Contents

Introduction

BEEM147: Topics in Microeconomic Theory II

Matching and Market Design

Spring Term 2021

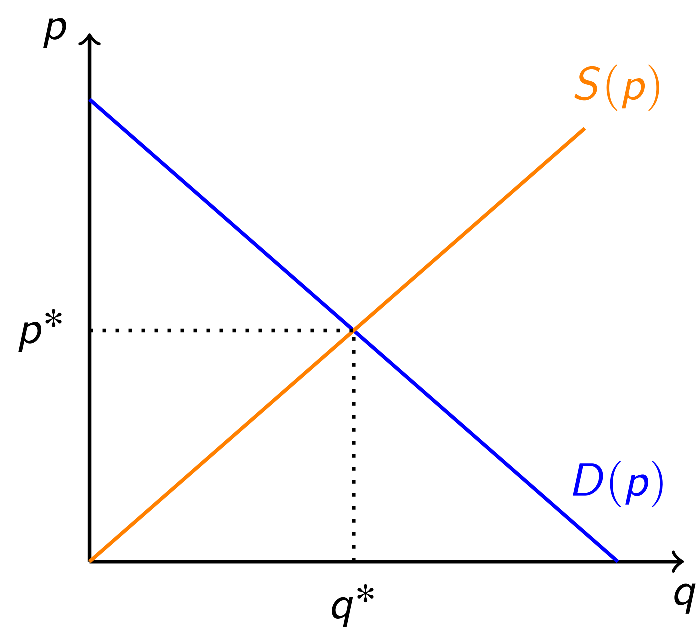
University of Exeter

Overview

- Matching and Market Design
- Syllabus
- Assessment
- Presentation topics

Standard Microeconomics

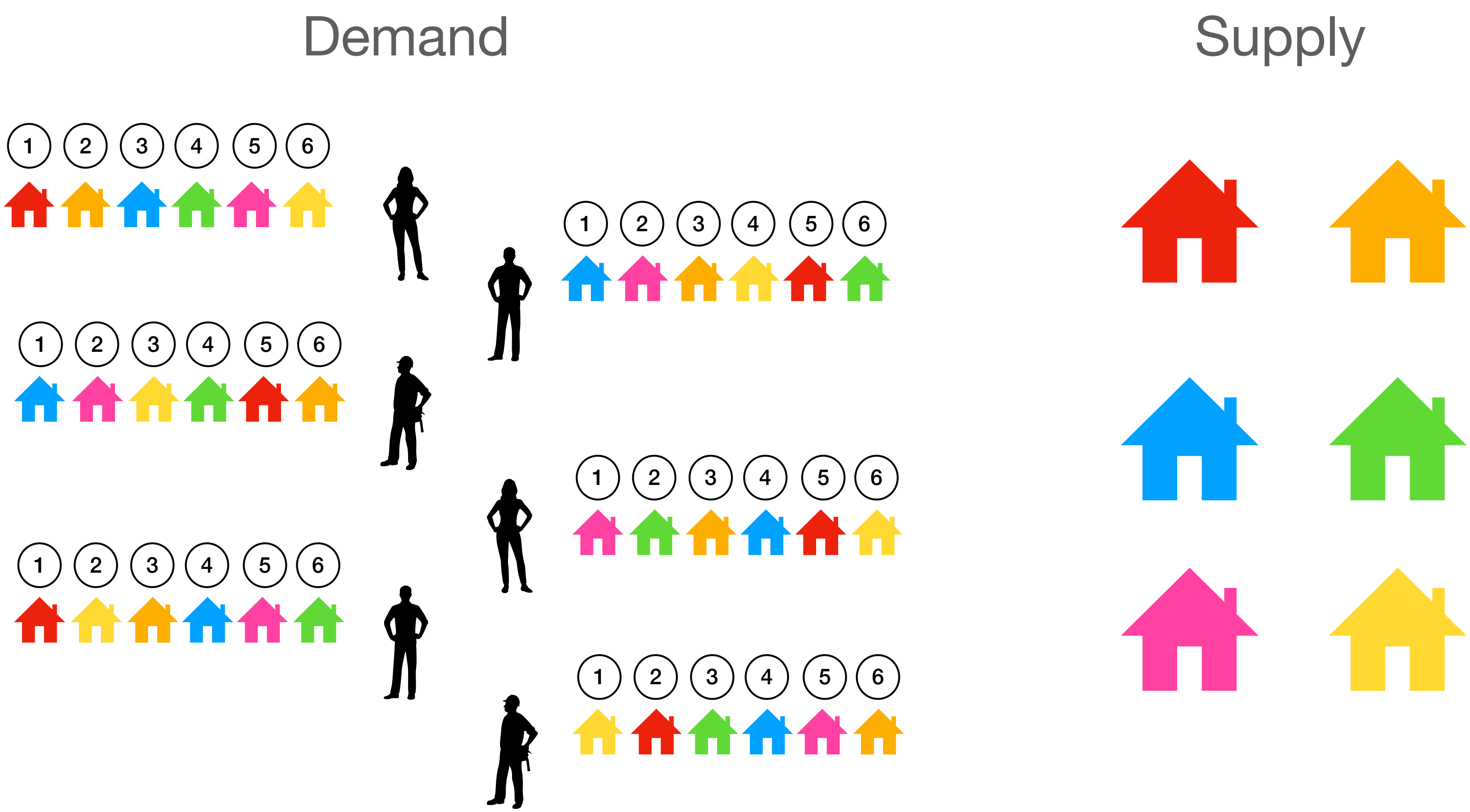
- Commodity markets
  - Unique (or few) homogeneous good(s)
  - Goods are infinitely divisible
  - Many buyers (usually homogeneous)
- Assignment mechanism: price system
  - Market price determines *"who gets what"*
- Focus on:
  - What determines demand and supply?
  - Comparative statics
  - Market structure, etc.



Matching

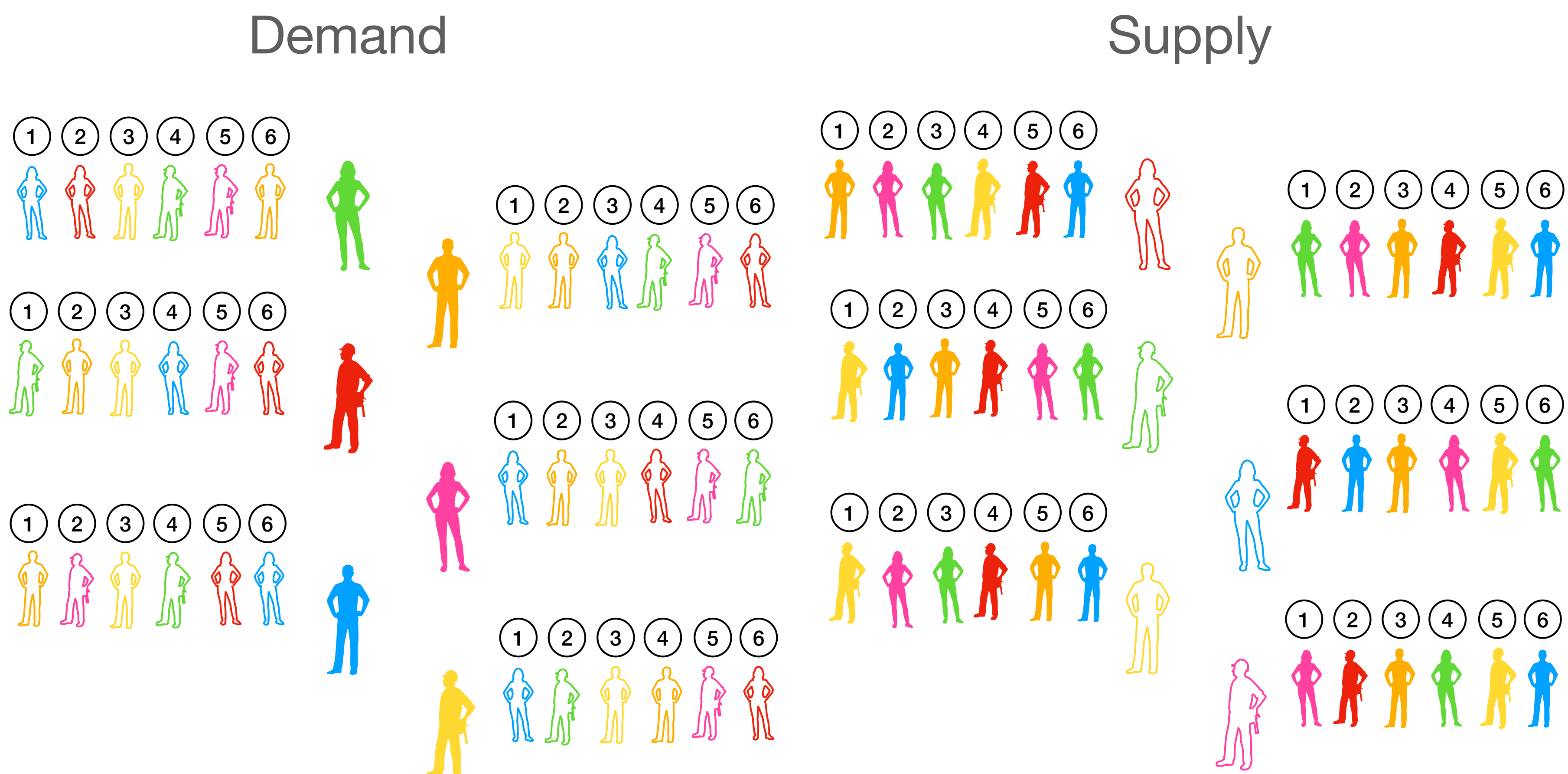
- High levels of heterogeneity
- Heterogeneous preferences over non-homogeneous goods
  - Preferences may be over goods or over who is selling the good
- One-sided: only buyers have preferences
- Two-sided: sellers also have preferences for buyers
- Focus on non-price mechanisms
  - Prices may not be used in these markets because of legal constraints (e.g., illegal to buy and sell human organs, or to pay for publicly provided services, such as public schooling)
  - In small markets, it may be hard to impose "equilibrium conditions" to find prices

Matching

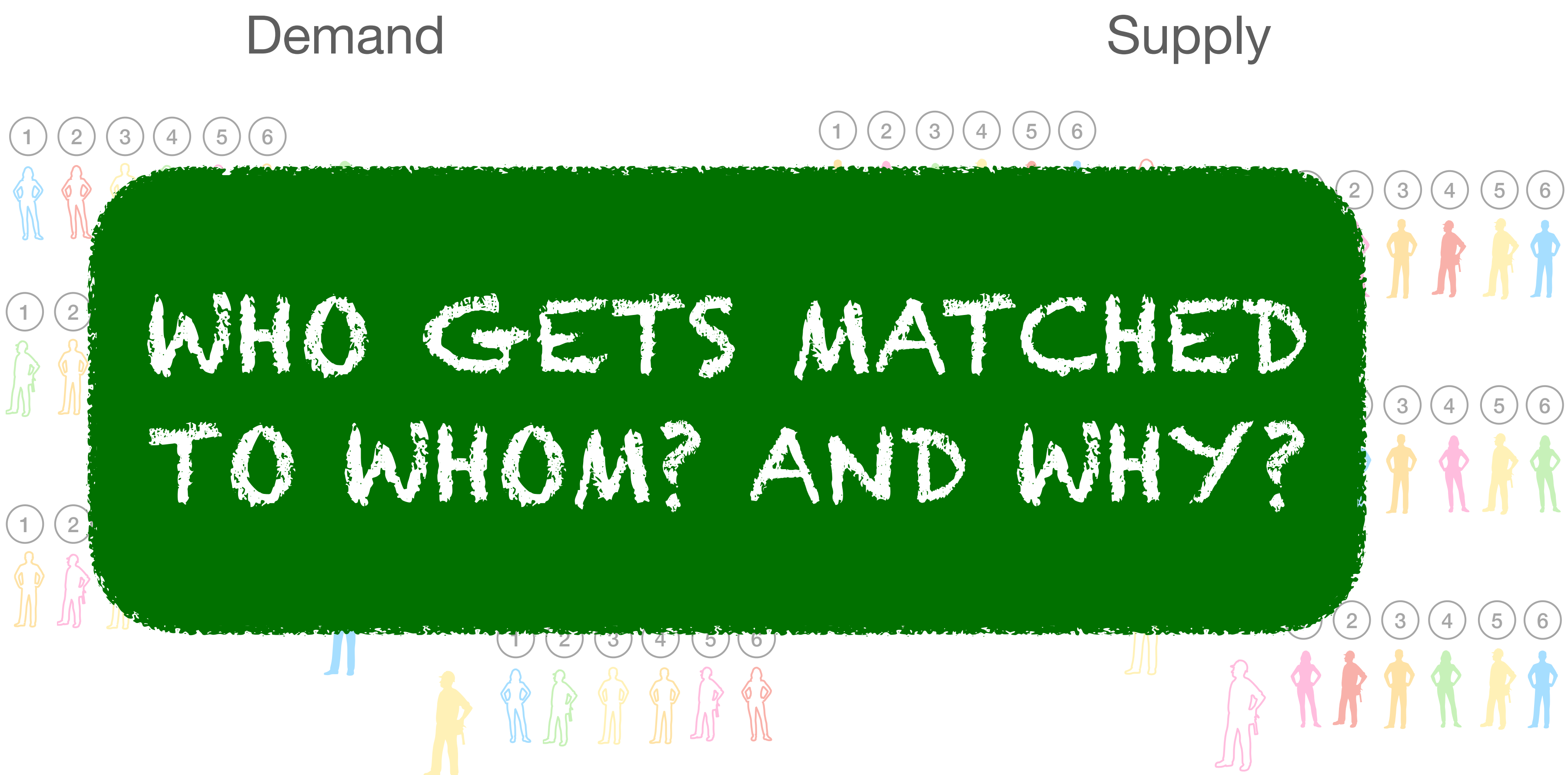


Matching

Matching



Matching



WHO GETS MATCHED TO WHOM? AND WHY?

Market design

- Market design focuses broadly on the application of a wide arrange of tools to study specific markets with the above characteristics
- Most famous examples:
  - School choice
  - Kidney exchange
  - Medical residency programs
- This course focuses on the theoretical aspects of market design (i.e., matching models)
  - Auctions are also considered a part of market design, but we will not cover auctions in this course.
- Active research areas also use other tools: structural econometrics, experiments, causal inference, computer science, policy design and evaluation, etc.

Syllabus

- Week 1: House allocation
- Week 2: Housing market
- Week 3: Kidney exchange
- Week 4: Random allocations
- Week 5: Marriage market
- Week 6: The medical match
- Week 7: School choice
- Week 8: Course allocation
- Week 9: Student presentations 1
- Week 10: Student presentations 2
- Week 11: Final take-home exam

Assessment

- Evaluation
  - 55%: final take-home exam
  - 25%: presentation(s)
  - 20%: two problem-sets
- The material for the module consists of prerecorded lectures and lecture notes.
  - The videos and notes for week t will be distributed on Monday of week $t-1$.
- Students are responsible to cover the material on their own before class
- Live sessions will be held online through Zoom (time TBD)
  - They will be dedicated to Q&A, discussion, and solving problems

Presentation topics

- Ideally, each student will give two presentations
- But this will depend on the number of students and the speed with which we cover the material
- The topic of the presentation(s) will be chosen by students upon consultation with the instructor
- Some recommended topics for presentations:
  - Fair division
  - Voting rules and social choice
  - Dynamic matching
  - Empirical matching models
  - Matching with network externalities
  - Global kidney exchange
  - Antitrust lawsuit against the medical match (NRMP)

Week 1: House Allocation

BEEM147: Topics in Microeconomic Theory II

Matching and Market Design

Spring Term 2021

University of Exeter

Overview

- Binary relations
- Preference relations
- House allocation problems
- Review of mechanism design
- Serial Dictatorship

Binary relations

- **Binary relation** R over set X is a subset of $X \times X$
- $(x, x') \in R \Rightarrow$ write xRx'
- $(x, x') \notin R \Rightarrow$ write not xRx'
- Example: "less than or equal" $\leqslant$ on $\mathbb{R}$
- Binary relation R on X is:
  1. **complete** if for every $x, x' \in X$, xRx', $x'Rx$, or both
  2. **transitive** if for every $x, x', x'' \in X$, xRx' and $x'Rx''$ imply xRx''
  3. **antisymmetric** if for every $x, x' \in X$, xRx' and $x'Rx$ imply $x = x'$
- Example: $\leqslant$ on $\mathbb{R}$ is complete, transitive and antisymmetric. $<$ on $\mathbb{R}$ is only transitive, not complete or antisymmetric.

Preferences

- I = finite set of n **agents**
- H = finite set of **houses**
- A **preference relation** $\succcurlyeq$ is a complete and transitive binary relation over H
- $\succcurlyeq_i$ = preference relation of $i \in I$ over H
  - $h \succcurlyeq_i h'$ means i *prefers house* h *at least as much as* h'
  - $h \succ_i h' \Leftrightarrow h \succcurlyeq_i h'$ and not $h' \succcurlyeq_i h$
  - $h \sim_i h' \Leftrightarrow h \succcurlyeq_i h'$ and $h' \succcurlyeq_i h$

Preferences

- "Complete" preferences: every agent thinks one of the following for every pair of houses h and h'
    1. *I like house h more than house* $h' \iff h \succ h'$
    2. *I like house h less than house* $h' \iff h' \succ h$
    3. *I am indifferent between houses h and* $h' \iff h \sim h'$
- "Transitive" preferences rule out inconsistencies:
    - Liking h_1 more than h_2, h_2 more than h_3, and h_3 more than h_1!
- $\succcurlyeq_i$ antisymmetric $\Rightarrow$ **strict preference relation** (aka. **linear order**)
- $\mathcal{P}(H)$ = set of linear orders over H
- $(\succcurlyeq_i)_{i \in I}$ = **preference profile**, also written shortly as $(\succcurlyeq_i)$
- $\mathcal{P}(H)^n$ = set of all strict preference profiles, i.e., $(\succcurlyeq_i) \in \mathcal{P}(H)^n$

House allocation problems

- $(I, H, (\succcurlyeq_i)_{i \in I}) =$ **house allocation problem**
- Assume $\succcurlyeq_i \in \mathcal{P}(H)$ for every $i \in I$
- A **matching** is a function $\mu : I \cup H \to I \cup H \cup \{\varnothing\}$ such that:
  1. $\mu(i) \in H \cup \{\varnothing\}$
  2. $\mu(h) \in I \cup \{\varnothing\}$
  3. $\mu(i) = h$ if and only if $\mu(h) = i$
- $\mathcal{M}(I, H) =$ set of all matchings
- A matching μ is **Pareto efficient** if there is no other matching $\nu \in \mathcal{M}(I, H)$ such that $\nu(i) \succcurlyeq_i \mu(i)$ for all $i \in I$ and $\nu(j) \succ \mu(j)$ for at least one $j \in I$.

House allocation problems

Example

Let $I = \{1, 2, 3, 4\}$ *and* $H = \{a, b, c, d\}$*. The preferences are given by:*

$$\succ_1: b, c, d, a; \quad \succ_2: a, b, c, d; \quad \succ_3: a, c, d, b; \quad \succ_4: a, d, b, c.$$

That is to say, agent 1 prefers house b *over all the houses, followed by house* c*, house* d*, etc.*

Consider the matching μ *given by:*

$$\mu(1) = d, \quad \mu(2) = a, \quad \mu(3) = c, \quad \mu(4) = b.$$

Is matching μ *Pareto efficient?*

Review of mechanism design

- Given the *Revelation Principle*, we restrict attention to direct mechanisms
- A **matching mechanism** is a function $\phi : \mathcal{P}(H)^n \to \mathcal{M}(H, I)$ mapping preference profiles into matchings
- It is **Pareto efficient** if the matching $\phi[(\succcurlyeq_i)]$ is Pareto efficient for every profile $(\succcurlyeq_i) \in \mathcal{P}(H)^n$
- It is **strategy proof** if, for every $(\succcurlyeq_i) \in \mathcal{P}(H)^n$ and every agent $i \in I$,

$$\phi[(\succcurlyeq_i, \succcurlyeq_{-i})](i) \succcurlyeq_i \phi[(\succcurlyeq', \succcurlyeq_{-i})](i)$$

for every $\succcurlyeq' \in \mathcal{P}(H)$

Serial Dictatorship

- $\pi : \{1, \ldots, n\} \rightarrow I$ = **priority order**
- Serial Dictatorship algorithm
  - Given $\succcurlyeq_{\pi(1)}$, assign $\pi(1)$ to their top choice in H
  - Given $\succcurlyeq_{\pi(2)}$, assign $\pi(2)$ to their top choice among the remaining houses: H minus $\mu(\pi(1))$

    ...
  - Given $\succcurlyeq_{\pi(k)}$, assign $\pi(k)$ to their top choice among the remaining houses ...
- Advantages: it is Pareto efficient and strategy proof
- Disadvantages: it is fair inasmuch as π is, usually randomly determined, but not clear how it affects the properties of the mechanism

Serial Dictatorship

Example

Let $I = \{1, 2, 3, 4\}$ *and* $H = \{a, b, c, d\}$. *The preferences are given by:*

$$\succ_1: b, c, d, a; \quad \succ_2: a, b, c, d; \quad \succ_3: a, c, d, b; \quad \succ_4: a, d, b, c.$$

Recall: $\mu(1) = d$, $\mu(2) = a$, $\mu(3) = c$, $\mu(4) = b$, *not PE since 1 and 4 swap. Is resulting matching PE?*

Yes, since it is the outcome of SD with $\pi = 1, 2, 3, 4$:

1. *Agent 1* $\rightarrow$ *b.*
2. *Agent 2* $\rightarrow$ *a.*
3. *Agent 3* $\rightarrow$ *c (since a no longer available).*
4. *Agent 4* $\rightarrow$ *d (since a no longer available).*

Serial Dictatorship

- **Theorem:** The Serial Dictatorship mechanism is Pareto efficient.
- **Proof:**
  - Suppose not: $\exists\ \nu \in \mathcal{M}(I, H)$ s.th. $\nu(i) \succcurlyeq_i \mu(i)$ for all $i \in I$ and $\nu(j) \succ_j \mu(j)$ for some $j \in I$
  - Wlog, assume $\pi(k) = k$ to simplify notation
  - Since $k = 1$ gets their top choice in μ, $\nu(1) = \pi(1)$
  - For $k = 2$, note that $\nu(2)$ must also be equal to $\mu(2)$, o/w it'd need to be $\mu(1)$, which cannot be since $\mu(1) = \nu(1)$
  - By induction, $\nu(k) = \mu(k)$ for every $k \geqslant 2$, hence, contradiction!

Serial Dictatorship

- **Theorem:** For every Pareto efficient matching μ, there exists a priority order π such that μ is the outcome of SD under π.
- **Note:** SD characterizes the set of Pareto efficient matchings
- **Proof:**
  - First, show that at least one agent must be getting their top choice under μ
  - Suppose not, then have everyone point to their top choice. And every house point to their owner under μ. There must by a cycle (no one points to their own house, but every chain comes to an end since there are finite houses and agents)
  - Have agents in a cycle exchange houses. New matching will Pareto dominate μ, which cannot be!
  - Assign those to pointing to their top choices to the highest priorities, and remove them along their houses from the market
  - Continue by induction . . .

Serial Dictatorship

- **Theorem:** The Serial Dictatorship mechanism is strategy proof.
- **Proof:**
  - The set of houses from which each agent chooses does not depend on their own preferences (it depends on folks with higher priorities!).
  - If they report other preferences, they will get the same set to choose from.
  - Hence, the best they can do is to report their true preferences. No incentives to lie.

Week 2: Housing Market

BEEM147: Topics in Microeconomic Theory II

Matching and Market Design

Spring Term 2021

University of Exeter

Overview

Last week

- House allocation problems
- Serial Dictatorship algorithm

This week

- Housing markets
- Individual rationality and the core
- Top-Trading Cycle (TTC) algorithm
- House allocation with existing tenants

Housing markets

- Same as house-allocation problem: $(I, H, (\succcurlyeq_i)_{i \in I})$, but **agents initially own a house**
  - Agent i owns house h_i
  - Aim is to study mechanisms that respect **property rights**
- Key takeaways:
  - Pareto efficiency **not** a sufficient criterion
  - Serial Dictatorship **not** fully desirable
  - Need new tools!

Housing markets

Example (1)

$I = \{1,2,3,4\}$ and $H = \{a,b,c,d\}$, with preferences (house initially owned underlined):

$$\succ_1: b, c, d, \underline{a}; \quad \succ_2: a, \underline{b}, c, d; \quad \succ_3: a, \underline{c}, d, b; \quad \succ_4: a, \underline{d}, b, c.$$

- *Initial allocation not Pareto efficient.*
- *Agents 1 and 2 would want to trade. Resulting allocation is PE.*
- *Alternative: run SD. But not always sensible. Say $\pi = \{3,1,2,4\}$. Resulting matching:*

$$\mu(1) = b, \quad \mu(2) = c, \quad \mu(3) = a, \quad \mu(4) = d.$$

- *μ is PE, but agent 2 has no incentives to participate in this mechanism: $h_2 = b \succ_2 c$.*

Housing markets

Example (2)

$I = \{1, 2, 3\}$ and $H = \{a, b, c\}$, with preferences (house initially owned underlined):

$$\succ_1: b, c, \underline{a}; \quad \succ_2: a, \underline{b}, c; \quad \succ_3: a, b, \underline{c}.$$

Consider the matching :

$$\mu(1) = c, \quad \mu(2) = b, \quad \mu(3) = a.$$

- *μ is PE: SD with $\pi = \{3, 2, 1\}$*
- *μ gives everyone a house at least as preferred as the one they initially own*
- *However, agents 1 and 2 would rather trade amongst themselves: $b \succ_1 c$ and $a \succ_2 b$.*

Individual rationality and the core

- Matching μ is **individually rational (IR)** if $\mu(i) \succcurlyeq_i h_i$ for every $i \in I$
- A mechanism is IR if it always generates IR matchings
- IR captures notion of property rights
- Agents do not participate **voluntarily** in mechanisms that are not IR

Individual rationality and the core

- However, IR not enough to guarantee that **groups of agents** wish to participate: Example (2)
- Matching μ is **blocked** by **coalition** $A \subseteq I$ if $\exists\ \nu \in \mathcal{M}(I, H)$ s.th.
  1. for all $a \in A$, $\nu(a)$ is initially owned by someone in A;
  2. $\nu(a) \succcurlyeq_a \mu(a)$ for all $a \in A$, and $\nu(a) \succ_a \mu(a)$ for some $a \in A$.
- Matching is in the **core** if it is not blocked by any coalition
- **Note:** the notion of the core is appealing when thinking on decentralized exchange
- Key questions:
  - Is the core none-empty?
  - Would agents "converge" to a matching by trading indefinitely amongst themselves?
  - How do we find matchings in the core?

Individual rationality and the core

- **Proposition.** Every matching in the core is individually rational and Pareto efficient.
- **Proof:**
  - IR: If not IR $\Longrightarrow$ coalition $\{i\}$ blocks
  - PE: If not PE $\Longrightarrow$ coalition I blocks
- **Note:** Example (2) shows that converse is not true: being in the core is stronger than IR + PE

Top-Trading Cycle

- Top-Trading Cycle Algorithm (TTC)
  - Start: all houses are available and all agents are unmatched
  - **Agents point to their favorite available house, houses point to their owner**
  - Find all **cycles**:

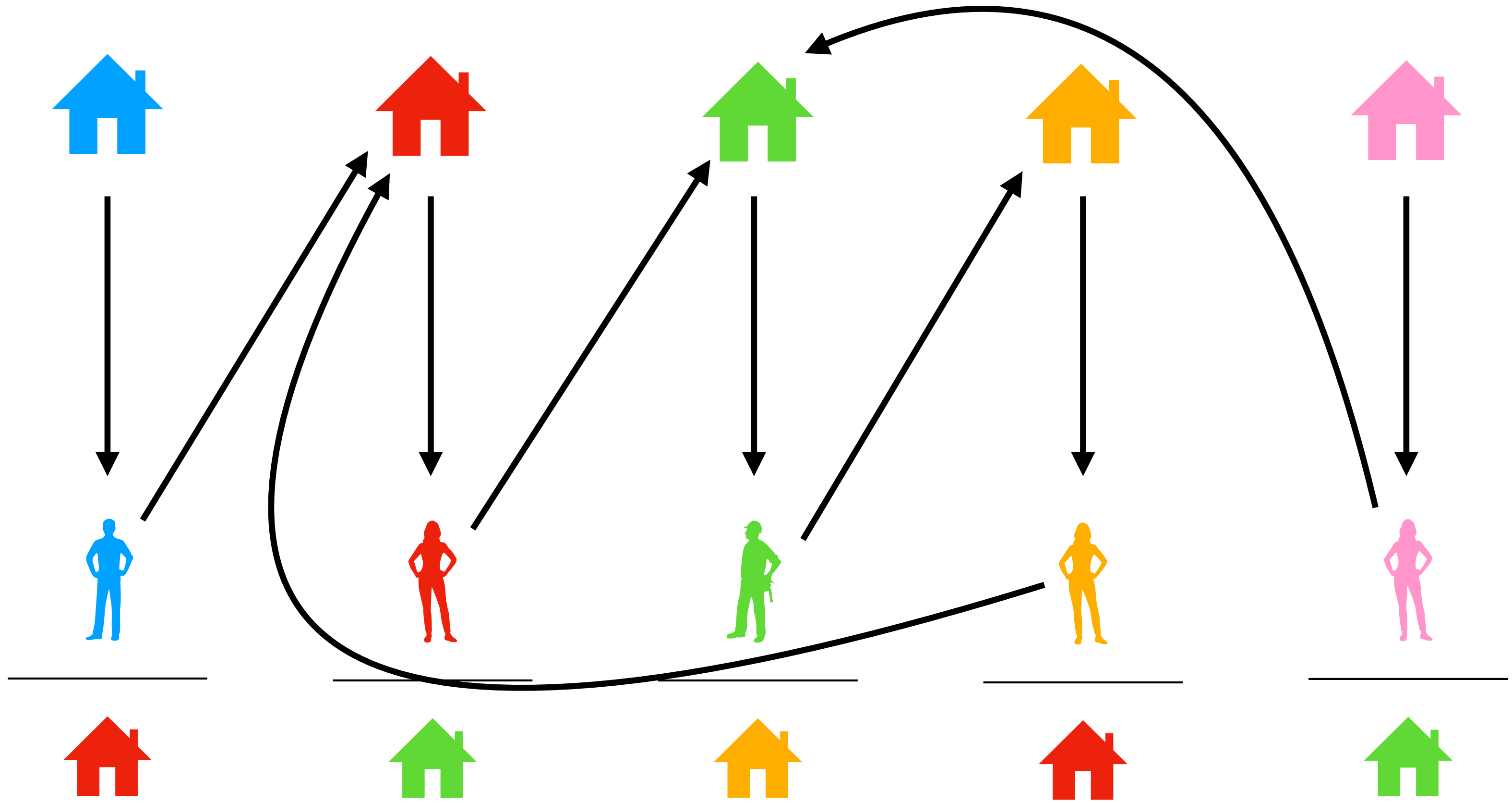
$$h_1 \to i_1 \to h_2 \to i_2 \to \cdots \to h_K \to i_K \to h_1$$

    where h_{k+1} = favorite house of i_k
  - **Assign all agents to houses in a cycle**, repeat until no more houses or agents.
- Advantages: TTC characterizes the core of a housing market, which has a unique matching. And it is the unique strategy-proof, IR and PE mechanism.
- Disadvantages: not very intuitive, people often do not understand it
- **Note**: you should prove that the mechanism is well defined: does not get "stuck"

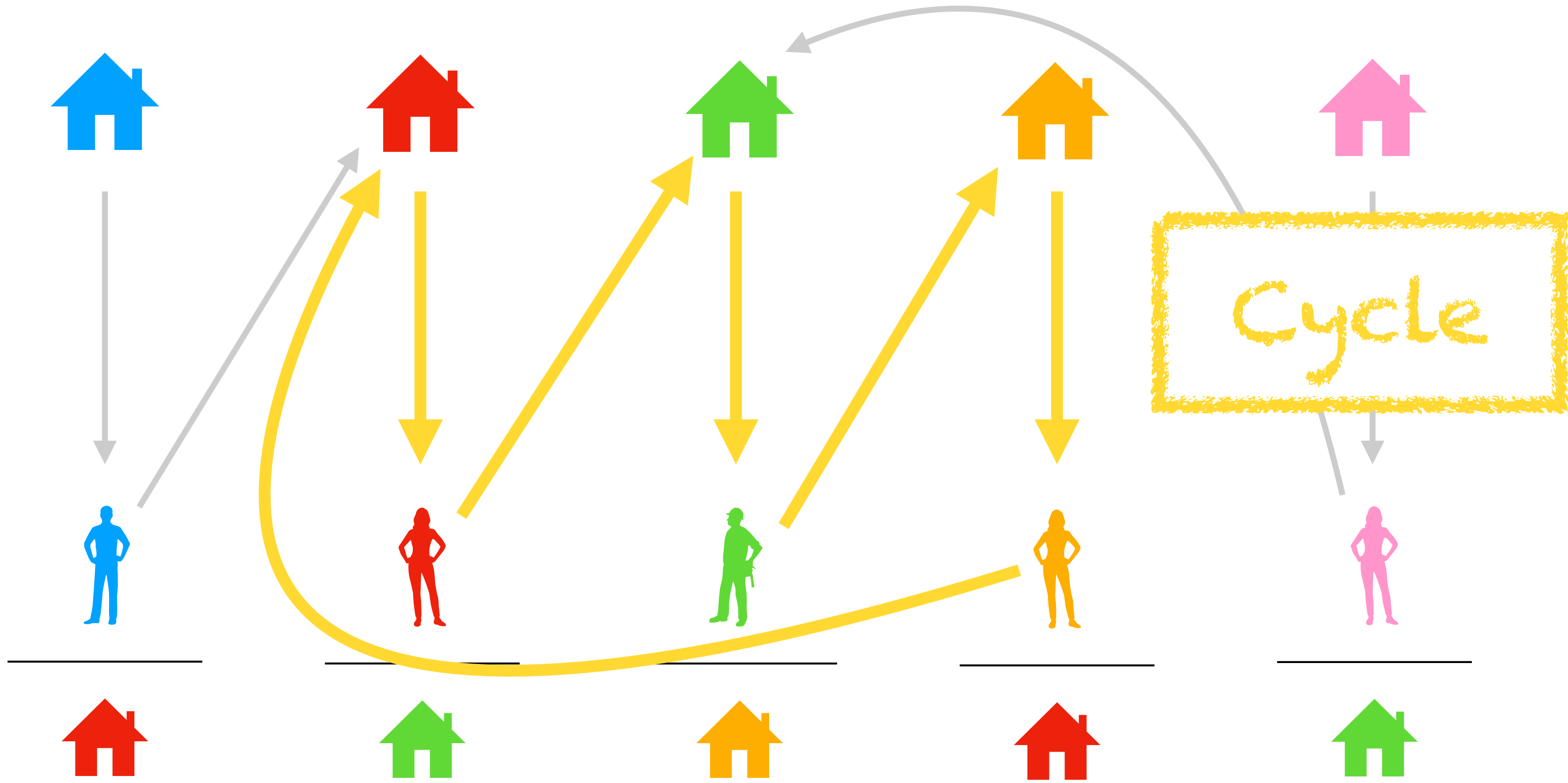
Top-Trading Cycle

Top-Trading Cycle

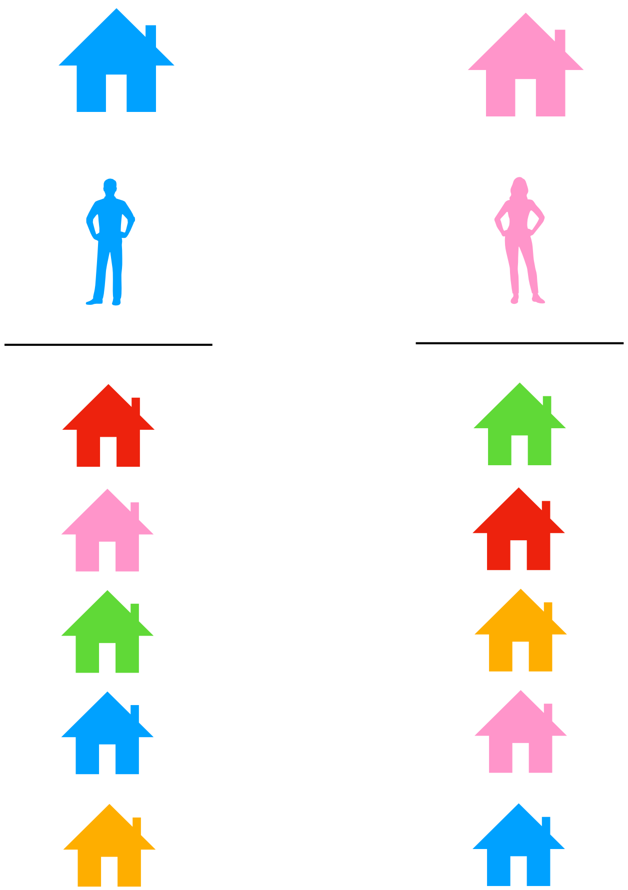
Top-Trading Cycle

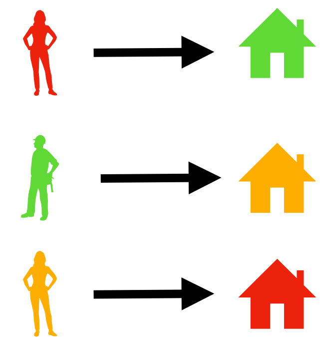


Top-Trading Cycle

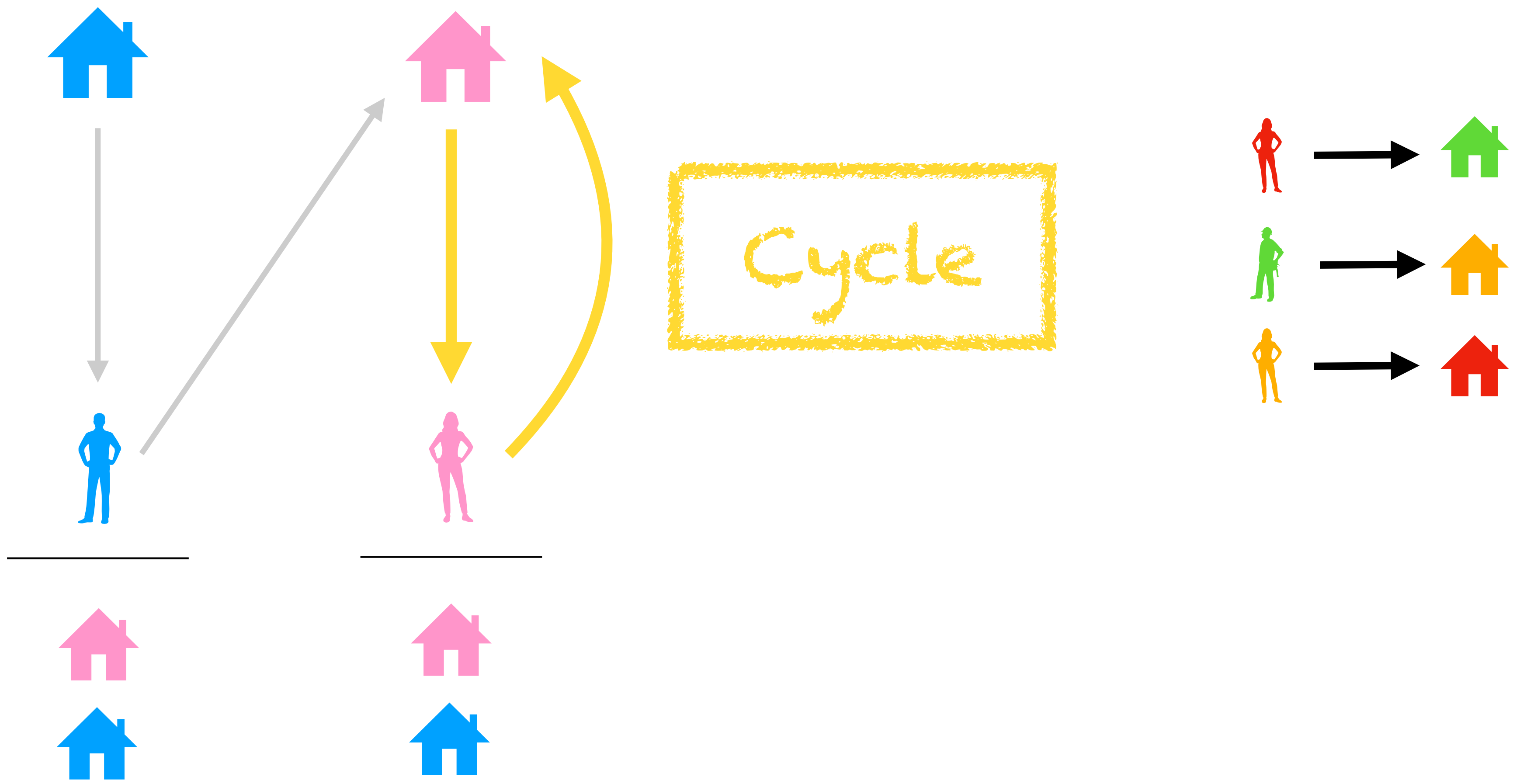


Top-Trading Cycle

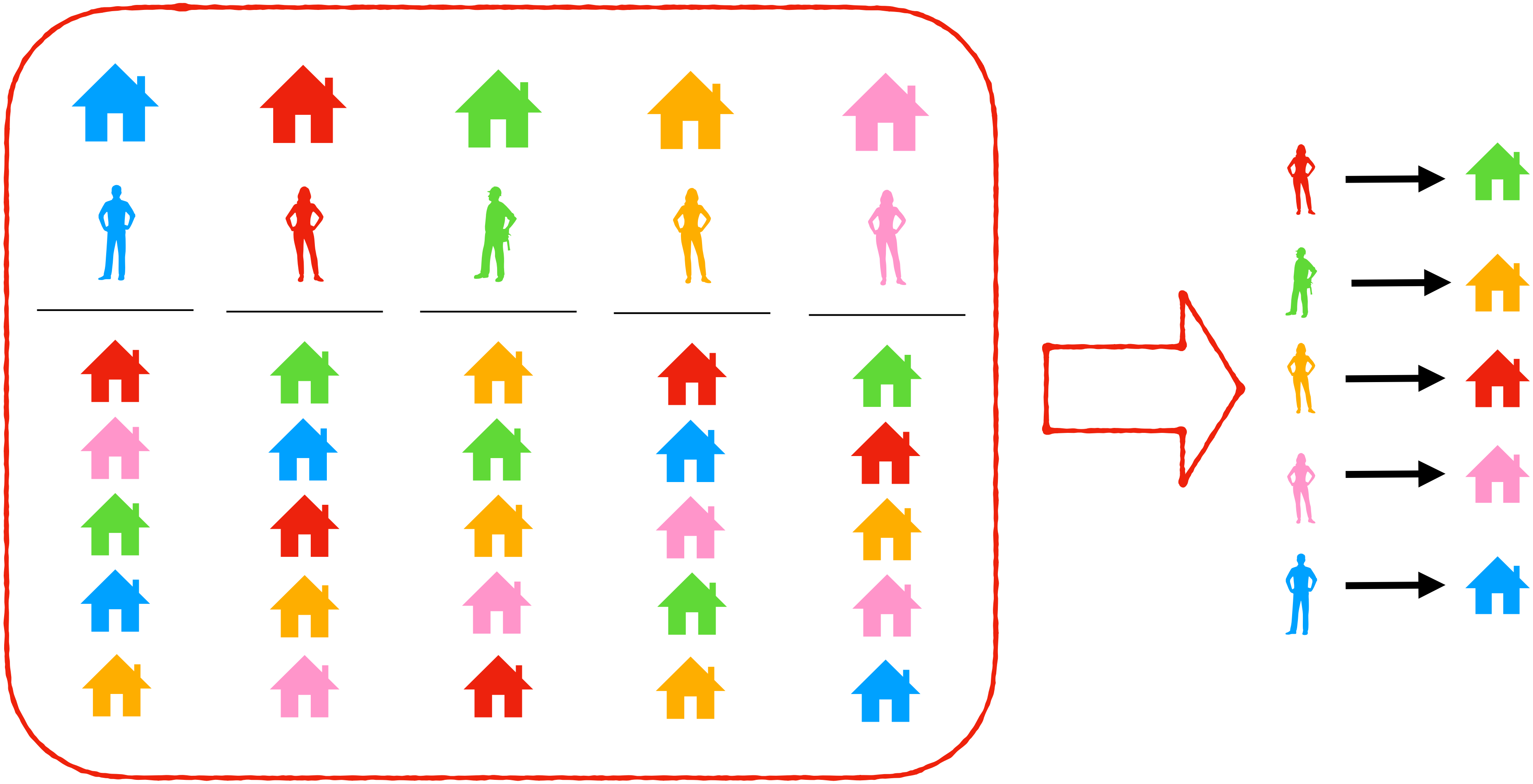




Top-Trading Cycle



Top-Trading Cycle



Top-Trading Cycle

- **Proposition:** TTC $\Longrightarrow$ IR and PE matching
- **Proof:**
  - PE: very similar to SD. Assume not: $\exists$ matching ν that Pareto dominates μ (TTC outcome)
  - All agents leaving in round 1 get their top choice in μ, so $\nu(i) = \mu(i)$ for $i \in I_1$
  - Agents leaving in round 2: cannot get sth strictly better in ν, since that would mean they are getting sth that went away in round 1. Hence, $\nu(i) = \mu(i)$ for $i \in I_2$. Continue by induction to reach contradiction.
  - IR: Houses never leave before their owners. Agents can always choose their house in the round in which they leave.

Top-Trading Cycle

- **Theorem:** Core is unique & core $= \{$TTC-outcome$\}$
- **Background:**
  - Shapley and Scarf (1974) proved core is non-empty w/o using TTC
  - David Gale proposed the TTC as a simpler way to prove result
  - Roth and Postlewaite (1977) proved that the core is unique

Top-Trading Cycle

- **Theorem:** Core is unique & core $= \{$TTC-outcome$\}$
- **Proof:**
  - First, show that TTC-outcome is in the core
  - Assume it is blocked by coalition A with ν. Take $j =$ first agent in A who leaves with $\nu(j) \succ_j \mu(j)$.
  - Then, $\nu(j)$ left before j. Let $a_1 \in A$ be the owner of $\nu(j)$. Consider the cycle in which a_1 leaves:

$$a_1 \to h_{a_2} \to a_2 \to h_{a_3} \to a_3 \to \cdots \to h_{a_m} \to a_m \to h_{a_1}.$$

  - Since $\nu(a_1) = \mu(a_1) \Rightarrow h_{a_2} = \nu(a_1)$, so a_2 is in the coalition.
  - Hence, $\nu(a_2) = \mu(a_2) = h_{a_3}$ and a_3 is in the coalition, $\ldots$ a_m is in the coalition.
  - However, $h_{a_1} = \mu(a_m) = \nu(a_m)$ is a contradiction, since we had $h_{a_1} = \nu(j)$

Top-Trading Cycle

- **Theorem:** Core is unique & core $=$ {TTC-outcome}
- **Proof:**
  - Second, show there is no other matching in the core. Let $\nu \neq \mu$.
  - Let i be first agent who leaves with $\mu(i) \neq \nu(i)$.
  - Every agent who leaves before gets the same house in μ and ν.
  - Every agent who leaves along i is getting a house from someone who leaves at the same time or afterwards.
  - Since $\mu(i)$ is i's top-choice among all these houses, $\mu(i) \succ_i \nu(i)$.
  - Then, agents who leave along i can form a coalition and block ν with μ.
  - Hence, ν is not in the core.

Top-Trading Cycle

- **Theorem (Roth, 1982):** TTC is strategy-proof
- **Proof:**
  - Key for the proof: an agent **cannot** affect the cycles who leave the market before by changing their preferences. Even if they point to a house in a cycle, no one in the cycle will point to their house.
  - No gain from leaving "early" since they can always keep pointing to their top choice, instead of "closing" a cycle that will go nowhere.
  - No gain in leaving "late" since they can only get something worse.

Top-Trading Cycle

- **Theorem (Ma, 1994):** TTC $\Longleftrightarrow$ SP + PE + IR
- **Proof:**
  - Let τ = TTC and $\phi \neq \tau$ s.th. ϕ is also SP, PE and IR.
  - The key is showing: $\phi[(\succcurlyeq_i)](i) = \tau[(\succcurlyeq_i)](i)$ for agents leaving in round 1, in I_1.
  - Suppose not $\Rightarrow \exists\ i_1$ with $\tau[(\succcurlyeq_{i_1})](i_1) \succ_{i_1} \phi[(\succcurlyeq_{i_1})](i_1) \succcurlyeq_i h_1$ (last from ϕ-IR)
  - Then, i_1 is trading with someone in roud 1, consider cycle:

$$i_1 \to h_2 \to i_2 \to h_3 \to \cdots \to i_m \to h_1 \to i_1, \quad \text{where } h_{k+1} = \text{top choice of } i_k.$$

  - Consider $\succcurlyeq_i'$ for $i \in I_1$: top-choice in $\succcurlyeq_i'$ is same as $\succcurlyeq_i$, but second choice in $\succcurlyeq_i'$ is h_i
  - When reporting $\succcurlyeq_i'$ in ϕ, agents in I_1 get their top choice (h_{k+1}), or their own house (h_k).
  - ϕ-SP $\Rightarrow i_1$ gets h_1 when reporting $\succcurlyeq_i'$ to ϕ; otherwise, i_1 would lie: they'd get h_2, which is better than what they are getting when reporting $\succcurlyeq_i$ to ϕ.

Top-Trading Cycle

- **Theorem (Ma, 1994):** TTC $\iff$ SP + PE + IR
- **Proof:**

  [...]

  - ϕ-SP $\Rightarrow$ i_1 gets h_1 when reporting $\succcurlyeq'_i$ to ϕ; otherwise, i_1 would lie: they'd get h_2, which is better than what they are getting when reporting $\succcurlyeq_i$ to ϕ.
  - Since i_1 keeps their own house at $\phi(\succcurlyeq'_i, \succcurlyeq_{-i})$, if i_m also misreports $\succcurlyeq'_m$, they would also keep their own house, h_m (sine the only other possibility is to get h_1, which is being assigned to i_1).
  - Consecutively, every agent in the cycle gets their own house when they report $\succcurlyeq'_i$. But this is a contradiction, since this is not PE: they can trade to the TTC allocation and be (strictly) better off.
  - Therefore, τ and ϕ assign the same houses to those in round 1 at $(\succcurlyeq_i)$. By induction, the same holds for all agents leaving in further rounds, and we have a contradiction.

House-allocation with existing tenants

- **House-allocation problem with existing tenants** $= (I_E, I_N, H_O, H_V, (\succcurlyeq_i)_{i\in I})$
  - I_E = existing tenants (already own a house)
  - I_N = new applicants (do not own a house)
  - H_O = houses owned by existing tenants
  - H_V = vacant houses
  - Assume $\succcurlyeq_i \in \mathcal{P}(H_O \cup H_V)$
- A house-allocation problem with existing tenants is the **middle ground** between a house-allocation problem and a housing market
  - Obtain house-allocation problem if $I_E = H_O = \varnothing$
  - Obtain housing market if $I_N = H_V = \varnothing$

House-allocation with existing tenants

- Most common **application**: assigning university housing to undergraduate students
  - Existing tenants are students who already own a room
  - New applicants are new students who do not have a room
- In the next slides, study three mechanisms used in **practice**
  1. Serial Dictatorship with Squatting Rights aka. "housing lottery" (used in CMU, Duke, Michigan, Northwestern, and Penn)
  2. Serial Dictatorship with Waiting List
  3. MIT NH4 (used in a residency at MIT)
- And an **extension of TTC** to existing tenants

House-allocation with existing tenants

- Serial Dictatorship with Squatting Rights ("housing lottery")
  - Run a Serial Dictatorship (assigning priorities randomly), but, before running the mechanism, allow existing tenants to opt out of the mechanism and be assigned to their current house
  - **Exercise**: is this mechanism Pareto efficient? Strategy-proof?
- Serial Dictatorship with Waiting List
  - Run a Serial Dictatorship with π, but give priority to existing tenants over their houses.
  - The house owned by existing tenant $\pi(k)$ is not available to any agent with a priority higher than $\pi(k)$, and is available to agents with lower priority than $\pi(k)$ only if the existing tenant chooses another house in round k.
  - **Exercise**: is this mechanism Pareto efficient? Strategy-proof?

House-allocation with existing tenants

- MIT NH4
  - $\pi =$ priority order. Assign houses **tentatively** according to π as in a Serial Dictatorship, until all houses are assigned or a **squatting conflict occurs**.
  - A squatting conflict occurs if it is the turn of an existing tenant, say $i \in I_E$, and they find all of the available houses worse than h_i.
  - Call the agent who was tentatively assigned to h_i the **conflicting agent**.
  - Erase all the assignments made up to the conflicting agent, and assign the existing tenant to the house they previously owned.
  - At this point, the squatting conflict is resolved. Restart the algorithm with the conflicting agent, and resolve all the subsequent squatting conflicts in the same manner.
  - **Exercise**: is this mechanism Pareto efficient? Strategy-proof?

House-allocation with existing tenants

- TTC with existing tenants
  - π = priority order. Initially, all agents are unassigned, and the set of **available** houses is the set of vacant houses.
  - Unassigned agents point to their favorite house in the market (available or unavailable), available houses point to the unassigned agent with the highest priority, and occupied houses point to their owners.
  - Find all cycles and remove them from the market by assigning the agents to the houses they are pointing to within each cycle
  - If an existing tenant is assigned to a house, and the house they previously owned is not part of any cycle, it becomes available in the next round.

House-allocation with existing tenants

- **Theorem (Abdulkadiroğlu and Sönmez, 1999):** The TTC with existing tenants is Pareto efficient, individually rational, and strategy-proof.
- **Exercise**: Prove this Theorem. The proof goes along the same lines as the ones we have already done.

Week 3: Kidney Exchange

BEEM147: Topics in Microeconomic Theory II

Matching and Market Design

Spring Term 2021

University of Exeter

Overview

- Last week
  - Housing markets
  - TTC and the core
  - House allocation with existing tenants
- This week
  - Kidney exchange
  - Kidneys as houses, tenants as donors
  - Pairwise kidney exchange

Kidney Exchange

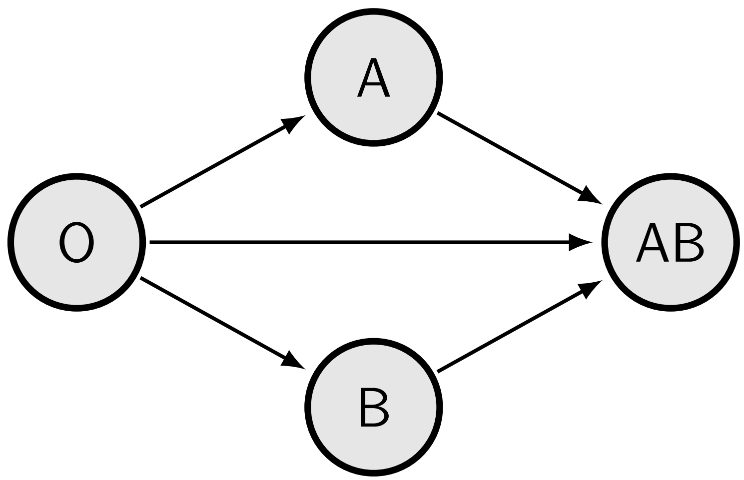
- Transplanted kidneys come from both deceased and living donors
- Cadaveric kidneys allocated by a queue or waiting list
- **Problem**: there is a shortage of kidneys
- In 2016, more than 100,000 people were waiting for a kidney transplant in the U.S.
- The median patient waits 3.5 years to receive a kidney
- In 2014, 17k+ kidney transplants in the U.S.: 67.6% cadaveric and 32.4% from living donors
- In every country in the world (except Iran), it is illegal to buy and sell human kidneys

Kidney Exchange

- Until the early 2000s, living donors were only relatives or loved-ones
- From the early 2000s, the exchange of kidneys has been growing
- **Why?** *Not everyone can donate a kidney to any one.* Donors and recipients need to have *compatible* blood and tissue types.
- **Two-way (pairwise) exchange**:
  - your donor gives their kidney to patient k, and the donor of patient k gives you their kidney
- **Multiple exchange**:
  - There are K pairs of donors and patients
  - Donor $k \in \{1, \ldots, K-1\}$ gives their kidney to patient $k+1$, and either (i) donor K gives their kidney to patient 1, or (ii) patient 1 is given a high priority in the cadaveric queue (aka. indirect exchange).

Blood and tissue type compatibility

- ABO **blood-types**: O, A, B, or AB
- Every blood-type can donate to itself, but not to everyone else:



- **Tissue-types** (known as HLA type) depend on 6 proteins
  - The stronger the match, the more likely that the transplant will be successful
  - Need to test for antibodies. If recipient has antibodies for tissue type of donor, transplant is not viable.

Kidneys as houses, tenants as donors

- Simplest case:
  - Patients with donors $\longleftrightarrow$ existing tenants with houses
  - Patients w/o donors $\longleftrightarrow$ new applicants
  - Kidneys from "altruistic" or deceased donors $\longleftrightarrow$ empty houses
- Slight caveats when applying TTC:
  - Kidneys in the cadaveric queue are ex-ante unknown: matching patients to waiting list is a lottery.
  - May create chains instead of cycles; a patient may be in multiple chains (each terminating in the waiting list).
  - Need to select one among multiple possible chains; see Roth, Sönmez, and Ünver (2004)
  - Transplants must be *simultaneous*. Hard to handle long cycles logistically.

Pairwise kidney exchange

- Restrict to two-way exchanges
- **Questions**: how to maximize the number of transplants? Is there a trade-off between priority and quantity of transplants?
- (I, R) = **pairwise kidney exchange problem**
- $I = \{1, 2, \ldots, n\}$ set of donor-patient pairs
- $R = (r_{ij})_{i \neq j}$ compatibility matrix
  - $r_{ij} = 1$ iff donor i can donate to j and donor j can donate to i
- Matching: $\mu : I \to I$ s.th. $\mu(i) = j$ iff $\mu(j) = i$ and $r_{ij} = 1$.
  - $\mu(i) = i$ indicates i receives no kidney under μ
- $\mathcal{M}(I, R)$ = set of all matchings

Pairwise kidney exchange

- Set of patients who receive a kidney under μ:

$$M_\mu = \{i \in I : \mu(i) \neq i\}\,.$$

- Matching μ is **efficient** if there does not exist matching ν such that

$$M_\mu \subseteq M_\nu \quad \text{and} \quad M_\mu \neq M_\nu.$$

- **Question**: to determine if a matching is efficient, is it enough to count the number of patients receiving a kidney?
  - **A**: Yes, but it is not obvious. Requires some tools to prove.

Matroids

- A **matroid** is a pair $(X, \mathcal{I})$ where X is a finite set, called the **ground set**, and $\mathcal{I}$ a collection of subsets of X, called the **independent sets**, that satisfy the following properties:
  (i) the subsets of independent sets are also independent, i.e., if $J \in \mathcal{I}$ and $J' \subseteq J$ then $J' \in \mathcal{I}$;
  (ii) if one independent set J is larger than another one J', i.e., $|J| > |J'|$, then there exists $x \in J \backslash J'$ such that $J' \cup \{x\}$ is an independent set.
- Examples:
  (a) The collection of all linearly independents columns of a matrix forms a matroid
  (b) Let $n \leqslant |X| < \infty$, and $\mathcal{I} = \{S \subseteq X : |S| \leqslant n\}$, then $(X, \mathcal{I})$ is a matroid

Matroids and matchable pairs

- Let $\mathcal{I}$ be the collection of all groups of patient-donor pairs that are **matchable**, i.e.,

$$\mathcal{I} = \{J \subseteq I : \exists\, \mu \in \mathcal{M}(I, R) \text{ s.t. } J \subseteq M_\mu\}.$$

- **Proposition**: $(I, \mathcal{I})$ is a matroid.
- **Proof:** see the lecture notes
- The key is property (ii) of a matroid:
  (ii) if there are two sets $J, J' \subseteq I$ with $|J| > |J'|$ and two matchings μ and μ' such that $J \subseteq M_\mu$, $J' \subseteq M_{\mu'}$, then there exists $i \in J \backslash J'$ and $\nu \in \mathcal{M}(I, R)$ such that $J' \cup \{i\} \subseteq M_\nu$.
- **Proposition**: If $\mu, \nu \in \mathcal{M}(I, R)$ are efficient, then $|M_\mu| = |M_\nu|$.
- **Proof**: Follows from (ii): if μ and ν are efficient with $|M_\mu| > M_\nu$, then there exists ν' and $i \in M_\mu \backslash M_\nu$ such that $M_\nu \cup \{i\} \subseteq M_{\nu'}$, which contradicts ν being efficient.

Priority mechanisms

- Fix a priority order over I, for simplicity: $\pi = \{1, 2 \ldots, n\}$
- Let $\mathcal{E}^0 = \mathcal{M}(I, R)$
- In step 1, $\mathcal{E}^1 =$ the set of all matchings μ under which patient 1 receives a kidney.
- In step 2, $\mathcal{E}^2 =$ all the matchings $\mu \in \mathcal{E}^1$ in which patient 2 also receives a kidney, and so on.
- Formally, for every $k \leqslant n$,

$$\mathcal{E}^k = \begin{cases} \{\mu \in \mathcal{E}^{k-1} : \mu(k) \neq k\} & \text{if } \exists\ \mu \in \mathcal{E}^{k-1} \text{ s.th. } \mu(k) \neq k \\ \mathcal{E}^{k-1} & \text{otherwise} \end{cases}$$

- The set of **priority matchings** is given by $\mathcal{E}^n$.
- **Intuition**: match as many patients as possible starting with the patient with the highest priority and following the priority ordering, never "skipping" or "sacrificing" a higher priority patient because of a lower priority patient.

Priority mechanisms

- **Proposition**: Every priority matching is efficient
- **Proof**: left as an exercise
- **Intuition**: There is no trade-off between priority allocation and the number of transplants that can be arranged

Priority mechanisms

- In practice, a patient could decline a medically compatible kidney (e.g., do not want to receive a kidney from an older donor, or a smoker)
- **Proposition**: Let $A_i \subseteq K_i$ be the subset of kidneys i reports as acceptable, where $K_i = \{j : r_{ij} = 1\}$. In a priority matching, it is strategy-proof for patients to report $A_i = K_i$.
- **Proof**: By misreporting compatible kidneys, unmatched patients can only decrease the chances of being matched under a priority mechanism: for patient k, the sets $\mathcal{E}^{k'}$ shrink if they misreport, for $k' < k$.

Week 4: Random allocations

BEEM147: Topics in Microeconomic Theory II

Matching and Market Design

Spring Term 2021

University of Exeter

Overview

- Last week
  - Kidney exchange
  - Kidneys as houses, tenants as donors
  - Pairwise kidney exchange
- This week
  - Random allocations
  - Stochastic dominance and ordinal efficiency
  - Random Serial Dictatorship
  - Top-Trading Cycle with Random Endowments
  - Probabilistic Serial Mechanism

Random allocations

- $(I, X =)$ **random allocation problem**
  - I = set of n agents
  - X = set of n objects
- Same ingredients as house allocation problem
- **Random allocation**: $n \times n$ bistochastic matrix P where

$$P_{ix} = \mathbb{P}\{\text{agent } i \text{ gets object } x\}$$

- **Bistochastic** matrix: rows and columns add up to 1: $\forall\ i \in I, x \in X$

$$\sum_{x' \in X} P_{ix'} = 1 \quad \text{and} \quad \sum_{i' \in I} P_{i'x} = 1$$

- **Note**: every agent is assigned to exactly one object w.p.1

Ordinal Preferences

Example

$X = \{\$0, \$10, \$18\}$. Agents prefer more money than less. Consider agent i and the random allocations P and Q:

$$P_{i,\$10} = 1 \quad \textit{and} \quad Q_{i,\$0} = Q_{i,\$18} = 1/2.$$

Under P, agent i gets \$10 for sure. Under Q they get \$0 with probability 1/2 and \$18 otherwise.

Does agent i prefer P or Q?

Given $\succcurlyeq_i$, we cannot tell. An agent may find Q too risky and prefer P, or may not mind the risk and prefer Q.

First-Order Stochastic Dominance

Example

$X = \{apple, banana, orange\}$. Assume

$$apples \succ_i oranges \succ_i bananas.$$

Consider random allocations P and P':

$$P_{i,oranges} = 1 \quad and \quad P'_{i,apples} = P'_{i,oranges} = 1/2.$$

Does agent i prefer P or P'?

Reasonable to assume i prefers P' over P. It gives higher probability to more preferred objects. No trade-off between preferences and risk.

FOSD and Ordinal Efficiency

- Random allocation P **(first-order) stochastically dominates** Q if for every $i \in I$ and every $x \in X$,

$$\sum_{y \in X : y \succcurlyeq_i x} P_{iy} \geqslant \sum_{y \in X : y \succcurlyeq_i x} Q_{iy}.$$

  Intuition: P FOSD Q if it gives more probability to objects higher in the preference ranking

- FOSD is *not* a complete order
- A random allocation is **ordinally efficient** if there is no other random allocation that stochastically dominates it
- Why not use Pareto efficiency? Would agents agree to participate in a mechanism that is not ordinally efficient?

Why not cardinal preferences?

- Assume each i has utility $u_i : X \to \mathbb{R}$
- Agent i prefers P over Q if
$$\sum_{x \in X} u_i(x) P_{ix} \geqslant \sum_{x \in X} u_i(x) Q_{ix}$$
- Expected-utility ranking is complete over all random allocations
- $\mathcal{U}(\succcurlyeq) =$ set of utility functions that represent $\succcurlyeq$
- **Proposition:** random allocation P first-order stochastically dominates Q if and only if for every $i \in I$ and every utility function $u_i \in \mathcal{U}(\succcurlyeq_i)$,
$$\sum_{x \in X} u_i(x) P_{ix} \geqslant \sum_{x \in X} u_i(x) Q_{ix}.$$
- **Intuition**: using FOSD as ranking criteria amounts to being agnostic about cardinal utilities

Random Serial Dictatorship

- Choose priority order π uniformly at random, then run Serial Dictatorship
- **Advantages**: very simple and fair (widely used)
- **Disadvantage**: it is not ordinally efficient
- **Counterexample**:
  - $I = \{1, 2, 3, 4\}$ and $X = \{x, y\}$
  - $x \succ_{1,2} y$ and $y \succ_{3,4} x$
  - P = random allocation from RSD
  - Can show (not obvious!): $P_{1x} = 5/12$ and $P_{2x} = 1/12$ (symmetric for other agents)
  - Q = flip two coins: first assigns x btw 1 and 2, second assigns y btw 3 and 4

    $\Rightarrow Q_{1x} = 1/2$ and $Q_{2x} = 0$
  - Both P and Q assign an object with 1/2 prob. But Q puts all mass on top choice: Q FOSD P.

Top-Trading Cycle with Random Endowments

- First, assign objects to agents uniformly at random. Then, run TTC.
- **Advantages**: TTC has good properties in the deterministic case
- **Disadvantage**: It suffers from the same shortcomings as RSD; actually, they are equivalent
- **Theorem** (Abdulkadiroğlu and Sönmez 1998): RSD and TTC with random endowments generate the same random allocation for every preference profile.

Probabilistic Serial Mechanism

- Is there a mechanism that is ordinally efficient? Yes!
- **Probabilistic Serial** (Bogomolnaia and Moulin 2001):
  - Think of each object as a cake of size 1
  - Time is continuous $t \in [0, 1]$.
  - Each agent "eats" from their favorite cake, among the ones that have not been finished.
  - All agents eat at speed 1, from one cake at a time.
  - P_{ix} = share of cake x eaten by i at time $t = 1$

Probabilistic Serial Mechanism: Example

- $I = \{1, 2, 3, 4\}$ and $X = \{w, x, y, z\}$
- Preferences:

$$\succ_1, \succ_2 \colon x, y, z, w, \quad \text{and} \quad \succ_3, \succ_4 \colon y, x, w, z.$$

- At time $t = 0$:
  - Agents 1 and 2 start eating x; agents 3 and 4 start eating y
- At time $t = 1/2$:
  - Agents 1 and 2 finish x and start eating z; agents 3 and 4 finish y and start eating w
- Hence,

$$P_{1x} = P_{2x} = P_{3y} = P_{4y} = 1/2$$

$$P_{1z} = P_{2z} = P_{3z} = P_{4w} = 1/2$$

$$\text{otherwise} = 0$$

Probabilistic Serial Mechanism

- **Theorem** (Bogomolnaia and Moulin 2001): For every profile of preferences, the Probabilistic Serial mechanism produces an ordinally efficient random allocation.
- Before proving, useful lemma . . .

Probabilistic Serial Mechanism

- Define graph: $y \rightarrow x$ if $\exists\ i \in I$ s.th. $x \succ_i y$ and $P(i, y) > 0$
- **Lemma**: If P is stochastically dominated by another random allocation, then the graph has a cycle.
- **Proof**:
  - Let Q FOSD $P \Rightarrow \exists\ i_1 \in I, x \in X, x_1 \in X$ s.th. $x_1 \succ_1 x$, $Q(i_1, x_1) > P(i_1, x_1)$ & $Q(i_1, x) < P(i_1, x)$ (Exists some agent for which Q gives more mass to more preferred objects than P)
  - $\Rightarrow x \rightarrow x_1$
  - Since $\sum_{i \in I} Q(i, x_1) = \sum_{i \in I} P(i, x_1) = 1$, exists $i_2 \in I$ s.th. $Q(i_2, x_1) < P(i_2, x_1)$.
  - Q FOSD $P \Rightarrow \exists\ x_2 \in X$ s.th. $x_2 \succ_2 x_1$ & $Q(i_2, x_2) > P(i_2, x_2)$
  - $\Rightarrow x_1 \rightarrow x_2$

    $\ldots\ \exists\ i_3 \in I$ s.th. $Q(i_3, x_2) < P(i_3, x_2)$ $\ldots\ \exists\ x_3 \in X$ s.th. $x_3 \succ_3 x_2$ & $Q(i_3, x_3) > P(i_3, x_3) \Rightarrow x_2 \rightarrow x_3 \ldots$
  - Since X is finite, eventually we must find a cycle.

Probabilistic Serial Mechanism

- **Theorem** (Bogomolnaia and Moulin 2001): For every profile of preferences, the Probabilistic Serial mechanism produces an ordinally efficient random allocation.
- **Sketch of proof**:
  - Idea: agents within a cycle would like to exchange probabilities. But this cannot happen under the Probabilistic Serial mechanism
  - Consider cycle: $x \leftrightarrow y \Rightarrow \exists\ i_1, i_2 \in I$ s.th.
    - i_1 prefers x to y, and gets y with positive probability
    - i_2 prefers y to x, and gets x with positive probability
  - $t_1 =$ time at which i_1 starts eating $y \Rightarrow x$ is finished by $t_1 \Rightarrow x$ is finished before y
  - $t_2 =$ time at which i_2 starts eating $x \Rightarrow y$ is finished by $t_2 \Rightarrow y$ is finished before x
  - Contradiction!

Weeks 5 and 6: Marriage market

BEEM147: Topics in Microeconomic Theory II

Matching and Market Design

Spring Term 2021

University of Exeter

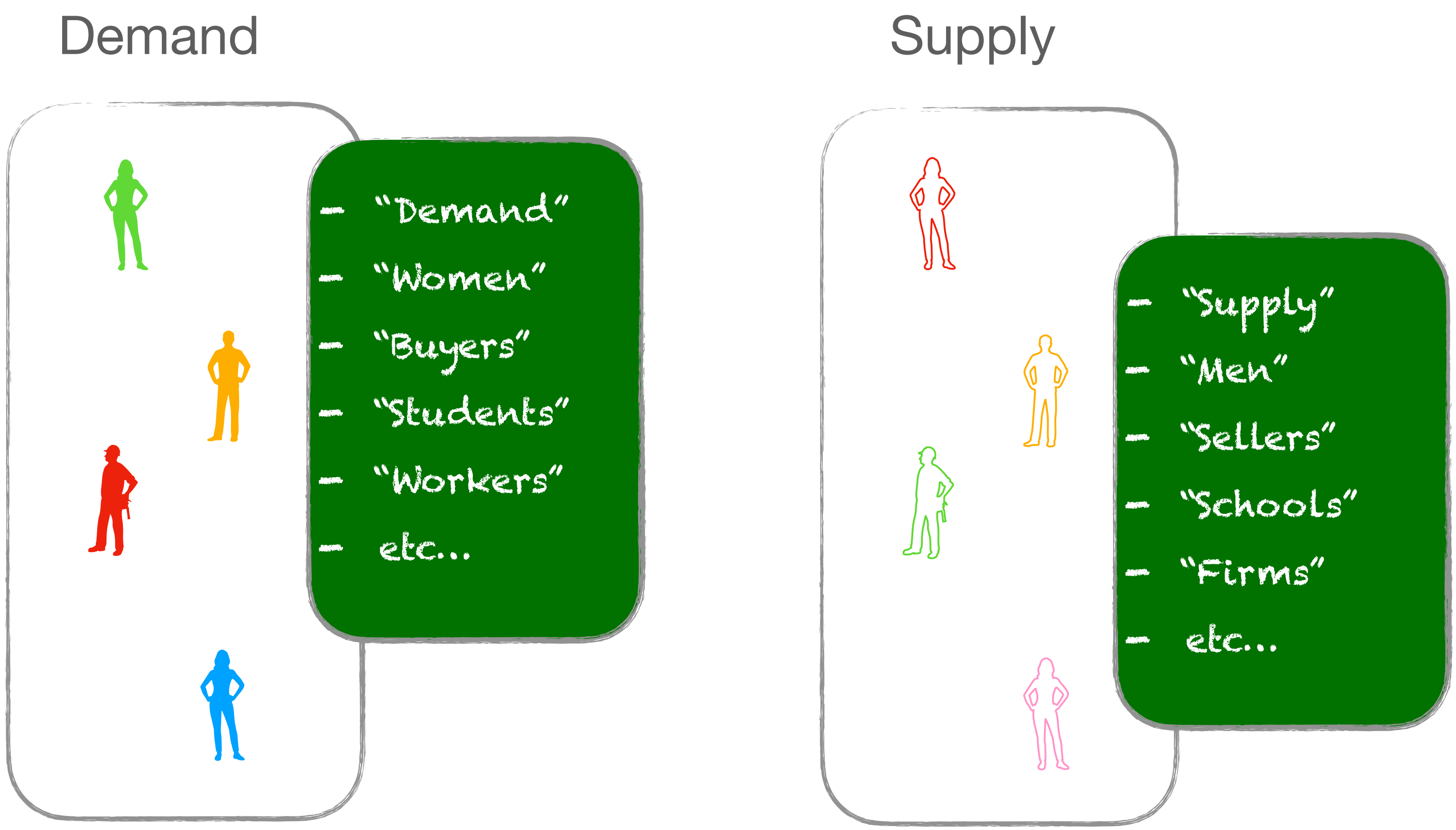
Overview

- Last week
  - Random allocations: stochastic dominance and ordinal efficiency
  - Random SD and TTC with random endowments
  - Probabilistic Serial Mechanism
- This week
  - Two-sided matching: marriage market
  - Stability and efficiency: Gale-Shapley algorithm
  - Structure of set of stable matchings
  - Incentives in the marriage market
  - Matching with transferable utility

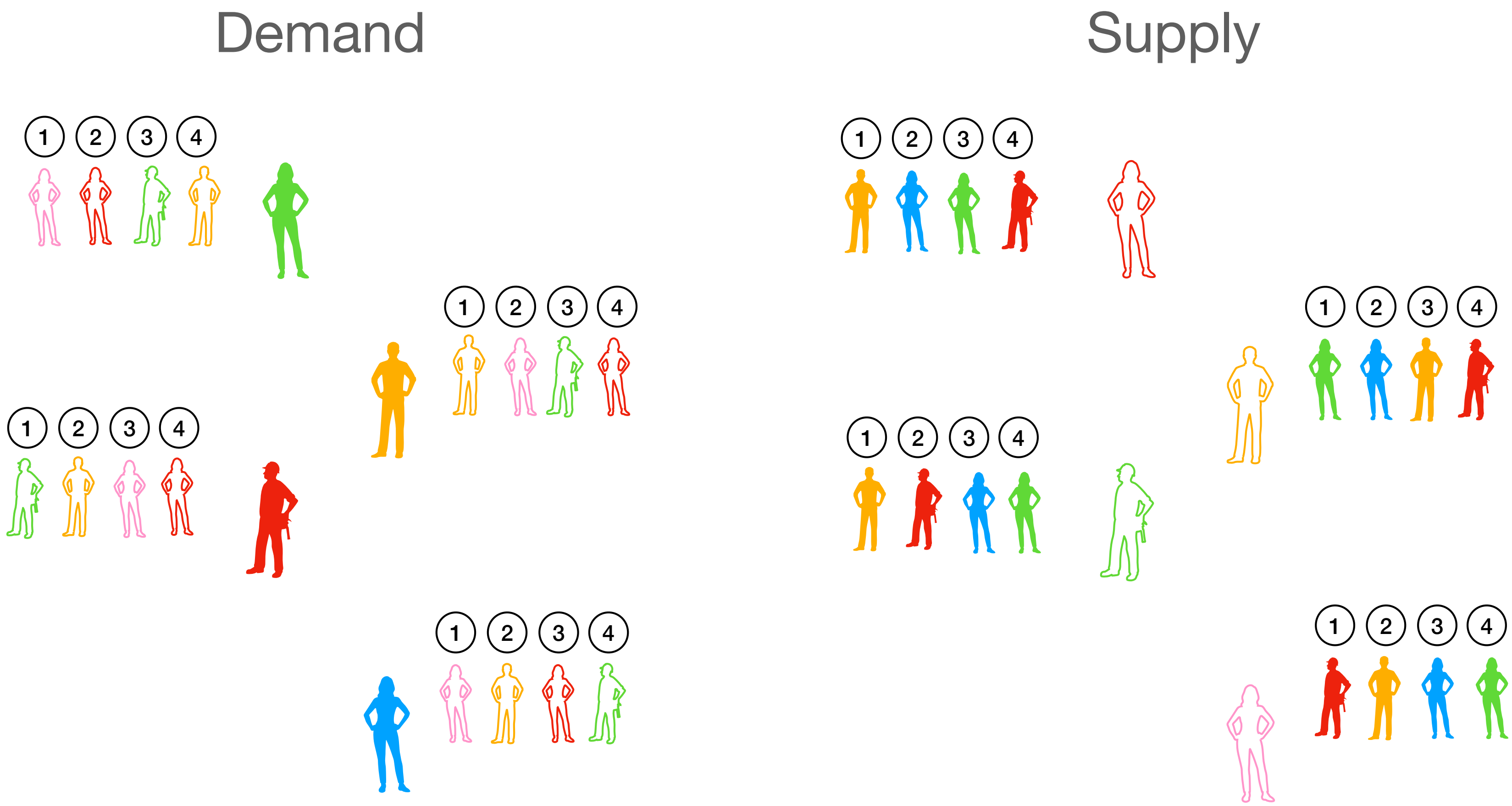
Two-sided matching

- **Marriage market**: $(M, W, (\succcurlyeq_m)_{m \in M}, (\succcurlyeq_w)_{w \in W})$
  - M = set of men (finite)
  - W = set of women (finite and $W \cap M = \varnothing$)
  - $\succcurlyeq_m \in \mathcal{P}(W \cup \{m\})$ for every $m \in M$
  - $\succcurlyeq_w \in \mathcal{P}(M \cup \{w\})$ for every $w \in W$
- Note: agents may wish to remain single, e.g. $w \succ_m w' \succ_m m \succ_m w''$
- **Matching**: $\mu : M \cup W \to M \cup W$ such that, for all $m \in M$ and $w \in W$
  (i) $\mu(m) \in W \cup \{m\}$
  (ii) $\mu(w) \in M \cup \{w\}$
  (iii) $\mu(m) = w$ if and only if $\mu(w) = m$
- $\mathcal{M}(M, W)$ = set of matchings between M and W

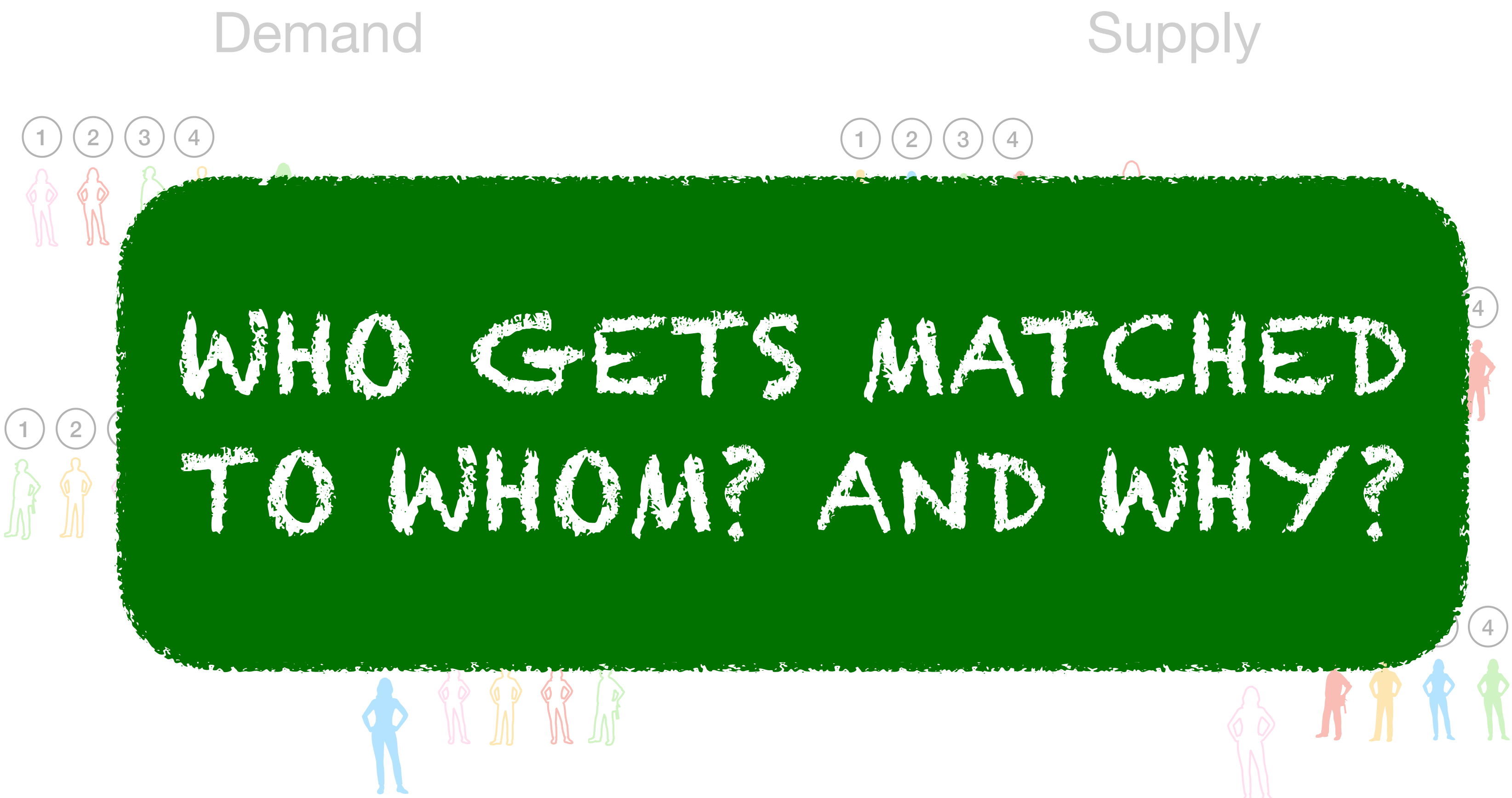
Marriage Market: Example



Marriage Market: Example



Marriage Market: Example



WHO GETS MATCHED TO WHOM? AND WHY?

Efficiency and Stability

- Matching μ is **Pareto efficient** if there is no other matching μ' such that $\mu'(i) \succcurlyeq_i \mu(i)$ for all $i \in M \cup W$ and $\mu'(i) \succ_i \mu(i)$ for some $i \in M \cup W$.
- Agent i is **acceptable** for agent j if $i \succ_j j$.
- Matching μ is **individually rational** if $\mu(i)$ is acceptable for every $i \in M \cup W$.
- A pair $(m, w) \in M \times W$ **blocks** a matching μ if $w \succ_m \mu(m)$ and $m \succ_w \mu(w)$.
- A matching is **stable** if it is individually rational and admits no blocking pair.
- $\mathcal{S}(M, W, \succcurlyeq) =$ set of all stable matchings in $(M, W, \succcurlyeq)$.
- **Question 1**: are stable matchings Pareto efficient and vice versa?
- **Question 2**: do stable matchings always exist? If so, how to find them?

Efficiency and Stability

- **Question 1**: are stable matchings Pareto efficient and vice versa?
- **Proposition**: Every stable matching is Pareto efficient
- Proof:
  - By contradiction: suppose μ' Pareto dominates stable matching μ
  - Wlog suppose $\mu'(m) \succ_m \mu(m)$
  - Let $w' = \mu'(m)$. Then $m = \mu'(w') \succ_{w'} \mu(w')$ since μ' Pareto dominates μ.
  - Then, (m, w') block μ, contradiction.
- **Exercise**: Is every Pareto efficient matching stable?
- **Question**: Are the First and Second Welfare Theorems satisfied in a marriage market?

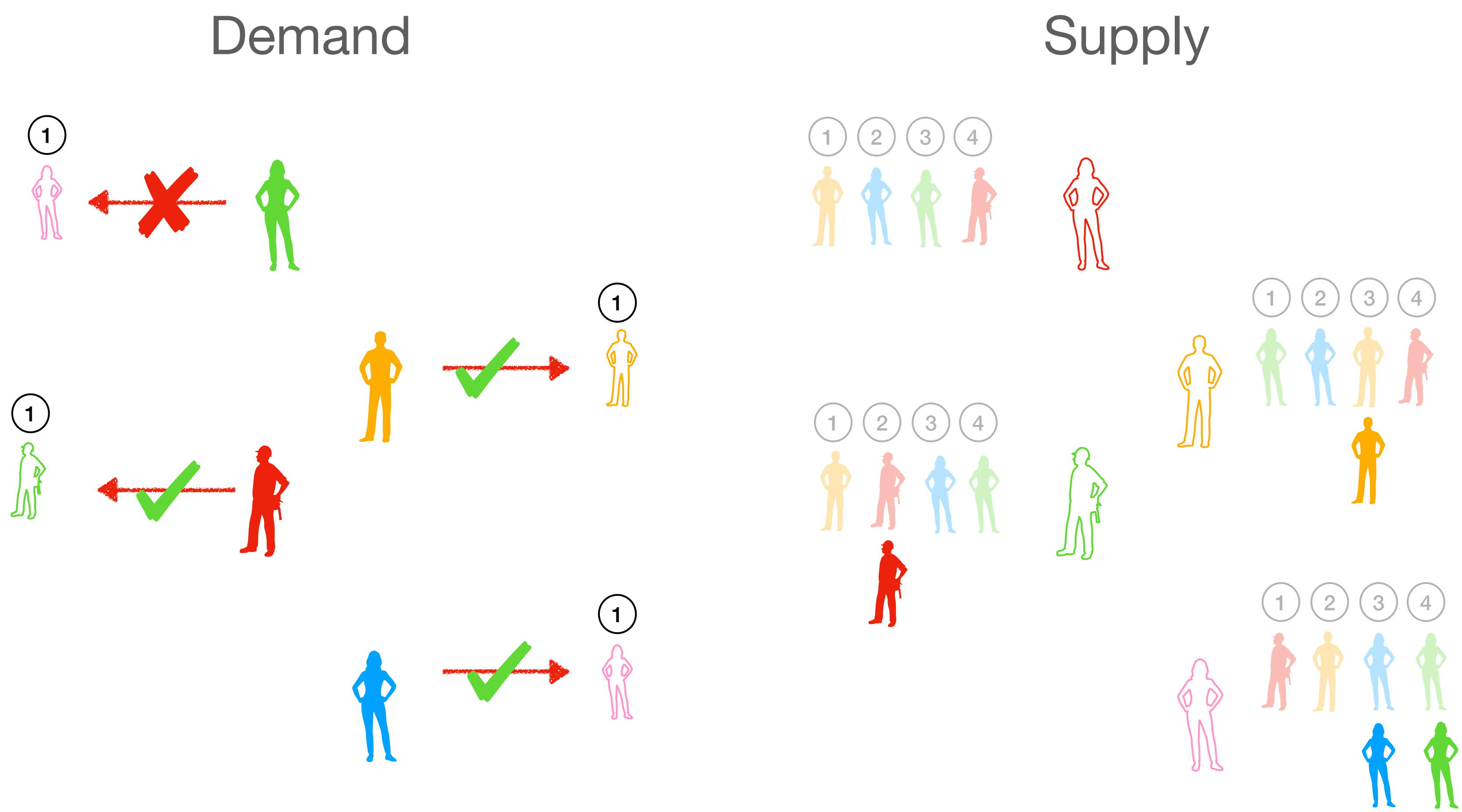
Efficiency and Stability

- **Question 2**: do stable matchings always exist? If so, how to find them?
- Before answering question, why do we care?
  - Stability is desirable in **centralized** settings. Participants will deviate from mechanisms that do not generate stable matchings.
  - **Decentralized** markets are likely to converge to stable matchings (same intuition as the core).
- **Answer to Q2**: Yes! Answer provided by David Gale and Lloyd Shapley in 1962. It relies on their celebrated Deferred Acceptance (DA) algorithm.

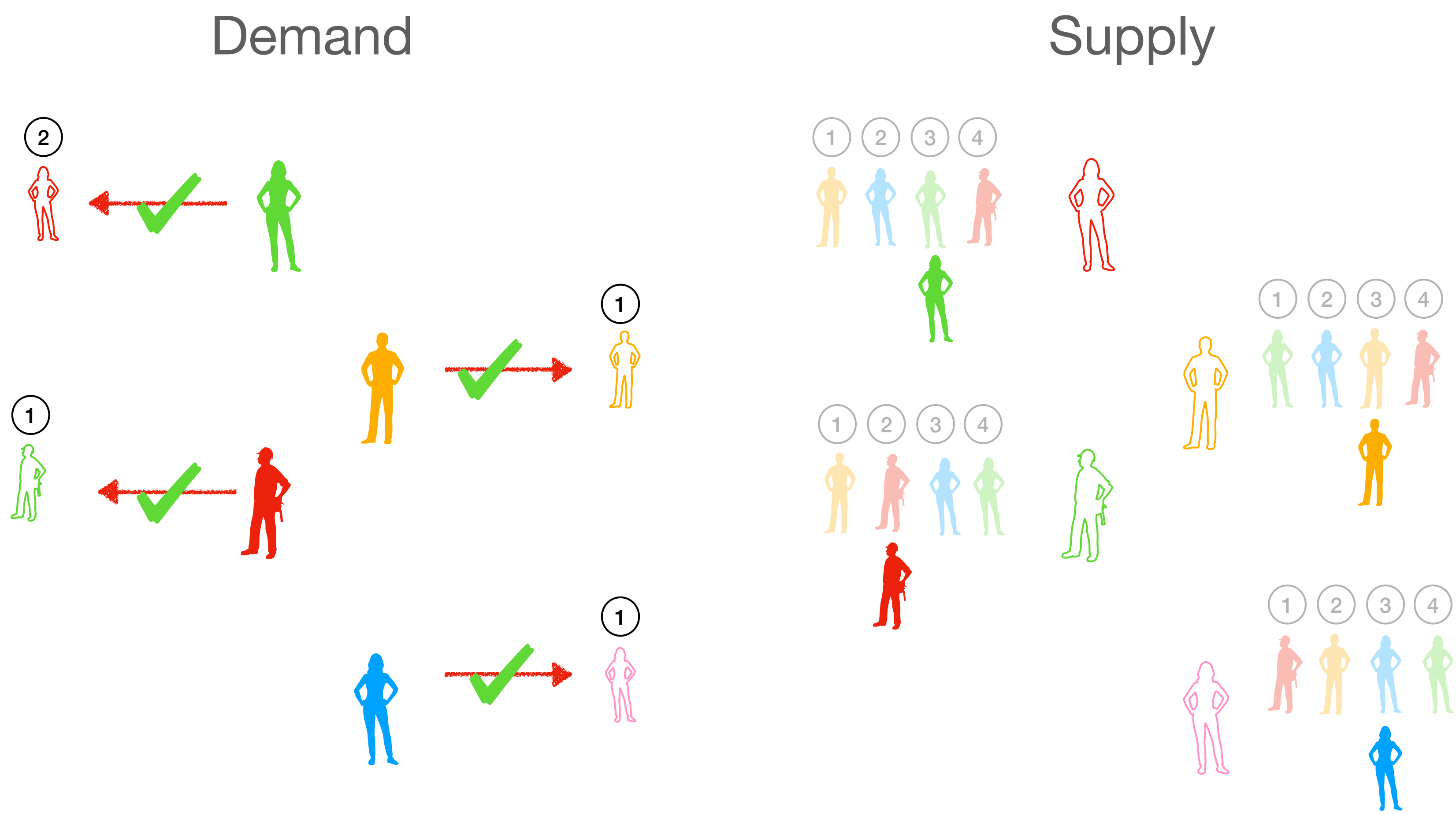
Gale-Shapley Algorithm

- Men-proposing version (woman-proposing is analogous)
- Initially, all men are active and no agent is provisionally matched. Proceed in steps:
  - All active men propose to their most preferred woman, among the ones they have not proposed to previously.
  - Each woman considers the set of men who have just proposed to her, and their provisional partner (if they have one). Women become provisionally matched to their favorite man among this set. All men who are not provisionally matched become active.
  - Stop if there are no active men, or if all active men have proposed to all acceptable women.
- **Theorem**: the output of the Gale-Shapley algorithm is a stable matching.
- **Key intuition**: Men go from top to bottom in their ranking, women go from bottom to top.

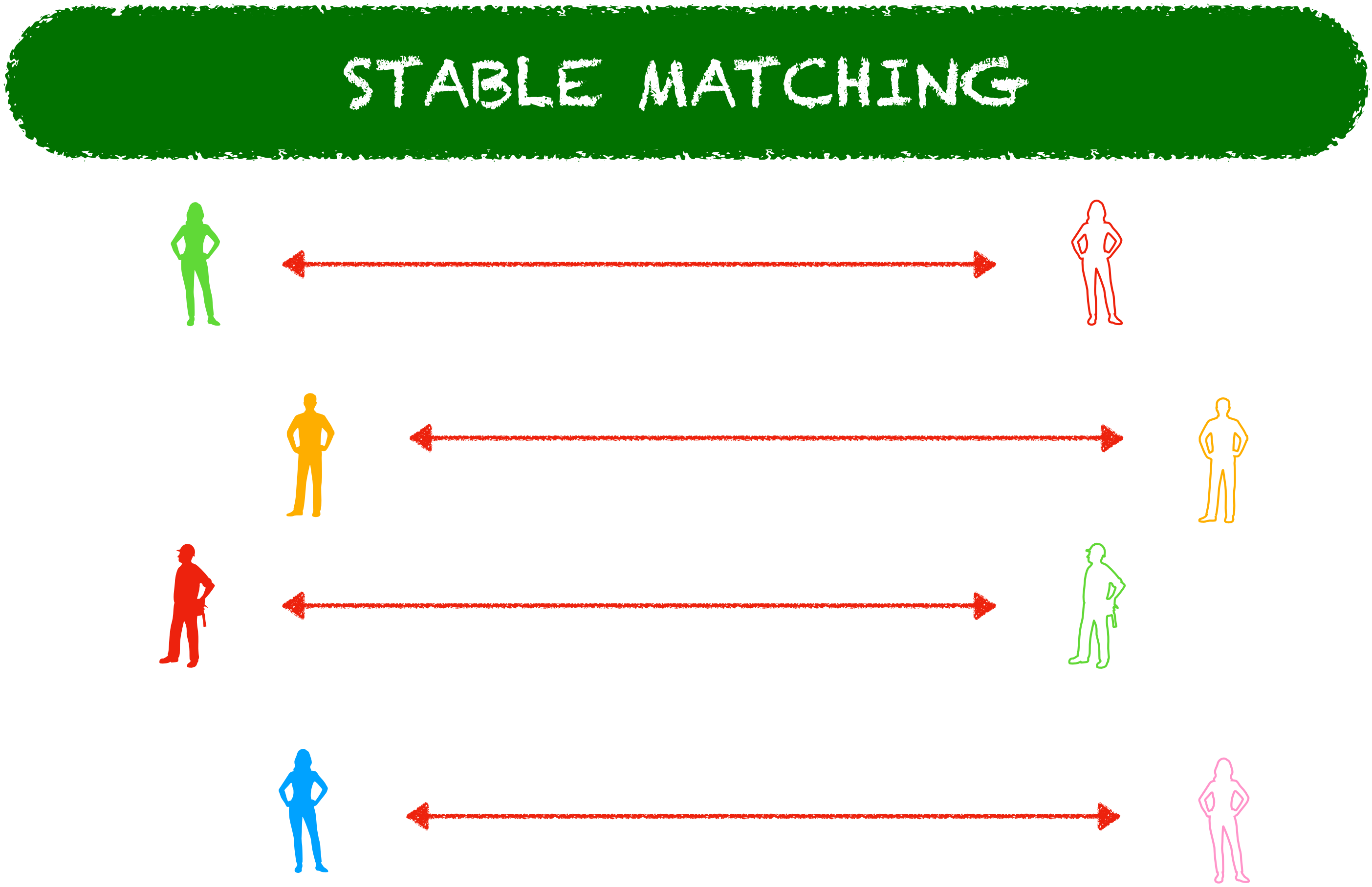
Gale-Shapley Algorithm: Example



Gale-Shapley Algorithm: Example



Gale-Shapley Algorithm: Example



Gale-Shapley Algorithm

- **Theorem**: the output of the Gale-Shapley algorithm is a stable matching.
- Proof:
  - μ = output of men-proposing GS-algorithm
  - Men only propose to acceptable women; women only accept proposals from acceptable men

  $\Rightarrow$ μ is individually rational
  - Let $m \in M$ and $w \in W$ be s.th. $w \succ_m \mu(m)$ (does w prefer m over $\mu(w)$? No!)
  - Why? m proposed to w in some iteration of the algorithm, and w rejected m for m'
  - Then, $\mu(w) \succcurlyeq_w m' \succ_w m$

  $\Rightarrow$ (m, w) do not block μ

  $\Rightarrow$ μ is stable

Brute-force stable matching

- **Question**: can we find a stable matching by "brute force"?
- **Answer**: Yes, we can start with any matching and randomly match blocking pairs. Eventually, we will reach a stable matching (though it may take too long!)
- **Theorem (Roth and Vande Vate, 1990)**: Let μ be any matching. There exists a finite sequence of matchings $\mu_1, \mu_2, \ldots, \mu_k$, such that $\mu_1 = \mu$, μ_k is stable, and for each $i = 1, \ldots, k-1$, there is a blocking pair (m_i, w_i) for μ_i such that μ_{i+1} is obtained from μ_i by satisfying the blocking pair (m_i, w_i).
- **Proof**: omitted (read the lecture notes)

Opposition of interests

- **Question**: does it matter who proposes in the DA algorithm?
- **Answer**: Yes! Make an example in which both versions of the algorithm do not reach the same matching.
- Actually, this is crucial ...
- **Theorem (Gale and Shapley, 1962)**: Let μ_M and μ_W be the outcomes of the men- and women-proposing Gale-Shapley algorithms, respectively. Then, for every $\mu \in \mathcal{S}(M, W, \succcurlyeq)$,

$$\forall m \in M, \quad \mu_M(m) \succcurlyeq_m \mu(m) \succcurlyeq_m \mu_W(m);$$

$$\forall w \in W, \quad \mu_W(w) \succcurlyeq_w \mu(w) \succcurlyeq_w \mu_M(w).$$

- Note: all men prefer μ_M over every other stable matching, and find μ_W to be the worst one. The women have exactly the opposite preferences!
- μ_M is the **M-optimal** stable matching; μ_W is the **W-optimal** stable matching

Opposition of interests

- **Theorem (Gale and Shapley, 1962)**: μ_M and μ_W outcomes of the men- and women-proposing GS-algorithms. Then, $\forall\ \mu \in \mathcal{S}(M, W, \succcurlyeq)$, $m \in M$, $w \in W$,

$$\mu_M(m) \succcurlyeq_m \mu(m) \succcurlyeq_m \mu_W(m) \quad \& \quad \mu_W(w) \succcurlyeq_w \mu(w) \succcurlyeq_w \mu_M(w).$$

- Proof:
  - First, show that $\mu_M(m)$ is the best partner for m out of
$$A_m = \{w \in W : \exists\, \mu \in S(M, W, \succcurlyeq) \text{ s.t. } w = \mu(m)\}.$$
  - Suppose not: $\exists\ m \in M$ who is the first to propose to $w \in A_m$ and is rejected (against m').
  - Then, $m' \succ_w m$, and m' finds w at least as good as everyone in $A_{m'}$ (o/w he would've been rejected by someone in $A_{m'}$).
  - Since $w \in A_m$, $\exists\ \mu \in \mathcal{S}$ s.th. $w = \mu(m)$.
  - However, note that (m', w) block μ since $w \succ_{m'} \mu(m')$. Contradiction!

Opposition of interests

- **Theorem (Gale and Shapley, 1962)**: μ_M and μ_W outcomes of the men- and women-proposing GS-algorithms. Then, $\forall\ \mu \in \mathcal{S}(M, W, \succcurlyeq)$, $m \in M$, $w \in W$,

$$\mu_M(m) \succcurlyeq_m \mu(m) \succcurlyeq_m \mu_W(m) \quad \& \quad \mu_W(w) \succcurlyeq_w \mu(w) \succcurlyeq_w \mu_M(w).$$

- Proof:
  - Second, we show that μ_M is the worst matching for women.
  - Suppose not: $\exists\ w \in W$ s.th. $\mu_M(w) \succ_w \mu(w)$ for some $\mu \in \mathcal{S}$
  - We know that $w \succ_{\mu_M(w)} \mu(\mu_M(w))$; that is, whoever is matched with w under μ_M, $\mu_M(w)$, ranks w above whoever they are matched with in any other stable matching, in particular, μ.

  $\Rightarrow$ $\exists\ m \in M$ s.th. $w \succ_m \mu(m)$ and $m \succ_w \mu(w)$.

  $\Rightarrow$ (m, w) are a blocking pair for μ, a contradiction.

Opposition of interests

- This notion of opposite interests goes beyond the **extremal matchings** μ_M and μ_W
- Define the binary relations $\succ_M$ and $\succ_W$ over $\mathcal{M}(M, W)$:
  - $\mu \succ_M \mu'$ if $\mu(m) \succcurlyeq_m \mu'(m)$ for all $m \in M$;
  - $\mu \succ_W \mu'$ if $\mu(w) \succcurlyeq_w \mu'(w)$ for all $w \in W$.
- **Theorem (Knuth, 1976)**: If μ and μ' are stable matchings, then $\mu \succ_M \mu'$ if and only if $\mu' \succ_W \mu$.
- Proof:
  - Let $\mu, \mu' \in \mathcal{S}$ be s.th. $\mu \succ_M \mu'$. Towards a contradiction: suppose not $\mu' \succ_W \mu$

  $\Rightarrow$ $\exists\ w \in W$ s.th. $\mu(w) \succ_w \mu'(w)$
  - Then, man $m = \mu(w)$ is matched to another woman under μ', which he prefers less to w (since $\mu \succ_M \mu'$).

  $\Rightarrow$ $m \succ_w \mu'(w)$ and $w \succ_m \mu'(m)$

  $\Rightarrow$ (m, w) block μ', contradiction.

Lattice Structure

- The above results suggest that the set of stable matchings $\mathcal{S}$ can be ordered in some way
- For any two matchings μ, μ', define the **join** of μ and μ' as the matching $\mu \vee_M \mu'$ such that, for every $m \in M$ and $w \in W$,

$$\mu \vee_M \mu'(m) = \begin{cases} \mu(m) & \text{if } \mu(m) \succ_m \mu'(m) \\ \mu'(m) & \text{if } \mu'(m) \succ_m \mu(m) \end{cases} \quad \& \quad \mu \vee_M \mu'(w) = \begin{cases} \mu'(w) & \text{if } \mu(w) \succ_w \mu'(w) \\ \mu(m) & \text{if } \mu'(w) \succ_w \mu(w) \end{cases}$$

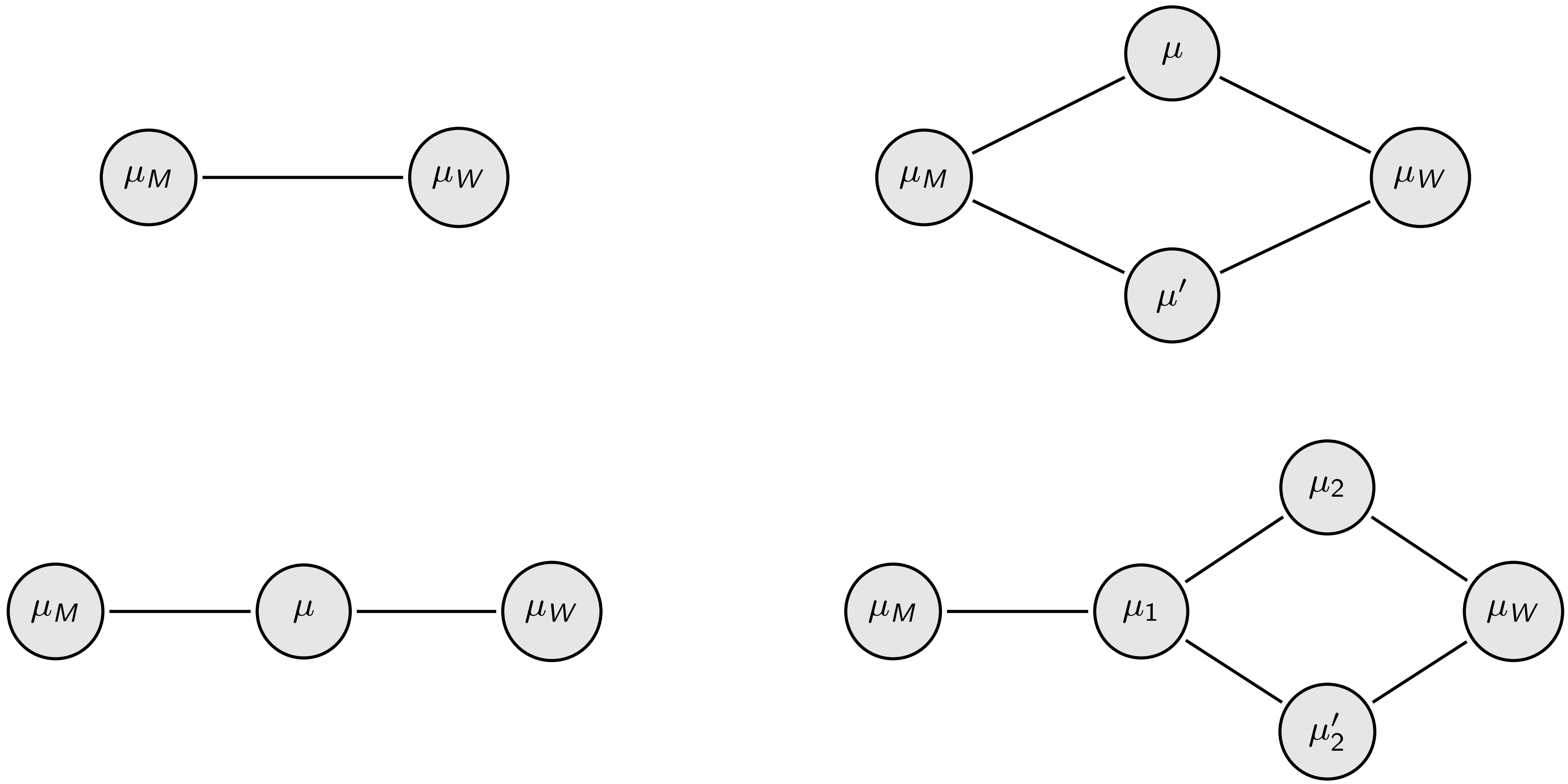
- Note: $\mu \vee_M \mu'(i)$ stands for the agent matched with i in matching $\mu \vee_M \mu'$.
- Define the **meet** of μ and μ' as the matching $\mu \wedge_M \mu'$ such that, for every $m \in M$ and $w \in W$,

$$\mu \wedge_M \mu'(m) = \begin{cases} \mu'(m) & \text{if } \mu(m) \succ_m \mu'(m) \\ \mu(m) & \text{if } \mu'(m) \succ_m \mu(m) \end{cases} \quad \& \quad \mu \wedge_M \mu'(w) = \begin{cases} \mu(w) & \text{if } \mu(w) \succ_w \mu'(w) \\ \mu'(m) & \text{if } \mu'(w) \succ_w \mu(w) \end{cases}$$

Lattice Structure

- **Theorem (Conway)**: If μ and μ' are stable matchings, then both $\mu \vee_M \mu'$ and $\mu \wedge_M \mu'$ are stable matchings.
- Proof: omitted (read the lecture notes)
- **Intuition**:
  - The set of stable matchings $\mathcal{S}$ is a **lattice**
  - Lattice = partially ordered set X in which every two-element subset $\{x, y\} \subseteq X$ has a "join" $x \vee y$ and a "meet" $x \wedge y$, both of them elements in X.
  - See the lecture notes for a formal definition, but it is very simple to illustrate . . .

Examples of Lattice Structures



Stable matchings as fixed points

- Elegant characterization of stable matchings that yields existence and lattice structure via a **fixed-point** argument, due to Adachi (2000)
- **Idea**:
  - Define **prematchings**: same as matchings but two people can be matched with the same agent (i.e., prematchings need not be "reciprocal")
  - For prematching (or "fantasy") ν define the sets

$$A(m,\nu) = \{w \in W : m \succcurlyeq_w \nu(w)\} \quad \& \quad A(w,\nu) = \{m \in M : w \succcurlyeq_m \nu(m)\}.$$

  - $A(i,\nu)$ is the set of agents that find i acceptable given their current "partner" under ν
  - Define mapping $\nu \mapsto T\nu$ as follows:
    - $(T\nu)(m) =$ most preferred woman in $A(m,\nu) \cup \{m\}$
    - $(T\nu)(w) =$ most preferred man in $A(w,\nu) \cup \{w\}$

Stable matchings as fixed points

- **Idea (cont'd)**:
  - Define $\leqslant$ as follows: $\nu \leqslant \nu'$ if

$$(\forall m \in M)\, \nu'(m) \succcurlyeq_m \nu(m) \quad \& \quad (\forall w \in W)\, \nu(w) \succcurlyeq_w \nu'(w).$$

  - **Lemma 1**: T is monotone increasing, $\nu \leqslant \nu'$ implies $T\nu \leqslant T\nu'$.
  - **Lemma 2**: A matching μ is stable if and only if it is a fixed point of T.
  - **Theorem (Tarski)**: The set of fixed points of a monotone function on a lattice is a nonempty and complete lattice.
  - **Note**: this provides an additional algorithm to find extremal stable matchings:
    - Let $\bar{\nu}$ be a prematching where $\bar{\nu}(w) = w$ for all w and $\nu(m)$ is the best alternative in $W \cup \{m\}$ for all m
    - Define $\nu^0 = \bar{\nu}$ and $\nu^{k+1} = T\nu^k$
    - Then, there is $K < \infty$ s.th. $\mu = \nu^K$ is a stable matching (indeed, it is μ_M from the def. of $\leqslant$)
    - The result follows from the monotonicity of T: $T(T\bar{\nu}) \leqslant T\bar{\nu} \leqslant \bar{\nu}$ implies $\cdots \nu^{k+1} \leqslant \nu^k \leqslant \cdots \leqslant \bar{\nu}$.
    - The sequence is decreasing and the set of prematchings is finite.

Incentives in the marriage market

- **Question**: Is the Gale-Shapley algorithm strategy-proof?
- **Answer**: No! Actually, there exists no strategy-proof mechanism that always generate stable matchings (but there are strategy-proof mech's that always generate Pareto efficient matchings).
- **Remark**: Serial dictatorship is strategy-proof and Pareto efficient
- Proof: try it as an exercise (hint: ignore one side of the market, treat it as if they were "objects," and run SD with the preferences of the other side)

Incentives in the marriage market

- **Theorem**: There is no mechanism that is stable and strategy-proof.
- Proof:
  - $M = \{m_1, m_2\}$ and $W = \{w_1, w_2\}$
  - Let $\succcurlyeq_{m_i}$ rank w_i over w_{3-i} over m_i. Let $\succcurlyeq_{w_i}$ rank m_{3-i} over m_i over w_i
  - Two stable matchings: $\mu_M(m_i) = w_i$ for $i = 1, 2$, and $\mu_W(m_i) = w_{3-i}$ for $i = 1, 2$
  - Let ϕ be a stable mechanism. Then, $\phi(\succcurlyeq)$ must coincide with either μ_M or μ_W. Say wlog that it coincides with μ_M.
  - Consider the preference $\succcurlyeq'_{w_1}$ that ranks m_2 over w_1 over m_1; thus making m_1 unacceptable.
  - There is a single stable matching in $(M, W, (\succcurlyeq_{m_1}, \succcurlyeq_{m_2}, \succcurlyeq'_{w_1}, \succcurlyeq_{w_2}))$ and it coincides with μ_W.
  - So ϕ is not strategy-proof as $m_2 = \mu_W(w_1) \succ_{w_1} \mu_M(w_1) = m_1$.

Incentives in the marriage market

- There is a silver lining!
- **Theorem**: The men-proposing Gale-Shapley mechanism is (group) strategy-proof for the men.
- Proof: omitted (read in Chapter 4 of Roth and Sotomayor, 1990)
- **Intuition**:
  - Men are already "going down" their preference list
  - Even if m misreports his preference by ranking a woman w who is better than $\mu_M(m)$, note that m is already being rejected by w in the course of the GS-algorithm.
  - The result is stronger since it implies that not even coalitions of men can obtain a better matching than μ_M by coordinating the way in which they misreport their preferences.

Matching with transferable utility

- The Assignment Game of Shapley and Shubik (1971)
- (B, S, α) = **matching market with transfers**
  - B = set of **buyers** (finite)
  - S = set of **sellers** (finite and disjoint of B)
  - $\alpha = (\alpha_{ij})_{i \in B, j \in S}$ is the **surplus matrix**
  - α_{ij} = surplus generated from match (i, j)
  - **Idea**: if (i, j) are matched they share α_{ij} (split it among themselves with a transfer)
- Final **utilities**:
  - u_i = utility of i
  - v_j = utility of j
  - if (i, j) are matched, then $u_i + v_j = \alpha_{ij}$

Matching with transferable utility

- **Matching**: matrix $x = (x_{ij})_{i \in B, j \in S}$ such that $x_{ij} \geqslant 0$, and for all $(i,j) \in B \times S$,

$$\sum_{s \in S} x_{is} \leqslant 1 \quad \text{and} \quad \sum_{b \in B} x_{bj} \leqslant 1.$$

- $x_{ij} = 1$ means that (i,j) are matched
- In principle, x_{ij} may be in $(0,1)$
  - Can interpret as the probability of (i,j) being matched (but, as we'll see, this will not play any role)
- **Assignment**: pair of vectors $u = (u_i)_{i \in B}$ and $v = (v_j)_{j \in S}$ such that $u_i \geqslant 0$, $v_j \geqslant 0$ and there exists a matching x satisfying

$$\sum_{i \in B} u_i + \sum_{j \in S} v_j = \sum_{i \in B, j \in S} \alpha_{ij} x_{ij}.$$

- In such case, say that matching x **supports** assignment (u,v)
- **Intuition**: an assignment is a redistribution of the total surplus generated by a matching

Matching with transferable utility

- Matching x is **efficient** it maximizes the total surplus in the economy, i.e., if it solves the linear program:

$$\max_{x_{ij} \geqslant 0} \sum_{i \in B, j \in S} x_{ij} \alpha_{ij} \quad \text{subject to:} \quad \begin{aligned} \sum_{j \in S} x_{ij} &\leqslant 1 \quad \forall i \in B \\ \sum_{i \in B} x_{ij} &\leqslant 1 \quad \forall j \in S. \end{aligned} \tag{$\star$}$$

- Assignment (u, v) is in the **core** if $u_i + v_j \geqslant \alpha_{ij}$ for every $(i, j) \in B \times S$.
- **Intuition**: If $u_i + v_j < \alpha_{ij}$, then (i, j) can **block** assignment (u, v) by trading amongst themselves and sharing α_{ij}.
- **Note**: a **feasible** assignment is in the core if and only if

$$u_i = \max \{\alpha_{is} - v_s : s \in S\}, \quad \text{and} \quad v_j = \max \{\alpha_{bj} - u_b : b \in B\}.$$

Efficiency and the core

- **Question 1**: What is the relationship between efficient matchings and core assignments?
- **Question 2**: Are core assignments supported by efficient matchings?
- **Answer**: Yes! There is a **duality** between efficient matchings and core assignments, in the sense that one implies the other.
- **Theorem (Shapley and Shubik, 1971)**: For every efficient matching x, there exists a core assignment (u, v) such that $\sum_i u_i + \sum_j v_j = \sum_{i,j} \alpha_{ij} x_{ij}$. Likewise, every core assignment is supported by an efficient matching.

Efficiency and the core

- **Theorem (Shapley and Shubik, 1971)**: For every efficient matching x, there exists a core assignment (u, v) such that $\sum_i u_i + \sum_j v_j = \sum_{i,j} \alpha_{ij} x_{ij}$. Likewise, every core assignment is supported by an efficient matching.
- Sketch of proof:
  - Proof relies on linear programming duality
  - $(\star)$ is a linear program
  - Its dual characterizes core assignments: the vectors (u, v) are the Lagrangean multipliers associated to the constraints $\sum_{j \in S} x_{ij} \leqslant 1$ and $\sum_{i \in B} x_{ij} \leqslant 1$.
  - Dual problem:

$$\min_{u_i, v_j \geqslant 0} \sum_{i \in B} u_i + \sum_{j \in S} v_j \quad \text{subject to} \quad u_i + v_j \geqslant \alpha_{ij} \quad \forall (i, j) \in B \times S \tag{$\star\star$}$$

  - The Lagrangean multipliers of $(\star\star)$ are given by x, the solution to $(\star)$

Efficiency and the core

- **Theorem (Shapley and Shubik, 1971)**: For every efficient matching x, there exists a core assignment (u, v) such that $\sum_i u_i + \sum_j v_j = \sum_{i,j} \alpha_{ij} x_{ij}$. Likewise, every core assignment is supported by an efficient matching.
- Sketch of proof:
  - By duality, the value of the objective function at the optimal solution in both the primal $(\star)$ and the dual $(\star\star)$ is the same.
  - That is, for every efficient matching x there exists a core assignment (u, v) such that

$$\sum_{i \in B} u_i + \sum_{j \in S} v_j = \sum_{i \in B, j \in S} x_{ij} \alpha_{ij}.$$

  - Finally, note that for the same reason every feasible (u', v') in the dual (i.e., any core assignment), must be supported by an efficient matching.

Efficiency and the core

- **Remark**: The complementary slackness conditions of the primal problem $(\star)$ imply that there exists an efficient matching x such that

$$x_{ij} \in \{0,1\} \quad \text{and} \quad x_{ij} = 1 \quad \Rightarrow \quad u_i + v_j = \alpha_{ij}.$$

- No need to worry about “fractional” matchings

Structure of the core

- Notably, the core has also the "opposition of interests" property (i.e., it is a lattice)
- **Theorem**: Let (u, v) and (u', v') be assignments in the core. Let $\bar{u}_i = \max\{u_i, u'_i\}$ and $\underline{v}_j = \min\{v_j, v'_j\}$. Then $(\bar{u}, \underline{v})$ is an assignment in the core.
- Proof: omitted (read in lecture notes)
- **Corollary**: There exists core assignments (u^*, v_*) and (u_*, v^*) such that for any core assignment (u, v), for every $i \in B$ and $j \in S$,

$$u_i^* \geqslant u_i \geqslant u_{*i}$$

$$v_j^* \geqslant v_j \geqslant v_{*i}$$

- Think of (u^*, v_*) and (u_*, v^*) as core assignments with minimal and, respectively maximal, prices. They are the buyer- and seller-optimal core assignments.

Week 7: The medical match

BEEM147: Topics in Microeconomic Theory II

Matching and Market Design

Spring Term 2021

University of Exeter

Overview

- Last week
  - Two-sided matching: marriage market
  - Stability and efficiency: Gale-Shapley algorithm
  - Structure of set of stable matchings
  - Incentives in the marriage market
  - Matching with transferable utility
- This week
  - The medical match
  - A brief history of *unraveling* and *jumping the gun*
  - The NIMP algorithm and its trial-run

The medical match

- Every year, more than 40,000 medical residents are assigned to more than 30,000 residency programs in the U.S.
- Assignment done through a centralized matching clearinghouse
  - National Resident Matching Program (NRMP), a.k.a. "The Match"
- The success of this sytem has prompted similar ones in Canda and the U.K.
- Key lessons from the history of the NRMP:
  (a) the importance of **stability** as a condition for the survival of an institutional design
  (b) real-life markets are complicated: sometimes they work and sometimes they don't
  (c) economists have a great-deal to learn from real-world markets

History of the NRMP

- Up to 1945, market of medical interns was unregulated:
  - Recently graduated interns applied and interviewed for positions at hospitals
  - Hospitals aimed to attract the most promising candidates
- First decades of the 20th century: the market **unraveled**
- In 1945, the Association of American Medical Colleges (AAMC) decided to step in
  - 1945–1952: experimented with several **rules** and **regulations**
- In 1952, they settled on a centralized matching clearinghouse: the **NIMP algorithm**
- This system was seen as a great success. It stayed in place until the mid-1990s.
- Prompted by calls for reform, in the mid-1990s, a group of economists led by Alvin Roth, **redesigned** certain aspects of the system.
  - The main building blocks of the system remain the same as they were in the 1950s

A history of unraveling

- Prior to 1945, hospitals competed in an **“arms race”** for the best graduates
- Hospitals offered **binding agreements** prior to graduation to lock them into their programs.
- Hospitals undercut each other agreements
- Their **dates** became earlier and earlier as years went by
- By the 1940s, the market had clearly **unraveled**
- The typical student would sign a binding agreement **two years** before graduating
- **Inefficient for students**: do not know preferences, not enough information
- **Inefficient for hospitals**: high uncertainty

Jumping the gun: exploding offers post-1945

- On 1945, the AAMC stepped in
- They **regulated the date** at which medical schools released students records
- Hospitals responded by making **"exploding offfers"**: shortly after the deadline, they would make offers that expired after very few days
- Goal: lock in students before another preferred program made them an offer
- Students face tough decisions:
  - Play it **"safe"**: could exit the market too soon, a better program might offer later
  - **"Risk"** it: could end up letting go the best offer
  - Either way: highly likely students end up matching with a less preferred program while a better one has an opening for them (unstable outcomes!)

Reform Towards the NIMP (aka. The Match)

- 1845–1950: AAMC **experimented** with the minimum date at which offers could be made, and the minimum time hospitals had to give students to decide. Process was very chaotic.
- On 1951, AAMC came up with a **centralized matching clearinghouse**
- **NIMP** = National Intern Matching Program
  - Students applied and held **interviews** with hospitals
  - Then, hospitals and students submitted a **rankings** of each other to the NIMP
  - The final allocation was determined through a **matching algorithm**
- On 1951, they did a trial run (aka. **NIMP trial-run**)
- Worked out "well enough," but there were some **complaints**
- On 1952, they adjusted the **NIMP algorithm** and used it
- Importantly, the system was **voluntary**

Redesigning the NRMP

- NIMP algorithm **stayed in place** until the 1990s (high participation rates)
- Eventually it became the **NRMP**
- On the 1990s, students called for **reform**
- They argued that the system could be **"gamed"**
- Complaints about how the algorithm allocated **couples** (which became more common)
- In the 1990s, group of **economists redesigned** some aspects of the system
  - But the main building blocks remained in place
  - It was noted that the interview process was a key part for the mechanism to work properly (reduces the number of potential blocking partners of each agent)

Key lessons

- The pre-1952 market **failed** to deliver stable outcomes
- Main strength of NIMP algorithm: **stability** (unlike trial-run)
- Necessary condition for **voluntary system to survive test of time**: stability
- The process through which the **institutional design** of this market "converged" to a stable mechanism was long, chaotic, and idiosyncratic.
  - Why then and not before or after?
  - Was it inevitable? Obvious? By chance? Seems hard to know
- **Non-economists** out there know a lot about how real-life markets actually work
- AAMC **administrators** had been designing NIMP for more than a decade by the time Gale and Shapley came by in 1962 (A. Roth was born in 1951!)

One-to-one medical match

- Same setup as marriage market, but with different labels
- $(S, H, \succcurlyeq)$ = **one-to-one medical match**
  - $s_j \in S$ = students
  - $h_i \in H$ = hospitals
  - $\succcurlyeq$: linear orders, one for each agent
- Note 1: Read Chapters 5 and 6 of Roth and Sotomayor (1990) to see how to extend this to many-to-one (one hospital hires several students)
- Note 2: We will see some many-to-one matching next week in school choice

NIMP trial-run

- Students submit ranking of hospitals
- Hospitals rank students into 5 groups in order of preference: rank 1, . . . , rank 5.
- Proceed in stages:
  - 1:1 stage. Students and hospitals are matched if they give each other a rank of 1.
  - 1:2 stage. The remaining students and hospitals are matched if the student has ranked the hospital 1 and the hospital has ranked the student 2.
  - 2:1 stage. Among the remaining students and hospitals, match students who ranked hospitals 2, and hospitals who ranked students 1.
  - 2:2 stage. . . . , followed by 1:3 stage, and so forth.

NIMP trial-run

- **Proposition**: The NIMP trial-run algorithm is not stable, nor strategy-proof for students.
- Proof:
  - Consider 4-by-4 example:

$$s_1 : h_1, h_2, h_3 \qquad s_2 : h_2, h_3, h_1 \qquad s_3 : h_1, h_3, h_2$$

$$h_1 : s_2, s_3, s_1 \qquad h_2 : s_1, s_2, s_3 \qquad h_3 : s_3, s_2, s_1$$

  - 1:1 stage: no matches
  - 1:2 stage: match (s_2, h_2) and (s_3, h_1)
  - Finally, match (s_1, h_3)
  - First, note that s_1 and h_2 form a blocking pair $\rightarrow$ not stable
  - Second, if s_1 ranks h_2 as their top choice (above h_1), (s_1, h_2) matched in the 1:1 stage.
  - Hence, s_1 has incentives to misreport $\rightarrow$ not strategy-proof for students

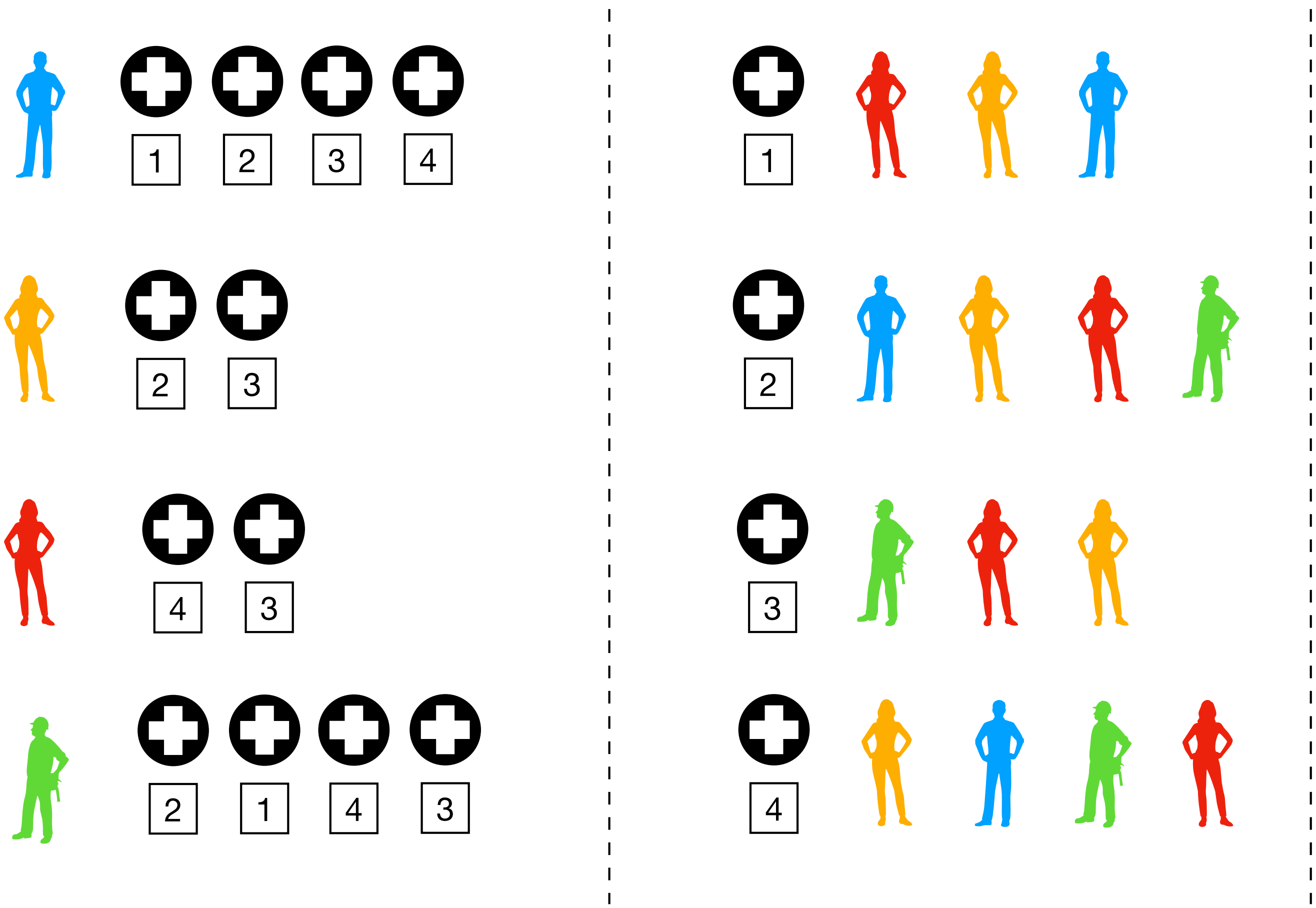
NIMP algorithm

- Students/hospitals submit ranking of hospitals/students (indicating unacceptable ones)
- From the list of s_j (resp. h_i), delete hospitals (resp. students) who find s_j (resp. h_i) unacceptable
- No one is tentatively matched. Proceed in stages:
  - **1:1 step**. Check if there are students and hospitals who rank each other as their top choice and are not tentatively matched. If no matches found, proceed to the 2:1 step. If found, proceed to tentative-assignment-and-update phase.
  - k**:1 step**. Check if there are students and hospitals such that: the student ranks the hospital as their k-th choice, the hospital ranks the student as their top choice, and they are not tentatively matched. If no matches found, proceed to the k+1:1 step. If found, proceed to the tentative-assignment-and-update phase.

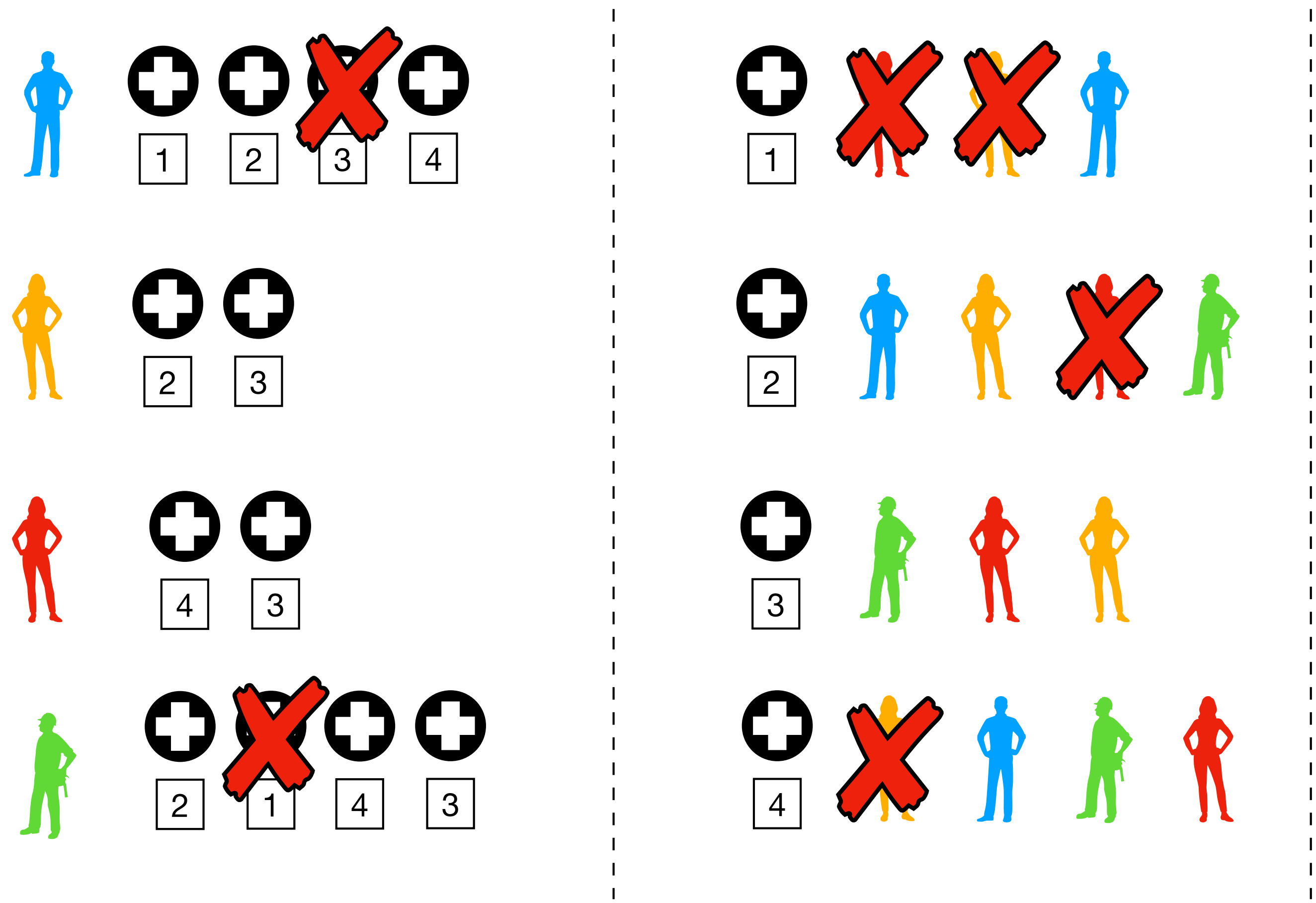
NIMP algorithm (cont'd)

- Students/hospitals submit ranking of hospitals/students (indicating unacceptable ones)
- From the list of s_j (resp. h_i), delete hospitals (resp. students) who find s_j (resp. h_i) unacceptable
- No one is tentatively matched. Proceed in stages:
  - **Tentative-assignment-and-update phase**. Assume algorithm entered phase from the k:1 step. Assign tentatively all k:1 matches. New matches replace previous ones. Update the rankings as follows:
    - From the ranking of student s_j, delete h_i if it is ranked lower than the current match of s_j. (If s_j is tentatively matched to their k-th choice, their ranking now only includes their first k choices.)
    - From the ranking of hospital h_i, delete s_j if h_i was just deleted from the ranking of s_j. (List of h_i only includes students who have not been tentatively assigned to a hospital they prefer over h_i.)
    - After updating the ranking lists, return to the 1:1 step.
  - **Terminate** the algorithm when no new tentative matches can be found, at which point the current tentative matches become final.

Example: NIMP and Gale-Shapley

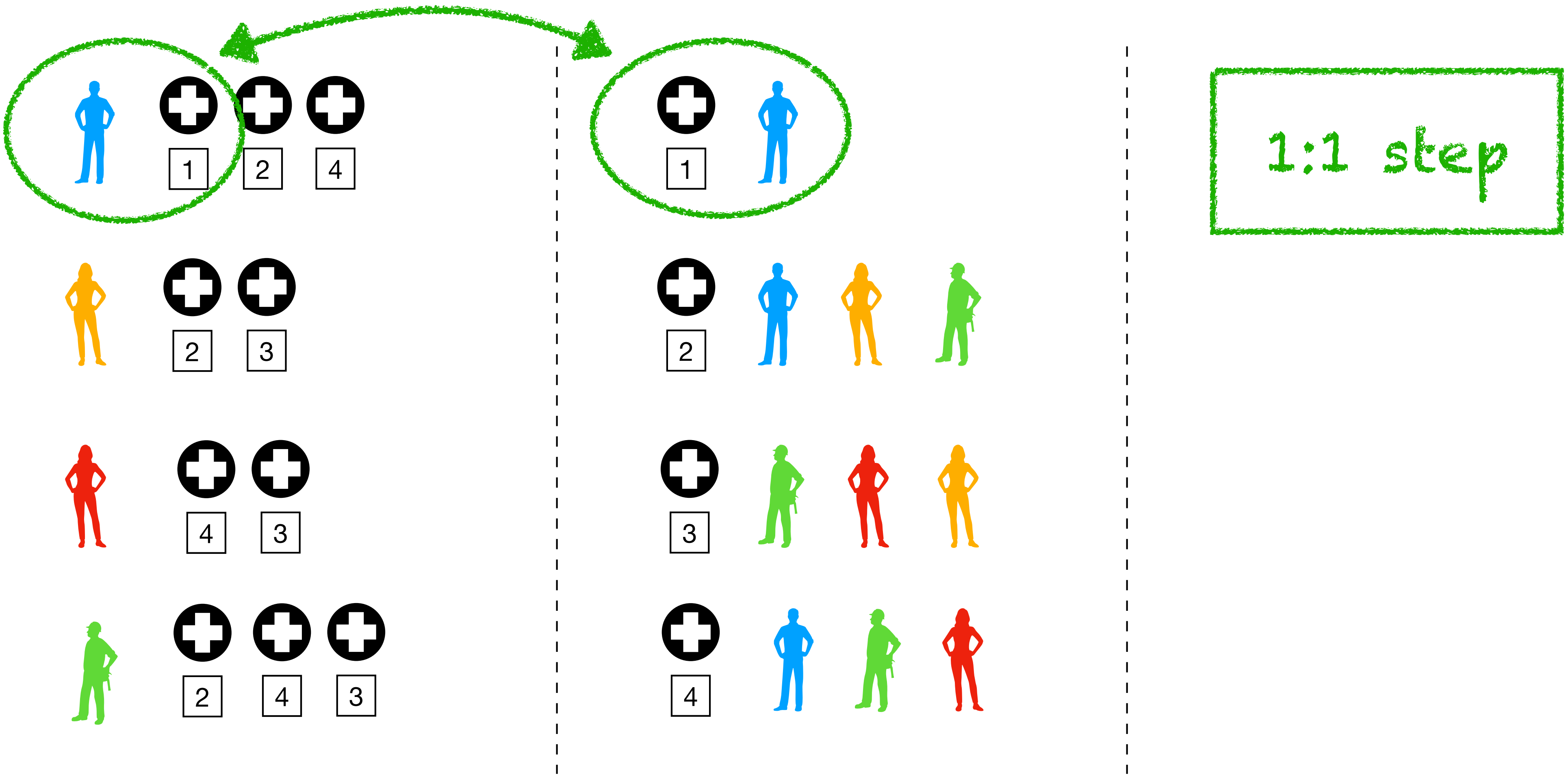


Example: NIMP and Gale-Shapley

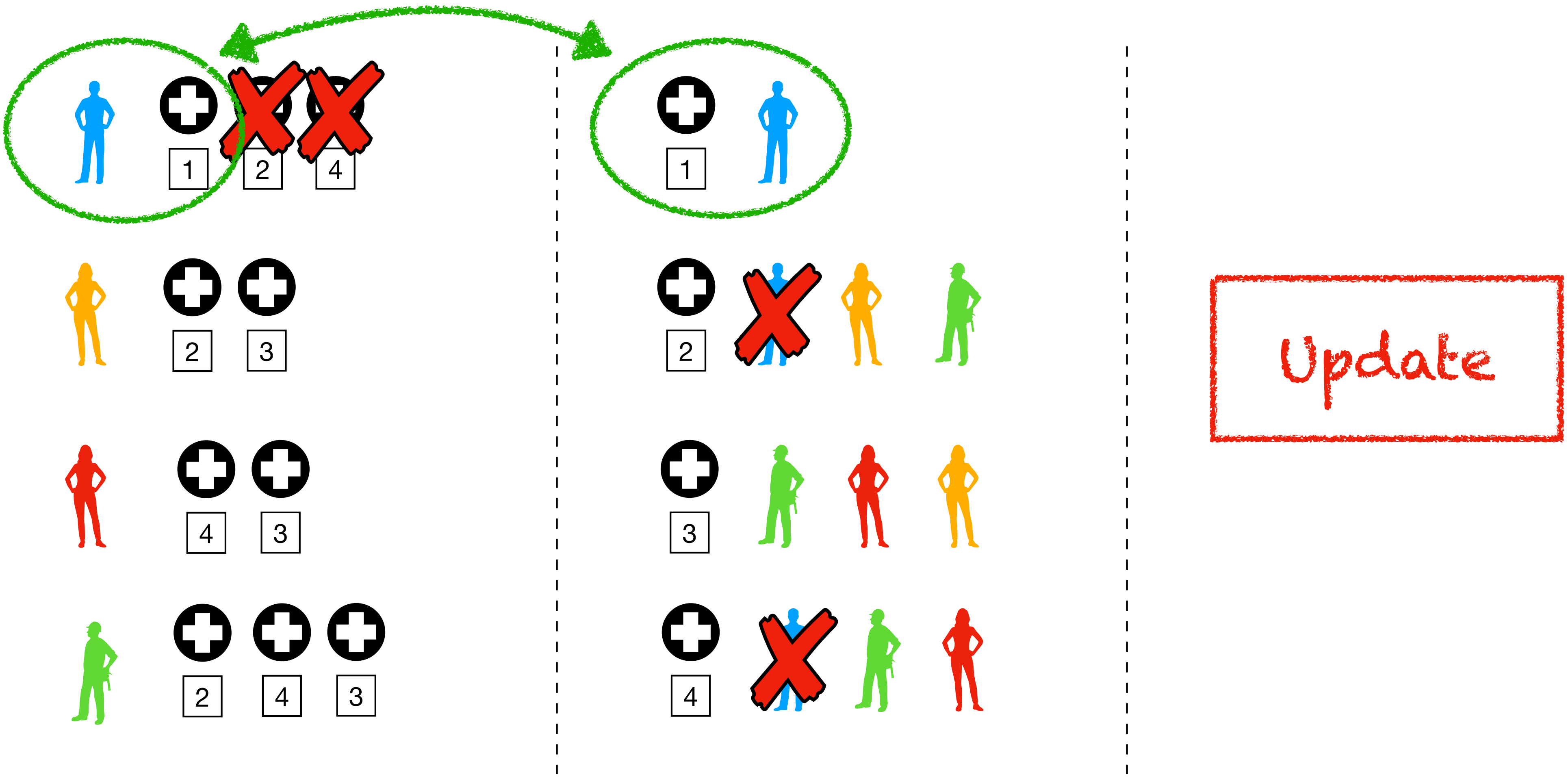


FIRST STEP: Remove those who find me unacceptable

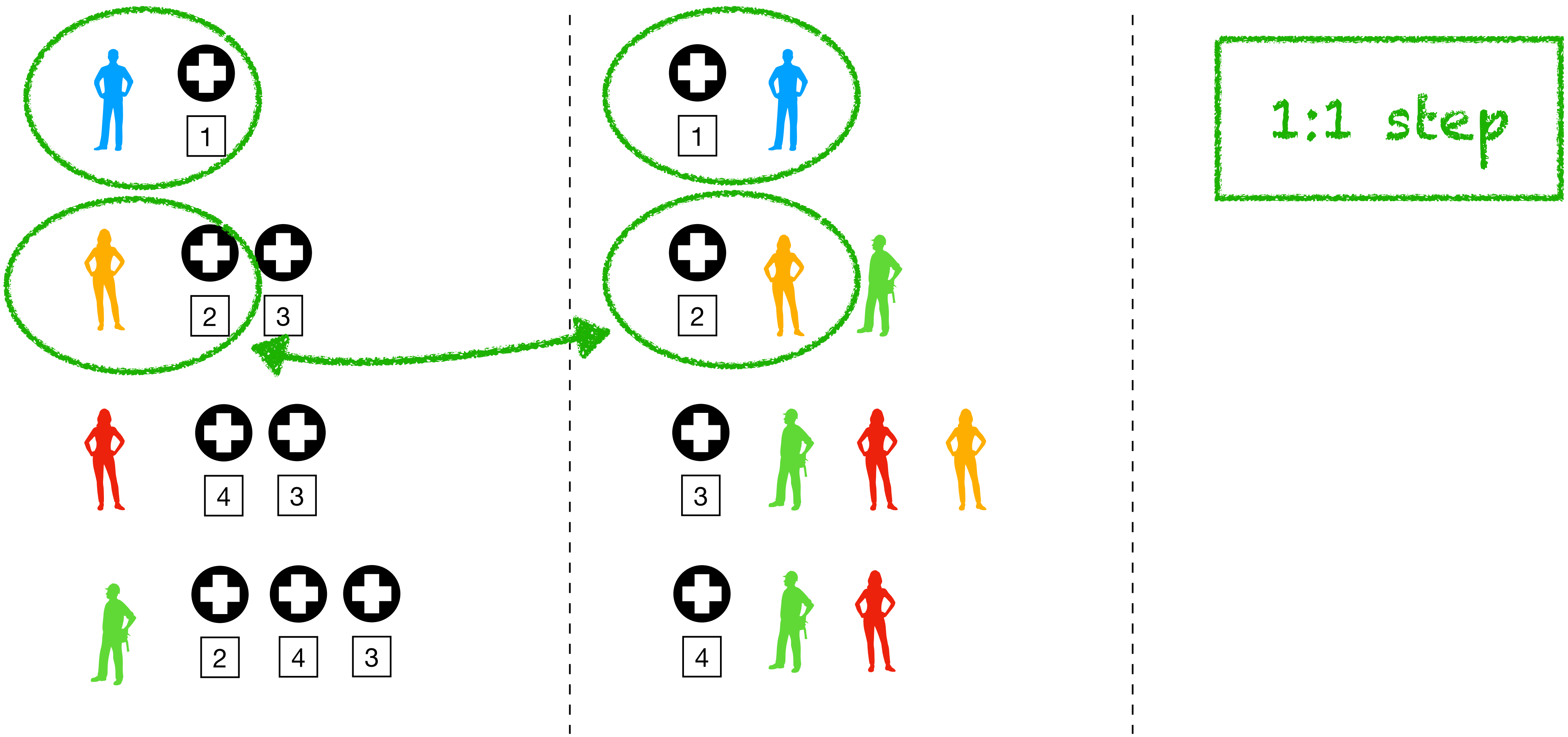
Example: NIMP and Gale-Shapley



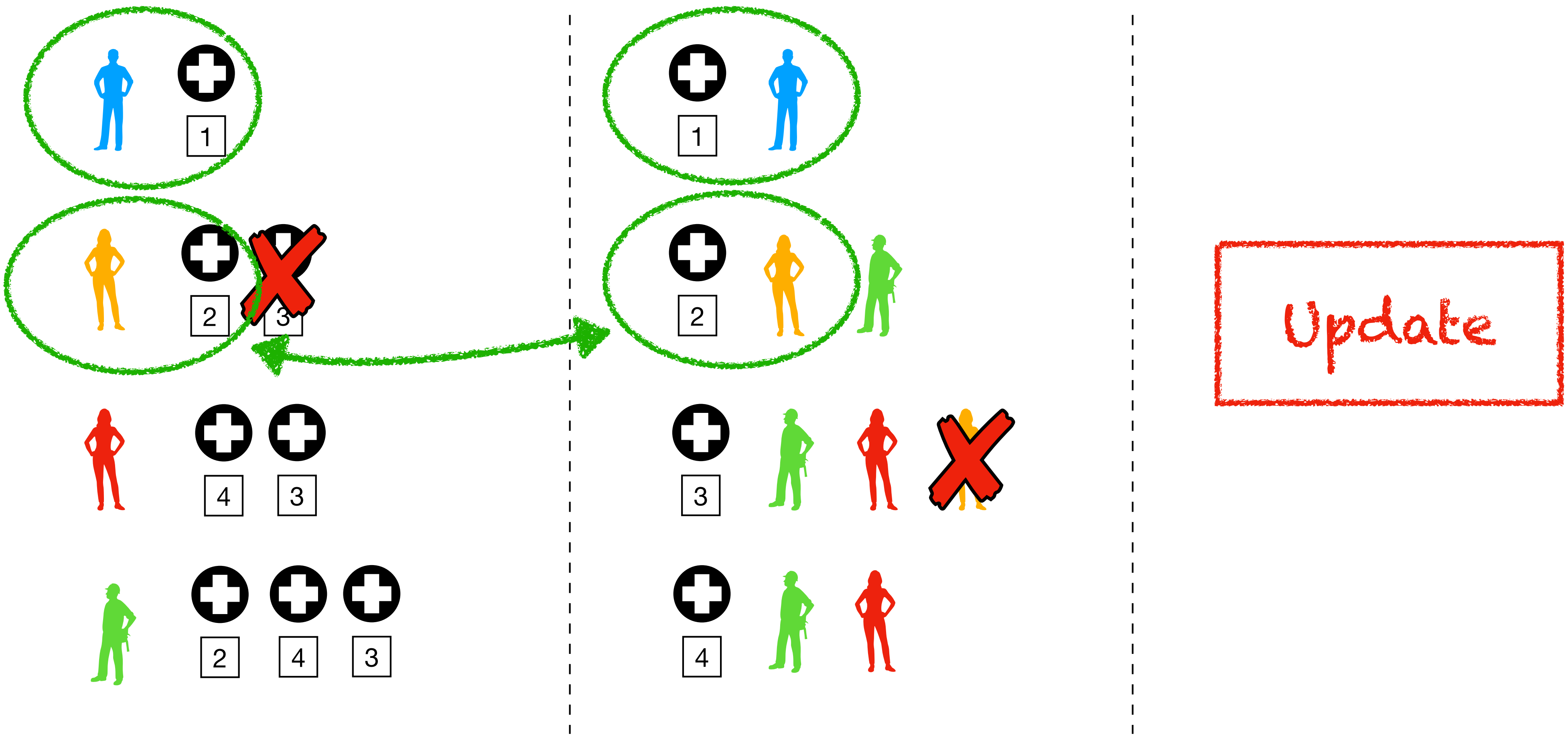
Example: NIMP and Gale-Shapley



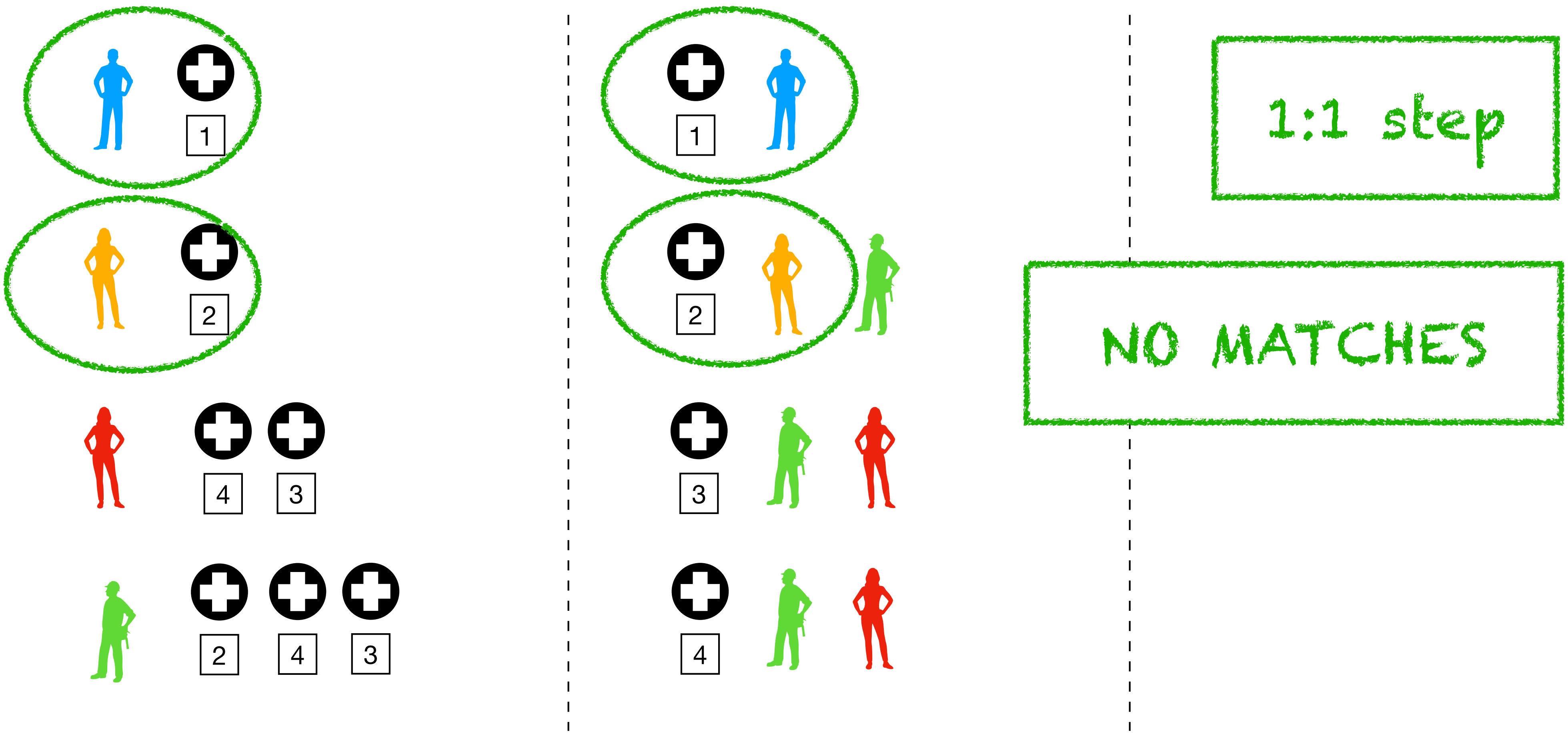
Example: NIMP and Gale-Shapley



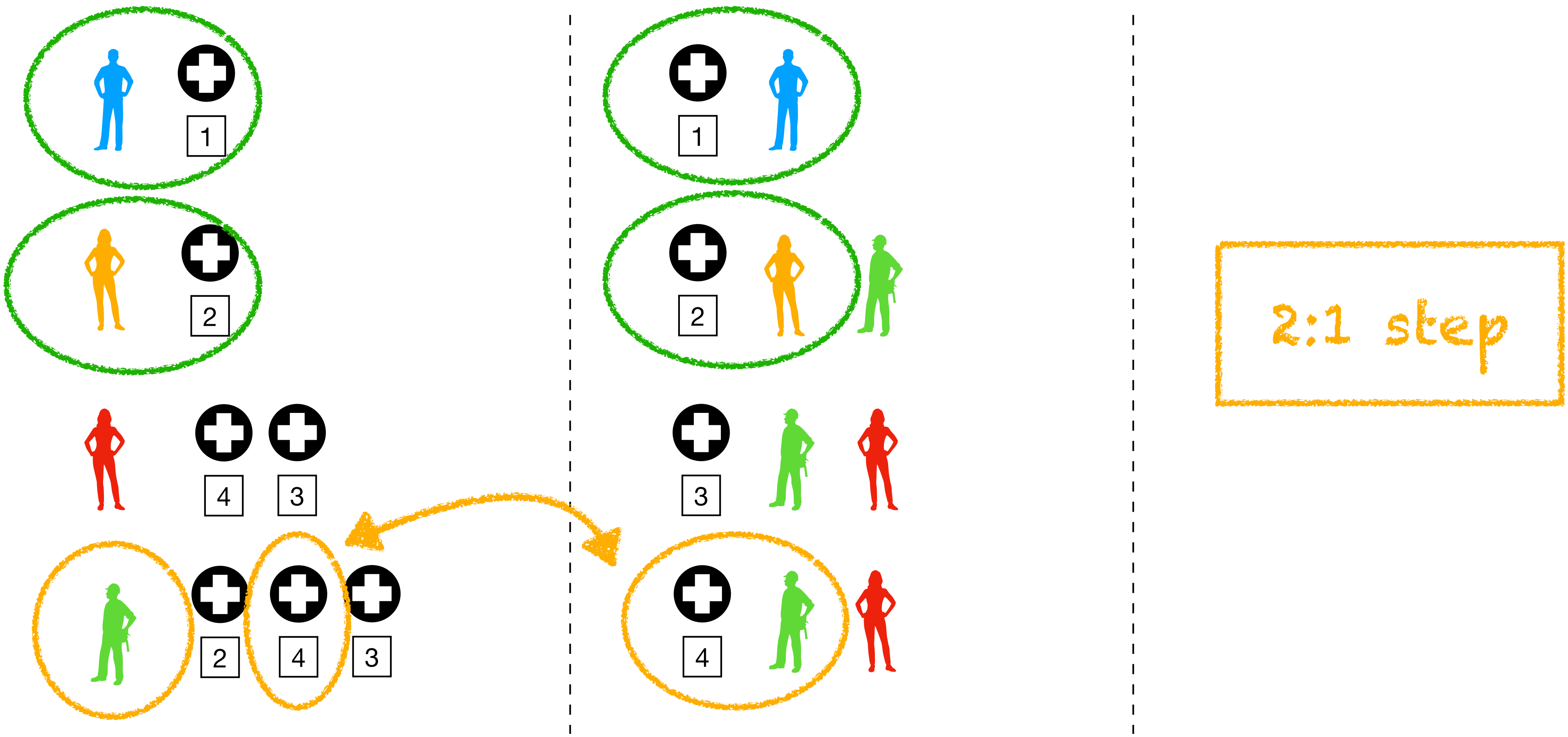
Example: NIMP and Gale-Shapley



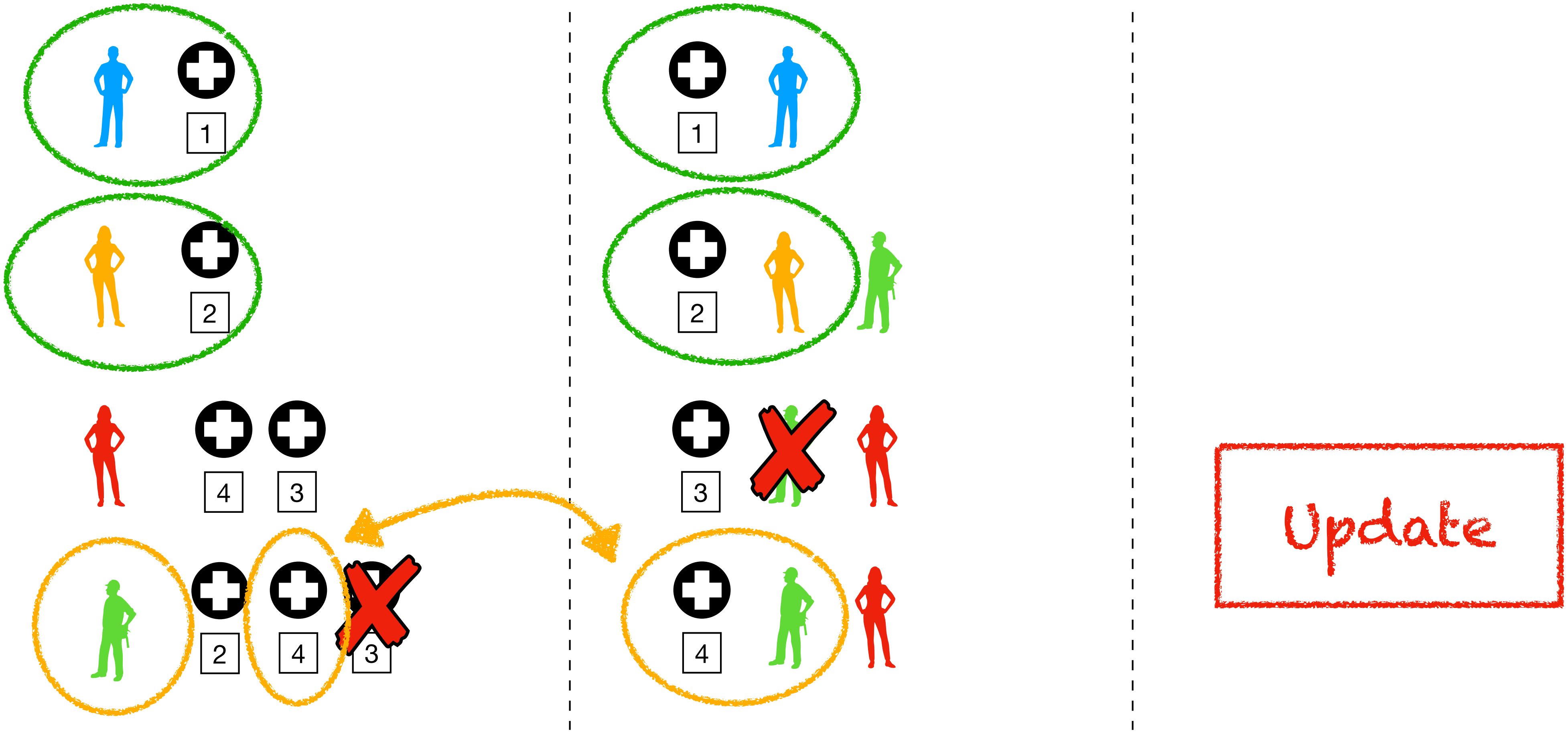
Example: NIMP and Gale-Shapley



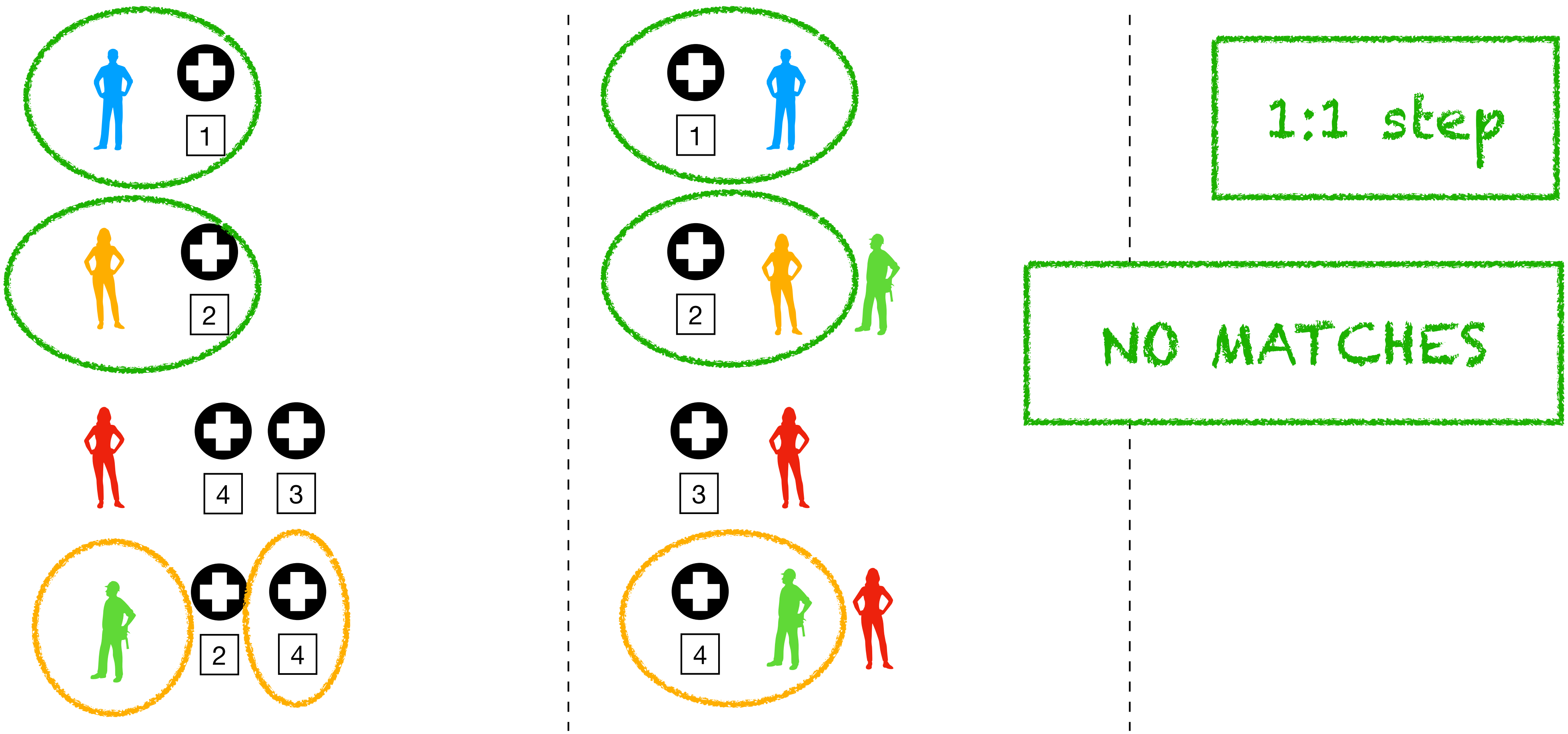
Example: NIMP and Gale-Shapley



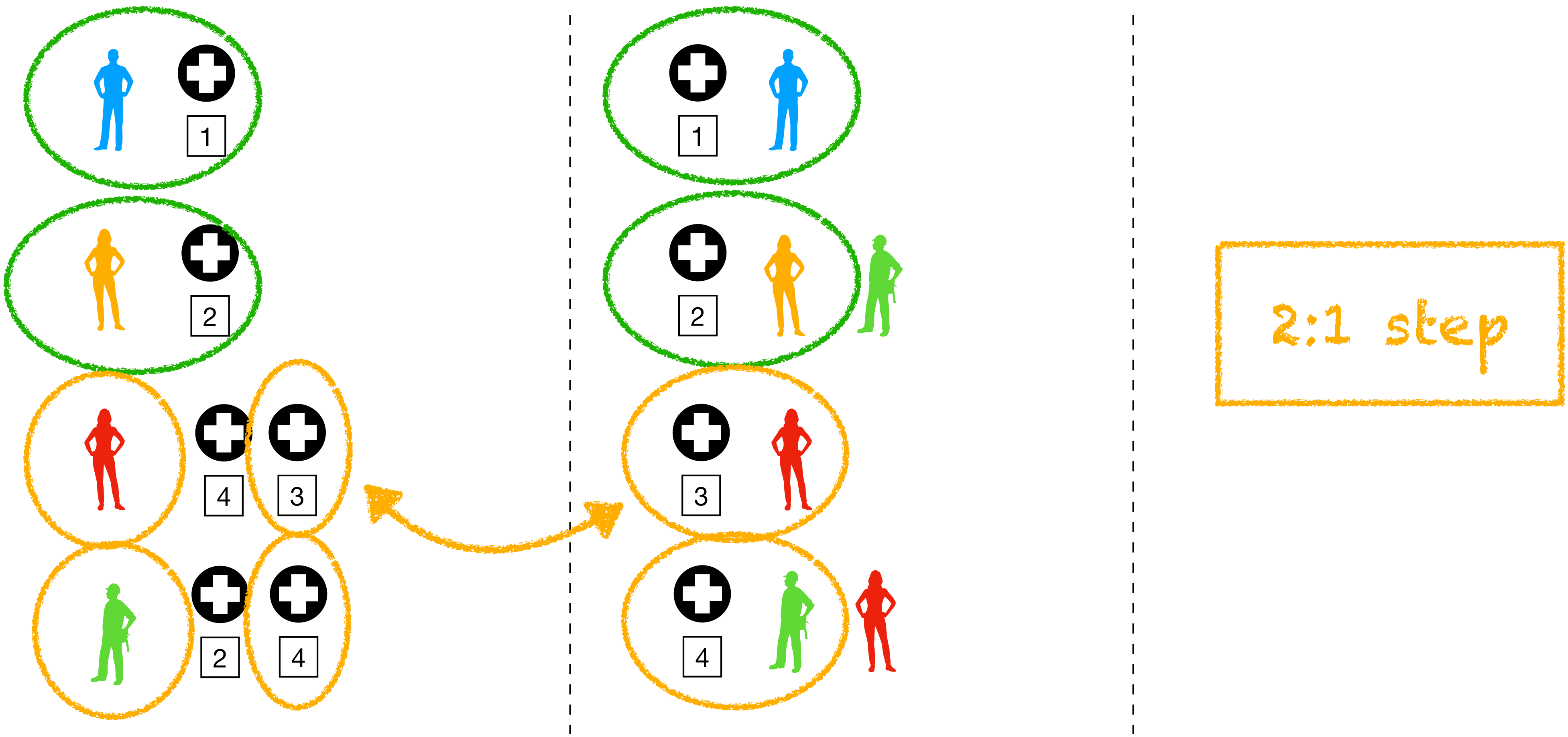
Example: NIMP and Gale-Shapley



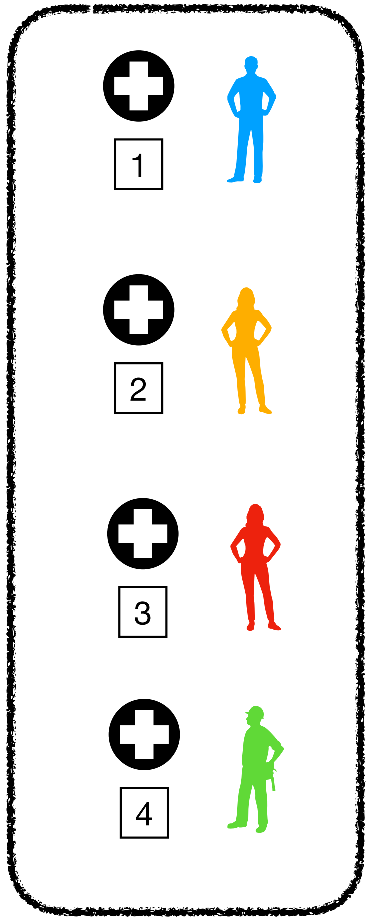
Example: NIMP and Gale-Shapley



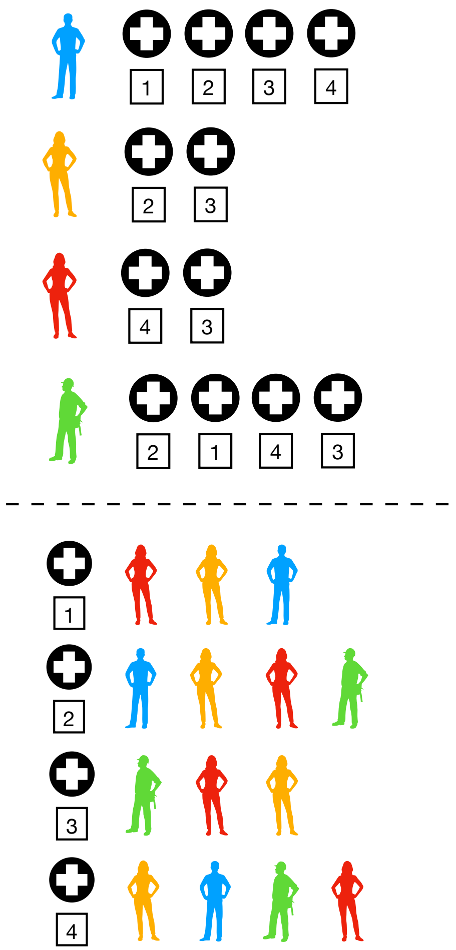
Example: NIMP and Gale-Shapley

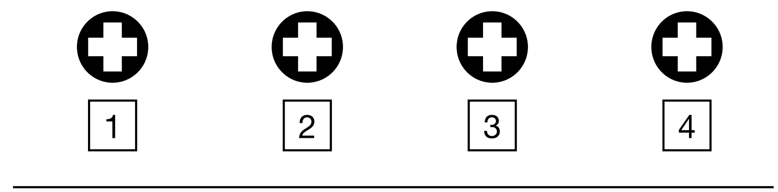


Example: NIMP and Gale-Shapley



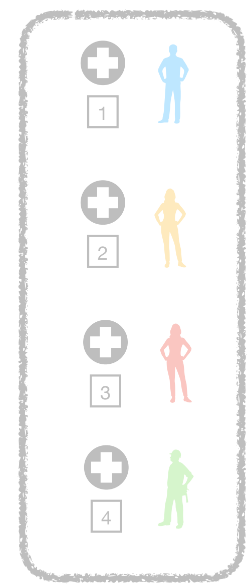
Example: NIMP and Gale-Shapley



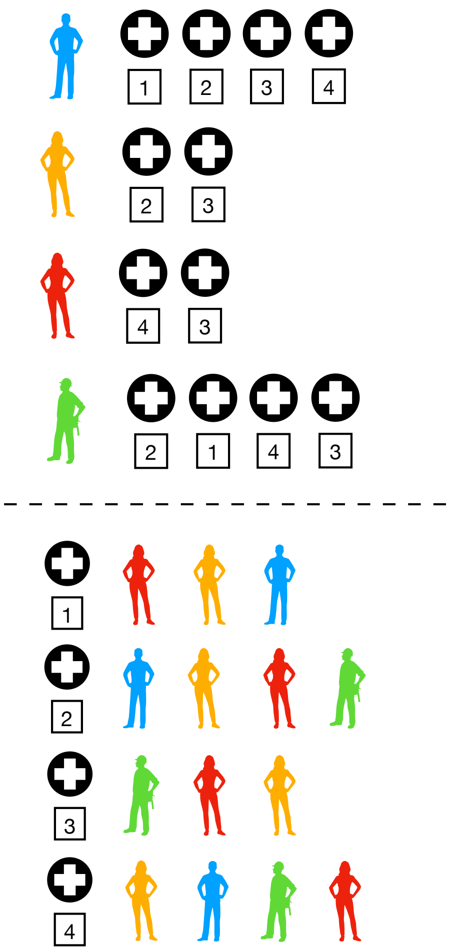


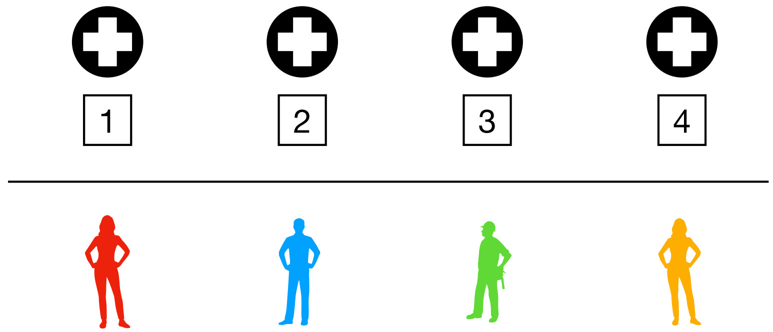
Hospital-proposing Gale-Shapley

NIMP

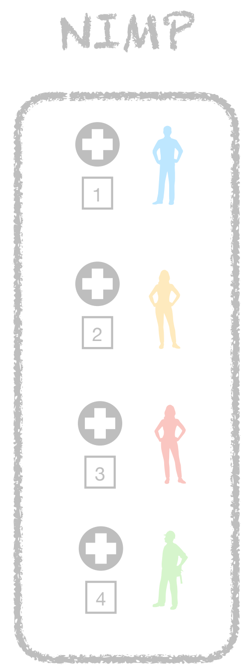


Example: NIMP and Gale-Shapley

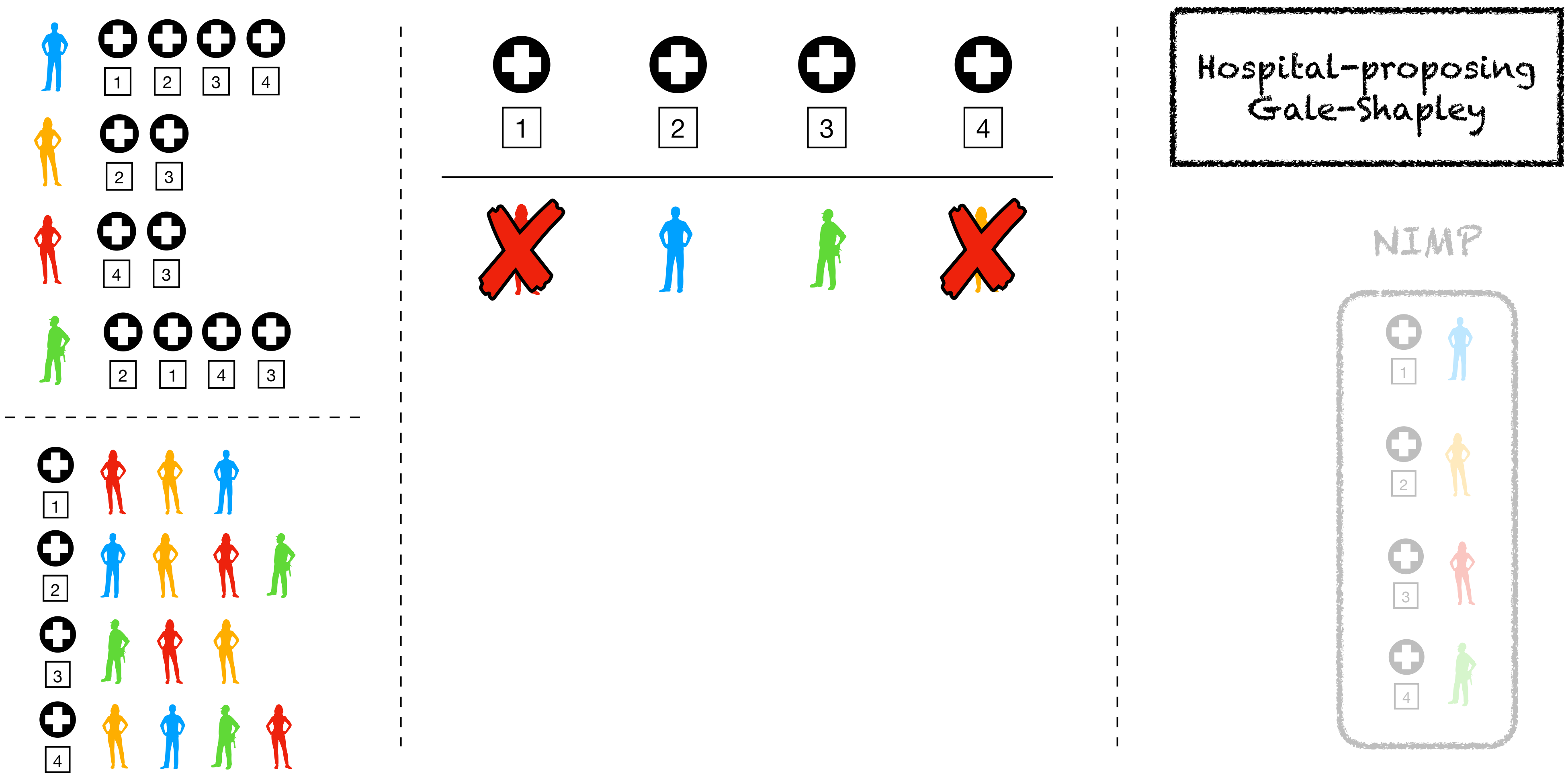




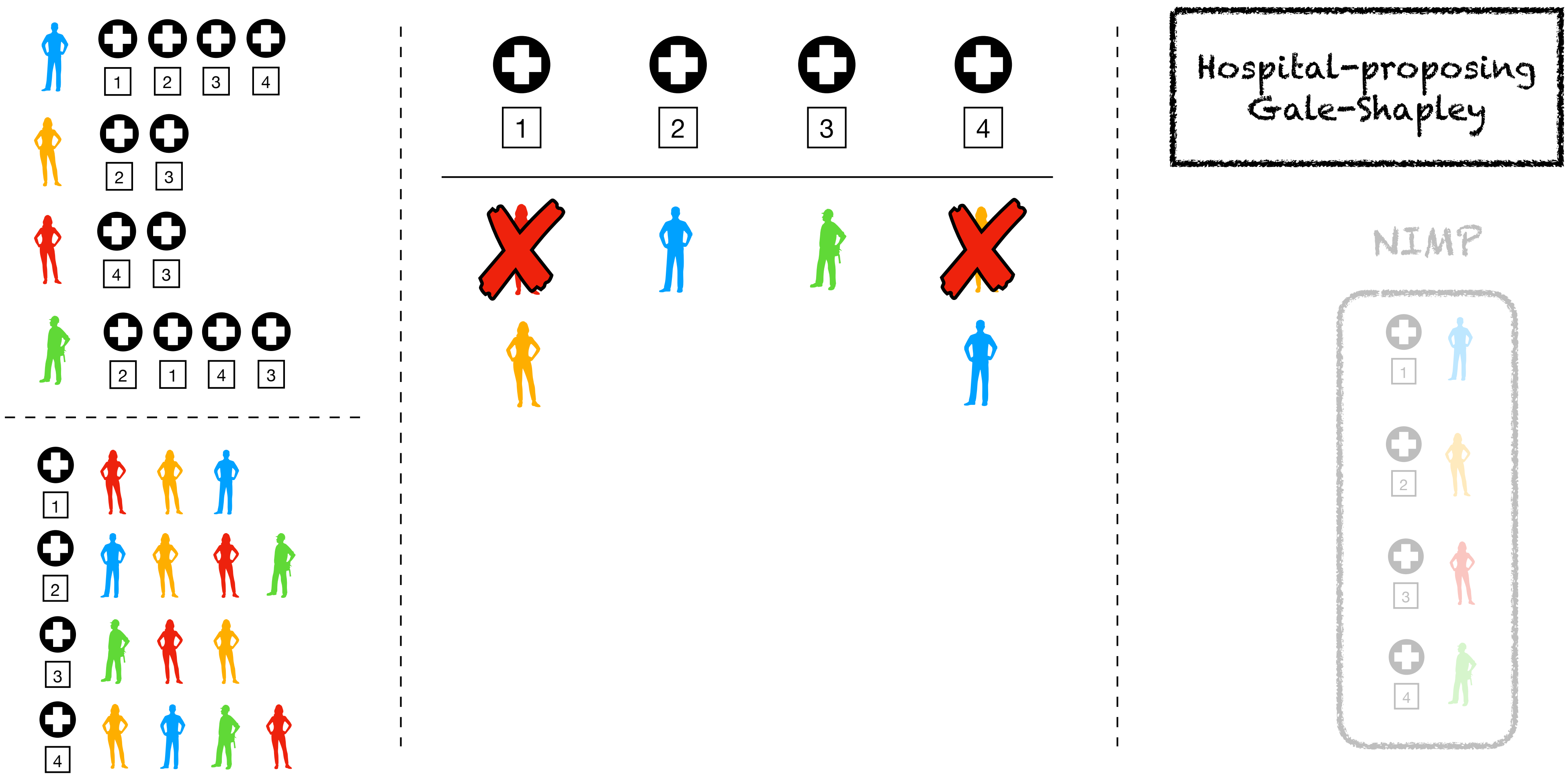
Hospital-proposing Gale-Shapley



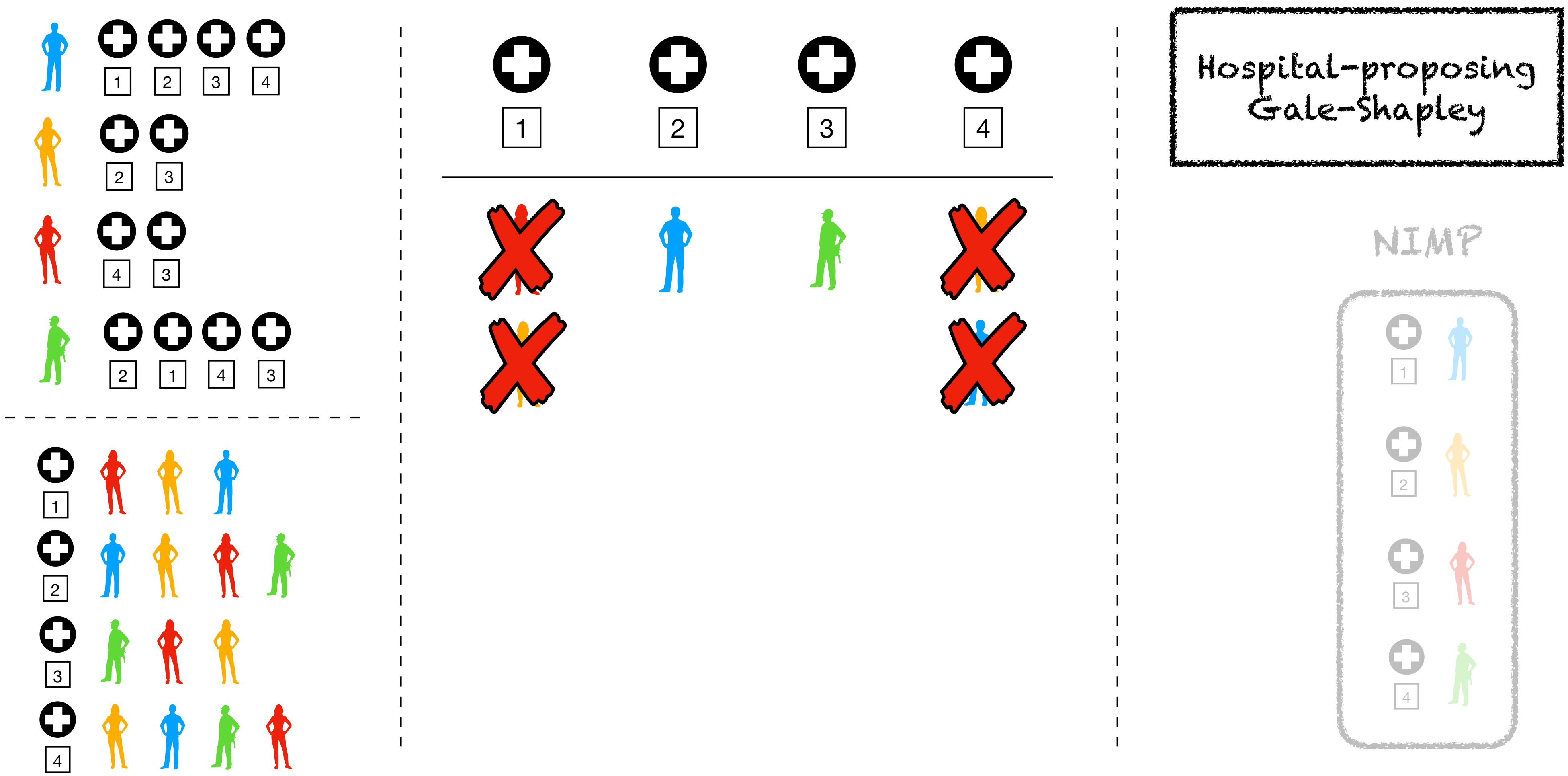
Example: NIMP and Gale-Shapley



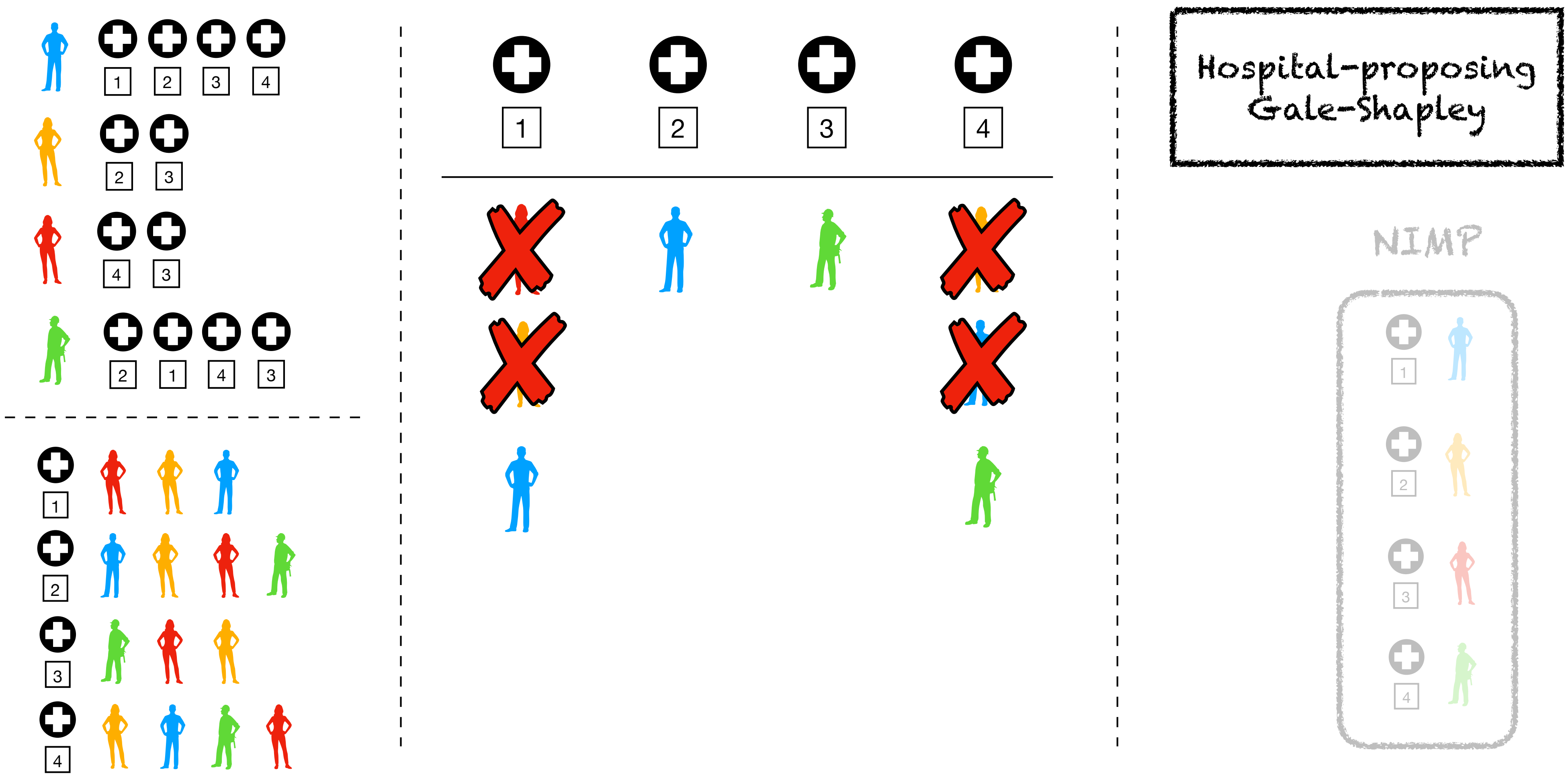
Example: NIMP and Gale-Shapley



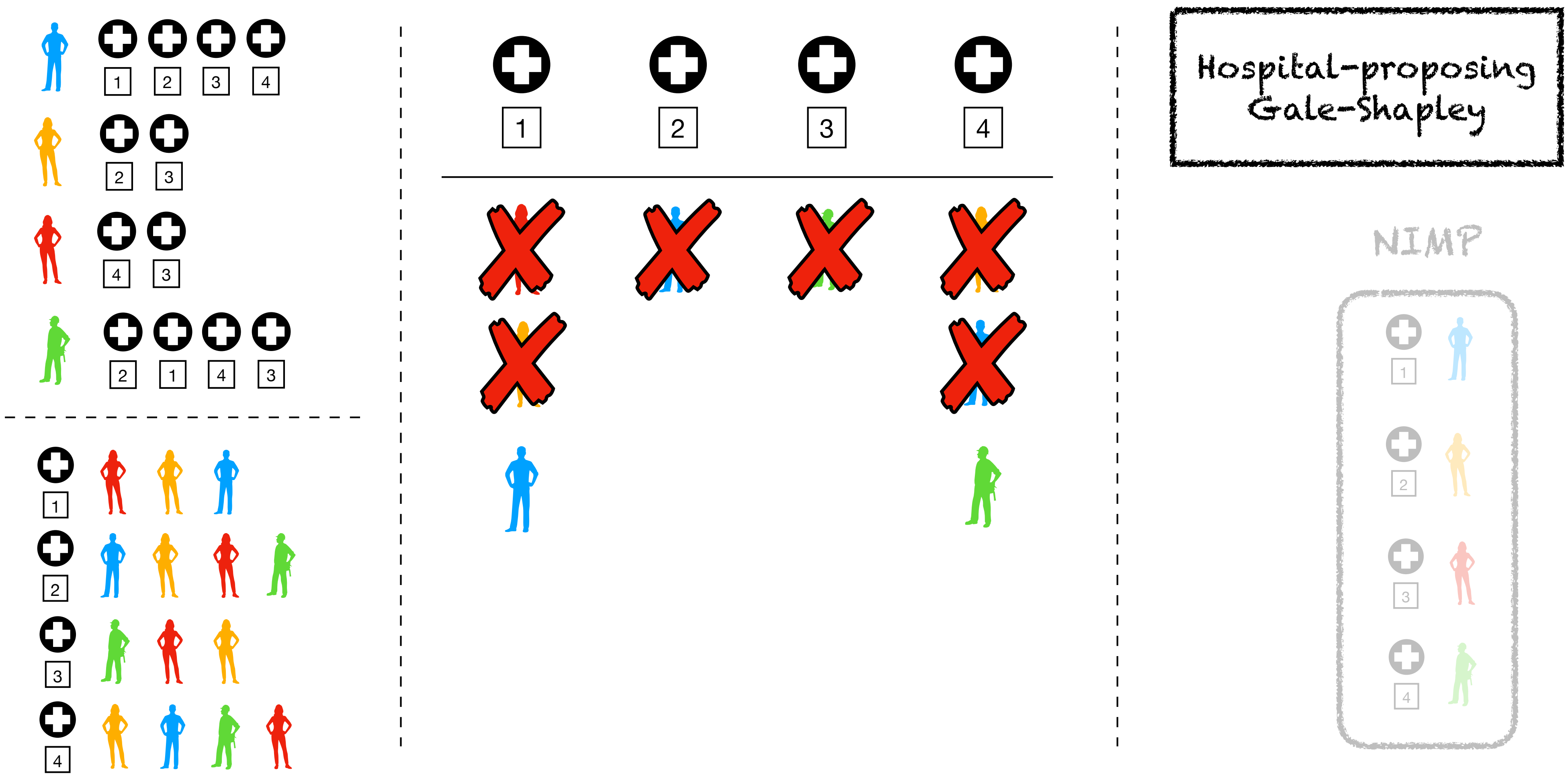
Example: NIMP and Gale-Shapley



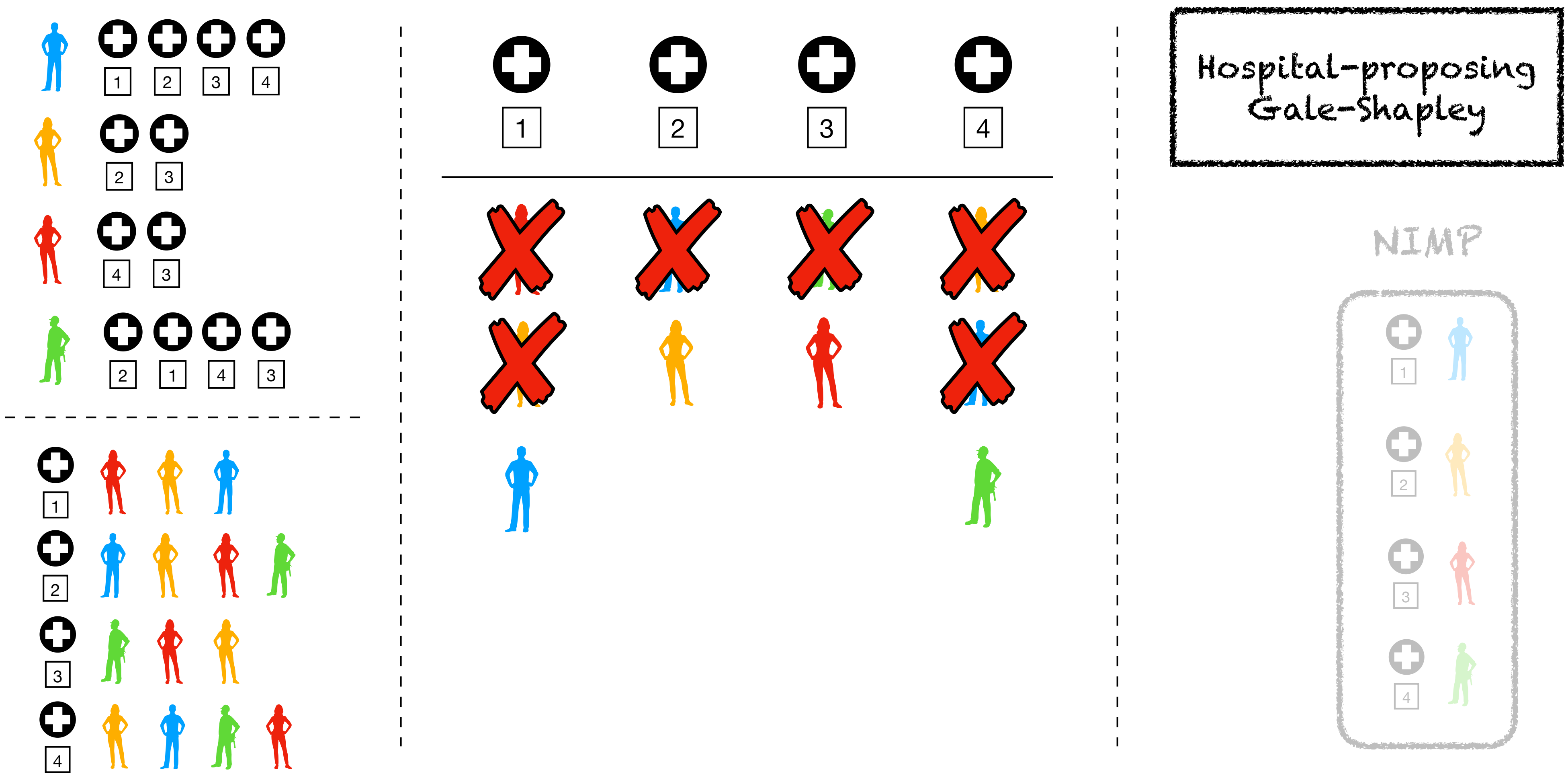
Example: NIMP and Gale-Shapley



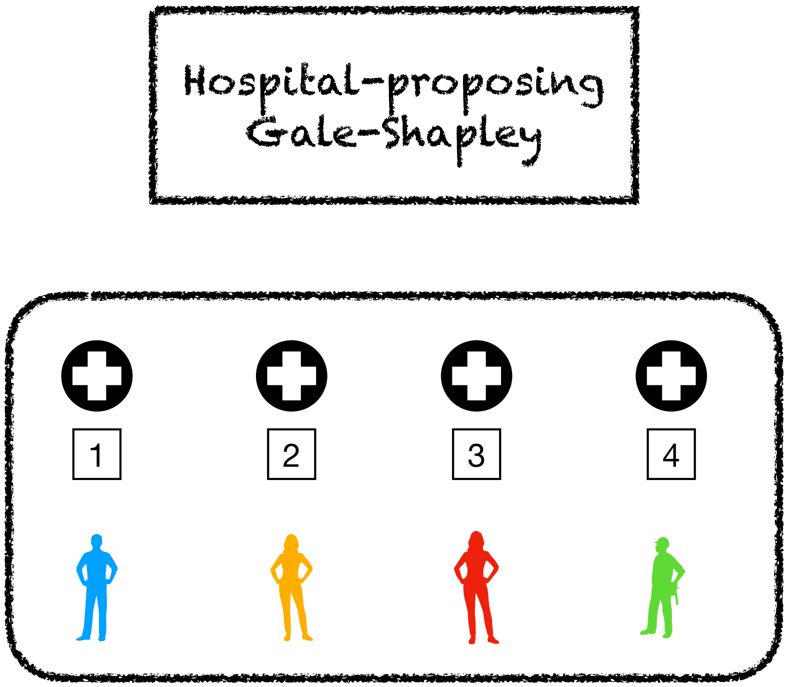
Example: NIMP and Gale-Shapley

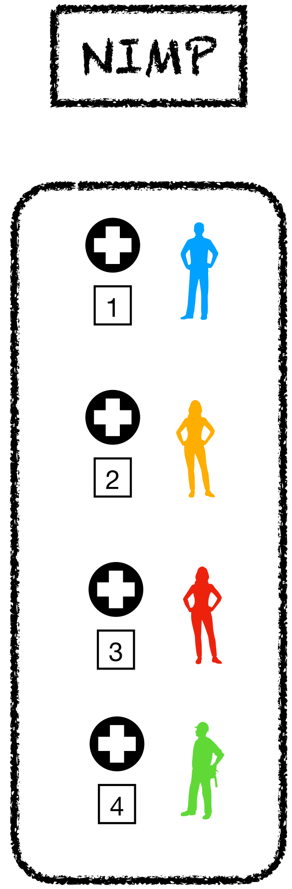


Example: NIMP and Gale-Shapley



Example: NIMP and Gale-Shapley





Example: NIMP and Gale-Shapley

NIMP = Hospital-proposing Gale-Shapley

- **Theorem**: The NIMP algorithm and the hospital-proposing Gale-Shapley algorithm are equivalent (they always generate the same matching).
- Proof: Try it as an exercise!
- **Exercise**: show with an example that the first step of the NIMP algorithm (removing partners who do not find me acceptable) is crucial. Give an example in which the algorithm generates a non-stable outcome when skipping the first step.

Week 8: School choice

BEEM147: Topics in Microeconomic Theory II

Matching and Market Design

Spring Term 2021

University of Exeter

Overview

- Last week
  - The medical match
  - A brief history of *unraveling* and *jumping the gun*
  - The NIMP algorithm and its trial-run
- This week
  - School choice
  - The Boston mechanism
  - DA and Pareto Efficiency
  - The school choice TTC

School choice

- Assign K-12 public school seats to students
- Typically, assign them according to location. But this is unfair (rich people can move, while poor cannot)
- Since a couple of decades ago, centralized allocation mechanisms: students are allowed to express their preferences for schools
- By nature, there are no equilibrium prices (tuition) in public school systems
- Literature started with Abdulkadiroğlu and Sönmez (2003, AER), "School Choice: A Mechanism Design Approach"
  - Identified problems in existing allocation mechanisms
  - Proposed fixes based on matching theory

School choice

- $(I, S, Q, \succ_I, \succ_S)$ = **school choice problem**
  - I = students
  - S = schools
  - $Q = (q_s)_{s \in S}$ capacity vector, with $q_s \in \mathbb{N}$ (how many seats in school s)
  - $\succ_I = (\succ_i)_{i \in I}$ students preferences with $\succ_i \in \mathcal{P}(S \cup \varnothing)$
  - $\succ_S = (\succ_s)_{s \in S}$ school priorities with $\succ_s \in \mathcal{P}(I)$
- **Matching** is a function $\mu : I \to S \cup \{\varnothing\}$ such that $|\mu^{-1}(s)| \leqslant q_s$ for every $s \in S$.
  - $\mu^{-1}(s)$ = set of students matched to school s
- Set of all matchings is $\mathcal{M}(I, S, Q)$.

Key properties

- Matching $\mu \in \mathcal{M}(I, S, Q)$ **eliminates justified envy** if there is no pair $(i, s) \in I \times S$ such that $i \succ_s j$ and $s \succ_i \mu(i)$ for some student j with $\mu(j) = s$.
  - Note resemblance of justified envy with stability
  - Different interpretation: students with justified envy may sue the school system
- Matching $\mu \in \mathcal{M}(I, S, Q)$ is **non-wasteful** if, for every $(i, s) \in I \times S$, $s \succ_i \mu(i)$ implies $|\mu^{-1}(s)| = q_s$.
  - Same idea as blocking partner who remains single, but interpretation is different
  - It is a "waste" for a seat to be unassigned while a student wants it
- Mechanism $\phi : \mathcal{P}(S \cup \{\varnothing\})^{|I|} \to \mathcal{M}(I, S, Q)$ is **strategy-proof** if every student finds it optimal to report their true preferences: for every $i \in I$ and every profile $(\succ_i) \in \mathcal{P}(S \cup \{\varnothing\})^{|I|}$,

$$\phi[(\succ_i, \succ_{-i})](i) \succcurlyeq_i \phi[(\succ', \succ_{-i})](i) \quad \text{for every } \succ' \in \mathcal{P}(S \cup \{\varnothing\}).$$

The Boston mechanism

- Used to be in place in Boston. Popular and easy to understand.
- **Algorithm**. Every student submits preference ranking, and each school determines priorities over students (home address, siblings, lottery ticket, etc.)
- **Step 1**: Assign the seats of each school to the students who rank it as their top choice, one at a time following its priority order. Proceed until either there are no seats left or until there is no student left who has listed it as their top choice.
- **Step** k: Assign the remaining seats of each school with remaining seats to the students who rank it as their k-th choice, one at a time following its priority order. Proceed until either there are no seats left or until there is no student left who has listed it as their k-th choice.
- Stop when every student has been assigned or there are no seats left.
- **Note:** mechanism also known as Immediate Acceptance (IA), compare with Gale-Shapley DA

The Boston mechanism

- Major problem with Boston mechanism is that it is **not** strategy-proof
  - It was well-known that it was better to “skip” popular schools in the ranking
  - There were parent associations advising parents how to “strategize” optimally
- As observed by Glazerman and Meyer (1994):

  It may be optimal for some families to be strategic in listing their school choices. For example, if a parent thinks that their favorite school is oversubscribed and they have a close second favorite, they may try to avoid "wasting" their first choice on a very popular school and instead list their number two school first.
- **Exercise**: provide the simplest example you can think of a school choice problem in which at least one student has incentives to misreport their true preference.

The Boston mechanism

- The Boston mechanism was highly criticized for giving an advantage to parents who were more familiar with the system, or that had the resources to "strategize" correctly.
- It reintroduced unfairness to a system which main objective was to eliminate it.
- **Exercise**: does the Boston mechanism eliminate justified envy?
- **Exercise**: is the Boston mechanism non-wasteful?

Deferred Acceptance (DA) and Pareto efficiency

- An alternative to the Boston mechanism is Gale and Shapley's DA (student-proposing):
  - At every step, a school s is tentatively matched to the best q_s students who have proposed or who were tentatively matched to it in the previous round, and rejects the rest.
- We know that DA:
  - eliminates justified envy
  - is non-wasteful
  - is strategy-proof for students
- Seems like the ideal mechanism, it was adopted in New York in 2003, and in Boston in 2005.
- However, there is a catch!

Deferred Acceptance (DA) and Pareto efficiency

- Student-proposing DA may generate matchings that are *not* **Pareto efficient for students**.
  - In this market, we have a valid reason for considering one-sided efficiency: can the students trade schools amongst themselves so that everyone is better off?
- **Example:** $I = \{1, 2, 3\}$ and $S = \{A, B, C\}$ with $q_s = 1$ for every school, preferences and priorities given by:

$$\succ_1: B, A, C \quad \succ_2: A, B, C \quad \succ_3: A, B, C$$

$$\succ_A: 1, 3, 2 \quad \succ_B: 2, 1, 3 \quad \succ_C: 2, 1, 3$$

- Student-proposing DA matches $(1, A)$, $(2, B)$, and $(3, C)$
- Note that student 3 is matched to C in every stable matching
- This matching is **Pareto dominated (amongst students)** if students 1 and 2 trade schools.
- Stability forces students 1 and 2 to "share" schools inefficiently

Deferred Acceptance (DA) and Pareto efficiency

- Matchings that are not Pareto efficient amongst students are hard to justify in practice.
- Introduce new notion of efficiency
- A mechanism ϕ **Pareto dominates** a mechanism ψ if, for all profile of preferences, ϕ results in a matching that all students prefer to the matching obtained by ψ, and for some profile $\succcurlyeq_I$ of students' preferences some of the students are strictly better off.
- **Question**: can we do better than DA in terms of efficiency without sacrificing any of its desirable properties?
- **Answer**: No!

Deferred Acceptance (DA) and Pareto efficiency

- **Proposition**: If ϕ is a strategy-proof (for students) and non-wasteful mechanism, then there is no strategy-proof mechanism that Pareto dominates ϕ.
- **Proof**:
  - $\phi =$ strategy-proof for students and non-wasteful. Two steps.
  - Step 1: Fix a preference profile $\succ_I \in \mathcal{P}(S \cup \{\varnothing\})^{|I|}$.
  - Consider matching ν that Pareto dominates $\mu = \phi(\succ_I)$: $\nu(i) \succcurlyeq_i \mu(i)$ for all i.
  - We show that the set of matched agents is the same under μ and ν
  - First, i matched under $\mu \Rightarrow i$ matched under ν
    - Otherwise, i would report $s = \mu(i)$ as unacceptable in ϕ

Deferred Acceptance (DA) and Pareto efficiency

- **Proposition**: If ϕ is a strategy-proof (for students) and non-wasteful mechanism, then there is no strategy-proof mechanism that Pareto dominates ϕ.
- **Proof**:
  - Second, assume i is matched under ν but not under μ: $\nu(i) \succ_i \varnothing = \mu(i)$
    - ϕ non-wasteful $\Rightarrow$ school $\nu(i)$ is full under μ
    - $\Rightarrow \exists\ i_1$ with $\mu(i_1) = \nu(i)$ and $\nu(i_1)$ is preferred by i_1 than $\mu(i_1)$ (preference are strict)
    - ϕ non-wasteful $\Rightarrow$ school $\nu(i_1)$ is full under μ
    - $\Rightarrow \exists\ i_2$ with $\mu(i_2) = \nu(i_1)$ and $\nu(i_2)$ is preferred by i_2 than $\mu(i_2)$ (preference are strict)
    - keep going on until run out of people, at some point reach a contradiction
  - Then, set of agents matched in μ and ν are the same

Deferred Acceptance (DA) and Pareto efficiency

- **Proposition**: If ϕ is a strategy-proof (for students) and non-wasteful mechanism, then there is no strategy-proof mechanism that Pareto dominates ϕ.
- **Proof**:
  - Step 2. Suppose $\exists$ mechanism ψ that Pareto dominates ϕ
  - We show that ψ is *not* strategy-proof for students
  - $\exists$ profile $\succ_I$ s.th. $\psi[\succ_I](i) \succcurlyeq_i \phi[\succ_I](i)$ for all i, strict for some j
  - Let $s = \psi[\succ_I](j)$. Consider that j reports $\succ'_j$, where s is the only acceptable school.
  - ϕ-SP $\Rightarrow \phi(\succ'_j, \succ_{-j})(j) = \varnothing$ (otherwise, j would have incentives to misreport in ϕ).
  - Step 1 $\Rightarrow$ the same students must be matched under both $\phi(\succ'_j, \succ_{-j})$ and $\psi(\succ'_j, \succ_{-j})$.
  - $\Rightarrow \psi(\succ'_j, \succ_{-j})(j) = \varnothing$.
  - $\Rightarrow \psi$ is not SP since, when j's true preference is $\succ'_j$, they would rather lie and report $\succ_j$. Q.E.D.

School choice TTC

- Alternative to DA, to obtain Pareto efficient matchings for students: Top-trading Cycle (TTC)
- **Algorithm.** Initially, every seat is empty.
  - Students point to their favorite schools. Schools that have remaining seats point to their favorite students.
  - There is at least one cycle, and no cycles overlap.
  - Assign students to the school they are pointing to within each cycle (and remove the corresponding seat from each school that was just assigned a student).
  - Remove schools that have no empty seats.
  - Repeat until no more students are assigned or there are no seats left.

School choice TTC

- TTC benefits students with a high priority but in a different way:
  - It allows students to trade a high priority in one school for a seat in another school.
  - Intuition: I am more likely to be in a cycle (and get my current top choice), if I have the highest priority for more schools
- **Proposition**: The school choice TTC mechanism is strategy-proof for students and Pareto efficient.
  - Prove the Proposition on your own as an exercise (very similar to housing markets)
- **Exercise**: Prove that the school choice TTC is non-wasteful
- **Question**: does the school choice TTC eliminate justified envy?
- **Answer**: No, there is a trade-off between stability (DA) and efficiency (TTC).

School choice TTC

- **Example:** Same as before: $I = \{1, 2, 3\}$ and $S = \{A, B, C\}$ with $q_s = 1$ for every school, preferences and priorities given by:

$$\succ_1: B, A, C \qquad \succ_2: A, B, C \qquad \succ_3: A, B, C$$

$$\succ_A: 1, 3, 2 \qquad \succ_B: 2, 1, 3 \qquad \succ_C: 2, 1, 3$$

- Student-proposing DA matches $(1, A)$, $(2, B)$, and $(3, C)$
- TTC matches $(1, B)$, $(2, A)$, and $(3, C)$, which Pareto dominates the DA assignment.
  - However, this is not the case in general (as we know from the Proposition)
- Note that the TTC matching *does not* eliminate justified envy: 3 envies 2

School choice TTC

- **Example:** Consider the following preferences:

$$\succ_1: B, C, A \quad \succ_2: A, B, C \quad \succ_3: A, B, C$$

$$\succ_A: 1, 3, 2 \quad \succ_B: 2, 3, 1 \quad \succ_C: 2, 1, 3$$

- Even though agent 1 ranks A as the worst choice now, they still have the highest priority in A (which matters for TTC).
- Verify that student-proposing DA matches $(1, C)$, $(2, B)$, and $(3, A)$.
- TTC matches $(1, B)$, $(2, A)$, and $(3, C)$.
- Agents 1 and 2 prefer to use TTC, while student 3 prefers student-proposing DA.
- $\Rightarrow$ Trade-off between efficiency and stability

School choice TTC

- In practice, the idea of "trading priorities" is hard to implement.
- According to school administrators:

There may be advantages to this approach... It may be argued, however, that certain priorities, e.g., sibling priority, apply only to students for particular schools and should not be traded away.

The trading mechanism can have the effect of "diluting" priorities' impacts, if priorities are to be "owned" by the district as opposed to being "owned" by parents; it shifts the emphasis onto the priorities and away from the goals the BPS is trying to achieve by granting these priorities in the first place; and could lead to families believing they can strategize by listing a school they don't want in hopes of a trade.