Two-Sided Matching with Partial Information

Kevin Leyton-Brown
University of British Columbia

Joint work with Baharak Rastegari,
Anne Condon, Nicole Immorlica

Samos Summer School on Algorithmic Game Theory:
July 16, 2012

Lecture Overview

1. Introduction
2. Our Model
3. Necessary & Impossible Matches
4. Optimality Certificates
5. Symmetric Partial Information Among All Applicants

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
- at the end, based on the interviews everyone matches up

Our 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
- 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 that is optimal for a given side of the market, by performing as few interviews as possible.

Example

- 2 employers: UBC, Athens
- 2 applicants: Alice, Vasilis
- Initial partially ordered preferences

Alice	**Vasilis**
UBC	Athens
Athens	UBC

UBC	**Athens**
Alice Vasilis	Alice Vasilis

Example

- 2 employers: UBC, Athens
- 2 applicants: Alice, Vasilis
- Initial partially ordered preferences

Alice	**Vasilis**
UBC	Athens
Athens	UBC

UBC	**Athens**
Alice Vasilis	Alice Vasilis

- All four possible total orderings for the employers.

UBC	**Athens**
Alice	Alice
Vasilis	Vasilis

(a)

UBC	**Athens**
Alice	Vasilis
Vasilis	Alice

(b)

UBC	**Athens**
Vasilis	Vasilis
Alice	Alice

(c)

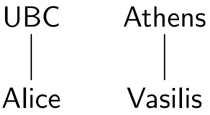
UBC	**Athens**
Vasilis	Alice
Alice	Vasilis

(d)

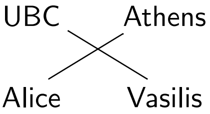
Example

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)
  - (Athens, Vasilis) blocks μ_2 under (b), (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)



μ_1



μ_2

Alice	**Vasilis**
UBC	Athens
Athens	UBC

Applicants

UBC	**Athens**
Alice	Alice
Vasilis	Vasilis

Employers (a)

UBC	**Athens**
Alice	Vasilis
Vasilis	Alice

Employers (b)

UBC	**Athens**
Vasilis	Vasilis
Alice	Alice

Employers (c)

UBC	**Athens**
Vasilis	Alice
Alice	Vasilis

Employers (d)

Lecture Overview

1 Introduction

2 Our Model

3 Necessary & Impossible Matches

4 Optimality Certificates

5 Symmetric Partial Information Among All Applicants

Policies

Definition (Information State)

The information state I_i of agent i after interviews with $\ell \geq 0$ candidates is a list of these ℓ candidates, ordered according to the underlying true preference profile. The global information state after a sequence of interviews is $I = \bigcup_i I_i$.

Definition (Policy)

A policy is a mapping from a global information state I either to an interview to perform or to a matching. A policy is sound if it is guaranteed to return an employer-optimal matching, regardless of the true preference order.

Minimizing the Number of Interviews

- Finding a sound 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 the Number of Interviews

- Finding a sound 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!

This is easy if we have a prior distribution over strict orderings (e.g., we believe all orderings are equally likely).

Definition (Optimal-in-expectation policy)

A policy f is optimal in expectation if it is sound and it minimizes the expected number of interviews performed, given a prior.

An optimal-in-expectation policy always exists.

Very Weak Domination

We'd prefer not to rely on a prior.

Definition (Very weakly dominant policy)

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

- "Very weak": two such policies can dominate each other.

Very Weak Domination

We'd prefer not to rely on a prior.

Definition (Very weakly dominant policy)

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

- "Very weak": two such policies can dominate each other.

Proposition

Very weakly dominant, sound policies do not always exist.

- Proof idea: the minimum set of interviews necessary to certify the employer-optimal matching can vary depending on the (unknown) underlying strict ordering.

Pareto Optimality

Definition (Pareto optimal policy)

A policy f is Pareto optimal if it is sound and there does not exist any other sound policy g that performs weakly fewer interviews for every underlying preference ordering, and strictly fewer interviews for some ordering.

Pareto Optimality

Definition (Pareto optimal policy)

A policy f is Pareto optimal if it is sound and there does not exist any other sound policy g that performs weakly fewer interviews for every underlying preference ordering, and strictly fewer interviews for some ordering.

Proposition

A Pareto optimal policy always exists.

Pareto Optimality

Definition (Pareto optimal policy)

A policy f is Pareto optimal if it is sound and there does not exist any other sound policy g that performs weakly fewer interviews for every underlying preference ordering, and strictly fewer interviews for some ordering.

Proposition

A Pareto optimal policy always exists.

Proposition

If a policy f is optimal in expectation and the prior has full support, then f is Pareto optimal.

Computing an optimal-in-expectation policy

Brute force: check every policy, keep the best one

- Let S denote the number of global information states.
- Thus, the number of distinct policies is $O\left((n^2)^S\right)$.
- The brute force algorithm is doubly-exponential.

Computing an optimal-in-expectation policy

Brute force: check every policy, keep the best one

- Let S denote the number of global information states.
- Thus, the number of distinct policies is $O\left((n^2)^S\right)$.
- The brute force algorithm is doubly-exponential.

Theorem (Policy computation)

An optimal in expectation policy can be computed in time polynomial in S.

- Encode the problem as a Markov decision process (a bit tricky).
- Compute cost-minimizing policy for the MDP (straightforward).

$\Rightarrow$ Exponential in input size; doesn't leverage matching structure.

Computing an optimal-in-expectation policy

Brute force: check every policy, keep the best one

- Let S denote the number of global information states.
- Thus, the number of distinct policies is $O\left((n^2)^S\right)$.
- The brute force algorithm is doubly-exponential.

Theorem (Policy computation)

An optimal in expectation policy can be computed in time polynomial in S.

- Encode the problem as a Markov decision process (a bit tricky).
- Compute cost-minimizing policy for the MDP (straightforward).

$\Rightarrow$ Exponential in input size; doesn't leverage matching structure.

Can we find an optimal in expectation policy in polynomial time?

Lecture Overview

1. Introduction
2. Our Model
3. Necessary & Impossible Matches
4. Optimality Certificates
5. Symmetric Partial Information Among All Applicants

Exploiting Structure

- A polynomial algorithm would have to leverage structural properties of our problem.
- One natural candidate: uninformative interviews
  - pairs that match for every underlying preference profile
  - pairs that likewise never match
- It could help an algorithm to remove such employer–applicant pairs from consideration, reducing problem size

Definition (Necessary (Impossible) match)

A pair that is (is not) matched in the employer-optimal matchings of all underlying preference orderings.

Can we tractably identify necessary or impossible matches?

Characterizing stable matchings

Theorem (Characterization)

Every matching that is stable w.r.t. some total ordering that refines the partial ordering is a vertex of the polytope:

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

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

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

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

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

- $j \succeq_e a$: 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.

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.

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

We can identify impossible matches analogously.

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 sound 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 that is identified by our LP.

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.

Lecture Overview

1. Introduction
2. Our Model
3. Necessary & Impossible Matches
4. **Optimality Certificates**
5. Symmetric Partial Information Among All Applicants

Optimality Certificates

Definition (Optimality certificate)

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

Definition (Minimum optimality certificate for $\succ$)

(I, μ) is a minimum optimality certificate for a total ordering $\succ$ if μ is the employer-optimal matching for $\succ$, $\succ$ refines I, and if there does not exist a smaller optimality certificate (I', μ) such that $\succ$ refines I'.

Optimality Certificates

Definition (Optimality certificate)

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

Definition (Minimum optimality certificate for $\succ$)

(I, μ) is a minimum optimality certificate for a total ordering $\succ$ if μ is the employer-optimal matching for $\succ$, $\succ$ refines I, and if there does not exist a smaller optimality certificate (I', μ) such that $\succ$ refines I'.

Theorem

A policy computes a minimum optimality certificate for every preference profile if and only if it is very weakly dominant.

Hardness of finding minimum optimality certificates

Theorem (Hardness; informal)

Finding a minimum optimality certificate is NP-hard.

- Formal statement of the theorem uses a decision version of the minimum optimality certificate problem
- The proof is a reduction from the feedback arc set problem.

Corollary

It is NP-hard to find a very weakly dominant policy if one exists.

What does this result mean?

- The fact that minimum certificates are hard to find seems like evidence against the existence of a polynomial-time algorithm for finding optimal-in-expectation or Pareto optimal policies
- However, we don't know if finding minimum certificates is necessary for such policies.
- Determining the hardness of computing an optimal-in-expectation or Pareto optimal policy remains an open problem.

Lecture Overview

1. Introduction
2. Our Model
3. Necessary & Impossible Matches
4. Optimality Certificates
5. Symmetric Partial Information Among All Applicants

Symmetric partial information on one side of the market

Consider two restrictions on our partial information setting:

1. all applicants start out with the same equivalence classes
   - but not necessarily the same underlying preference orderings
   - and, if there is a prior, not necessarily the same distributions
2. your boss won't let you hire someone you haven't interviewed

- A Pareto optimal policy can take $O(S)$ space to write down.
  - i.e., space exponential in the size of the input
- However, in this restricted setting it turns out that we can execute an optimal policy tractably.

A Polytime Algorithm

Asynchronous Gale-Shapley. Repeat until everyone is matched or has been rejected by all agents on the other side of the market:

- Every unmatched employer who knows his top choice among the remaining applicants proposes; remaining employers wait.
- Applicants receive proposals, tentatively accept their best matches, and reject employers who are inferior.
- If all unmatched employers are waiting, some unmatched employer from the applicants' top remaining equivalence class interviews his entire top remaining equivalence class.

A Polytime Algorithm

Asynchronous Gale-Shapley. Repeat until everyone is matched or has been rejected by all agents on the other side of the market:

- Every unmatched employer who knows his top choice among the remaining applicants proposes; remaining employers wait.
- Applicants receive proposals, tentatively accept their best matches, and reject employers who are inferior.
- If all unmatched employers are waiting, some unmatched employer from the applicants' top remaining equivalence class interviews his entire top remaining equivalence class.

Theorem (Polytime Algorithm for the Restricted Setting)

Asynchronous Gale-Shapley executes a very weakly dominant policy—and hence both an optimal-in-expectation and a Pareto optimal policy—in polynomial time.

Conclusions

We extended classical two-sided matching to a model in which agents are endowed with partial preference information.

- A very weakly dominant policy may not exist.
- Both an optimal-in-expectation policy and a Pareto optimal policy always exist; both can be computed in exponential time.
- We can tractably identify necessary and impossible matches, but nevertheless can't avoid these interviews
- Finding a minimum optimality certificate is NP-hard, and thus so is finding a very weak dominant policy, if one exists.
- When all applicants begin with the same equivalence classes, we can execute a very weakly dominant policy in polytime.

Key open questions: hardness of executing optimal policies in general; hardness of approximation; characterizing settings where a linear number of interviews suffices; studying decentralized policies.