Announcements

- HW #20:
  - Due now.

- HW #20.5
  - Due Monday

- Project #6: Linear Programming
  - In preparation: HW #21
    - Formulate problem for Proj. #6
    - NO 2-hour time limit
    - Due Wednesday

Objectives

- Better understand the idea of “changing the problem” in the max flow algorithm
- Understand the relationship between flows and cuts
- Solve Bipartite Matching
- See example of Padded matrix-vector representation

Max Flow Example

1. Pick a simple path \(s \rightarrow t\) and increase the flow along that path as much as possible
2. Create the residual graph \(G^r\) including reverse edges
3. Repeat until there is no path with remaining capacity.

Solve

Summarize the max flow algorithm.
Max Flow Example

Key Points about our Max Flow Algorithm
- When we transform the problem we create "imaginary edges"
- However, in the end our selected paths always produce a flow schedule that involves only the original edges.

Cut Property
- Max flow is less than or equal to the flow across any cut.
- Max-Flow Min-Cut Theorem: The maximum flow through a network is equal to the minimum flow across all cuts.
Algorithms

- Our focus has been on gaining intuition toward understanding the Simplex algorithm.
- The general algorithm for solving max. flow:
  - Ford-Fulkerson
  - An implementation thereof:
    - Edmonds-Karp: $O(V \cdot E^2)$

Match-making

Match-making

Is there a perfect matching?

Bipartite Matching

Is there a perfect matching?

LP: **Padded** Matrices & Vectors

```
Max 3x1 + 2x2 = Z
Subject to:
3x1 + 2x2 <= 5
5x1 + 4x2 <= 6
x1, x2 >= 0

A = [1 2]
B = [3 2]
C = [4 5]
D = [6 7]
```

End
Assignment

- **HW #20.5**
  - Due Monday

- **HW #21:**
  - Part 1 of Project #6
  - Just the problem formulation and representations
  - No time limit
  - Due Wednesday