

The case where $L = 2$ is also tractable. Imagine a graph with cycles of any length. Suppose we were to only focus on the cycles of length 2; we can then replace each 2-cycle with an undirected edge that represents a pairwise exchange. Then, this problem reduces to a maximum cardinality matching problem in an undirected graph. See Figure 2 for an illustration.

However, given that many modern exchanges still have cycle length 3, we will discuss a practical way to solve this problem for $L \geq 3$ using integer programming.

Definition 3 (Kidney Exchange with $L \geq 3$ using Integer Programming)**.** For each cycle c of length $l_c \leq L$, variable $x_c \in \{0, 1\}, x_c = 1$ iff cycle c is included in the cover.



(a) Given a graph with cycles of any length

(b) Focus on cycles of length 2

(c) Now it's a max matching problem!

Figure 2: Cycle-Cover with $L = 2$

The objective is to maximize the number of matched vertices, which we take to be the product of the cycle lengths and whether the cycle is included in the cover. We ensure that each vertex is only in one cycle that has been taken in our cover.

$$\begin{aligned} \textbf{max} \quad & \sum_c x_c \ell_c \\ \textbf{s.t.} \quad & \forall v \in V, \sum_{c\,:\,v \in c} x_c \leq 1 \\ & \forall c,\ x_c \in \{0, 1\} \end{aligned}$$

Example 1 (United Network of Organ Sharing (UNOS))**.** UNOS has been using these IP-based approaches since the inception of the exchange in 2002. The algorithms were originally developed by the Tuomas Sandholm group at Carnegie Mellon. There is still much to be done with regards to optimization; for example, altruistic donors starting chains further complicate the computational problem.

A discussion on game-theoretic incentive problems. In the past, kidney exchanges were carried out by individual hospitals, so there was no incentive issue here. Today, there are nationally organized exchanges, and participating hospitals have little interaction with the operation. It was observed that hospitals find it easy to match patients and donors internally, and they tend to not enroll these internally matched patients and donors to the national network. Instead, hospitals only report the harder-to-match patients to the national network, which creates an inefficiency. As such, the goal is to incentivize hospitals to enroll all their patient-donor pairs to the national network.

Definition 4 (The Strategic Model)**.** The model is defined as follows:

1. It is an undirected graph, where each undirected edge represents a pairwise exchange.
2. The players are hospitals, and each player controls a subset of vertices (the internal patients).
3. The mechanism receives a grpah and returns a matching on that grpah.
4. The utility of a player is the number of vertices they control that have been matched. (This captures the idea that each hospital wants to match as many of its own patients.)
5. The objective is to maximize the utilitarian social welfare, or the number of matched vertices.
6. **Strategy:** The players choose which subset of revealed vertices under their control to reveal to the mechanism. (This captures the enrollment into the national program.) Note the edges are public knowledge; anyone can check if a pair is compatible.
7. The mechanism is strategyproof if it is a dominant strategy to reveal all vertices.

We consider an example that illustrates that the optimal solution is not strategyproof in Figure 3.

Now, we want to try to design a mechanism that is strategyproof. We'll have two takes on this.

Definition 5 (Match$_{\{\{1\},\{2\}\}}$ Mechanism)**.** Consider matchings that maximize the number of *internal edges*, which are edges where both vertices are controlled by the same player. Among these returned matchings will be a matching with maximum cardinality.

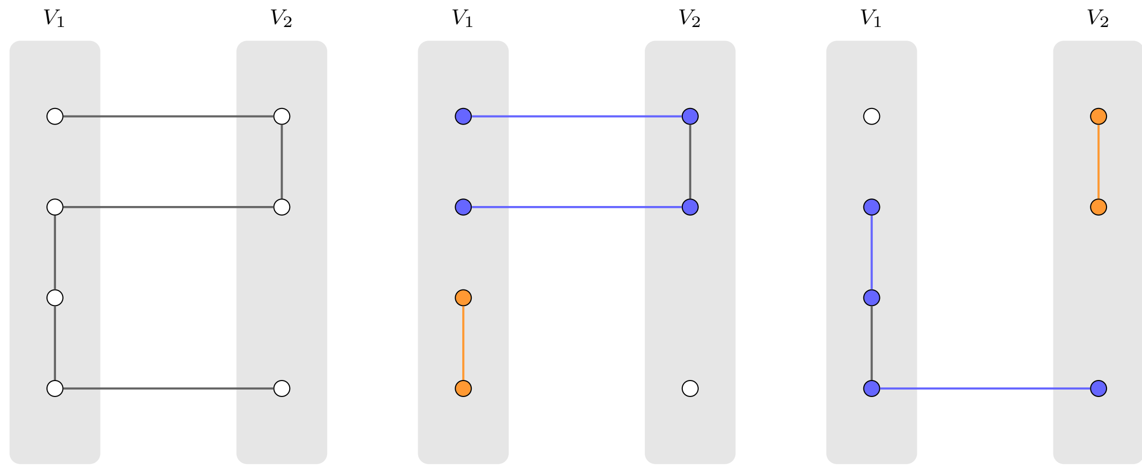


Figure 3: Left is an illustration of an unmatched starting graph where either a vertex in V_1 must remain unmatched, or a vertex in V_2 must remain unmatched. We can see this by simply tracing the possible edges: start matching with the topmost vertex in V_1, and the last vertex in V_2 will be unmatched; start with the topmost vertex in V_2, and the topmost vertex in V_1 will be unmatched. The middle figure shows a useful deviation for the case where V_1 has an unmatched vertex, and the right shows a useful deviation for the case where V_2 has an unmatched vertex. In both deviations, the player with the unmatched vertex can withhold internal matches (orange) and end up improving their total utility.

This strategyproof approximation is helpful, as we help mitigate the incentives problem, while being within some bound away from the optimal solution. Here are some guarantees to the $\textsc{Match}_{\{\{1\},\{2\}\}}$ Mechanism:

1. $\textsc{Match}_{\{\{1\},\{2\}\}}$ gives a 1/2-approximation.
   1.1. In Figure 4, we see that this mechanism gives at best 1/2-approximation.
   1.2. We can also argue that it also has a lower bound of one half. This resulting matching is an inclusion maximal matching, meaning you are not able to add more edges to the matching without taking edges off. We know this is true because it already tries to maximize internal edges subject to and, subject to that, maximize external edges; therefore, you cannot add either internal nor external edges.
   1.3. Similar to the greedy algorithm, for any edge in the optimal matching, at least one of its two vertices must be matched. If both of them were unmatched, then you would have been able to add this edge to the matching.
   1.4. *Note: While the worst-case is 1/2, in practice, the approximation is more like 90%.*
2. **Theorem (special case)**: $\textsc{Match}_{\{\{1\},\{2\}\}}$ is strategyproof for 2 players, which we proof below.

Theorem 2. $\textsc{Match}_{\{\{1\},\{2\}\}}$ *is strategyproof for two players.*

Proof. Let M be the matching output by the mechanism when Player 1 reports all their vertices truthfully, and M' the matching when Player 1 hides some vertices and possibly matches some internally.

We examine the symmetric difference $M \Delta M'$, which consists of vertex-disjoint alternating paths and cycles, where edges alternate between M and M'. The paths must alternate between matchings because, given two edges on one vertex, the edges cannot be part of the same matching using the vertex twice. See Figure 5 for reference.

We want to argue that M is better (has at least as many matched vertices) for Player 1 than $M^{'}$. In other words, Player 1 did not gain by hiding some of their vertices. We do this by showing M is at least as

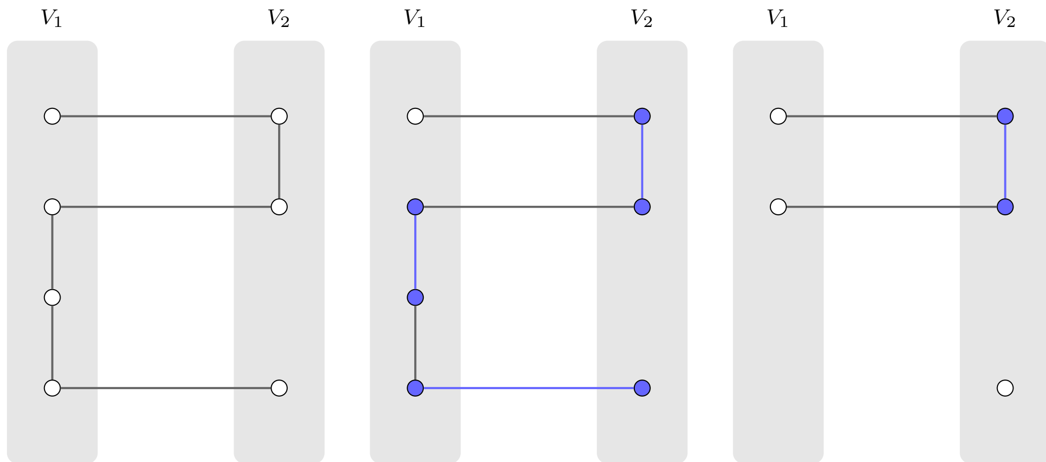


Figure 4: $\textsc{Match}_{\{\{1\},\{2\}\}}$ Mechanism: Given the edges as shown in the left subfigure, the mechanism will choose the matchings that maximize the number of internal edges. The middle subfigure shows such a matching: both V_1 and V_2 get one internal matching each, which is the max possible. In the left figure, we show that V_1 choosing to hide an internal match is not a useful deviation: because the mechanism would still maximize the internal matching of V_2, only the two vertices in V_2 would be matched, and V_1 would still only have 2 vertices matched. This is in contrast to the optimal solution, where V_1 could still externally match its last vertex to V_2's last vertex.

good as $M^{'}$ for every component (every path and cycles) of the symmetric difference. On cycles, M and M' match the same set of vertices, so Player 1's utility is the same; we therefore focus on paths.

Consider a single path in the symmetric difference $M \Delta M^{'}$, and denote its edges in M by P and its edges in $M^{'}$ by $P^{'}$. For $i, j \in \{1, 2\}$,

$$P_{ij} = \{(u, v) \in P : u \in V_i, v \in V_j\}$$
$$P^{'}{}_{ij} = \{(u, v) \in P^{'} : u \in V_i, v \in V_j\}$$

In other words, P_{ij} represents the edges in P that are between i and j. Therefore P_{11} is internal to Player 1, P_{22} is internal to Player 2, P_{12} is external. Same goes for $P^{'}$.

First, we claim $|P_{11}| \geq |P^{'}_{11}|$. This is true because, when it has all the private information, the mechanism already maximizes internal edges. This relation translates to apply to every path separately: if you had some path where $M^{'}$ had more edges, then we could switch the edges from M to $M^{'}$ and get more internal edges for M. Given this weak inequality, let us consider two cases:

Case 1: $|P_{11}| = |P'_{11}|$ In this case, both matchings contain the same number of internal edges for Player 1. Since Player 2's vertices are unchanged, the mechanism has access to the same information of vertices and edges within V_2 under both M and M'. Hence, it also holds that:

$$|P_{22}| = |P'_{22}|$$

Now, recall that the mechanism selects a matching of maximum cardinality after maximizing internal edges. $M, M^{'}$ have the same constraints, but M has more information, so it can do better in terms of external edges. Therefore:

$$|P_{12}| \geq |P'_{12}|$$

Now we calculate the utility of Player 1. For every edge in P_{11}, there are two matched vertices of Player 1, and for every edge in P_{12}, there is one matched vertex of Player 1. Putting it together, the utility of Player 1 under M is:

$$U_1(P) = 2|P_{11}| + |P_{12}| \geq 2|P'_{11}| + |P'_{12}| = U_1(P')$$

Hence, Player 1 does not gain by deviating in this case.

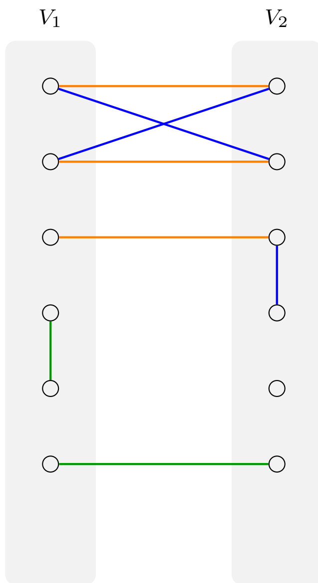


Figure 5: Supplement for proof of Theorem 2: $\textsc{Match}_{\{\{1\},\{2\}\}}$ is strategyproof for two players. This bipartite graph shows matchings M (orange), M' (blue), and their symmetric difference (green).

Case 2: $|P_{11}| > |P'_{11}|$ This case is more delicate. While M has more internal matches for Player 1, it is possible that M' has more external edges. We now argue that any gain from extra external matches in M' cannot compensate for the loss in internal matches.

The key observation is that

$$|P_{12}| \geq |P'_{12}| - 2$$

This bound arises because (see Figure 6 for reference):

- Each subpath in V_2 must be of even length. If we had an odd-length subgraph (such as orange-blue-orange), we would be able to switch the edges of one matching on the subpath with the edges of the other matching and increase the number of matched internal edges in V_2 (instead of matching blue, we match the two orange edges, thus getting another internal edge).
- Given a subpath of even length, then we know there is the edge "entering" V_2 must be the opposite color/matching than the edge "exiting" V_2. (If subpath is orange-blue, then entering is blue, exiting is orange.)
- It follows that we can pair up external edges in P_{12} and P'_{12} along the path, except potentially the first and last external edges.
- Therefore, the number of unpaired external edges in P' is at most 2 more than in P.

Using this, we compute:

$$\begin{aligned} U_1(P) &= 2|P_{11}| + |P_{12}| \\ &\geq 2(|P'_{11}| + 1) + (|P'_{12}| - 2) \\ &= 2|P'_{11}| + |P'_{12}| \\ &= U_1(P') \end{aligned}$$

So even in this worst-case alignment, Player 1 cannot gain utility by hiding vertices.

□

Unfortunately, the strategyproofness of $\textsc{Match}_{\{\{1\},\{2\}\}}$ does not generalize to settings with more than 2 players. In particular, when there are 3 or more players, maximizing internal edges is no longer strategyproof.

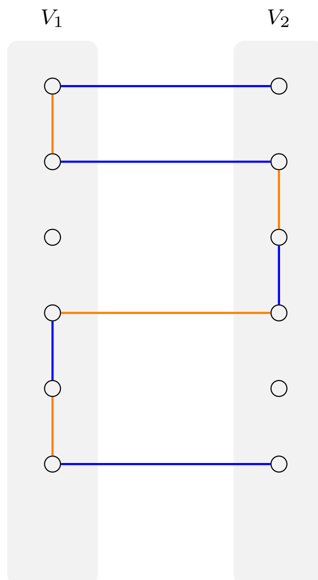


Figure 6: Supplement for proof of Theorem 2: $\text{Match}_{\{\{1\},\{2\}\}}$ is strategyproof for two players. Specifically, this figure shows that if $|P_{11}| > |P_{11}^{'}|$, then $|P_{12}| \geq |P_{12}^{'}| - 2$. Again, the bipartite graph has matchings M (orange) and M' (blue). Each subpath in V_2 must be of even length (see v_1, v_2), as otherwise we would be able to switch the edges of one matching on the subpath with the edges of the other matching, and thus increase the number of matched internal edges in V_2. Because we know the subpaths are even, then the edge "entering" must be a different color than the one "exiting" (blue going into v_1, orange coming out of v_3. Therefore, we can pair up edges in P_{12} and $P_{12}^{'}$, except for maybe the first and last external edges (at v_0 and v_5

. Thus, the bound is $|P_{12}| \geq |P_{12}^{'}| - 2$.

Example 2 ($\text{Match}_{\{\{1\},\{2\}\}}$ is not SP for 3 players)**.** Suppose we have 3 players with vertex sets V_1, V_2, V_3, and the mechanism selects a matching that maximizes the number of internal edges. Then, as shown in the example in Figure 7, it may happen that one player (say, V_1) can benefit by withholding an internal edge and matching it privately. This can result in additional external matchings that increase their utility, violating strategyproofness.

This motivates us to define a more robust mechanism.

Definition 6 (Match_{Π} Mechanism)**.** Let $\Pi = (\Pi_1, \Pi_2)$ be a bipartition of the set of players. Then Match_{Π} proceeds as follows:

1. Consider all matchings that:
   - Maximize the number of internal edges
   - Contain no edges between different players on the same side of the partition
2. Among these, return one with maximum cardinality.

The key idea is that the bipartition prevents players from colluding or influencing the mechanism by hiding joint matches with players on their side.

Theorem 3. Match_{Π} *is strategyproof for any number of players and any partition* Π.

The proof for this theorem is a generalization of the two-player mechanism from before. $\text{Match}_{\{\{1\},\{2\}\}}$ puts one player on one side of the partition and the other player on the other side. When there are just two players, one on either side, there are no forbidden edges. In other words, for $n = 2$, and $\Pi = (\{1\}, \{2\})$, this reduces to $\text{Match}_{\{\{1\},\{2\}\}}$, which we showed gives a 1/2-approximation.

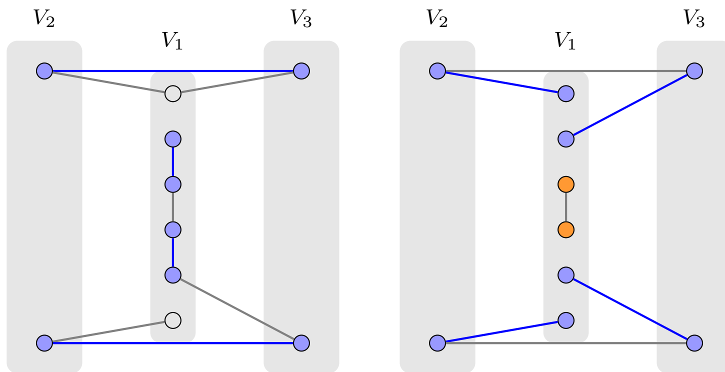


Figure 7: An example where maximizing internal edges is not strategyproof for 3 players.

Example 3. What are the approximation guarantees given by $\textsc{Match}_\Pi$ for $n \geq 3$ and $\Pi = \{\{1\}, \{2, 3\}\}$? Imagine a graph where the only edges are only between Players 2 and 3. The mechanism is not allowed to match anything, since it is not allowed to match edges between players on the same side of the partition. Therefore, it could match nothing, whereas the optimal solution is however many edges you want. The approximation ratio is 0!

To reconcile the goals of strategyproofness and approximation quality, we define a randomized mechanism.

Definition 7 (Mix-and-Match Mechanism)**.** The Mix-and-Match mechanism operates as follows:

1. **Mix:** Choose a random partition Π of the players (uniformly at random).
2. **Match:** Run $\textsc{Match}_\Pi$ on the chosen partition.

Theorem 4. *The Mix-and-Match mechanism is strategyproof and guarantees a $\frac{1}{2}$-approximation to the maximum matching.*

This is a surprising approximation, since there are 2 sources of loss here. First, you may be forced to take one internal edge instead of two external edges. Second, you are not allowed to match players on the same side of the partition, and every external edge is thus forbidden with probability 1/2. You would think that, given that we are losing a factor of 2 twice, we would get a 1/4 approximation. However, we have a 1/2 approximation because (as it turns out) these two sources of loss are mutually exclusive.