Network Design: Network Loading and Pup Matching

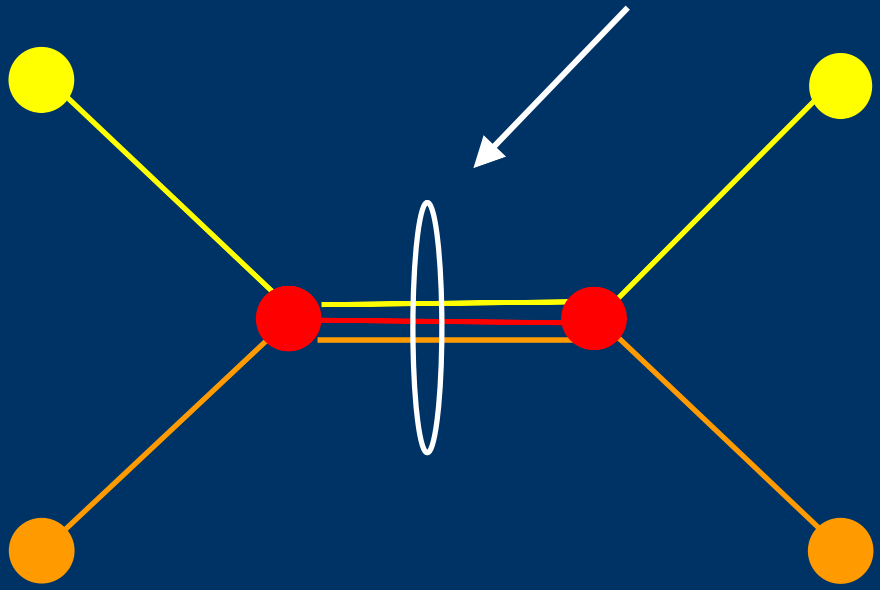
Thomas L. Magnanti

Today's Agenda

- Network design in general
- Network loading
- Solution approaches
  - Polyhedral combinatorics
  - Heuristics
- Pup matching

Network Design: Basic Issue

Total (Fixed) Cost on Each Arc



Commodity k:
- Origin O(k)
- Destination D(k)
- Flow req. r^k

Link (i,j):
- Fixed cost F_{ij}
- Flow cost c_{ij}

Possibly:
- Capacity C per unit installed on any edge

Network Design Applications

- Telecommunications systems
- Airline route maps
- Chip design
- Facility location
- Even TSP!

Multicommodity Flow Model with Complex Costs

$$\text{minimize } c(f)$$

$$\text{subject to } Nf^k = b^k \quad \text{for } k = 1, 2, \ldots, K$$

$$f = \left(f^1, f^2, \ldots, f^K\right) \geq 0$$

$$b_i^k = \begin{pmatrix} r^k \text{ if } i = O(k) \\ -r^k \text{ if } i = D(k) \\ 0 \text{ otherwise} \end{pmatrix}$$

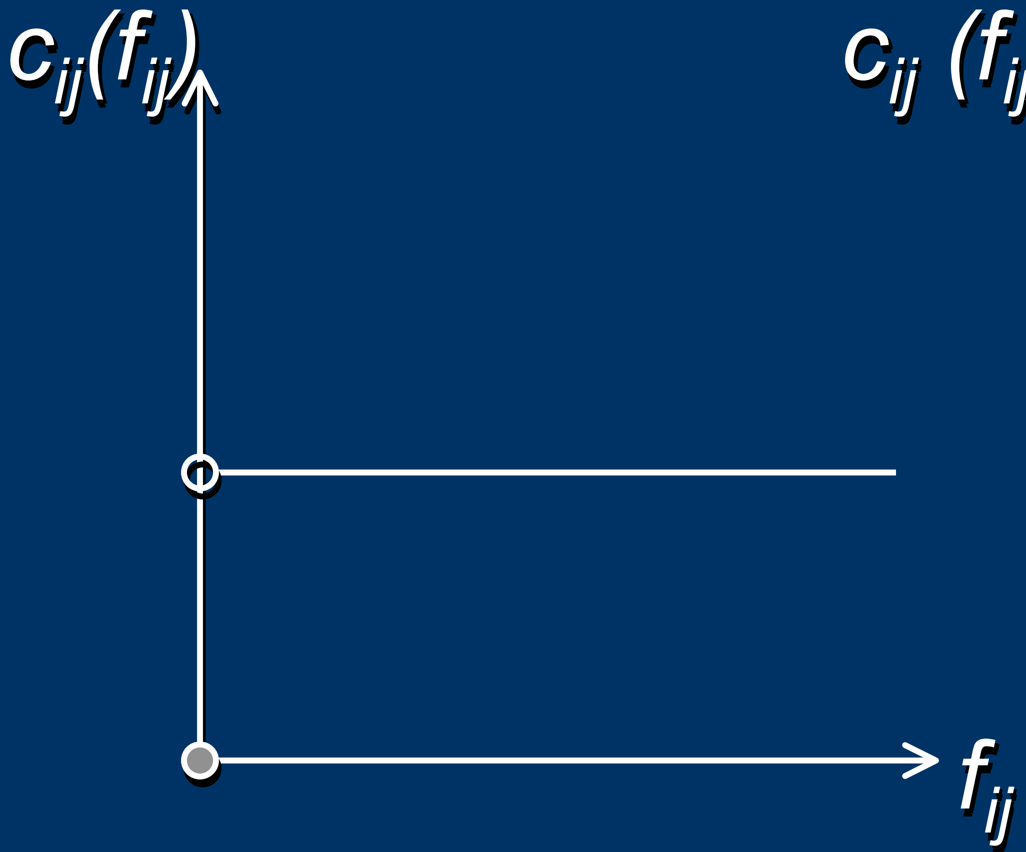
(possible flow bounds on f_{ij}^k)

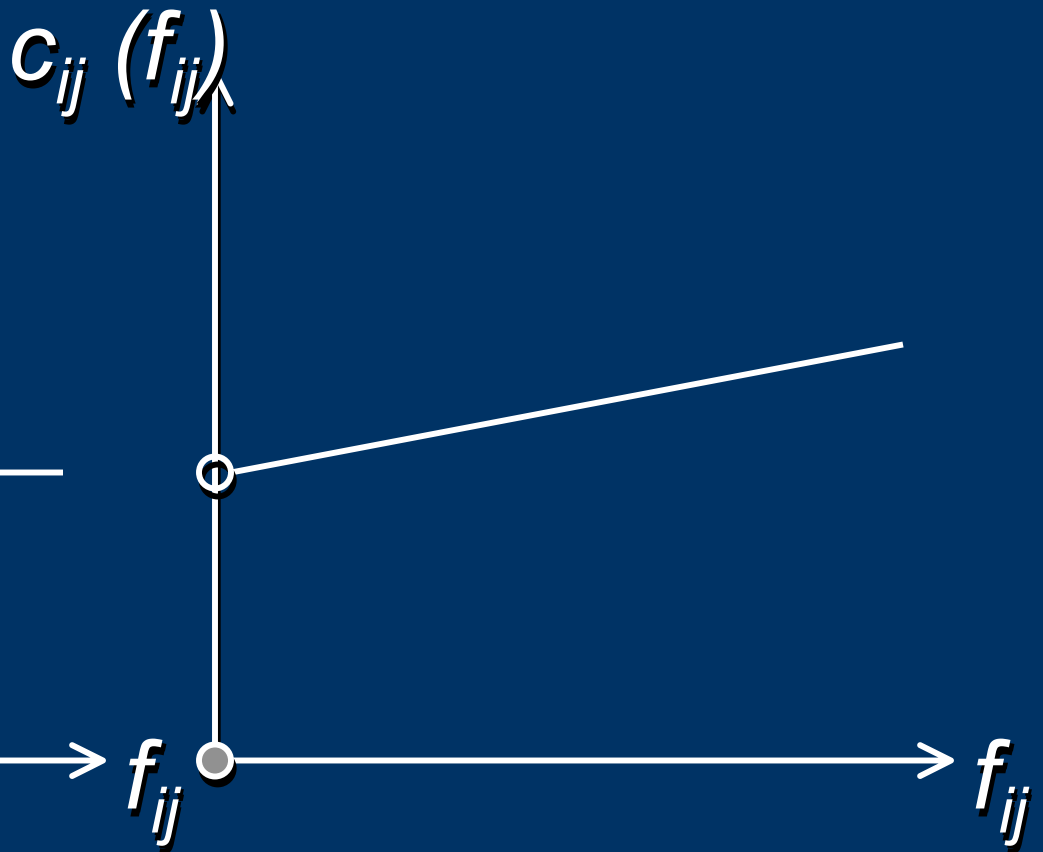
$$c(f) = \sum_{(i,j)\in A} c_{ij}\left(f_{ij}\right) \quad \text{separability}$$

$$c(f) = \sum_{(i,j)\in A} l_{ij} c\left(f_{ij}\right) \quad \text{proportionality}$$

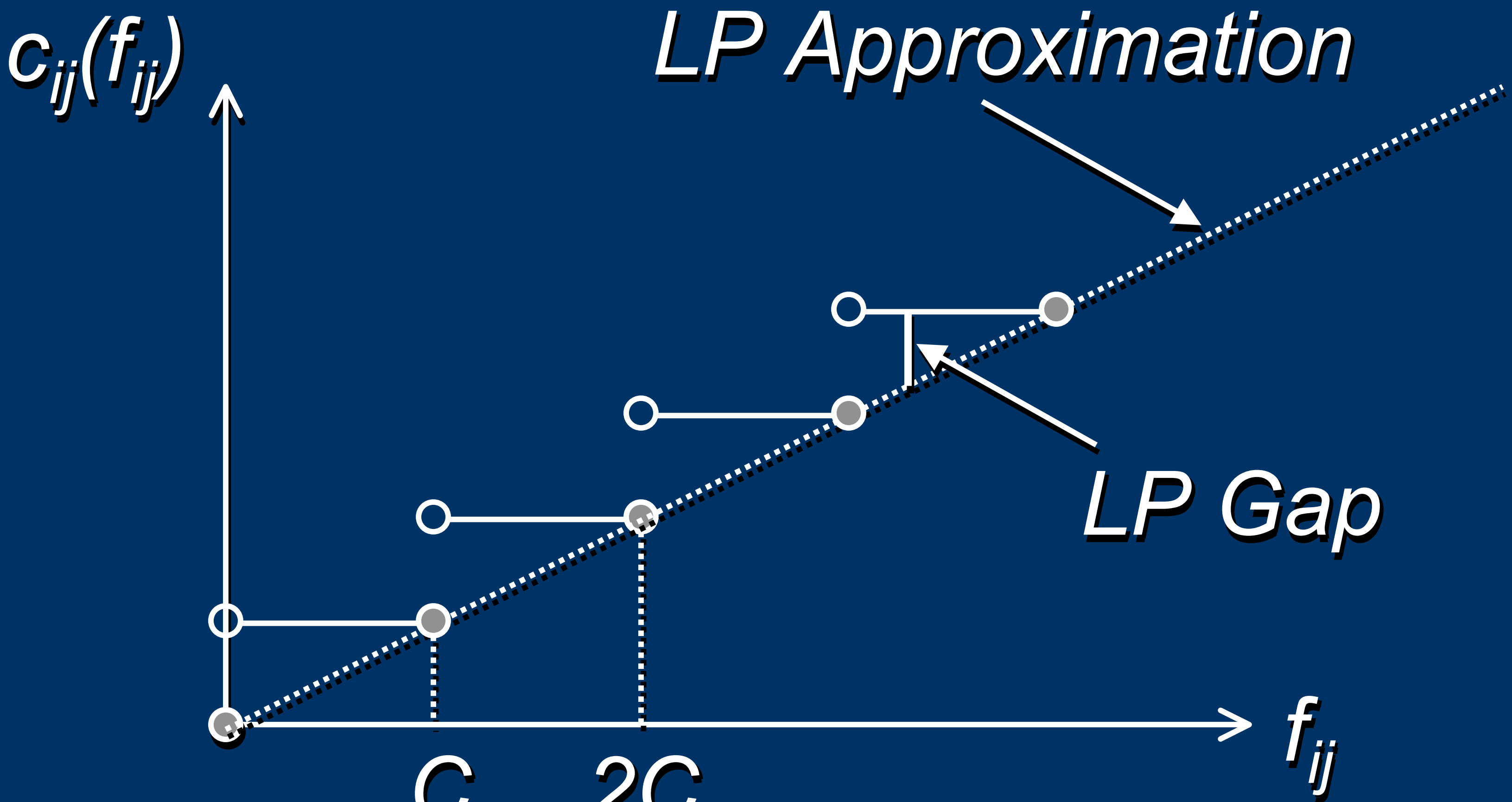
$$f_{ij} = \sum_k f_{ij}^k \quad \left(f_{ij} = \sum_k w^k f_{ij}^k\right)$$

Basic Cost Structures

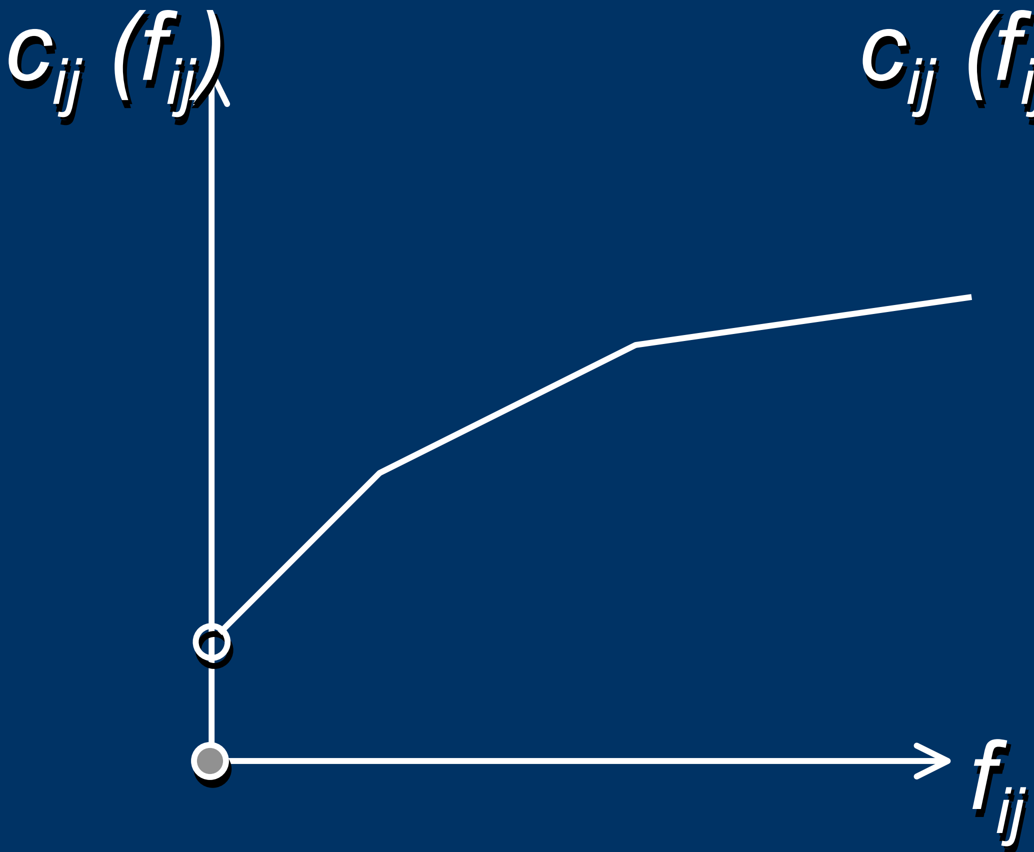


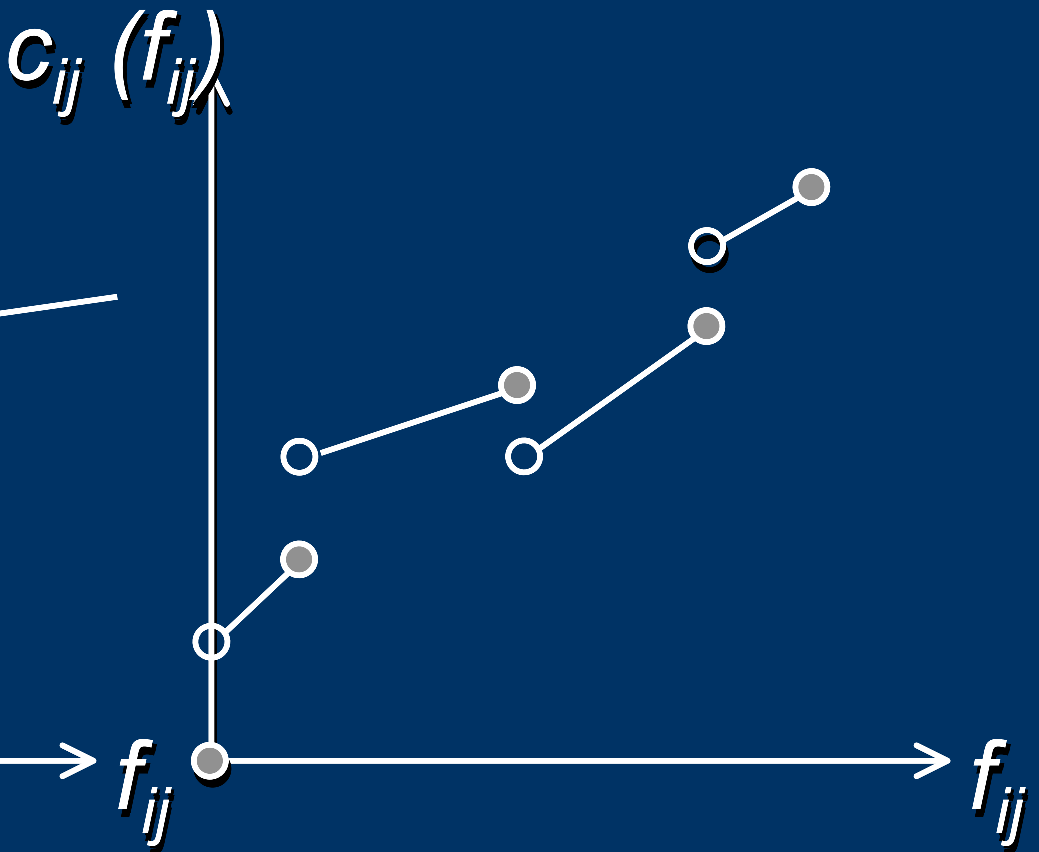


Network Loading Cost



Other Cost Structures





Integer Programming Model

$$\text{minimize } \sum_k c^k f^k + \sum_{(i,j)\in E} F_{ij} y_{ij}$$

$$\text{subject to } Nf^k = b^k \quad k = 1, 2, \ldots, K$$

$$\sum_k \left(f_{ij}^k + f_{ji}^k \right) \le C y_{ij} \quad \{i, j\} \in E \qquad (1)$$

$$f_{ij}^k \le \overline{r}^k y_{ij} \quad \{i, j\} \in E, \text{ all } k \qquad (2)$$

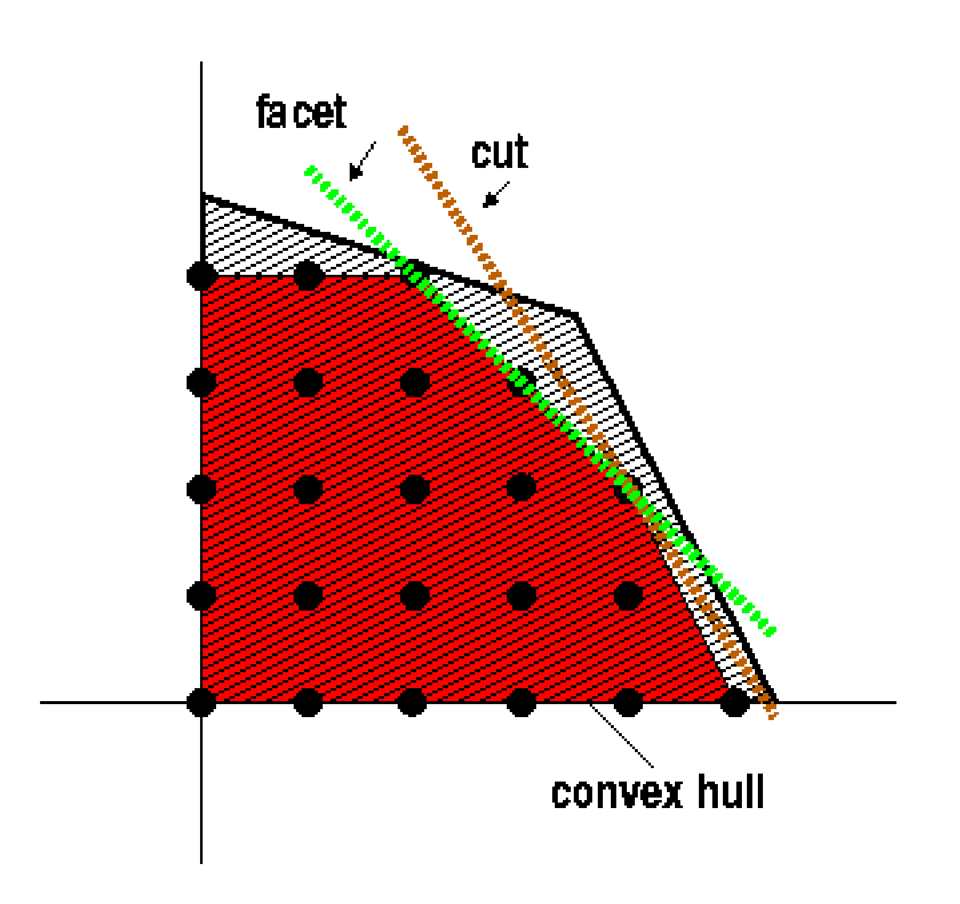
$$f = \left(f^1, f^2, \ldots, f^K \right) \ge 0$$

$$y_{ij} \ge 0 \text{ and integer all } \{i, j\} \in E$$

(configuration constraints and y)

Cuts for Lower Bounds

- LP relaxations yield lower bounds
- Addition of cuts can tighten bounds
  - Cut away solutions to the LP relaxation but leave all feasible integer points



Network Loading Model

$$\min \sum_{\{i,j\}\in E} F_{ij} y_{ij}$$

subject to

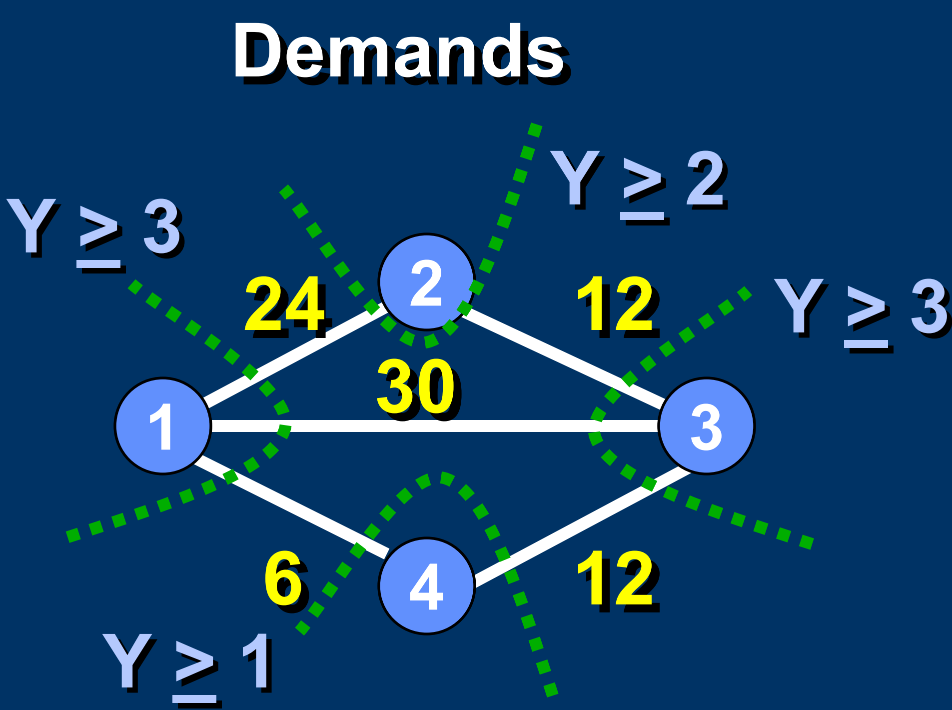
$$\sum_{\{j:(i,j)\in A\}} f_{ij}^k - \sum_{\{j:(j,i)\in A\}} f_{ji}^k = \begin{cases} 1, & i = O(k) \\ -1, & i = D(k) \\ 0, & \text{otherwise} \end{cases} \quad \text{all } k \in K, i \in N$$

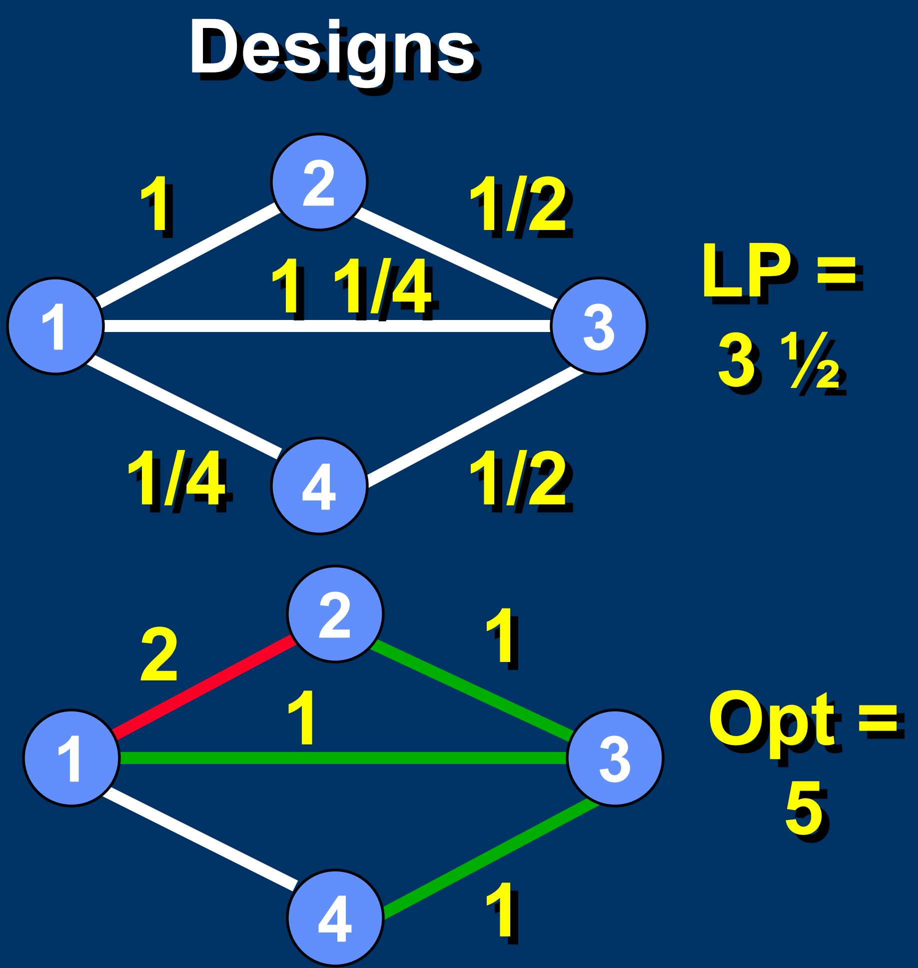
$$\sum_{k\in K} \left(f_{ij}^k + f_{ji}^k \right) \le C y_{ij} \quad \text{all } \{\text{i,j}\} \in \text{E}$$

$$f_{ij}^k \ge 0, y_{ij} \ge 0 \text{ and integer}$$

Improved Modeling (Cutset Inequalities)

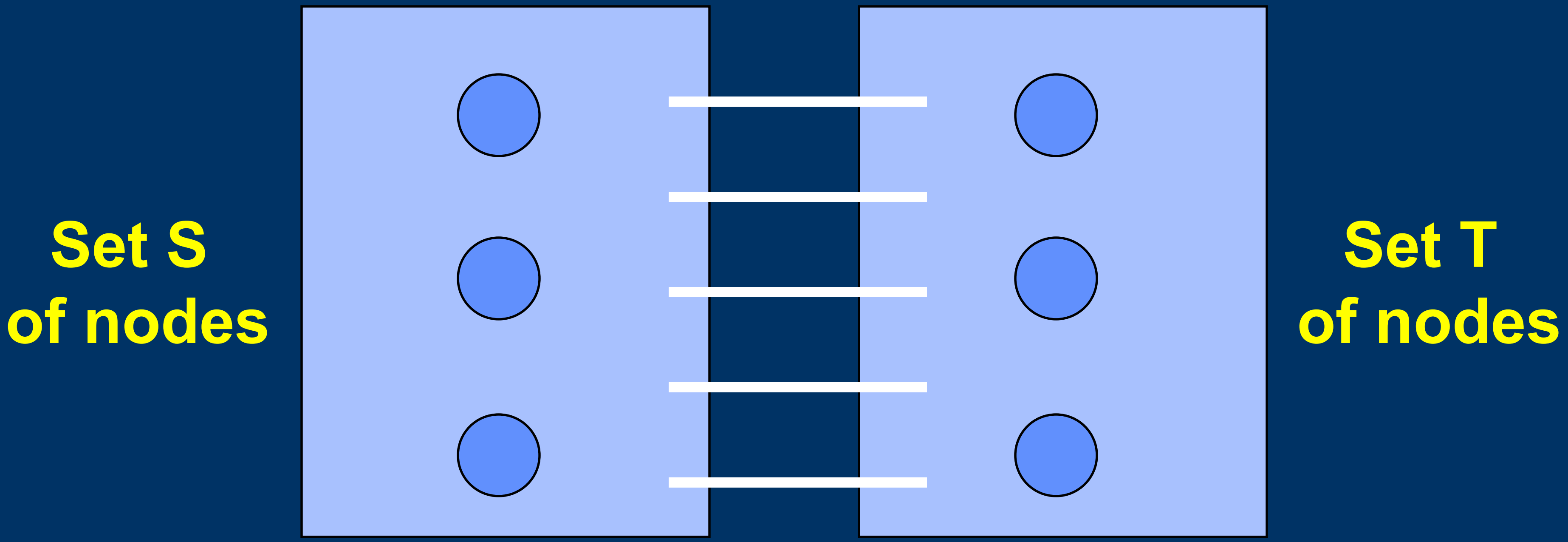
$F_j = 1$ all edges, all $c_{ij} = 0$
$C = 24$





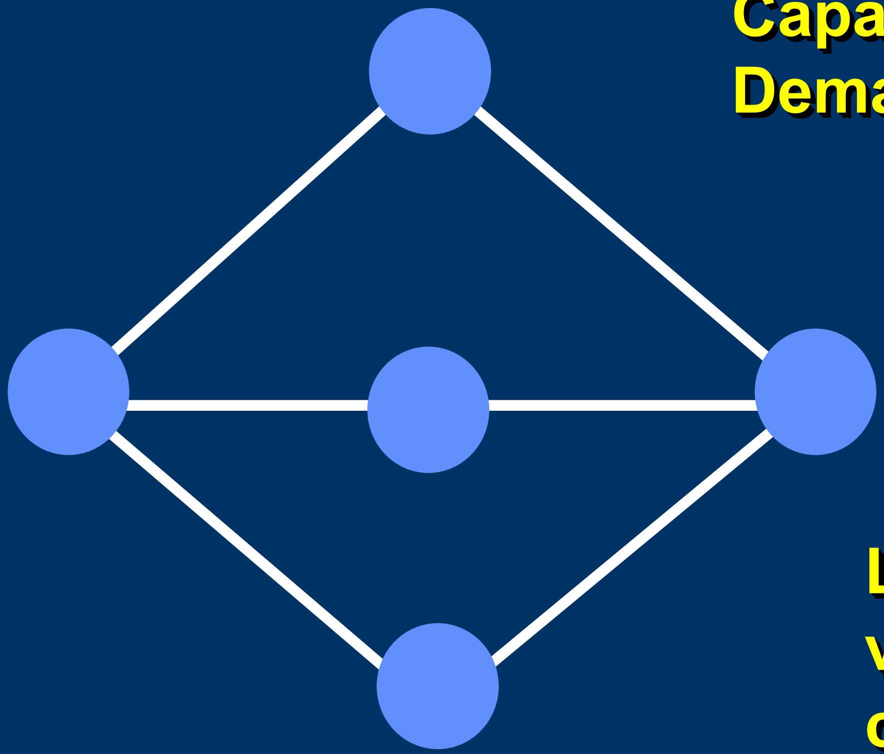
General Cutset Inequality

D_{ST} = total demand (nodes in S to nodes in T)



$$Y_{ST} \geq \left\lceil \frac{D_{ST}}{C} \right\rceil$$

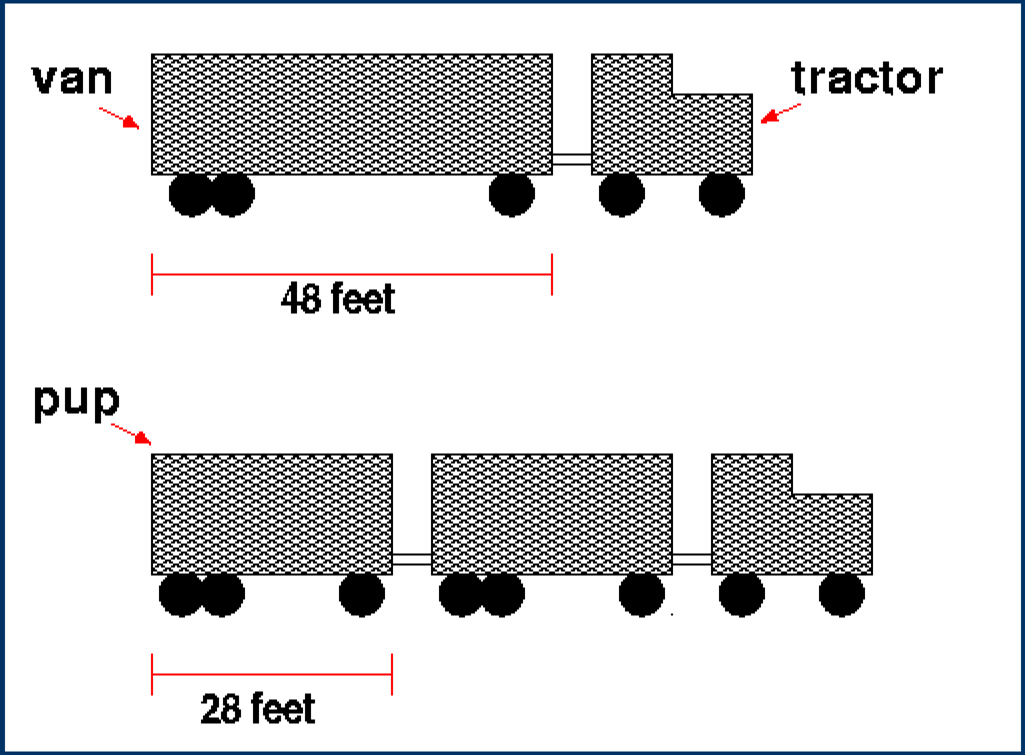
Cutset Inequalities Aren’t Sufficient



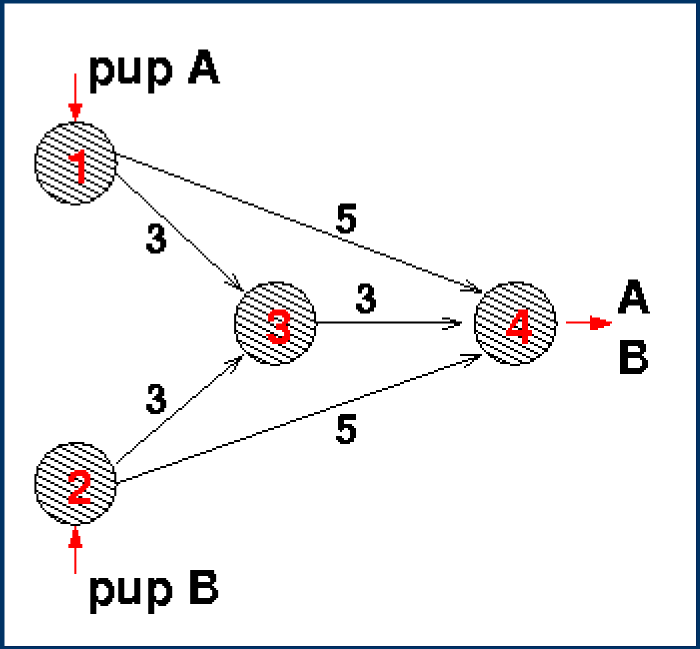
Capacity C = 1
Demand = 1 between all non-adjacent nodes

Loading 1 unit on all visible edges satisfies cutset inequalities, but not feasible

Pup Matching



Example

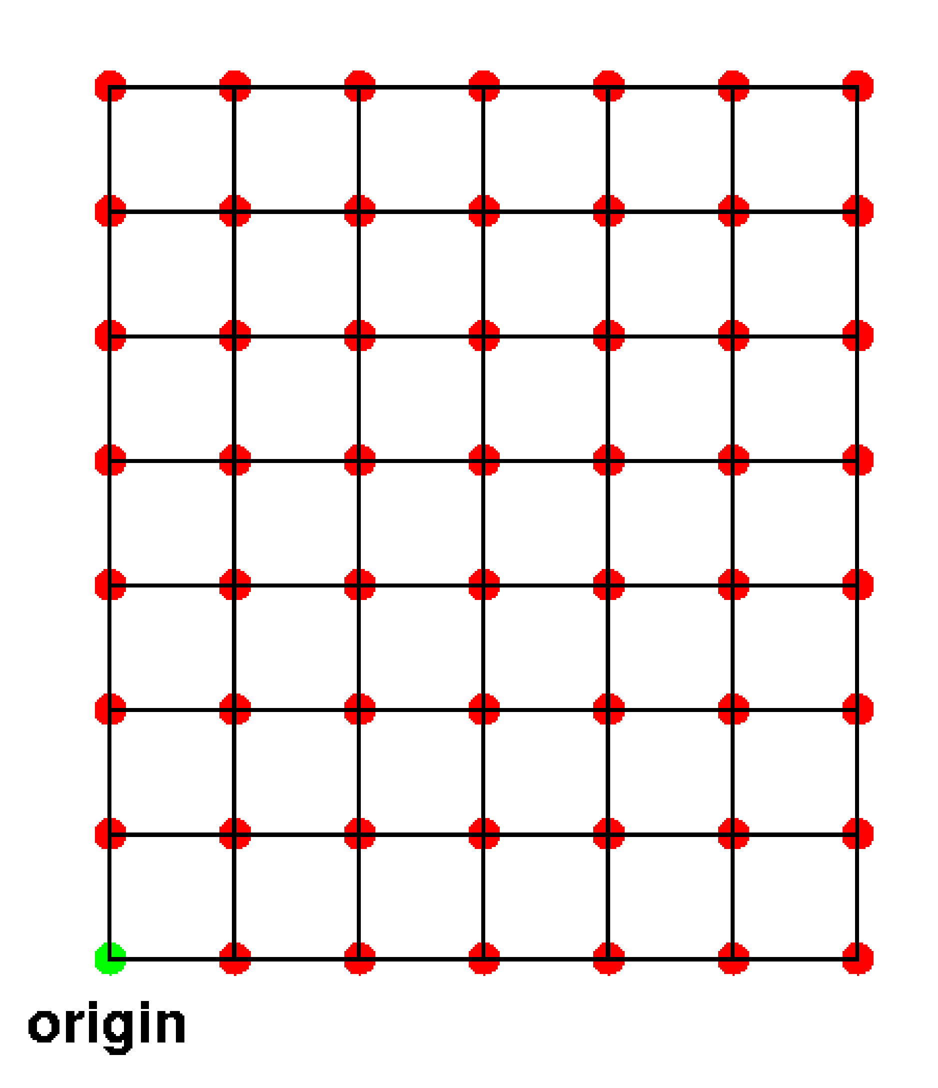


optimal solution is 9

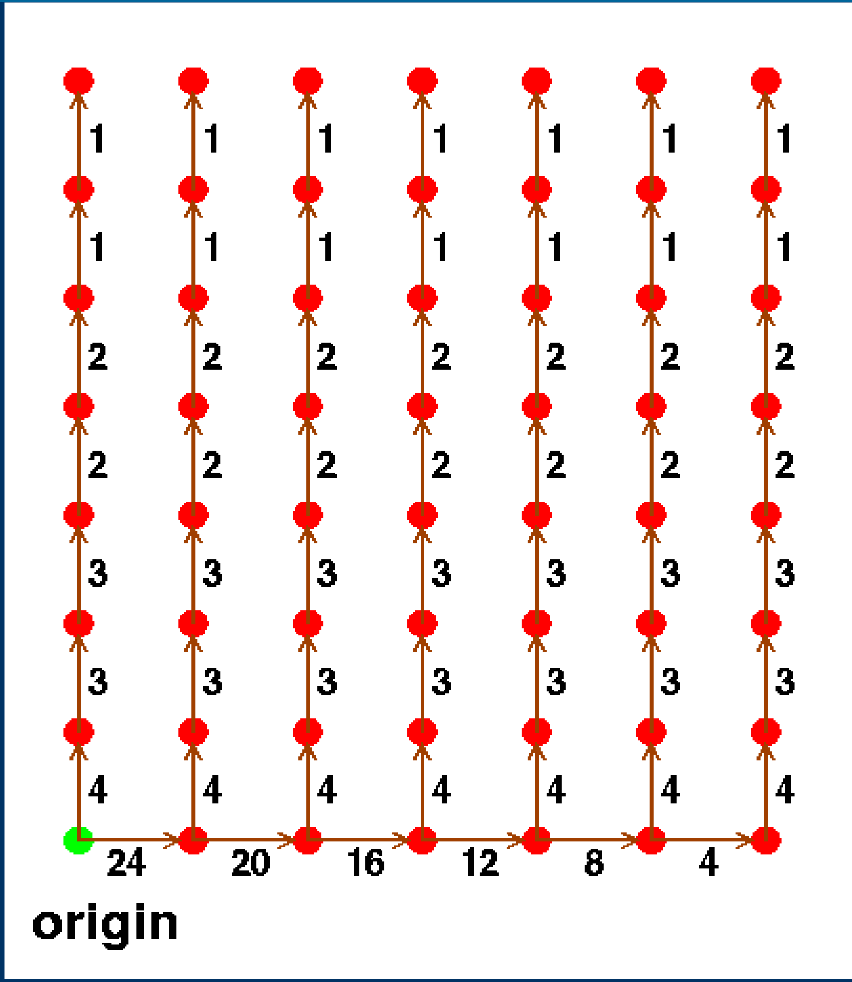
Pup Matching

- **Instance:** A directed network $G = (N,A)$, a set of K pairs of elements from N, and a cost function $c: A \rightarrow R+$.
- **Problem:** Find the minimum cost loading of G permitting unit flow from the first to the second node of each of the K pairs such that 1 unit or 2 units together can traverse an arc for each unit of loading. One unit of loading on $a \in A$ costs $c(a)$.

Example on City Blocks



Solution with cost 196



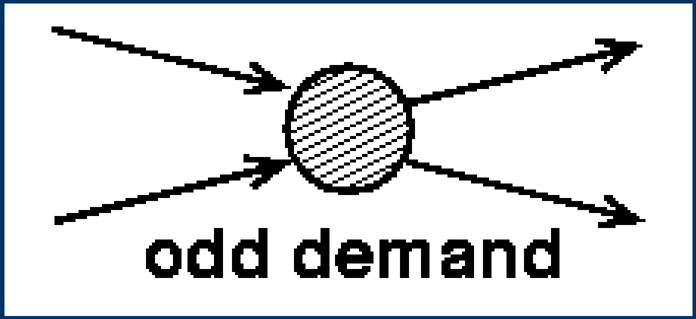
Several days computation can prove only that the objective is at least 184 (LP lower bound = 182). Can we do better?

Heuristics for Upper Bounds

- Matching Heuristic
  - permits each pup to be paired with at most one other pup
  - solved with a weighted matching routine
- Shortest Path Heuristics
  - three variations
- Each heuristic provides a 2-approximation to the NLP formulation

Odd Flow Inequality

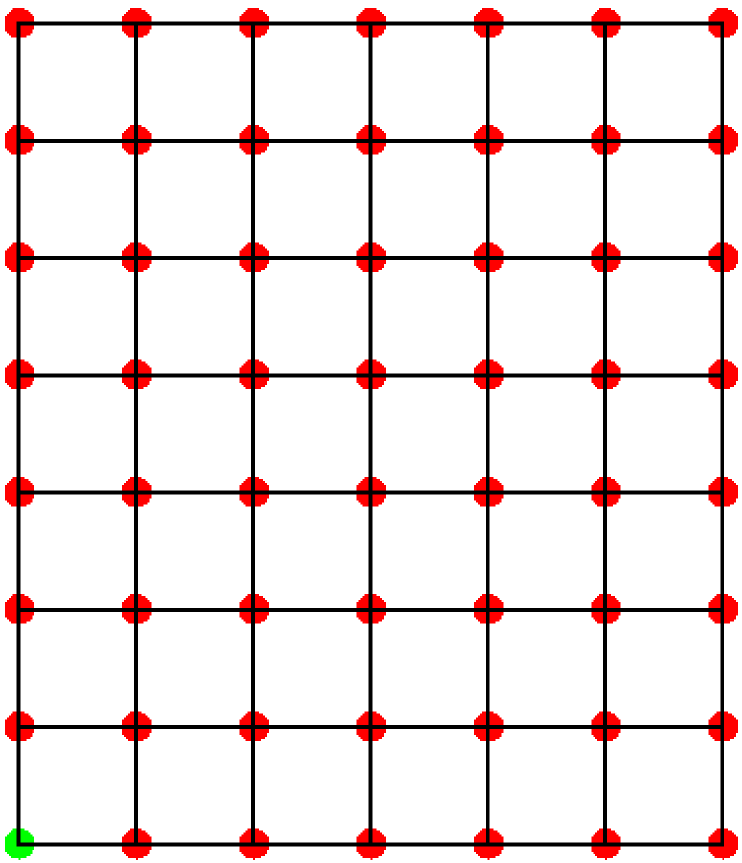
- If the flow on an arc is odd, one unit of loaded capacity will be unused



- If the net demand of a node is odd, the total inflow or total outflow is odd

$$\sum_{j}\left(y_{ij}+y_{ji}\right)-\frac{1}{2}\left(\sum_{k}\sum_{j}\left(f_{ij}^{k}+f_{ji}^{k}\right)\right)\geq\frac{1}{2}$$

Odd Flows on the City Block



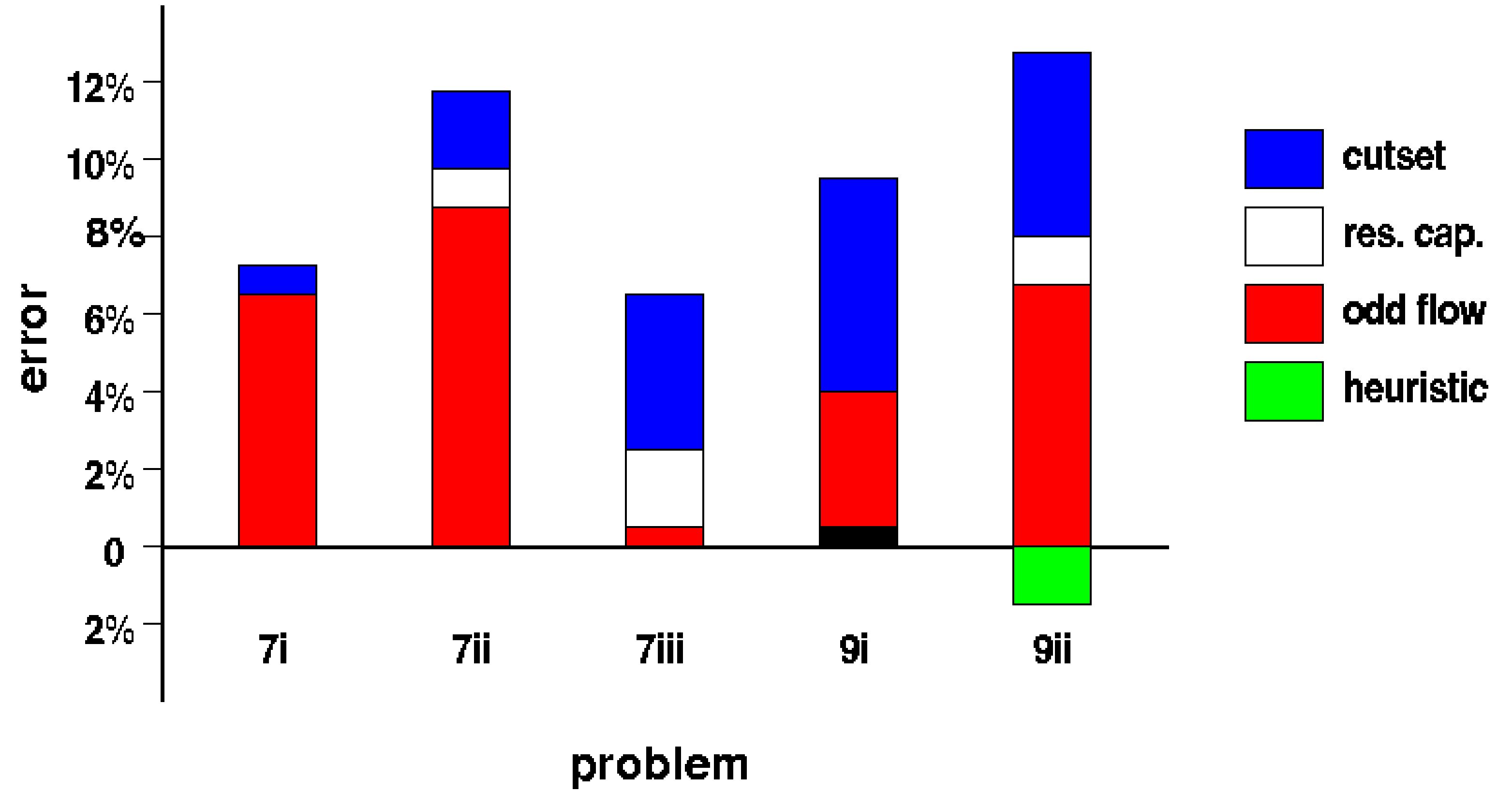
Each of the 56 nodes must be incident to at least one arc with $\frac{1}{2}$ unit of spare capacity

⇓ solution requires at least $\left\lceil \frac{1}{2} \cdot \frac{56}{2} \right\rceil = 14$ cabs' worth of empty capacity

LP relaxation of 182 gives lower bound on required used capacity

⇓ 196 is optimal

Gap Reductions on City Block Problems



Trials Using Realistic Data

- Node set given in (latitude, longitude) format based on a real logistics network
- Defined problems by choosing a subset of nodes, calculating arc lengths, and randomly selecting O-D pairs
- 30 problems, about half single origin
- Complete graphs, 12-25 nodes, 6-50 pups

Results

- Branch and Bound limited to 2 hours CPU time and a 220M tree
- With all 3 cut families, 67% were solved to optimality with an average gap reduction of 18.8% to 6.4%
- Without odd flow cuts, 30% were solved, and the gap was reduced to 7.8% on average
- With no cuts, 17% were solved
- Among solved problems, average heuristic error was 1.3%

Conclusions and Extensions

- Extensions apply to compartmentalized problems
- Cuts seem critical to provably solving the PM problem
- Odd flow inequalities define what seems an important set of facets
  - generalize to arbitrary capacity
  - can generalize to several facilities?
- Are there other cuts based on even-odd type arguments?

Many, Many Network Design Variants

- Network loading for compartmentalized capacity
  - airline capacity planning, tanker trucks
- Network survivability
- Network restoration
- Hierarchical designs
- …

Today's Lessons

- Network design arises in numerous applications
- Problem is a large-scale integer program
- Introduction to cutting planes (polyhedral combinatorics)
- Cutting planes valuable in tightening formulations and in problem solving

Module on Large-Scale Integer Programming & Combinatorial Optimization

Three Lectures

- Traveling salesman problem
- Facility location
- Network design

Games/Challenges

Applications, Models, and Solution Methods