Stable Matching Continued

Hardness of Max Cardinality WSMTI
(Weakly Stable Matching with Ties and Incomplete Preferences)
Outline for today

• SMI: Stable Matching with Incomplete (Partial) Preferences: proof of correctness of Gale-Shapley (GS) algorithm (using slides from previous lecture)
• A new variant with ties: Max Cardinality WSMTI
• NP-completeness: a quick review
• Proof that Max Cardinality WSMTI is NP-complete

• Yet other variants of stable matching
Stable Matching with Ties and Incomplete Preferences (SMTI)

- In this variant of the problem, professors and students may have ties in their preference lists.

- Example: p1: \{s2, s4\}, s3, \{s1, s5\} means that p1 is indifferent between s2 and s4, but prefers both to s3; moreover p1 prefers s3 to both s1 and s5, but is indifferent between s1 and s5.
Stable Matching with Ties and Incomplete Preferences (STMI)

With respect to a matching \( M \), pair \( (p,s) \) is 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 weakly stable if \( M \) has no weakly blocking pairs.
Stable Matching with Ties and Incomplete Preferences (STMI)

Goal: given a SMTI instance, find a weakly stable matching
NP-Completeness
NP-Completeness

• Helps identify which problems are unlikely to have efficient, i.e., polynomial-time, algorithms

• It's convenient to work with decision problems, which have yes or no answers
Max Cardinality SMI (stable matching with incomplete preferences)
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.
Does the instance have a stable matching of size at least \( k \)?

(We saw in the last class that there is an efficient algorithm for this problem.)
NP-Completeness: Decision Problem Examples

Max Cardinality WSMTI (*weak* SM with *ties* and *incomplete Prefs*)
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 $k$?

(This problem is our main focus today.)
Graph Colouring

Instance: An undirected graph $G = (V,E)$ and a positive integer $k$

Can the nodes of $G$ be coloured with at most $k$ colours, so that no two adjacent nodes have the same colour?

(This is one of the best known “NP-complete” problems.)
NP-Completeness: Decision Problem Examples

Maximum Bipartite Matching

Instance: A bipartite graph $G = (U,V,E)$ and a positive integer $k$

Does the maximum matching of $G$ have size at least $k$?

(There is an efficient algorithm for this problem, which we’ll study in the next class.)
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
NP-Completeness

We can group decision problems into classes (sets)

**P**: class of problems with polynomial time algorithms

Examples of problems in P:
- SMI: Stable Matching with Incomplete (Partial) Preferences
- Maximum Bipartite Matching (upcoming lecture)
NP-Completeness

**NP**: class of problems with polynomial time verifier algorithms

A verifier $V$ for a decision problem $D$ takes both an instance $I$ of $D$ and a witness $W$, such that:

- If $I$ is a yes-instance, then for some $W$, $V(I,W) = \text{yes}$
- If $I$ is a no-instance, then for all $W$, $V(I,W) = \text{no}$
NP-Completeness

Known examples of problems in $\textbf{NP}$:

• Graph Colouring
• Maximal Bipartite Matching
NP-Completeness

Known examples of problems in \textbf{NP}:

• Graph Colouring
• Maximal Bipartite Matching

\textit{Important point}: problems in P are also in NP!

• Max Cardinality SMI
• Maximum Bipartite Matching
A decision problem D' is *NP-complete* if

- D' is in NP, and
- For all D in NP, there is a *polynomial-time reduction* from D to D'. (We write D \( \leq_p D' \))

The reduction maps instances I of D to instances I' of D' such that I is in D if and only if I' is in D'.

Examples of NP-complete problems:

- Graph Colouring
- Exact Maximal Bipartite Matching
A decision problem $D'$ is \textit{NP-complete} if

- $D'$ is in NP, and
- For all $D$ in NP, there is a \textit{polynomial-time reduction} from $D$ to $D'$. (We write $D \leq_p D'$)
A decision problem $D'$ is *NP-complete* if

- $D'$ is in NP, and
- For all $D$ in NP, there is a *polynomial-time reduction* from $D$ to $D'$. (We write $D \leq_p D'$)

If a problem $D'$ is NP-complete, this is strong evidence that $D'$ does not have a polynomial time algorithm: if it did, *all* problems in NP would have polynomial time algorithms.
A decision problem $D'$ is *NP-complete* if

- $D'$ is in NP, and
- For all $D$ in NP, there is a *polynomial-time reduction* from $D$ to $D'$. (We write $D \leq_p D'$)
A decision problem $D'$ is $NP$-complete if
- $D'$ is in $NP$, and
- For all $D$ in $NP$, there is a \textit{polynomial-time reduction} from $D$ to $D'$. (We write $D \leq_p D'$)

Claim: If $D$ is in $NP$, $D'$ is $NP$-complete, and $D' \leq_p D$, then $D$ is also $NP$-complete.
The proof follows easily by transitivity of polynomial time reductions ($\leq_p$).

This claim enables us to show that new problems of interest to us are $NP$-complete.