Outline

- Reductions to network flow
- Reductions between search and decision problems

Reductions to Network Flow Problem

- Bipartite Matching $\propto$ Network Flow
- The Gee-ball Problem $\propto$ Network Flow

Matching

- Let $G=(V,E)$ be a graph.

  - $E' \subseteq E$ is a matching if every vertex of $V$ is incident to at most one edge of $E'$.

Matching Example

Bipartite Graph

- Let $G=(V,E)$ be a graph.

  - $G$ is bipartite if $V$ can be partitioned into $V_1$ and $V_2$ such that no pair of vertices in $V_i$ $(i=1,2)$ have an edge.
Bipartite Example

G

G

CHECK

Bipartite Matching

• Input: Bipartite graph $G$
• Output: A largest matching of $G$

Bipartite Matching $\propto$ Network Flow

Algorithm for Problem Bipartite Matching

Algorithm for Problem Network Flow

Bipartite Matching $\propto$ Network Flow

• Transform Input – Step 1

Bipartite Matching $\propto$ Network Flow

• Transform Input – Step 2

Bipartite Matching $\propto$ Network Flow

• Transform Input – Step 2
Bipartite Matching $\propto$ Network Flow

- Transform output

Reduction

- Is it correct?
- Is it efficient?

Integrality theorem

- If the capacities in a network are integral, then the max flow can be achieved with integral flows on each edge.
- Further the Ford-Fulkerson method yields an integral solution.

Proof of correctness

There is a 1-1 correspondence between 0/1 flows in the network and matchings in the input graph.

Reduction

- Is it correct?
- Is it efficient?

Reductions to Network Flow Problem

- Bipartite Matching $\propto$ Network Flow
- The Gee-ball Problem $\propto$ Network Flow
The Gee-ball Problem

- The southwestern conference of the gee-ball league consists of \( n+1 \) teams. Team \( n+1 \) is from HMC.
- We want to know whether it is possible for HMC to win more games this season than any other team in the conference.
- No ties allowed.

Example

- The teams are Pitzer, CMC, Pomona, and HMC
- Games won so far:
  - Pitzer 4, CMC 3, Pomona 2, HMC 2
- Games to play:
  - 1 game: Pitzer vs. HMC
  - 2 games: Pomona vs. HMC

The Gee-ball Problem

- Teams \( t_1, t_2, \ldots, t_n, t_{n+1} \)
- So far this year team \( i \) has won \( w_i \) games.
- Teams \( i \) and \( j \) will play each other \( g_{i,j} \) more times this season (\( g_{i,j} = g_{j,i} \)).

Gee-ball \( \propto \) Network Flow

Transform Input

1. Create a source \( s \).
2. Create vertex \( v_{i,j} \) for \( 1 \leq i < j \leq n \).
3. Create edge from \( s \) to \( v_{i,j} \) with capacity \( g_{ij} \).
4. Create vertices \( u_i \) for \( 1 \leq i \leq n \).
5. Create edge from \( v_{i,j} \) to \( u_i \) and \( u_j \) with infinite capacity.
5. Create a sink $t$.
6. Create edge from $u_i$ to $t$ with capacity $W_i$.

Let $W = \sum_{i=1}^{n+1} \sum_{j=1}^{n+1} g_{n+1,i}$.

Transform Input

$V_{1,2}$

$V_{1,3}$

$V_{1,n}$

$u_1$

$u_2$

$u_{n-1}$

$u_n$

$t$

$W_{u_i,t}$

$W_{u_n,t}$

$W_{u_{n-1},t}$

$W_{u_2,t}$

$W_{u_1,t}$

$V_{t,1}$

$V_{t,2}$

$V_{t,n-1}$

$V_{t,n}$

$S_{1,1}$

$S_{1,2}$

$S_{1,n}$

$S_{1,t}$

$S_{2,1}$

$S_{2,2}$

$S_{2,n}$

$S_{2,t}$

$S_{n,1}$

$S_{n,2}$

$S_{n,n}$

$S_{n,t}$

Algorithm for Gee-ball

Algorithm for Network Flow

Input

transform

Output

Gee-ball $\propto$ Network Flow

Transform Output

There is a way for HMC to win the season if and only if

Decision Problems

Decision problems are computational problems for which the output is "yes" or "no"

Search: Vertex Cover

• Input: Graph $G$
• Output: A smallest subset of vertices $W$ such that every edge of $G$ is incident to some vertex in $W$. 
**Decision: Vertex Cover**

- **Input:** Graph $G$ and integer $K$
- **Question:** Is there a subset $W$ of $K$ or fewer vertices of $G$ such that every edge is incident to some vertex in $W$?

**Polynomial Time Reductions**

- Decision-VC $\preceq_p$ Search-VC
- Search-VC $\preceq_p$ Decision-VC

**Independent Set**

- **Input:** Graph $G$ and integer $K$
- **Question:** Is there a subset $W$ of $K$ or more vertices of $G$ such that no two vertices of $W$ are adjacent?
- **Does Search-IS $\preceq_p$ Decision-IS?**

**3-Colorability**

- **Input:** Graph $G$
- **Question:** Can the vertices of $G$ be colored with Red, Green, and Blue so that no adjacent vertices have the same color?
- **Does Search-3Col $\preceq_p$ Decision-3Col?**