Matching Markets

Seminar on Markets Algorithms Incentives Networks

Mihir Mahajan
Department of Informatics
Technische Universität München
Email: mihir.mahajan@tum.de

***Abstract*— Matching Markets have been studied for years in several fields, like economics, operations research, and game theory. We focus on the design of matching markets and handle problems of assignments where each participant gets matched (either to another participant or to an item), while participants differ in preferences over their matches. We assume there is no monetary exchange involved, which means the participants' incentives and strategies rely exclusively on their personal preferences.**
Out of a variety of said problems, the focus will lay on the stable marriage problem, house allocation problem, and housing markets problem, while discussing their requirements, solutions, key properties, and importance. We compare the latter two problems which raise some interesting properties and probabilistic similarities.

I. Introduction

First, we will define some concepts that we will use below.

A. Definitions

1) Agents: We define a set of agents N, who participate in our market.

2) Outcome: The result of a matching mechanism is called an outcome $o \in O$, while O describes all possible outcomes.

3) Preference order: Each agent $i \in N$ has a weak preference order $\succeq_i \in R$ over different matching outcomes O.
Agent i weakly prefers outcome o over o': $o \succeq_i o'$.
Agent i strictly prefers outcome o over o': $o \succeq_i o'$ but not $o \preceq_i o'$ ($\implies o \succ_i o'$).
Agent i is indifferent about o and o': $o \sim_i o'$.

4) Preference profile: We call $\succeq := (\succeq_1, \dots \succeq_n) \in R^n$ a preference profile, while $n = |N|$.

5) Choice: The choice $X_i(R)$ of agent $i \in N$ over a set of objects or other agents R is the most preferred agent/object among R based on $i's$ preferences.

6) Reports: While those true preferences are private to their respective agents, Let $\hat{\succeq}_i$ be a report of agent i, which might differ from $\succeq_i$, which we will see throughout the topic.

7) Mechanism: An algorithm, which computes a suitable outcome $g(\hat{\succeq}) \in O$ based on **simultaneous** reports by each agent $i \in N$.
It is mentionable that the mechanism solely acts on the rules g and has $\hat{\succeq}$ as its only input.

8) Dictatorial: A matching mechanism is dictatorial if the outcome of the mechanism mirrors a single participants preferences, without consideration of others.

9) Dominant strategy: A dominant strategy is the best move for an individual agent to make regardless of how other agents act.

10) Strategy proofness: If it is a dominant-strategy equilibrium for each agent to report their real preferences ($\succeq_i = \hat{\succeq}_i$), we call that mechanism strategy proof.

11) Blocking pair: A blocking pair is a pair of agents that prefer eachother over their respective matches. An example for a blocking pair would be (w, m) if $m \succ_w \mu(m) \wedge w \succ_m \mu(w)$. If $\phi \succ_w \mu(w)$, we also call (ϕ, w) a blocking pair, since w could still break away after the matching.

12) Stable matching: A matching is stable if no pair of participants prefer each other over their assigned matches. More formally, matching μ is stable iff there is no blocking pair.

13) Optimal matching: A matching is optimal when each agent is matched with their most prefered *achievable* preference, while agent m is achievable to agent w if a stable matching μ exists with $\mu(w) = m$.

14) Pareto optimality: A assignment is pareto optimal if there are no two agents $i, k \in S$, who would prefer to exchange their assigned items with each other and therefore break away ($r'_i \succ_i r_i \wedge r'_k \succeq_k r_k$).

15) Core: The core is a set of outcomes, which can't be improved by coalition of a subset of agents in the market.

16) Ex post Pareto optimality: Is a property for randomized mechanisms and means that it is pareto optimal for each possible outcome.

17) Blocking coalition: We define a blocking coalition as a subset of agents, that can do better if they trade amongst themselves (outside of the market).

18) Core assignment: An assignment is in core if there is no blocking coalition.

19) Chain: A chain in a directed graph is a succession of edges (and vertices) that connect two vertices in the graph. In our definition, vertices can occur at most once in a chain. The head of the chain is the last vertex in direction of the graph, while the tail is the first vertex.

20) Cycle: We define cycles in graphs as a chain, where head and tail are the same vertex.

B. Assumptions

To keep things simple and escape the results of the impossibility theorem [1], there are a set of assumptions we have to make. In two-sided matchings and assignment problems, we assume agents to be indifferent about parts of the matching or assignment in which they are not directly involved. Furthermore, payments are not available thus agents strictly act based on their personal preferences. Additionally, all problems are simultaneous move games where all agents report their (alleged) preferences at once.

II. Two-Sided Matching Markets

There are two disjoint sets of agents, in which we want to match each participant from one set to at most one participant from the other set, where the outcome is a matching μ. To make this more intuitive, let M be a group of men, which need to be matched to a set of women W for example for marriage. Let $\mu(m) \in W \cup \{\phi\}$ and $\mu(w) \in M \cup \{\phi\}$ denote a match between man $m \in M$ and woman $w \in W$, while ϕ means that an agent is left unmatched. Clearly, if m is matched with w, then w also has to be matched with m. More formally: $\mu : M \cup W \rightarrow M \cup W \cup \{\phi\}$, while $\mu(m) = w \iff \mu(w) = m$. We assume, that agents have strict preference orders over agents on the other side. Let $\succ_m$ and $\succ_w$, denote the strict preference order of man m over women and of woman w over men respectively. E.g. $m_1 \succ_w m_2$ means that woman w had a strict preference of m_1 over m_2. If woman w prefers to stay unmatched, than being matched with man m (m is unacceptable to w), we denote $\phi \succ_w m$. Strict preference orders of participants over agents, lead to weak preference orders over matchings, e.g. $\mu \sim_m \mu'$, if $\mu(m) = \mu'(m)$ for man m (based on our first assumption). First of all, we want our market to be safe to participate in. For that, we need our one-on-one, two-sided matching mechanism to be stable. To solve this so called *stable marriage problem* we use the deferred acceptance procedure aka. Gale Shapley algorithm [2].

A. Deferred Acceptance Procedure (Gale Shapley Algorithm)

- *Input, Output:* The deffered acceptance procedure (DA) takes a preference profile over each agent as an input and returns a stable matching.
- *Description:* It has one set of agents that iteratively make offers (propose) to agents of the other set, while they tentatively accept the best offers until they get a better one.
- *Time complexity:* DA has a time complexity in $O(nm)$, while n is the number of men and m is the number of women.

This algorithm offers two variants, which differ in which set of agents carry out the proposals. The resulting matchings might differ but are guaranteed to be stable. We will use our running example and describe the **women-proposing** variant.

1) Woman proposing DA:

- *First iteration ($k = 1$):* Each woman proposes to her most preferred man, who tentatively accepts his most preferred proposal (or stays with ϕ) and rejects all other ones.
- *Further iterations ($k > 1$):*Each woman, who is not engaged in a pair, proposes to her most preferred man, who hasn't rejected her offer yet (or ϕ if that is most preferred) while men tentatively accept their most preferred proposal (or ϕ), while rejecting others. Consider that men can break off accepted pairs if they receive a better offer. DA terminates, when there is no more proposal.

2) Woman optimal matching: The outcome of the *woman proposing* DA is the *woman's optimal* matching, which means this mechanism is *optimal* for women. Thus there are no incentives for women to falsely report their preferences in woman proposing DA.

3) Lattice property: Stable matchings form a so-called *lattice*. A lattice is an abstract algebraic structure or can also be defined as a partially ordered set. While we are not interested in the detailed mathematical properties of a lattice here, it gives us a few interesting results.

- The *join* of two stable matchings $\mu \vee \mu'$ also forms a stable matching, which is as good or an even more preferred outcome for each *woman*, since each *woman* gets matched with the more preferred man out of both matchings.
- The *meet* of two stable matchings $\mu \wedge \mu'$ also forms a stable matching, which is as good or an even more preferred outcome for each *man*, since each *man* gets matched with the more preferred woman out of both matchings.
- The woman-optimal stable matching (which is the result of the woman proposing DA) is the least preferred outcome for the men. If it is also the most preferred, then it is the only stable matching.

It is important to understand, that agents don't carry out the actions of proposing and rejecting/accepting, but this is handled by the algorithm only. The participants of the market can only report a preference order. Based on the results we have drawn above, one can see that it is a *dominant strategy* for *women* to truthfully report their preferences (in women proposing DA). This is not the case for men. Unfortunately, there is no matching algorithm that would be *stable* and *strategy-proof* for all agents [1]. So we can only guarantee stability (over reported preferences).

III. Assignment Problems

Assignments are problems, where agents $s \in S$ with preference orders are now matched with indivisible items $r \in R = \{1, \ldots, n\}$ who don't have any preferences (e.g. students being assigned to dorm rooms). We use the same notation for preference orders, while each agent s has a strict preference order $\succ_s$ over items, which results in a weak preference order over outcomes of the assignment (equivalent to matching). Let $r_s \in R$ be the item (room) assigned to agent (student) $s \in S$

A. House Allocation Problem

In the house allocation problem, agents, who have no initial assignments are assigned to items. We will explain this based on students $s \in S$ being assigned to dorm rooms $r \in R$. We want our assignment to be pareto optimal. Let n be the number of students. We assume there are just as many students as rooms available in our market ($n := |S| = |R|$).

1) Serial dictatorship (SD):

- *Input:* Priority order π of students and their respective preference reports. Since our priority order π is a permutation of set S there are exactly $n!$ different possible priority orders.
- *Description:* The mechanism iterates through given priority order and in each step $k \in \{1, \ldots, n\}$ assigns the k'th agent their most preferred available item.
- *First iteration ($k = 1$):* We assign the first student $s_1 = \pi(1)$ their most preferred dorm room $r_{s_1} = X_{s_1}(R)$. Note that this means that the algorithm is dictatorial for the first student.
- *Further iterations ($k > 1$):* Now out of the still available rooms $R_k = R \setminus \{r_{s_1}, \ldots, r_{s_{k-1}}\}$, we give the k'th student $s_k = \pi(k)$ their most preferred room out of R_k: $r_{s_k} = X_{s_k}(R_k)$
- *Time complexity:* $O(n)$, while n is the number of agents.

The SD mechanism is *strategy-proof* and *pareto optimal*.

2) Random serial dictatorship (RSD): The serial dictatorship is unfair or at least poses the question of how to choose priority order π. So we use the random serial dictatorship mechanism.

- *Input:* Preference order of each agent.
- *Description:* The random serial dictatorship (RSD) runs the SD mechanism on a random priority order. The probability distribution is a uniform distribution over all possible priority orders (permutations).
- *Time complexity:* Also $O(n)$, with n as the number of agents, since creating a random permutation also is in $O(n)$ (e.g. Fisher-Yates shuffle algorithm).

RSD is also *strategy-proof* and *ex-post pareto optimal* (pareto optimal for each permutation) and *anonymous*.

B. Housing Markets Problem

Now we handle the assignment problem, where agents already hold initial items, and we try to reassign those based on each agent's reported preferences. We want our assignment to fulfill the core property.

1) Top Trading Cycles (TTC): The TTC mechanism essentially detects trading cycles of agents and executes the trades. This is done iteratively until there is no agent left to trade.

- *Input:* We take the simultaneous reports of each student, as well as their currently held room as input.
- *First iteration ($k = 1$):* We build a directed graph with students as nodes, while each student has one outgoing edge pointing to the student holding their most preferred item. We detect cycles (trading cycles), assign each agent (from the cycle) the item of the agent they were pointing to (trade) and remove them from the graph (with their respective items). Let us denote S_1 for the subset of students, who traded in the first round and denote D_1 for the dorm rooms that were traded in the first round.
- *Further iterations ($k > 1$):* We update the edges of the remaining students to their most preferred items, which are still available in the graph. Once again we detect trading cycles and execute the trade for these students S_k and remove them with their respective room D_k from the graph. We repeat the process until there are no more agents left in the graph. Note that S_k and D_k are the set of students/rooms who *trade/are traded* in round k. This is not to be confused with R_k from our SD mechanism, which were the rooms *left available* in round k!
- *Time complexity:* $O(n^2)$ while n is the number of agents.

In each round of TTC, there is at least one cycle in the constructed graph and no node is engaged in more than one cycle.

In round $k-1$, any student in S_k will only be part of a cycle in the next round. These students form at least one chain, while the head of the chain h points to a student in S_{k-1}, since it can not point to a student outside of these both sets $S \setminus (S_{k-1} \cup S_k)$, because then this edge would not change in the next round and thus not close the cycle. Naturally the head can also not point to a student in S_k, because then they would form a cycle that could be traded in round $k-1$, resulting this cycle being part of S_{k-1}.

C. RSD and TTC

While the serial dictatorship sounds unfair the randomized version also seems rudimentary. TTC seems more promising. Still, RSD is used widely to tackle assignment problems. One could get the idea to simulate the house allocation problem with the housing markets problem (running the TTC mechanism) using randomized initial assignments. This randomization would also be a uniform distribution over all possible assignments. Note that in our running example with n students and dorm rooms, there would be $n!$ possible initial assignments. It turns out that the RSD is the same lottery mechanism as our randomized TTC [3]. An intuition for this is that in our TTC mechanism the different rounds form a kind of priority order. Students from S^1 have more priority than S^2 and students S^2 have more priority than the ones in S^3 etc. Thus using the RSD for house allocation problems seems to be more justified now despite its simple approach.

IV. Summary and Outlook

The presented mechanisms and their properties are essential to having a safe or even fair market to participate in. Historically it shows that the right choice of matching mechanisms had a huge impact on society. The unraveling of the medical labor markets in the UK in the 1960s and 70s was fought through a stable market design [4]. The mechanisms offer flexibility, so further constraints and changes can be added. For instance, a variance of the TTC was used in the student-to-school assignment in the US. Also, the centralized matching procedure in the US for matching medical students to hospitals uses a student-proposing DA with small variations to consider matching multiple students to one hospital [1]. Even though the presented mechanisms are powerful solutions and are used widely, there are still several relevant matching problems left. To name a few there is the stable roommates problem, which differs from the stable marriage problem in the fact that all agents belong to the same set and can not be split into "men" and "women". Then there are also $1 : n$ matchings like the college admissions problem, where one college can accept multiple applicants. A more complex problem would be the kidney paired donation, where a kidney recipient enters the market along with a donor and these pairs have to be matched with another compatible pair (e.g. by blood group). While all these problems differ in their statement, the approach to solving those is very similar and the properties which are used are very close.

References

[1] D. Parkes and S. Seuken, *Economics and Computation*. Cambridge University Press, 2018. [Online]. Available: http://economicsandcomputation.org/

[2] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962. [Online]. Available: http://www.jstor.org/stable/2312726

[3] A. Abdulkadiraglu and T. Sonmez, "Random serial dictatorship and the core from random endowments in house allocation problems," *Econometrica*, vol. 66, no. 3, pp. 689–701, 1998. [Online]. Available: http://www.jstor.org/stable/2998580

[4] A. E. Roth, "A natural experiment in the organization of entry-level labor markets: Regional markets for new physicians and surgeons in the united kingdom," *The American Economic Review*, vol. 81, no. 3, pp. 415–440, 1991. [Online]. Available: http://www.jstor.org/stable/2006511