Two-Sided Matching with Partial Information

Kevin Leyton-Brown
University of British Columbia

Joint work with Baharak Rastegari, Anne Condon, Nicole Immorlica

Lecture Overview

1. Introduction
2. LP Formulation
3. Optimality Certificates

Two-sided Matching

- A widely applicable game theoretic paradigm:
  - university admission, hiring
  - corporate hiring (e.g., law firms)
  - hospital residents
- Important assumption in virtually all work on matching: agents have complete (and common) knowledge of their preferences (which are usually assumed to be strict).
- Key solution concept: stable matching
  - Gale–Shapley algorithm finds one of these in polynomial time

Reality Check

Think about how academic hiring really works...

- candidates mentally rank schools into top tier, second tier, etc, but don't really know how they would choose between schools within the same tier
- likewise, schools (often explicitly) rank candidates into tiers
- schools interview a small number of candidates
  - interviews are informative for both candidates and schools
  - schools don't necessarily interview from their top tier, because they're not sure they have a chance with these candidates
- at the end, based on the interviews everyone matches up

My goal: build a model to explain why this process works as well as it does (and perhaps to identify ways that it can fail).

Our Model

We consider a relaxed model in which:

- Agents start out unsure of their own preferences
  - They know a (true) partition of agents on the other side of the market into strictly ranked equivalence classes
  - "Pre-Bayesian" model: no probabilities
- In reality agents do have strict preferences
- Initial information can be refined through interviews, which are informative to both parties to the interview
- Goal: find a (true) stable matching by performing as few interviews as possible.

Upshot: I have many results to present. But this is still ongoing work; the biggest questions remain open.

Example

- 2 employers: UBC, McGill
- 2 applicants: Alice, Bob
- Initial partially ordered preferences

Alice	**Bob**
UBC	McGill
McGill	UBC

UBC	**McGill**
Alice	Alice
Bob	Bob

Example

- 2 employers: UBC, McGill
- 2 applicants: Alice, Bob
- Initial partially ordered preferences

Alice	Bob
UBC	McGill
McGill	UBC

UBC	McGill
Alice	Alice
Bob	Bob

- All four possible total orderings for the employers.

UBC	McGill
Alice	Alice
Bob	Bob

(a)

UBC	McGill
Alice	Bob
Bob	Alice

(b)

UBC	McGill
Bob	Bob
Alice	Alice

(c)

UBC	McGill
Bob	Alice
Alice	Bob

(d)

Stability

Definition (Matching)

A pairing of applicants and employers such that each applicant is paired with at most one employer and vice versa.

Stability

Definition (Matching)

A pairing of applicants and employers such that each applicant is paired with at most one employer and vice versa.

Definition (Blocking pair)

A pair who prefer each other to their current matches.

Definition (Stable matching)

A matching with no blocking pair and no agent matched to an unacceptable partner.

Stability

Definition (Matching)

A pairing of applicants and employers such that each applicant is paired with at most one employer and vice versa.

Definition (Blocking pair)

A pair who prefer each other to their current matches.

Definition (Stable matching)

A matching with no blocking pair and no agent matched to an unacceptable partner.

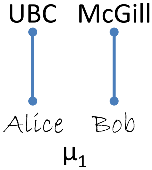
Definition (Employer-optimal matching)

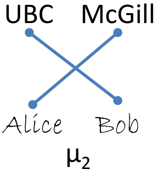
A stable matching that is weakly preferred by all employers to every other stable matching.

Example

Our example has two possible matchings: μ_1, μ_2

- μ_1 is stable under all orderings
- μ_2 is only stable under (d)
  - (UBC, Alice) blocks μ_2 under (a), (b)
  - (McGill, Bob) blocks μ_2 under (c)
- Employer optimality:
  - μ_1 is the only matching under (a), (b), (c), so here it's employer optimal
  - μ_2 is employer optimal under (d)





Alice	Bob
UBC	McGill
McGill	UBC

Applicants

UBC	McGill
Alice	Alice
Bob	Bob

Employers (a)

UBC	McGill
Alice	Bob
Bob	Alice

Employers (b)

UBC	McGill
Bob	Bob
Alice	Alice

Employers (c)

UBC	McGill
Bob	Alice
Alice	Bob

Employers (d)

Policies

Definition (Information; Information State)

The information $I_{i,k}$ available to agent i after interviews with $\ell \geq 0$ candidates in his kth equivalence class is an list of these ℓ candidates, ordered by i's degree of preference for each candidate. The global information state after a sequence of interviews is $I = \bigcup_{i,k} I_{i,k}$.

Definition (Policy)

A policy is a mapping from a global information state I to either an interview to perform or a matching that is guaranteed to be stable for all preference orderings consistent with the equivalence classes and I.

Definition (Employer-optimal Policy)

A policy is employer optimal if any matching it returns is guaranteed to be employer optimal for all preference orderings consistent with the equivalence classes and I.

Minimizing interviews

- Finding an employer-optimal policy is easy: perform every interview, then run Gale–Shapley.
- Our goal: perform as few interviews as possible.
- But... as few interviews as possible on which underlying preference ordering?
  - The policy depends on the results of the interviews!

Minimizing interviews

- Finding an employer-optimal policy is easy: perform every interview, then run Gale–Shapley.
- Our goal: perform as few interviews as possible.
- But... as few interviews as possible on which underlying preference ordering?
    - The policy depends on the results of the interviews!

Definition (Very weakly dominant policy)

A policy is very weakly dominant if it performs a minimal number of interviews on every underlying total ordering.

Dominant policies aren't going to work

Proposition

There does not exist a very weakly dominant policy.

- The minimal set of interviews necessary to certify the employer-optimal matching can depend on the (unknown) underlying strict ordering.
- We can already see this from our example:
  - To certify that μ_1 is employer optimal for (1), we only need to interview both candidates at UBC (to distinguish (1) from (4))
  - To certify that μ_1 is employer optimal for (2) or (3), we only need to interview both candidates at McGill (to distinguish (2) or (3) from (4))
  - Thus, any policy that conducts interviews at UBC is dominated in (2); any policy that conducts interviews at McGill is dominated in (1)

Non-weak Domination

Definition (Non-weakly dominated policy)

A policy p is non-weakly dominated if there does not exist any other policy p' that performs weakly fewer interviews for every underlying preference ordering, and strictly fewer interviews for some ordering.

Proposition

A non-weakly-dominated policy always exists.

Intuition: can't have cycles in the weak-domination relation between policies, because of the strict inequality.

Computing a non-weakly dominated policy

Brute force: check every policy

- Let n be the number of agents on the larger side of the market
- Each agent has $O(n!)$ possible preference orderings over agents on the other side of the market, and in the worst case, rules none of these out with prior information.
- Thus, there are $O((n+1)!)$ possible global information states.
- A policy maps from global information states to interviews. There are $O(n^2)$ possible interviews.
- Thus, the number of distinct policies is $O\left((n^2)^{(n+1)!}\right)$ which is $O\left((n^2)^{(n+1)^{(n+1)}}\right)$.
- So, the brute force algorithm is doubly-exponential.

Computing a non-weakly dominated policy

Theorem (Policy computation)

A non-weakly dominated policy can be computed in time $O((n+1)!)$, or $O\left((n+1)^{n+1}\right)$.

Key ideas:

- Formulate our setting as a Markov Decision Problem, with a uniform prior over the results of interviews and a uniform cost for performing interviews
- Compute an optimal policy for this MDP (slightly modified value iteration) in time linear in the size of the state space.
- Prove that this policy (which maximizes expected reward) is not weakly dominated.

Computing a non-weakly dominated policy

Theorem (Policy computation)

A non-weakly dominated policy can be computed in time $O((n+1)!)$, *or* $O\left((n+1)^{n+1}\right)$.

Key ideas:

- Formulate our setting as a Markov Decision Problem, with a uniform prior over the results of interviews and a uniform cost for performing interviews
- Compute an optimal policy for this MDP (slightly modified value iteration) in time linear in the size of the state space.
- Prove that this policy (which maximizes expected reward) is not weakly dominated.

This algorithm is still exponential in n, and doesn't leverage the fact that we're solving a matching problem. Can we find a non-weakly-dominated policy in polynomial time?

Lecture Overview

Characterizing stable matchings

Theorem (Characterization)

A matching is stable w.r.t. some total ordering that refines the partial ordering if and only if it is a vertex of the polytope:

$$\sum_{j \in A} x_{e,j} \le 1 \qquad \forall e \in E \qquad (1)$$

$$\sum_{i \in E} x_{i,a} \le 1 \qquad \forall a \in A \qquad (2)$$

$$\sum_{j \succeq_e a} x_{e,j} + \sum_{i \succeq_a e} x_{i,a} + x_{e,a} \ge 1 \qquad \forall e \in E,\ \forall a \in A \qquad (3)$$

$$x_{e,a} \ge 0 \qquad \forall e \in E,\ \forall a \in A \qquad (4)$$

$$x_{e,a} = 0 \qquad \forall \textit{unacceptable } (e,a) \textit{ pairs} \qquad (5)$$

- $j \succeq_e a$: either $j \succ_e a$ or e is uncertain about his ranking over j, a
- Constraint (3): either at least one of e and a is matched to someone (possibly) more preferred, or e and a are matched to each other.

Avoiding some interviews

Definition (Necessary match)

A pair that is matched in the employer-optimal matchings of all underlying preference orderings.

Definition (Impossible match)

A pair that does not match in the employer-optimal matching of any of the total orderings that refine the partial information.

Can we tractably identify necessary or impossible matches?

Is it necessary for e_i to match with a_j?

Proposition

(e_i, a_j) is a necessary match if (but not only if) the following program is infeasible.

$$\sum_{j \in A} x_{e,j} \leq 1 \qquad \forall e \in E$$

$$\sum_{i \in E} x_{i,a} \leq 1 \qquad \forall a \in A$$

$$\sum_{j \succeq_e a} x_{e,j} + \sum_{i \succeq_a e} x_{i,a} + x_{e,a} \geq 1 \qquad \forall e \in E,\ \forall a \in A$$

$$x_{e,a} \geq 0 \qquad \forall e \in E,\ \forall a \in A$$

$$x_{e,a} = 0 \qquad \forall\, unacceptable\ (e,a)\ pairs$$

$$x_{e_i,a_j} = 0$$

Is it impossible for e_i to match with a_j?

Proposition

(e_i, a_j) *is an impossible match if (but not only if) the following program is infeasible.*

$$
\begin{aligned}
\sum_{j \in A} x_{e,j} &\le 1 && \forall e \in E \\
\sum_{i \in E} x_{i,a} &\le 1 && \forall a \in A \\
\sum_{j \succeq_e a} x_{e,j} + \sum_{i \succeq_a e} x_{i,a} + x_{e,a} &\ge 1 && \forall e \in E,\ \forall a \in A \\
x_{e,a} &\ge 0 && \forall e \in E,\ \forall a \in A \\
x_{e,a} &= 0 && \forall \textit{unacceptable } (e,a) \textit{ pairs} \\
x_{e_i,a_j} &= 1
\end{aligned}
$$

Impossibility Claim

Although we can *find* necessary and impossible matchings tractably, this information isn't as useful as it might seem. It is sometimes still necessary for these pairs to interview when we aim to identify the employer-optimal matching.

Theorem (Impossibility)

No employer-optimal policy can:

- *avoid all interviews between necessary matches; and/or*
- *avoid all interviews between impossible matches.*

Proof

$\mathbf{e_1}$	$\mathbf{e_2}$	$\mathbf{e_3}$
a_1		
a_2	a_2	a_1
a_3		
	a_1	a_2

$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{a_3}$
e_2	e_3	e_1
e_1	e_1	
e_3	e_2	

Proof.

(e_1, a_3) is a necessary match.

1. If e_1's top choice is a_3 then all employers get their top choice.
2. otherwise, e_2 matches with a_1 and e_3 matches with a_2.
    - (1) is blocked by (e_1, a_1) and/or (e_1, a_2).

In order to distinguish between cases (1) and (2), we need to know whether e_1 has a_3 at the top of his ranking. Thus, e_1 has to interview both necessary and impossible matches.

What can be done?

Definition (Super-Stable Matching)

A matching that is stable w.r.t. all underlying preference orders.

Theorem

We can use linear programming to find a super-stable matching or prove that none exists in polynomial time, and without performing any interviews. Furthermore, we can likewise find the super-stable matching most preferred by employers. However, this latter matching is not guaranteed to be employer-optimal for all preference orders.

Lecture Overview

Restricted Setting and a Polytime Algorithm

Consider two restrictions on our partial information setting:

- all applicants have the same equivalence classes
  - but not necessarily the same underlying preference orderings
- your boss won't let you hire someone you haven't interviewed

Theorem (Polytime Algorithm for the Restricted Setting)

There exists a polynomial time algorithm that generates a very weakly dominant and thus non-weakly-dominated policy.

- The algorithm works like asynchronous Gale–Shapley, but "stalls" when an agent doesn't have required information.
- When the whole algorithm stalls, it has one employer from the applicants' top (remaining) equivalence class interview his entire top equivalence class.

Optimality Certificates

Definition (Optimality certificate)

A pair (I, μ) is an optimality certificate if μ is the employer-optimal matching for every preference ordering consistent with global information state I. The size of (I, μ) is the number of interviews performed in I.

Definition (Minimum optimality certificate for $>$)

(I, μ) is a minimum optimality certificate for a total ordering $>$ if μ is the employer-optimal matching for $>$, and if there does not exist a smaller optimality certificate (I', μ) that has I' consistent with $>$.

Theorem (Minimal optimality of our algorithm)

Our algorithm for the restricted setting terminates after identifying a minimum optimality certificate for the true underlying total ordering, subject to the constraint that all matched agents must have interviewed.

Hardness of finding small optimality certificates

Bad news: once we drop the assumption that all applicants have the same equivalence classes, finding a minimum optimality certificate is hard, even if we still assume that matched agents must interview.

Definition (Optimality Certificate decision problem)

Given partially ordered preferences and a total ordering $>$, decide whether there exists an optimality certificate of size at most k for $>$.

Theorem (Hardness)

The optimality certificate decision problem is NP-complete.

Proof Sketch

- *In NP: check the certificate using a (new) LP*
- *NP-hard: reduction from Feedback Arc Set*

What does this result mean?

It seems like bad news that minimal certificates are hard to find.

- However, we haven't proven that finding minimal certificates is necessary for every non-weakly-dominated policy, though they are certainly necessary for *some* such policies
- Also, even if finding minimal certificates *is* necessary, we also haven't shown that the solution to FAS we embedded in the optimality certificate decision problem needs to be found by a non-weakly-dominated policy
  - The policy could instead target the minimal optimality certificate for some $\succ$ that wasn't embedded by the reduction.
- The hardness of computing a non-weakly-dominated policy remains open...

Conclusions

- A non-weakly-dominated policy exists and can be computed in exponential time.
- Linear programming is a useful tool for characterizing and reasoning about matchings in our setting.
- When all applicants have the same initial information, we can identify a non-weakly-dominated policy in polynomial time
- This algorithm identifies a minimum optimality certificate for the true underlying ordering in this restricted setting
- Generally, finding a minimum optimality certificate is hard

Open questions:

- How hard is it to find a non-weakly-dominated policy?
- Can our algorithm for the restricted setting be extended to the case where matched agents don't have to interview?