Mechanism Design

7: Matching Mechanisms

Egor Starkov

Københavns Unversitet
Fall 2024

Matching

- We are now completely leaving the realm of "mechanism design" as it is usually defined.
  - (Please keep your arms, hands, legs, feet and head inside the ride vehicle at all times.)
- Matching: huge field, old field, broad field, still somewhat active field. We are merely dipping toes into this vast sea.
- These notes follow Roth and Sotomayor, ch.2 and 4.

This slide deck:

1. Classic matching model
2. Matching with private information about preferences
   - v1
   - v2
   - v3
3. Item allocation as a matching problem
   - Common preferences on one side
   - No preferences on one side
4. Matching and money
5. Dynamic matching

Classic Model: Two-Sided Matching ("Marriage")

- Set $M = \{m_1, m_2, ..., m_n\}$ of men, set $W = \{w_1, w_2, ..., w_p\}$ of women.
- Every man m_i has strict [ordinal] preferences $\succ_{m_i}$ over $W \cup \{m_i\}$; every woman w_j has preferences $\succ_{w_j}$ over $M \cup \{w_j\}$.
  - the latter options in either case mean "staying single".
  - All preferences are commonly known.

Definition

A matching is a one-to-one mapping $\mu : M \cup W \to M \cup W$ such that:

- $\mu(m) \in W \cup \{m\}$ for any $m \in M$ – men are matched to women or stay single;
- $\mu(w) \in M \cup \{w\}$ for any $w \in W$ – women are matched to men or stay single;
- $\mu(\mu(x)) = x$ for any $x \in M \cup W$ – every person is matched to the person who is matched to them.

- A matching μ prescribes who should be married to whom.
- Goal: finding a "good" matching given players' preferences.

Stories

- One of the original motivations for the model was U.S. medical internships market (was a horrible mess in the 1940s-50s).
- Current academic job markets in Econ and related fields also fit the model quite nicely (and are a less horrible mess).
- Another application: allocating heterogeneous workers across projects (think consulting firm) or resources across branches (think large production company).
  - Different projects require for ("prefer") different skills / resources;
  - workers have preferences over projects (e.g., like/dislike certain topics);
  - resources' "preferences" can take form of, e.g., transportation costs to various production plants.
- Special case: when only one side has meaningful preferences, while other side's preferences are either absent, or perfectly aligned. Examples:
  - school admissions;
  - kidney exchange;
  - allocation of items without transfers/payments.

Properties of Matchings

What constitutes a good matching mechanism?

Definition

Matching μ is individually rational if any player $x \in M \cup W$ prefers their match $\mu(x)$ to staying single, i.e., $\mu(x) \succsim_x x$.

Definition

Fix matching μ. A blocking pair is (m, w) such that both $m \in M$ and $w \in W$ prefer each other to their prescribed matches, i.e., $m \succ_w \mu(w)$ and $w \succ_m \mu(m)$.

Definition

Matching μ is stable if it is individually rational and has no blocking pairs.

Stability and IR will be our main requirements.

Stability: example

Example

- Consider the following example.
- Three men $M = (m_1, m_2, m_3)$; three women $W = (w_1, w_2, w_3)$.
- Preferences (types):

$$m_1 : w_2 \succ_{m_1} w_1 \succ_{m_1} w_3 \qquad w_1 : m_1 \succ_{w_1} m_3 \succ_{w_1} m_2$$
$$m_2 : w_1 \succ_{m_2} w_3 \succ_{m_2} w_2 \qquad w_2 : m_3 \succ_{w_2} m_1 \succ_{w_2} m_2$$
$$m_3 : w_1 \succ_{m_3} w_2 \succ_{m_3} w_3 \qquad w_3 : m_1 \succ_{w_3} m_3 \succ_{w_3} m_2$$

  (no one wants to stay single)
- Matching $\mu = ((w_1, m_1), (w_2, m_2), (w_3, m_3))$ is not stable – (m_1, w_2) is a blocking pair ((m_3, w_2) is another).
- Matching $\mu' = ((w_1, m_1), (w_2, m_3), (w_3, m_2))$ is stable.

Stability

- Stability is effectively an equilibrium concept for matching problems.
  - No one wants to "deviate" from the "equilibrium outcome".
  - Although this is not a game – we have not introduced actions/strategies, so "deviations" are not in game-theoretic sense.
- Note that we are not dealing with any private information.
  - The idea is same as in Social Choice: suppose types θ (preferences) of all players are known – what is the "best" outcome (stable/"equilibrium" matching) for this collection of θs?
  - But we'll get to incomplete information eventually.

Stability and existence

Theorem (Gale and Shapley)

A stable matching exists for every marriage market.

- Nice, but specific to the exact model. Breaks down in most extensions, e.g.:
  - **roommate problem/one-sided matching** – group of people need to split in twos to be roommates, have preferences over whom to live with;
  - **three-sided matching** – tanks, damage-dealers, and healers need to split in triples for quest dungeons, have preferences over party-members;
  - **many-to-one matching** – firms and workers; firms have preferences over sets of workers rather than individual employees; need strong conditions on firms' preferences for theorem to hold.
  - **peer effects** – as above, but now workers also have preferences over potential colleagues, rather than the firm itself.
  - See RS ch.2.3 for particular examples of nonexistence.
- Also, we would want to know how this stable matching looks and how to implement it...

How to Find Your Stable Matching

[Men-Proposing] Deferred Acceptance Algorithm

- Consider a dynamic environment with stages $t = 0, 1, \ldots$.
- At stage 0 all men propose to their favorite women.
- If a woman has received one offer, she holds onto it.
- If a woman has receiver more than one offer, she chooses her favorite, holds onto this proposal, and rejects all other men. (Can reject all if staying single is better than all offers.)
- At stage 1 all rejected men propose to their next-favorite women. Men who were not rejected do nothing.
- Women compare all new offers to what they have from the previous stage; pick best, reject the rest.
- The algorithm iterates until no new offers are made. At that stage marriages are finalized. The resulting matching is stable.

Deferred Acceptance: Example

Example

- Preferences:

$$m_1 : w_2 \succ_{m_1} w_1 \succ_{m_1} w_3 \qquad w_1 : m_1 \succ_{w_1} m_3 \succ_{w_1} m_2$$
$$m_2 : w_1 \succ_{m_2} w_3 \succ_{m_2} w_2 \qquad w_2 : m_3 \succ_{w_2} m_1 \succ_{w_2} m_2$$
$$m_3 : w_1 \succ_{m_3} w_2 \succ_{m_3} w_3 \qquad w_3 : m_1 \succ_{w_3} m_3 \succ_{w_3} m_2$$

- At stage 0
  - m_1 proposes to w_2; m_2 and m_3 propose to w_1.
  - w_2 has one offer – keeps it.
  - w_1 has two offers – keeps m_3, rejects m_2.
- At stage 1:
  - m_1 and m_3 have outstanding offers, so do nothing.
  - m_2 proposes to w_3.
  - all women have one offer each – keep all.
- At stage 2:
  - All men have outstanding offers, so no new offers made.
  - Matching is finalized.
- The resulting matching $\mu = ((w_1, m_3), (w_2, m_1), (w_3, m_2))$ is stable.

Deferred Acceptance and Stability

Claim

Mathing μ produced by DA algorithm is stable.

Proof.

- Note that in DA, a man would never propose to a woman he doesn't like, and a woman would always reject a man she doesn't like.
  - So matching produced by DA is individually rational.
- Now suppose there is a blocking pair (m, w) in the resulting matching μ.
  - Man m likes w more than his match $\mu(m)$. According to the algorithm, he must have proposed to w and got rejected.
  - Woman w rejected man m, so she must have had a better offer in hand. Her resulting matching must be better than what she had at that stage, so $\mu(w) \succ m$, a contradiction. $\square$

So Many Stable Matchings

- In the algorithm above men were proposing to women.
- But we can run the algorithm in reverse, with women proposing to men.

So Many Stable Matchings

- In the algorithm above men were proposing to women.
- But we can run the algorithm in reverse, with women proposing to men.
- ...and possibly get another stable matching.
- Can we say anything about the whole set of stable matchings?
- Yes, quite a lot.

Best Matchings Worst Matchings

Definition

- A stable matching μ is M(W)-optimal if every man (woman) likes [their match in] it at least as well as [in] any other stable matching.
- A stable matching μ is M(W)-worst if every man (woman) likes it less than any other stable matching.

Theorem

- *Matching μ_{MDA} produced by men-proposing DA algorithm is M-optimal and W-worst.*
- *Matching μ_{WDA} produced by women-proposing DA algorithm is W-optimal and M-worst.*

Best Matchings Worst Matchings

- The theorem above contains some insights that generalize nicely:
  1. all men agree on which stable matchings are best and worst, same for all women;
  2. men's and women's preferences over stable mathings are opposed.
- There exist other stable matchings, which cannot be obtained through DA algorithm.
- Very surprisingly, they organize in a very nice lattice structure w.r.t. players' preferences...

Lattice Structure of the Set of Stable Matchings

- If there exist stable matchings μ' and μ'' then there also exist stable matchings $\bar{\mu}$ and $\underline{\mu}$ such that for all $m \in M$ and $w \in W$:

$$\bar{\mu}(m) = \max_{\succ_m}\{\mu'(m), \mu''(m)\} \qquad \bar{\mu}(w) = \min_{\succ_w}\{\mu'(w), \mu''(w)\}$$

$$\underline{\mu}(m) = \min_{\succ_m}\{\mu'(m), \mu''(m)\} \qquad \underline{\mu}(w) = \max_{\succ_w}\{\mu'(w), \mu''(w)\}$$

(i.e., in $\bar{\mu}$ every m gets the best match among those he can have in μ' and μ'', and every w gets the worst of the two; vice versa for $\underline{\mu}$).

- See RS ch.2.3 and ch.3 for more details.
- Some corollaries of this lattice structure:
  - All stable matchings are contained "between" μ_{MDA} and μ_{WDA} in terms of players' preferences.
  - If $\mu_{MDA} = \mu_{WDA}$ then this is the unique stable matching.

Structure of the Set of Stable Matchings

- Plenty of other fun results regarding structure of this set, e.g.:
  - Set of singles is the same in all stable matchings.
  - Adding a new man to the market harms all men and helps all women.
  - Iteratively satisfying blocking pairs leads to a stable matching.
  - And many more, see RS.

This slide deck:

This slide deck:

Strategic Aspects

- Let us finally return to mechanism design angle.
- Suppose players' preferences $\succ$ are not known to the designer.
- We have a nice algorithm (DA) which obtains stable matchings and is decentralized by design. A reasonable question then is:

Question 1

In a game induced by DA algorithm, is it optimal for players to play according to their true preferences?

- Stability implies that no one wants to rematch after the outcome is announced. But misrepresenting own preferences may affect which stable matching is selected.

Good News and Bad News

Good News

In a game induced by the Deferred Acceptance algorithm, it is a dominant strategy for players on the proposing side to play truthfully.

Bad News

It is generally not optimal for players on the receiving side to play truthfully.

DA is not DSIC: Example

Example

- Two men, two women
- Preferences (including the options to stay single):

$$m_1 : w_1 \succ_{m_1} w_2 \succ_{m_1} m_1 \qquad w_1 : m_2 \succ_{w_1} m_1 \succ_{w_1} w_1$$

$$m_2 : w_2 \succ_{m_2} w_1 \succ_{m_2} m_2 \qquad w_2 : m_1 \succ_{w_2} m_2 \succ_{w_2} w_2$$

- Two stable matchings:
  - $\mu = ((m_1, w_1), (m_2, w_2))$;
  - $\nu = ((m_1, w_2), (m_2, w_1))$;
- M-DA leads to matching μ.
- However, w_2 can play as if her preferences are $m_1 \succ'_{w_2} w_2 \succ'_{w_2} m_2$, i.e., she could rejects m_2's offer.
- Easy to see that then the algorithm will produce matching ν, preferred by w_2 to μ, so this is a profitable deviation.

Private types: Beyond DA

- So, DA does not work as a mechanism that extracts players' private information.
- We have to go beyond it and ask:

Question 2

Does there exist a DSIC mechanism that leads to a stable matching in a marriage model?

- Answer: NO.
- The previous example is a universal counterexample:
  - If in that problem your mechanism chooses matching μ, then w_2 will always have that same deviation that switches outcome to ν instead.
  - If your mechanism chooses ν, then m_2 will have a similar deviation that switches the outcome to μ.

Private types: going formal

Theorem (Roth)

No stable matching mechanism exists for which stating the true preferences is a dominant strategy for every agent.

Theorem

No stable matching mechanism exists for which stating the true preferences is a best response for every agent if all other agents are reporting truthfully.

The above statements mean that for every mechanism there is some collection of players and their preferences where a deviation exists (that was our example). The below is a stronger statement, that applies to any such collection.

Theorem

Under any mechanism, if more than one stable matching exists in a given marriage problem, then at least one player can profitably misrepresent their preferences.

“Private” Types...?

- Let’s look closer at the logic used in our reasoning.
- Each player knew the consequences of misreporting their preferences!
  - It’s as if all players knew everyone’s preferences! And only the designer is ignorant...
  - But we know that this kind of environment is not a problem for the designer! (Remember cross-verification mechanisms? The only subtlety is that we need a suitable threat for mismatching reports.)
- While DSIC amounts to exactly the above (i’s report must be optimal regardless of everyone else’s reports), BIC allows for more:
  - In BIC mechanism, truthtelling must be optimal for i on average, across all possible preferences of other players (given that they report the truth).
  - A Bayesian player chooses their report before learning others’ preferences.

This slide deck:

Private Types! (Version 2)

- Let us consider the following, more familiar environment.
  - Each player i has private type $\theta_i \in \Theta_i$;
  - all players report their types to a direct mechanism Γ;
  - Γ prescribes a matching $\mu(\theta)$;
  - we want $\mu(\theta)$ to be stable.
  - (Previously we implicitly had that θ_i was known by all players, but not by designer.)
- Now when player chooses what type $\hat{\omega}_i$ to report, they no longer know for sure the consequences of their deviation.
- To justify stability as a requirement, assume that all types are revealed after the mechanism is run. Then everyone learns everyone's realized preferences (types), and blocking pairs can be formed. We want no blocking pairs to exist.

Impossibility (version 2)

Theorem (Roth, RS Th.4.23)

There is no mechanism such that [at least one of] its equilibrium outcome is stable for all realizations of players' types.

- Require at least two players on each side of the market.
- Proof uses an example similar to what we used to show DA is not DSIC.

Example (exact same example as before)

- Two men, two women
- The most likely (modal) preference profile:

$$m_1 : w_1 \succ_{m_1} w_2 \succ_{m_1} m_1 \qquad w_1 : m_2 \succ_{w_1} m_1 \succ_{w_1} w_1$$

$$m_2 : w_2 \succ_{m_2} w_1 \succ_{m_2} m_2 \qquad w_2 : m_1 \succ_{w_2} m_2 \succ_{w_2} w_2$$

- With small probability, w_2 can also have type with $m_1 \succ_{w_2} w_2 \succ_{w_2} m_2$; similar for m_2; m_1 and w_1 have no other types.
- Two stable matchings under modal preferences:
  - $\mu = ((m_1, w_1), (m_2, w_2))$;
  - $\nu = ((m_1, w_2), (m_2, w_1))$;
- Suppose your mechanism leads to matching μ under modal preferences.
- Consider w_2's deviation to report her alternative type $(m_1 \succ'_{w_2} w_2 \succ'_{w_2} m_2)$.
- This deviation will likely (for w_2) produce matching ν, preferred by w_2 to μ, so this is a profitable deviation.

Impossibility (version 2)

- Issue: stability defined as "stability w.r.t. true preferences", not w.r.t. reported types – i.e., need types to be revealed after the mechanism is run.

  If we only announce the final matching or the reports, the result can break.

This slide deck:

Stability and Private Types (version 3)

- The last remark in prev.slide makes you wonder: is stability an appropriate concept under incomplete information?
- It doesn't make sense to assume that true types are revealed. Reports maybe, but that is not the same.
- *Sometimes* nothing stops a player from trying to form a blocking pair even if they do not know whether the potential "partner in crime" would want this (i.e., even if types not announced).
  - If this is a threat, we do actually need stability w.r.t. true preferences.
  - But why do we even need a mechanism in this case? Just let the agents match on their own.
- **But** if there are costs to such proposals and nothing is announced ex post except for the resulting matching μ, then some weaker form of stability may suffice.
  - Idea: even if m and w prefer each other to their prescribed matches $\mu(m)$ and $\mu(w)$, they do not know that they prefer each other – and so do not form a blocking pair.

Stability and Private Types (version 3)

- There exists a notion of stability that incorporates players' incomplete knowledge of each other's preferences when trying to form blocking pairs...
  - see Liu [2020]
- but there are much fewer results as to whether & how these "stable" matchings can be obtained in a mechanism, what their properties are, etc

Private Types: Summary

- If all players know all preferences but the designer knows none:
  - in DA some players on the receiving side want to misreport, so DA is not DSIC;
  - more generally, there cannot exist any DSIC stable mechanisms.
- If players only know own preferences ex interim, but learn others' preferences ex post:
  - result for DSIC stable mechanisms carries over from the previous case;
  - there is no BIC stable mechanism either.
- If players only know own preferences ex interim and only learn the matching μ ex post:
  - if the cost of trying to make a [blocking] pair is small then the results from the previous case apply;
  - otherwise the standard notion of stability may be too strong and relaxing it may allow us to come up with an IC slightly-less-stable mechanism, but we don't really know.

This slide deck:

1. Classic matching model
2. Matching with private information about preferences
   - v1
   - v2
   - v3
3. Item allocation as a matching problem
   - Common preferences on one side
   - No preferences on one side
4. Matching and money
5. Dynamic matching

This slide deck:

Item Allocation

- Consider now a special case of the marriage problem.
- There are a set N of players and set X of objects.
  - Player $i \in N$ has preferences $\succ_i$ over X;
  - Objects have common preferences $\succ_X$ over players N.
- The designer knows $\succ_X$, does not know $\succ_i$.
- The designer wants to allocate objects to players (at most one item per player) so as to create a stable matching.

Common preferences

- Here $\succ_X$ can be seen as the designer's preference over players.
  - Players with higher priority get to choose first.
- Alternative interpretation:
  - N are slots available in schools/universities. Universities usually have common preferences over school graduates, given by exam scores.

Deferred Acceptance

- We already know an algorithm that solves this problem: Deferred Acceptance (N-DA).
  - Let players pick objects in the order of their (players') priority $\succ_X$. First the $\succ_X$-best player i picks his most preferred $x \in X$, then the second player chooses among the remaining $x \in X$ etc.
  - Can omit the "holding on to an offer" part – later offers are never preferred by $\succ_X$ to early offers.
  - We know it is a dominant strategy for players to pick truthfully.
- But we also know that the resulting matching is X-worst among stable. If $\succ_X$ is the designer's preference then it is also the worst for the designer. Can do better?
  - According to the general results from before, X-DA would be better, but it is not IC for players N.

Uniqueness

Theorem

If one side of the marriage market has common preferences then the stable matching is unique.

- X-DA yields the same outcome as N-DA: first all objects send their offers to $\succ_X$-best player, who chooses his favorite, then all remaining objects send offers to $\succ_X$-second player etc
- So the lattice structure of the set of stable matchings implies that the stable matching produced by X-DA / N-DA is the unique stable matching.
- So nothing can beat N-DA in this problem.

This slide deck:

1. Classic matching model
2. Matching with private information about preferences
   - v1
   - v2
   - v3
3. Item allocation as a matching problem
   - Common preferences on one side
   - No preferences on one side
4. Matching and money
5. Dynamic matching

Fair Allocations

- What if we do not have any priority list $\succ_X$ and instead want to treat all players $i \in N$ equally?
  - Note: this takes us beyond the standard marriage model, where many results required that all preferences on both sides are strict.
- Substitute stability with Pareto-Optimality:
  - Mathing μ is Pareto-Optimal if no set of players can trade among themselves to [weakly] improve all of their utilities.
  - Related to stability; is a weaker form of efficiency (as we defined it in mechanism design).

Fair Allocations: Shapley-Scarf model

- There are a set N of players and set X of objects.
  - Player i has preferences $\succ_i$ over X;
  - Objects have no preferences over whom to be allocated to.
- The designer does not know $\succ_i$.
- The designer wants to allocate objects to players (at most one object per player) so as to create a Pareto-optimal matching.
- There is some initial matching μ_0.

TTC Algorithm

The algorithm we can use here: Top Trading Cycles.

- In marriage model, we were looking for a stable matching – one with no blocking pairs. One way to get it was to start from a random matching and iteratively resolve blocking pairs.
- Same idea: start from the initial allocation and let players trade.

TTC Algorithm

Top Trading Cycles algorithm

1. Begin with matching μ_0 (if there's no initial matching - pick μ_0 at random).
2. Pick any player i_0. Ask them to point to person i_1 who currently holds i_0's favorite object. Analogously, ask i_1 to point to i_2 etc. When some i_N points to i_k ($0 \leq k < N$), the cycle closes. Conduct the trades: i_k gives their object to i_N, who gives their object to i_{N-1} etc.
3. Remove players $\{i_k, ..., i_N\}$ and their objects from the game. Start the next cycle with the remaining players and objects.
4. Any player is allowed to point to themselves if they already have the most desired object. This would be a one-player cycle.

TTC Algorithm

Theorem (Roth [1982])

It is a dominant strategy for all players to follow their true preferences in TTC.

Theorem

TTC algorithm is the only mechanism in Shapley-Scarf model which is:

1. *incentive compatible;*
2. *Pareto-optimal;*
3. *individually rational relative to μ_0.*

Item Allocation in Matching: Conclusion

- TTC algorithm is widely used in the real world.
  - Has been applied to school choice and kidney exchange.
- Rule of thumb:
  - if you have a marriage market and both sides have preferences, use DA (not ideal but as good as it gets);
  - if only one side of the market has meaningful preferences (and you are actively unwilling to assume some preferences on the other side), use TTC.

This slide deck:

Matching and Money

- The models we had so far (marriage, object allocation) were framed without transfers/money.
- Many people think that this non-monetary nature is one of the characteristic properties of matching models.
- Few know that you can actually do matching with money.
- One example follows (using RS ch.6.2).
- See RS ch.7-9 for more.

Kelso and Crawford [1982]

- Set of firms $F = \{f_1, ..., f_n\}$; set of workers $W = \{w_1, ..., w_m\}$.
- Many-to-one: each firm can hire any number of workers; each worker can only work for one firm.
  - This is for simplicity; firms can have limited number of spots in principle.
- Worker i has disutility σ_{ij} from working for firm j – i.e., accepts this job only if offered wage of at least σ_{ij}.
  - Worker i's utility from working for j is quasilinear: $u_i(j, s_{ij}) = s_{ij} - \sigma_{ij}$, where s_{ij} is the wage paid by firm j to the worker.
- Firm j receives profit y_{ij} from hiring worker i.
  - Firm's profit function is additively separable: profit from hiring set $C \subseteq W$ of workers at respective wages $\{s_{ij}\}_{i \in C}$ is $\pi(C, s) = \sum_{i \in C}(y_{ij} - s_{ij})$.
  - This is restrictive. We can look at slightly more general profit functions, but for stable matchings to exist we need profit functions to satisfy certain (restrictive) assumptions.
- Assume further that money/wages are integer (rather than real) – i.e., there is no unit smaller than 1DKK.

KC '82: Modified DA algorithm

A stable matching can be obtained using a modified DA algorithm which proceeds as follows.

1. Let $s_{ij}(0) = \sigma_{ij}$ be the initial wage offers.
2. At every stage t every firm j makes offer $s_{ij}(t)$ to every worker i such that $y_{ij} > s_{ij}(t)$
3. Every i holds on to the best offer, rejects the rest.
4. If j's offer to i was rejected, set $s_{ij}(t+1) = s_{ij}(t) + 1$.
5. (Otherwise, set $s_{ij}(t+1) = s_{ij}(t)$.)
6. Iterate the algorithm until no new offers are made.

KC '82

- This modified DA algorithm yields a stable matching.
  - Offers from j to i dry out when firm j can no longer afford worker i – i.e., after i has a better offer from someone else – i.e., when hiring i is no longer IR for j.
  - No blocking pairs by the usual Gale-Shapley argument.
  - Note also that in the end, every i is matched to $j^*(i) = \arg\max_j(y_{ij} - \sigma_{ij})$ – i.e., matching is efficient (surplus-maximizing).

KC '82

- This modified DA algorithm yields a stable matching.
  - Offers from j to i dry out when firm j can no longer afford worker i – i.e., after i has a better offer from someone else – i.e., when hiring i is no longer IR for j.
  - No blocking pairs by the usual Gale-Shapley argument.
  - Note also that in the end, every i is matched to $j^*(i) = \arg\max_j (y_{ij} - \sigma_{ij})$ – i.e., matching is efficient (surplus-maximizing).
- Apply this model to item allocation (where now any player can get any number of items):
  - let y_{ij} be each player j's valuation of item i;
  - let $\sigma_{ij} = 0$ for all i, j.
  - Then the algorithm above is an English (ascending-price) auction
  - which, in turn, is equivalent to the second-price auction (VCG mechanism).
- See RS 6.2 for a more general treatment of this model and RS 7-9 for models with money as a continuous variable (use very different tools to obtain similar results).

This slide deck:

1. Classic matching model
2. Matching with private information about preferences
   - v1
   - v2
   - v3
3. Item allocation as a matching problem
   - Common preferences on one side
   - No preferences on one side
4. Matching and money
5. Dynamic matching

Dynamic matching

- Modern literature has shifted a bit towards dynamic matching settings.
  - E.g., buyers and sellers arrive at the market over time and need to be matched with each other.
- Main trade-off is between making good matches and making fast matches.
- See, e.g., Baccara, Lee, and Yariv [2020] as one model of dynamic matching.

References I

M. Baccara, S. Lee, and L. Yariv. Optimal dynamic matching. *Theoretical Economics*, 15(3): 1221–1278, 2020. ISSN 1933-6837. doi: 10.3982/TE3740. URL https://econtheory.org/ojs/index.php/te/article/view/3740.

A. S. Kelso and V. P. Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica: Journal of the Econometric Society*, pages 1483–1504, 1982. Publisher: JSTOR.

Q. Liu. Stability and Bayesian Consistency in Two-Sided Markets. *American Economic Review*, 110(8):2625–66, Aug. 2020. ISSN 0002-8282. doi: 10.1257/aer.20181186. URL https://www.aeaweb.org/articles/pdf/doi/10.1257/aer.20181186?etoc=1&perm_aeawebcookie=d7zoVbd4a6yjaZYaU5gOJeVoiaiUwgdG4uA&PHPSESSID=1c6026d29c5110f253ae0a85b7a999fa.

References II

A. E. Roth. Incentive compatibility in a market with indivisible goods. *Economics Letters*, 9(2): 127–132, Jan. 1982. ISSN 0165-1765. doi: 10.1016/0165-1765(82)90003-9. URL `https://www.sciencedirect.com/science/article/pii/0165176582900039`.