Algorithmic Game Theory, Summer 2025 — Lecture 21 (5 pages)

Allocations without Money

Thomas Kesselheim

Last Update: June 28, 2025

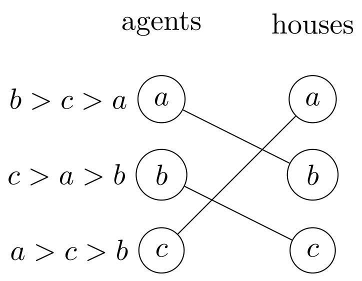
We have learned quite a bit about mechanism design *with money.* The general theme was to set rules of a game so that everybody is happy with what we do. Today, we will turn to similar questions *without money.* Often, we face agents behaving strategically but imposing payments is infeasible or would be unethical.

1 House-Allocation Problem

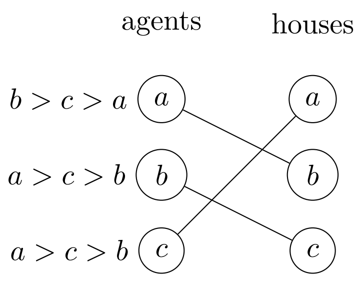
We start with a simple problem that has a lot of similarities with the auction problems we considered before. There are *n agents* N. Each of them brings an item to an exchange. We refer to the items as *houses.* Agents might prefer other houses to the one they live in. Our goal is to (possibly) reassign the houses so that all agents want to participate and are reasonably satisfied with the outcome. That is, we look for an allocation $\pi\colon N \to N$ that is a one-to-one correspondence (or a permutation) between agents and houses. After the reassignment, agent i gets the house that used to belong to agent $\pi(i)$.

In settings with money, we always expressed agents' preferences in terms of monetary units and called this valuation. In a world without money, this makes no sense. Therefore, instead each of the agents now has a complete preference list without ties over the houses. This is a total order on all agents' houses, including his own. It may be arbitrary and need not be consistent with other agents' preferences. We write $a > b > c$ if an agent prefers a's house over b's but likes both better than c's.

Let us start with an almost trivial example. There are three agents a, b, and c. Each of them likes the clockwise neighbor's house better. So, this is clearly a good allocation because all agents get their most preferred house.



Usually, exchanges will not be this smooth. Consider the following example, in which only the second agent's preference changes.



This allocation is what we call *unstable.* Agent b only has to convince agent a to leave the exchange and swap their houses outside. Agent a doesn't mind because he still gets b's house but b likes a's house better than c's.

Let us define this property formally.

Definition 21.1. *A subset of the agents $A \subseteq N$ is a* blocking coalition *of an allocation π, if there is an allocation $\sigma\colon A \to A$ only involving the agents in A such that:*

1. *No $i \in A$ prefers $\pi(i)$ over $\sigma(i)$ and*
2. *At least one $i \in A$ prefers $\sigma(i)$ over $\pi(i)$.*

So, in other words, a blocking coalition would rather reassign the houses among themselves: It would make nobody less happy but somebody would be happier than in allocation π.

Definition 21.2. *We call an allocation* stable *if there is no blocking coalition.*

We are interested in finding stable allocations because they ensure that all agents actually benefit from taking part in the exchange. Nobody would be better off by asking a couple of friends to run a secondary exchange.

2 Top Trading Cycle Algorithm

Interestingly, there is always a stable allocation and it can be found using a very elegant algorithm.

- Initially set $N' = N$
- While $N' \neq \emptyset$
  - Construct a directed graph on vertices N'. Each vertex i has one outgoing edge to the owner of his preferred house among all houses owned by N'. (This can also be i himself. In this case, there will be a loop at vertex i.)
  - Find an arbitrary directed cycle in this graph. To each i in this cycle, assign the house owned by i's successor in the cycle.
  - Remove all agents in the cycle from N'.

To make the algorithm well-defined we need that there always is a directed cycle in the constructed graph. Fortunately, this is an easy observation. (There might be multiple, but this is no problem.)

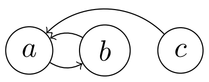
Lemma 21.3. *Every directed graph without sinks (that is, every vertex has an outgoing edge) has a cycle.*

Proof. Start from any vertex and follow an arbitrary outgoing edge. While we see new vertices, keep following outgoing edges. At some point, we have to come back to a vertex we have been to before because there are only finitely many vertices. Then a cycle has been completed. □

Let us come back to the initial example with the following preferences:

$$a : b > c > a, \quad b : a > c > b, \quad c : a > c > b$$

The graph in the first iteration is



In the second iteration, it is only the vertex c with a self-loop.

2.1 Stability of the Allocation

Next, we show that the allocation we compute using this algorithm is actually stable, meaning that there is no blocking coalition.

Theorem 21.4. *The allocation π outputted by the Top Trading Cycles Algorithm is stable.*

Proof. Suppose the allocation is not stable. Then, there has to be a coalition $A \subseteq N$ and an allocation $\sigma\colon A \to A$ within A with the following property. Let $A_{\neq}$ denote the set of agents $i \in A$ for which $\pi(i) \neq \sigma(i)$. We have to have $A_{\neq} \neq \emptyset$ and all agents $i \in A_{\neq}$ prefer $\sigma(i)$ over $\pi(i)$. (Here we use that there are no ties in the preferences.)

There has to be an iteration of the algorithm in which $\sigma(i)$ gets removed from the set N' for $i \in A_{\neq}$. Consider the first such iteration and let N^* denote the state of N' before it and let $i^* \in A_{\neq}$ such that $\sigma(i^*)$ gets removed.

Now consider $a_1 = \sigma(i^*), a_2 = \sigma(\sigma(i^*)), \ldots$. As i^* has to appear in this sequence, there has to be a smallest j such that $\sigma(a_j) \neq \pi(a_j)$. Note that $\pi(a_{j'}) = \sigma(a_{j'})$ for all $j' < j$. So, a_j is removed in this iteration. So, we know that a_j prefers $\pi(a_j)$ most in N^*. This is a contradiction to a_j preferring $\sigma(a_j)$ over $\pi(a_j)$. □

2.2 Uniqueness of the Stable Allocation

Quite surprisingly, the stable allocation is unique. That is, if we took a different cycle in the graph or even a totally different algorithm, the result would still be the same.

Theorem 21.5. *The allocation π outputted by the Top Trading Cycles Algorithm is the unique stable allocation.*

Proof. Let π' be any stable allocation. Furthermore, let N_t be the set of agents who are allocated in the t-th iteration of the algorithm. We will show that $\pi(i) = \pi'(i)$ for all $i \in N_t$ for all t by induction on t.

The base case of our induction is $t = 1$. By definition all $i \in N_1$ get their favorite houses. Furthermore $\sigma\colon N_1 \to N_1$ with $\sigma(i) = \pi(i)$ is a feasible allocation within N_1 because π is derived from a cycle in N_1. So, if $\pi'(i) \neq \pi(i)$ for any $i \in N_1$, then N_1 is a blocking coalition.

For the induction step, consider any $t > 1$. By induction hypothesis, we already know that $\pi'(i) = \pi(i)$ for all $i \in \bigcup_{t'<t} N_{t'}$ and that this allocation is among agents and houses who are all in $\bigcup_{t'<t} N_{t'}$. This means that both π and π' have to reallocate houses within $N' = \bigcup_{t' \geq t} N_{t'}$. So, pretend agents and houses in $N \setminus N'$ never existed in the first place. Then, the algorithm would first reassign houses within N_t. This is exactly what we discussed in the base case. □

2.3 Dominant-Strategy Incentive Compatibility

Not only is there exactly one stable allocation, the agents also have no incentive to misreport their preferences: No agent gets a better house by reporting a preference list other than the true one, regardless of what the other agents do. That is, the mechanism is dominant-strategy incentive compatible. (In the context of mechanism design without money, this property is also called strategyproofness.)

Theorem 21.6. *No agent can improve his allocation by misreporting the preference list.*

Proof. Fix some agent i and the other agents' preferences. Furthermore, let N_t be the set of agents who are allocated in the t-th iteration of the algorithm. Let t^* be the iteration in which agent i is allocated a house if he reports the true preference list.

By misreporting the preferences, agent i could be served in an iteration before t^*. However, he will not get any of the houses from $\bigcup_{t<t^*} N_t$ because there is no incoming edge from any of these vertices. That is, no matter what he reports, his allocation will be one of the houses in $\bigcup_{t \geq t^*} N_t$. By definition, his house when reporting the true preferences is exactly the most preferred one among these. □

3 Kidney Exchange

Possibly the most compelling motivation for mechanism design without money are organ donations. There are good reasons why it is forbidden to ask for or pay money for these.

You may have heard of examples of living organ donors. For example, people can donate one of their kidneys to save a relative's life. For example, Germany's president Frank-Walter Steinmeier donated one of his kidneys to his wife in 2010. Such a donation requires the tissues to be compatible. But what if all your relatives who would be willing and able to donate are incompatible? Ideally, you would find another patient/donor pair that has the same problem but the kidneys are compatible with the respective other patient. You could even extend this to a chain of exchanges.

Indeed, Roth, Sönmez, and Ünver proposed in 2004 to use the Top Trading Cycles Algorithm for this problem. Each patient/donor pair becomes an agent. An agent's preferences express how likely it is that the other agent's kidney is compatible. Then, we get a cycle of kidneys to be transplanted.

Unfortunately, there is one flaw in this argument: All surgeries have to take place simultaneously, close to each other. The reason is that nobody can be forced to donate an organ. So, if you are a donor and your relative that brought you in the exchange already got a kidney, you could simply walk away and this way break the cycle.

For this reason, one has to keep the cycles short. We will now consider a problem in which the cycles are limited to length 2. Also, the complete preference list is replaced by a simpler model.

4 Pairwise Kidney Exchange by Matching

An easier approach to kidney exchange relies on finding a matching in a graph. The agents (that is, patient-donor pairs) are vertices V in a graph. There is a set of (undirected) edges E on V indicating the possible pair-wise exchanges. Our goal is to compute a maximum-cardinality matching $M \subseteq E$.

Let E_i denote the set of agent i's neighbors according to E. Note that every agent can deny the surgery without stating a reason. Therefore, effectively, they *report* edge sets $F_i \subseteq E_i$. We would like our mechanism to ensure that no agent can get himself into the matching by misreporting $F_i \neq E_i$. That is, reporting $F_i = E_i$ should be a dominant strategy; the mechanism should be dominant-strategy incentive compatible.

- Collect a report F_i from every agent i.
- Let $F = \{\{i, j\} \mid i \in F_j, j \in F_i\}$ be the set of *reported* edges.
- Let M_0 denote the set of maximum matchings on (V, F).
- For $i = 1, \dots, n$ in any order that is independent of the reports
  - Let Z_i denote the matchings in M_{i-1} that match vertex i.
  - If $Z_i \neq \emptyset$, let $M_i = Z_i$; otherwise let $M_i = M_{i-1}$.
- Return an arbitrary matching in M_n.

Observe that there might be multiple matchings in M_n. However, they all match the same agents: Exactly the ones for whom $Z_i \neq \emptyset$.

Theorem 21.7. *Given any reports of the other agents, if agent i gets matched when reporting any $F_i \subseteq E_i$, he also gets matched when reporting E_i.*

Proof. First of all, observe that there are two cases that we distinguish. It might be that under report E_i the maximum matching is strictly larger than under reports F_i. Then, i is matched in every maximum matching when reporting E_i and therefore always gets matched.

The other case is that the size of the maximum matching does not change. Now let $Z_0, \ldots, Z_n$ be the sets generated if agent i reports F_i and $Z'_0, \ldots, Z'_n$ if he reports $E_i \supseteq F_i$. We may have $Z_{t'} \subseteq Z'_{t'}$ for all $t' \leq i$. In this case, $Z'_i \supseteq Z_i \neq \emptyset$ and we are done.

Otherwise, there has to be a point t at which $Z_t = \emptyset$ but $Z'_t \neq \emptyset$. This can only happen because Z'_t contains matchings that match i. Therefore, for all $t' \geq t$, $Z_{t'}$ only contains matchings that match i and so we are done. □

Further Reading

- Section 10.4 in the Karlin/Peres book
- Section 10.3 in the AGT book
- Tim Roughgarden's lecture notes `http://theory.stanford.edu/~tim/f13/l/l9.pdf` and lecture video `https://youtu.be/zV6yH3-AdEg?t=58m4s` on House Allocation
- Tim Roughgarden's lecture notes `http://theory.stanford.edu/~tim/f13/l/l10.pdf` and lecture video `https://youtu.be/NT07sILhsv4` on Kidney Exchange
- Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Kidney Exchange. Quarterly Journal of Economics, 119(2):457–488, 2004. (Top Trading Cycles for Kidney Exchange)
- Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver, Pairwise Kidney Exchange, Journal of Economic Theory, 125:151–188, 2005. (Matching for Kidney Exchange)