Reductions to Network Flow Problem

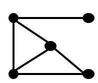
- Bipartite Matching $\propto$ Network Flow

Matching

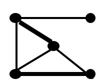
- Let G=(V,E) be a graph.
- E' $\subseteq$ E is a matching if every vertex of V is incident to at most one edge of E'.

Matching Example

G



E'

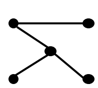


Bipartite Graph

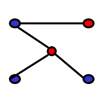
- Let G=(V,E) be a graph.
- G is bipartite if V can be partitioned into V_1 and V_2 such that no pair of vertices in V_i (i=1,2) have an edge.

Bipartite Example

G



G

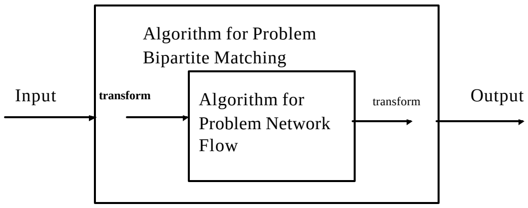


CHECK

Bipartite Matching

- Input: Bipartite graph G
- Output: A maximum matching of G

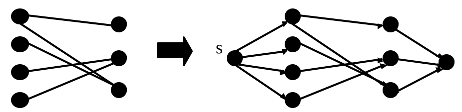
Bipartite Matching ∝ Network Flow



CS140 27-7

Bipartite Matching ∝ Network Flow

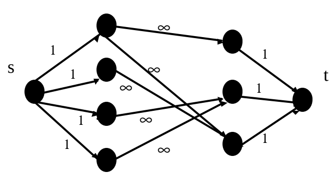
- Transform Input – Step 1



CS140 27-8

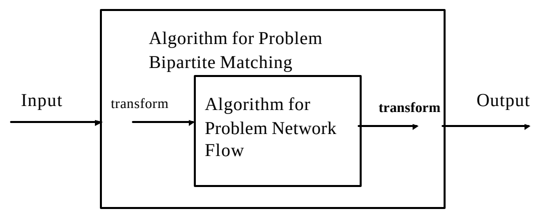
Bipartite Matching ∝ Network Flow

- Transform Input – Step 2



CS140 27-9

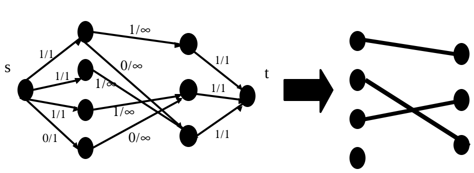
Bipartite Matching ∝ Network Flow



CS140 27-10

Bipartite Matching ∝ Network Flow

- Transform output



CS140 27-11

Reduction

- **Is it correct?**
- Is it efficient?

CS140 27-12

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 graph:

The flow into/out of a node in the network is 1 if and only if the corresponding node in the graph is incident to an edge of the matching.

Reduction

- Is it correct?
- **Is it efficient?**

$T_{BM}(n)=cn+T_{FF}(n+2)$

Reductions to Network Flow Problem

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

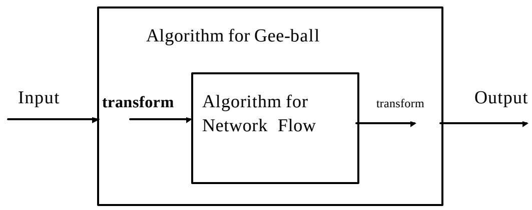
The Gee-ball Problem

- 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.

The Gee-ball Problem

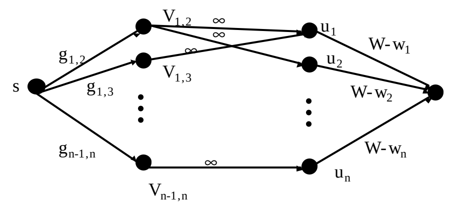
- So far this year team i has won w_i games.
- Teams i and j will play each other $g_{i,j}$ more times this season ($g_{i,j}=g_{j,i}$). There are no ties; each game ends with one winner.

Gee-ball ∝ Network Flow

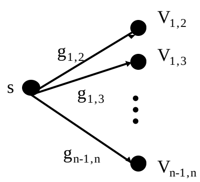


Transform Input

W+1=number of games HMC will win by the end of the season if it wins all of its remaining games

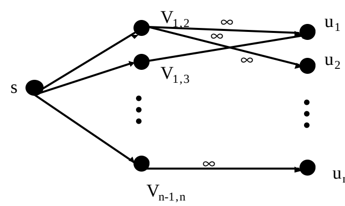


Transform Input



1. Create a source s and sink t.
2. Create vertex $v_{i,j}$ for $1 \le i < j \le n$
3. Create edge from s to $v_{i,j}$ with capacity $g_{i,j}$

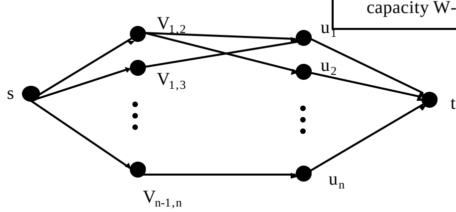
Transform Input



4. Create vertices u_i, $1 \le i \le n$
5. Create edge from $v_{i,j}$ to u_i and u_j with infinite capacity.

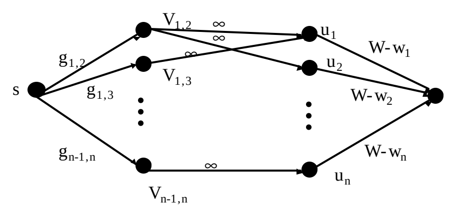
Transform Input

6. Create edge from u_i to T with capacity $W-w_i$.

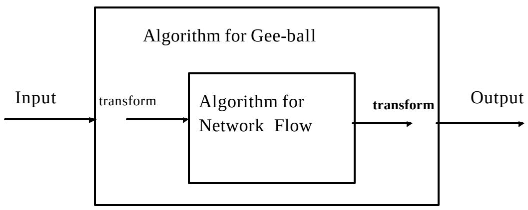


Transform Input

W+1=number of games HMC will win by the end of the season if it wins all of its remaining games



Gee-ball ∝ Network Flow



Transform output

- If max flow equals $\Sigma_{1\le i<j\le n} g_{i,j}$ then YES
- Else NO

Reduction

- Is it correct?
- Is it efficient?