Stable Matching Completed
Max Flow
Hard vs easy versions of matching

NP

NP-complete

Exact Maximal Bipartite Matching

Max Cardinality WSMTI

Max Cardinality SMI

Maximum Bipartite Matching

P
An instance consists of
• a list of preference orders for profs
• a list of preference orders for students
• a positive integer $k$

The preference orders may incomplete and have ties
Does the instance have a weakly stable matching of size at least $k$?

(Note that all capacities are 1.)
Exact Maximal Bipartite Matching

Instance: A bipartite graph $G = (U, V, E)$, where $|U| = |V|$, and a positive integer $k$
Does $G$ have a maximal matching of size exactly $k$?

A matching is *maximal* if no edges can be added to the matching
Outline for today

- Proof that Max Cardinality WSMTI is NP-complete
- Yet other variants of stable matching
NP-completeness of Max Cardinality WSMTI

We need to show two things:

• Max Cardinality WSMTI is in NP (discussed last time)

• There is a polynomial time reduction from a known NP-complete problem to Max Cardinality WSMTI

• We’ll show that

  Exact Maximal Bipartite Matching \( \leq_p \) Max Cardinality WSMTI
Exact Maximal Bipartite Matching $\leq_p$ Max Cardinality WSMTI

The reduction: Given an instance

$I = (U, V, E, k)$ of Exact Maximal Bipartite Matching, we need to show how to construct an instance

$I’ = (P\text{-prefs}, S\text{-prefs}, k’)$ of Max Cardinality WSMTI such that

$I$ is a yes-instance of Exact Maximal Bipartite Matching if and only if

$I’$ is a yes-instance of Max Cardinality WSMTI.
Exact Maximal Bipartite Matching $\leq_p$ Max Cardinality WSMTI

$I = (U, V, E, k) \quad \rightarrow \quad I' = (P\text{-prefs}, S\text{-prefs}, k')$
Exact Maximal Bipartite Matching $\leq_p$ Max Cardinality WSMTI

$I = (U, V, E, k) \rightarrow I' = (P\text{-prefs}, S\text{-prefs}, k')$

Let $n = |U| = |V|$

Then in $I'$, $k' = 2n-k$ and the preference lists are:

- $p_i : S_i, S_e$
- $s_i : P_i, P_e$, for $1 \leq i \leq n$
- $p_i : \{s_1, s_2, ... s_{2n-k}\}$
- $s_i : \{p_1, p_2, ... p_{2n-k}\}$ for $n+1 \leq i \leq 2n-k$

where

- $S_i = \{s_j \mid \{v_j, u_i\} \text{ is in } E\}$
- $P_i = \{p_j \mid \{u_j, v_i\} \text{ is in } E\}$
- $P_e$ is a set of “extra” professors $p_{n+1}, ... p_{2n-k}$
- $S_e$ is a set of “extra” students $s_{n+1}, ... s_{2n-k}$
For the proof of correctness, recall weakly blocking:

With respect to a matching $M$, pair $(p, s)$ is \textit{weakly blocking} if the following are all true.

(i) Pair $(p, s)$ acceptable and is not in $M$.
(ii) Either $s$ is unassigned, or $s$ prefers $p$ to $M(s)$.
(iii) Either $M(p) < c(p)$, or $p$ prefers $s$ to at least one member of $M(p)$.

A matching $M$ is \textit{weakly stable} if $M$ has no weakly blocking pairs.
NP-completeness: things to keep in mind

- Always reduce from the known NP-complete decision problem $D$ to the decision problem $D'$ of interest (not the other way)

- While the reduction goes one way $D \leq_p D'$ the proof of correctness has two directions: $I$ is a yes-instance of $D$ if and only if $I'$ is a yes instances of $D'$
NP-completeness: things to keep in mind

• For a problem D to be NP-complete, D must be a decision problem and D must be in NP.

• The term "NP-hard" can be used to include optimization problems, and problems that are not in NP.

• Since it's possible that P = NP, remember that NP-completeness is still just evidence that a problem does not have an efficient (polynomial time) algorithm, not a proof.
NP-completeness: True, False, Open

In what follows, assume that D and D’ have both yes and no instances.

1. Graph Colouring is in P

2. If D is in P and D’ is in P then $D \leq_p D'$

3. If D is in P and D’ is in NP then $D \leq_p D'$

4. SMI (SM with Incomplete Preferences, but no ties) is NP-complete

5. If D is NP-complete and $D' \leq_p D$ then $D'$ is NP-complete.

6. If D is NP-complete and $D'$ is a subset of D then $D'$ is NP-complete.
One more variant of Stable Matching

• Goal: Stable matching of profs and students.

• However, profs and students initially have only partial information about their (strict, complete) preference lists.

• Only through expensive interviews can they uncover the complete strict preference order (which is consistent with their initial partial orders.)

One more variant of Stable Matching

- Problem: design an adaptive, centralized interviewing algorithm which, given a partial information state, either selects an interview to perform or terminates with an optimal stable matching. As interviews are costly, the algorithm ideally should minimize the number of interviews performed.

One more variant of Stable Matching

• Problem: design an adaptive, centralized interviewing algorithm which, given a partial information state, either selects an interview to perform or terminates with an optimal stable matching. As interviews are costly, the algorithm ideally should minimize the number of interviews performed.

• A dominant algorithm performs a minimal number of interviews for all underlying true preference profiles consistent with the partial information. Unfortunately, such policies may not exist.

One more variant of Stable Matching

- Problem: design an adaptive, centralized interviewing algorithm which, given a partial information state, either selects an interview to perform or terminates with an optimal stable matching. As interviews are costly, the algorithm ideally should minimize the number of interviews performed.

One more variant of Stable Matching

• Problem: design an adaptive, centralized interviewing algorithm which, given a partial information state, either selects an interview to perform or terminates with an optimal stable matching. As interviews are costly, the algorithm ideally should minimize the number of interviews performed.

• A *Pareto optimal* algorithm may not be minimal for all underlying preference profiles, but is guaranteed not to be dominated by any other policy.

One more variant of Stable Matching

• Problem: design an adaptive, centralized interviewing algorithm which, given a partial information state, either selects an interview to perform or terminates with an optimal stable matching. As interviews are costly, the algorithm ideally should minimize the number of interviews performed.

• A Pareto optimal algorithm may not be minimal for all underlying preference profiles, but is guaranteed not to be dominated by any other policy.

• Pareto optimal policies are guaranteed to exist, but may run in exponential time. Rastegari et al. provide evidence that finding Pareto optimal interview policies may be NP-hard.

Summary: Stable Matching

• *SMI (Stable Matching with Incomplete Preferences)* is an old problem with an efficient “iterative improvement” algorithm

• *Max Cardinality WSMTI* is a decision variant of stable matching that avoids weakly blocking pairs and where preferences can contain ties, is NP-complete

• *Stable matching with partial information* is a new variant with a more game-theoretic problem formulation
We’ve also introduced two bipartite matching problems:

- Maximum Bipartite Matching
- Exact Maximal Bipartite Matching (NP-complete)

We’ll see algorithms for Maximum Bipartite Matching shortly... but first, we’ll take a detour to study Network Flow problems.
Summary: NP-completeness

• We’ve seen one NP-completeness proof; the reduction is pretty specific to the problem at hand, namely Max Cardinality WSMTI.

• If you want to show some other problem is NP-complete, the details of the reduction will likely be different. You’ll likely want to start with a known NP-complete problem that is “similar” to your problem.

• A classic guide to NP-completeness is a book by Garey and Johnson. Today, a google search is likely to be just as useful.