Module on Large-Scale Integer Programming & Combinatorial Optimization

Three Lectures

- Traveling salesman problem
- Facility location
- Network design

Games/Challenges

Applications, Models, and Solution Methods

Traveling Salesman Problem

Thomas L. Magnanti

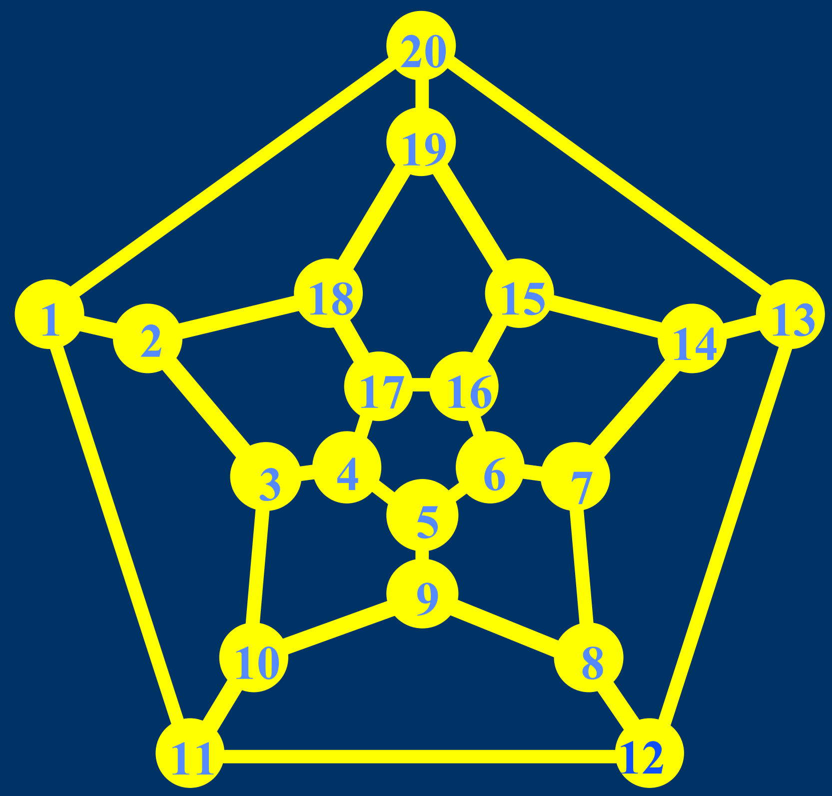
Agenda

- Origins
- Electronic Component Placement
- TSP Model
- Turbine Vane Placement
- Other Applications of TSP
- Solution Methods
  - Heuristic Methods
  - Lagrangian relaxation (bounding methods)
- Some large scale instances (computations)

William Rowan Hamilton

Icosian game

Icosian game



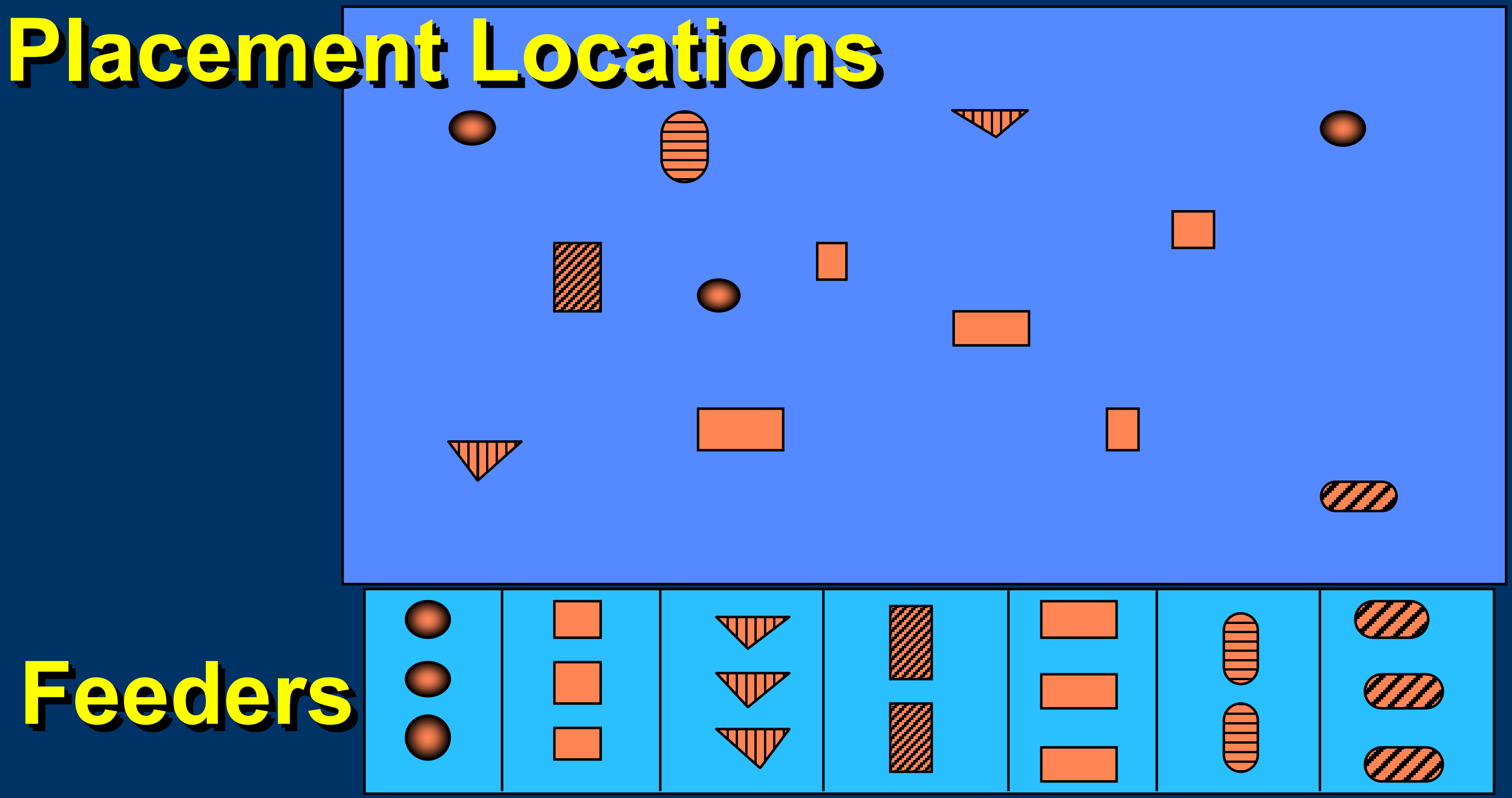
Hamiltonian Path and the TSP

Interest in Traveling Salesman Problem (TSP)

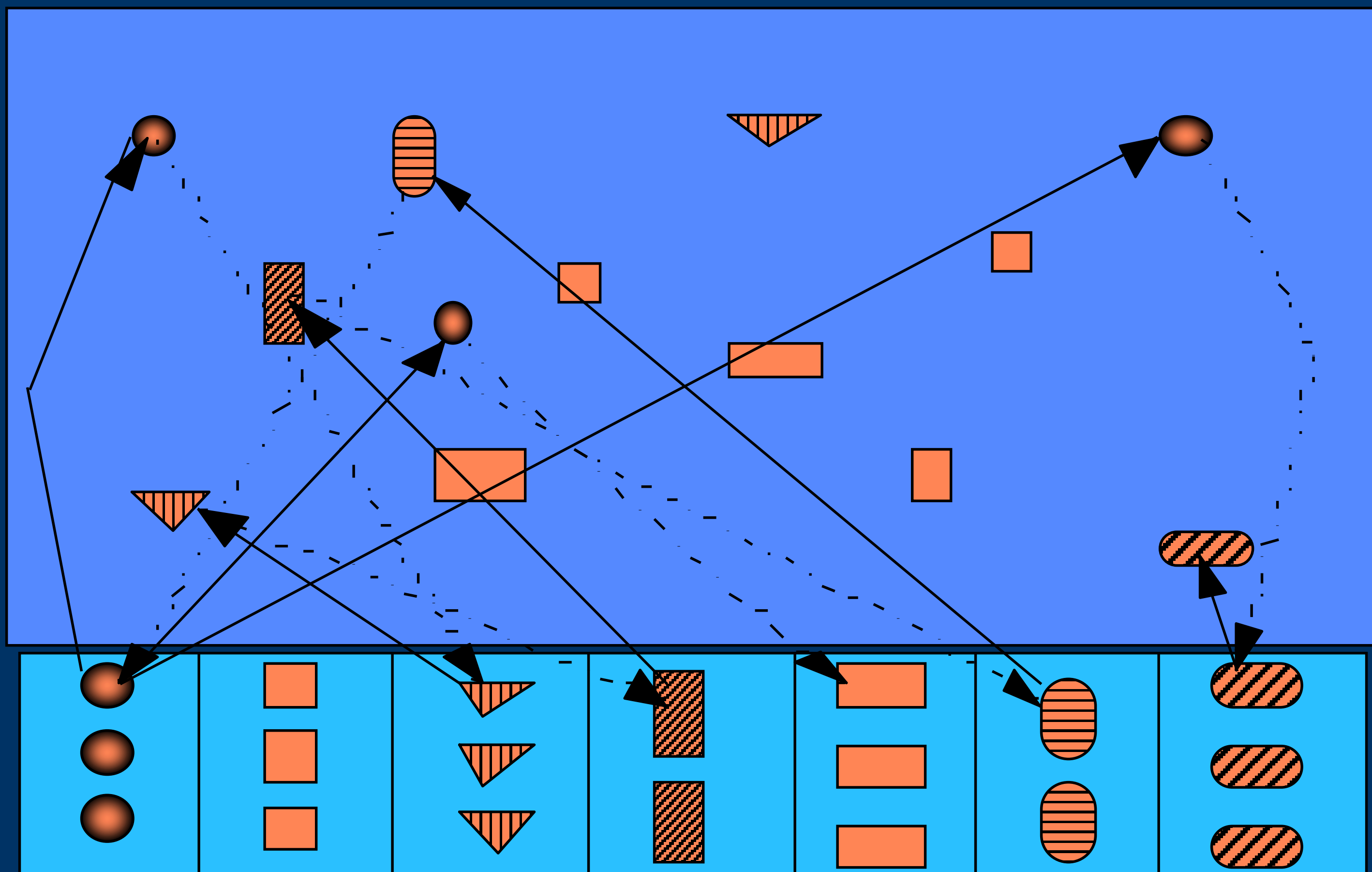
- Arises in Many Applications
- Alluring (Captures Imagination)
- Notoriously Difficult to Solve
- Has Attracted Best Minds in Math/CS/OR for 40 Years

The Traveling Salesman Problem and Electronics Assembly

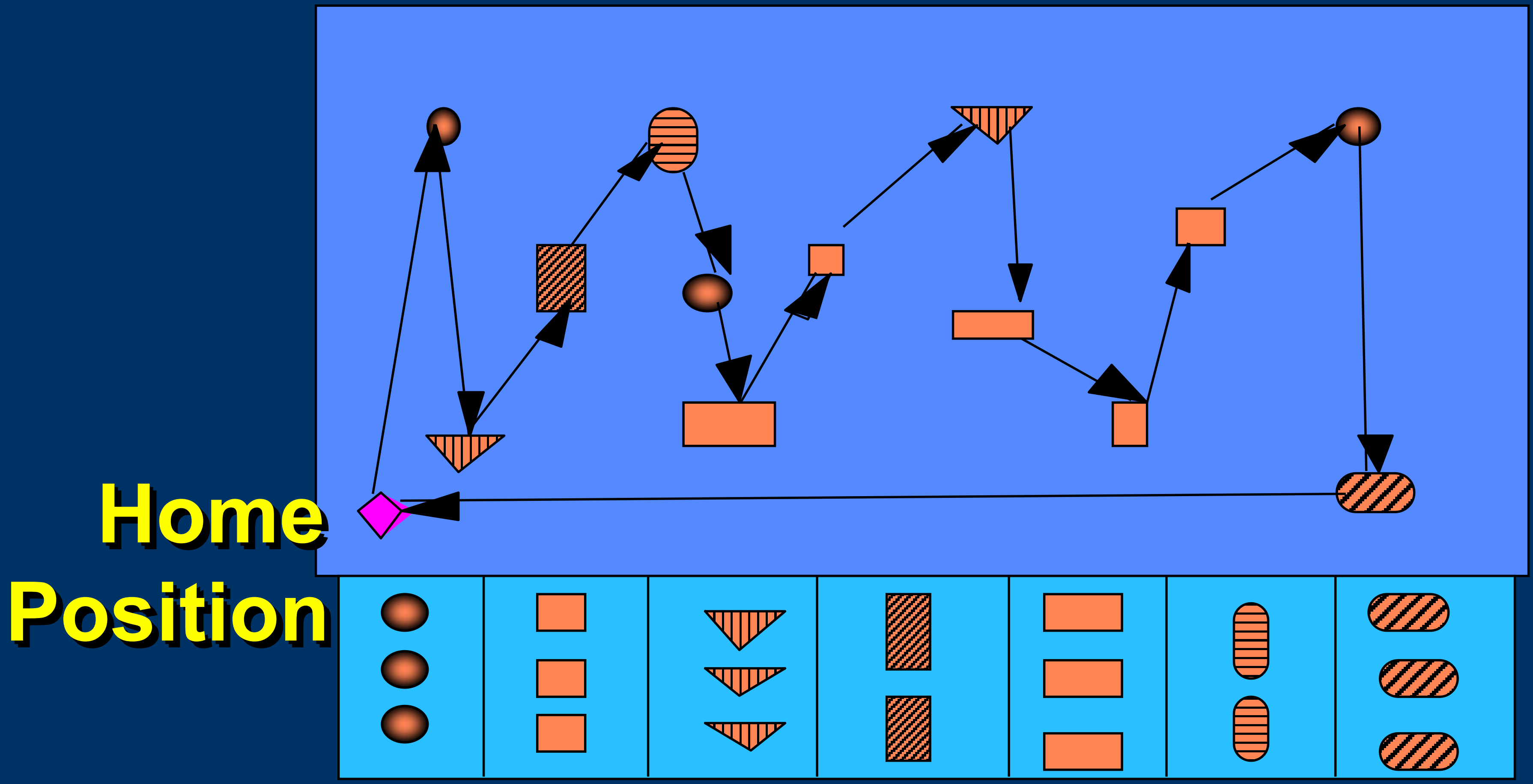
Placement Problem



Placement Sequence



Traveling Salesman Interpretation



Model Ingredients

c_{jk} cost of placing module k after placing module j

x_{jk} =1 if placement k follows placement j

0 otherwise

Assignment Problem

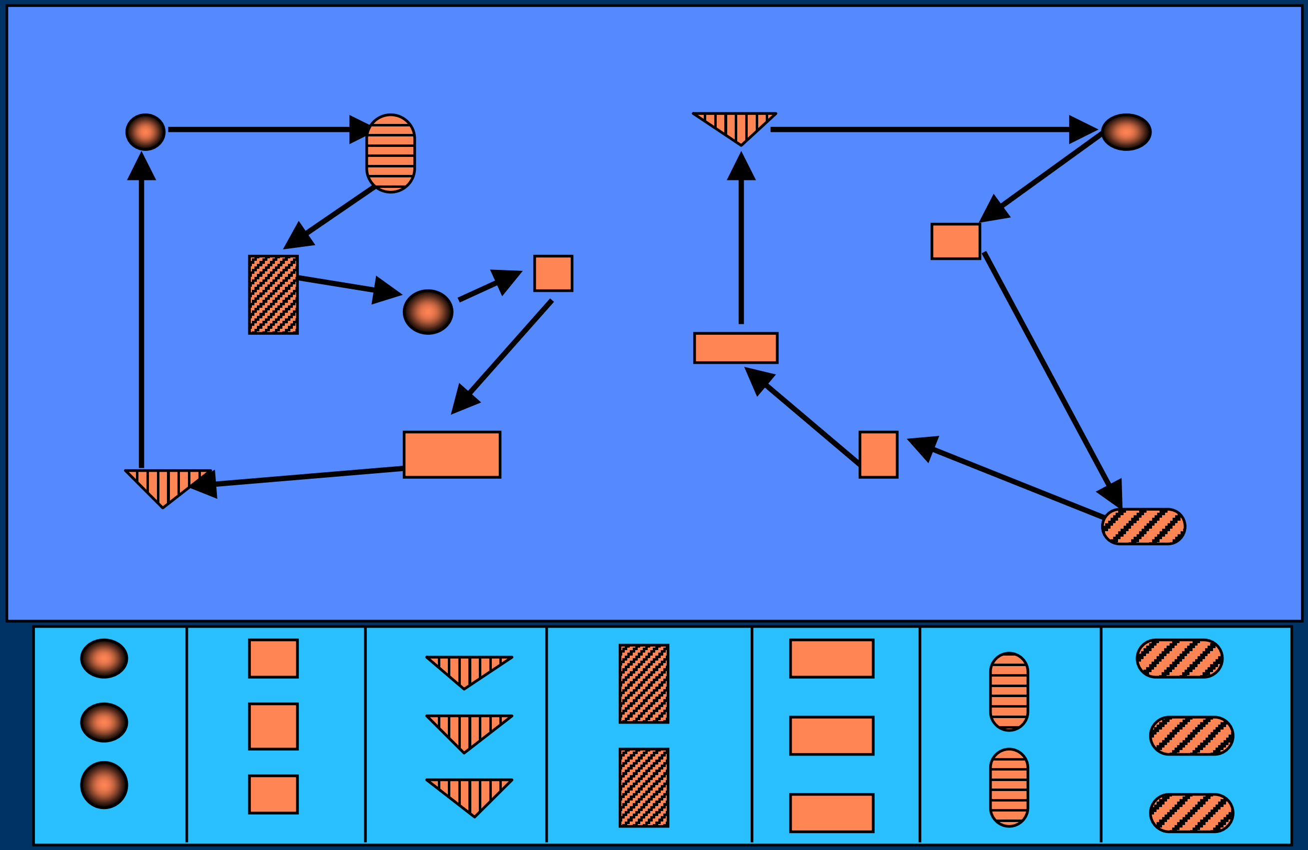
Minimize $\Sigma_j \Sigma_k c_{jk} x_{jk}$

subject to

$$\Sigma_k x_{jk} = 1 \text{ for each } j$$
$$\Sigma_j x_{jk} = 1 \text{ for each } k$$
$$x_{jk} \geq 0 \text{ for all } j \;\&\; k$$

Proper TSP Model?

Subtour Solution



TSP Model

Assignment Model

+

Subtour Breaking Constraints

$\Sigma_{j \in S} \Sigma_{k \in S} x_{jk} \leq |S| - 1$ **for all subsets S of nodes {2,3,…,n}**

Proper TSP Model?

Implications for IC Insertions

- Manual Designs $\Rightarrow$ Long Time
  - 10 hours for 70 to 100 components
- Better Solutions
  - 10-25% improvements by optimization

Other Applications Similar

Feeder Placement

- Modeling?
- Solution Methods?
  - Heuristic
  - Optimization

Other Applications of TSP?

Other Applications of TSP

Machine Scheduling

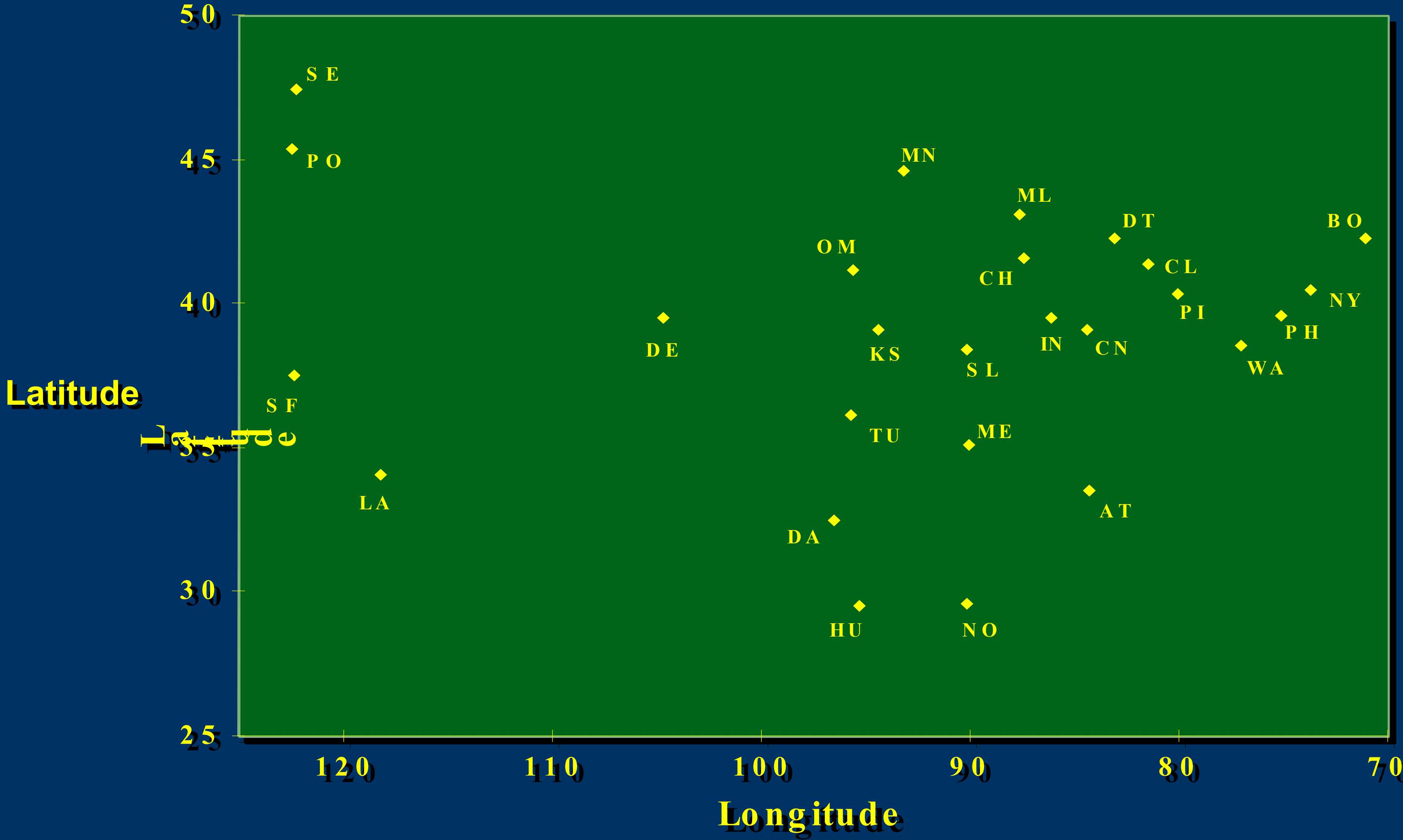
Machine "Visits" Jobs
Travel Time = Set up Time

Other Applications

- Analyzing the structure of crystals
- Material handling in a warehouse
- Clustering of data arrays
- Cutting stock problems
- Genome sequencing
- Starlight interferometer satellite positioning
- DNA universal strings
- Collecting coins from payphones

Solving the TSP

26 City Traveling Salesman Problem



Finding a Good Solution

- How?
- How good is good?
  - LP bounds
  - Combinatorial bounds

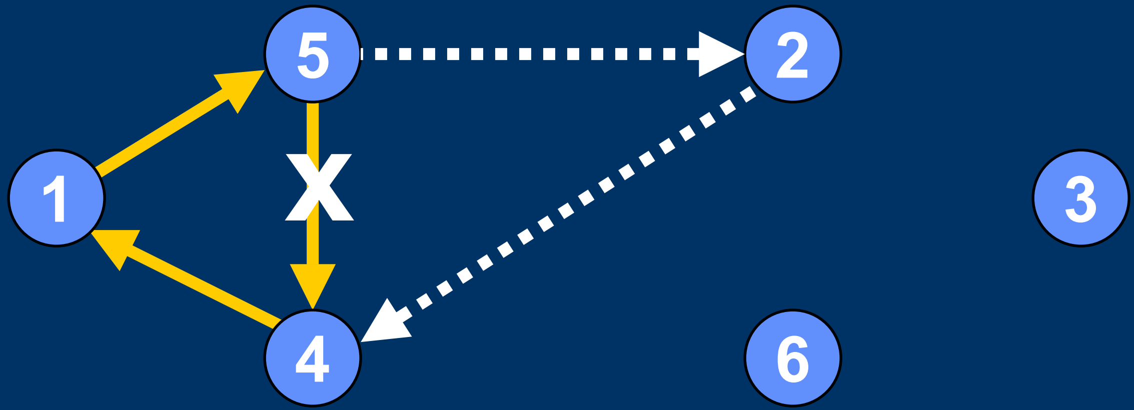
Solution Methods

- Heuristics
  - Growing solutions: nearest neighbor, farthest neighbor, nearest insertion
  - Improvement procedures: 2-opt, 3-opt
- Optimization Methods
  - Bounding: LP relaxation, Lagrangian dual
  - Polyhedral methods (cutting planes)

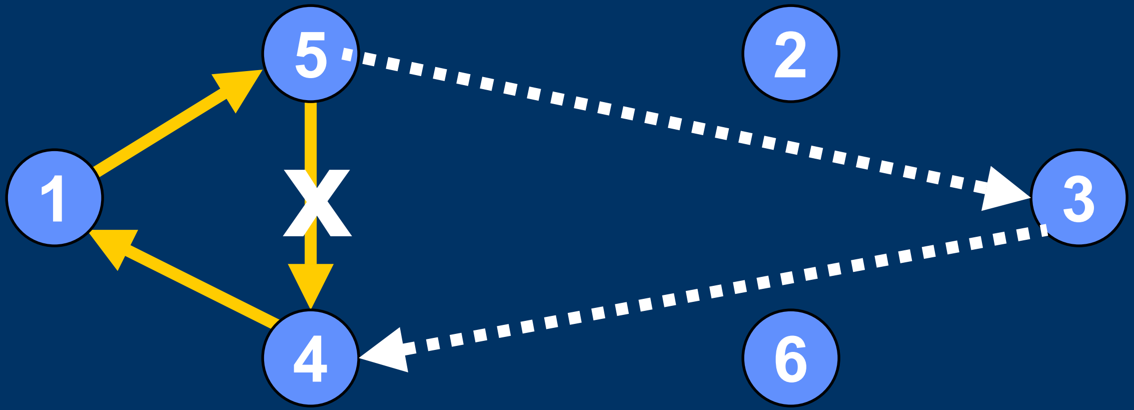
Heuristics

- Build Tour
  - Nearest Neighbor
  - Nearest/Farthest Insertion
- Improve Tour
  - Swapping Edges

Insertion Heuristics

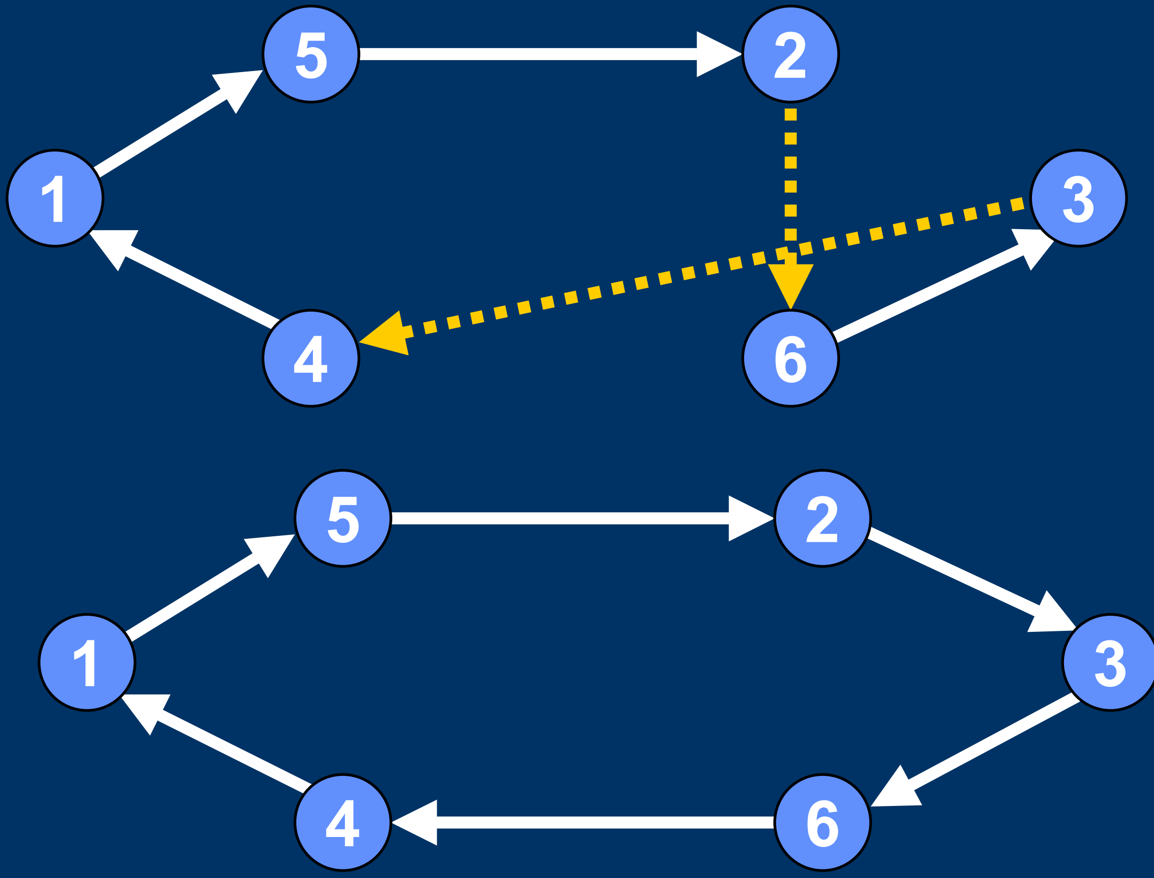


Nearest Insertion



Farthest Insertion

Tour Improvements



2-opt

Eliminate 2 arcs and reconnect

Choose best alternative at each step

Exploiting Embedded Structure

$$\text{minimize } \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\text{subject to } \cancel{\sum_{j=1}^{n} x_{ij} = 1 \text{ for all } i = 1, 2, \ldots, n}$$

$$\cancel{\sum_{i=1}^{n} x_{ij} = 1 \text{ for all } j = 1, 2, \ldots, n}$$

$$\sum_{i=2}^{n} \sum_{j=2}^{n} x_{ij} = n - 2$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \text{ for all } S \subseteq \{2, 3, \ldots, n\}$$

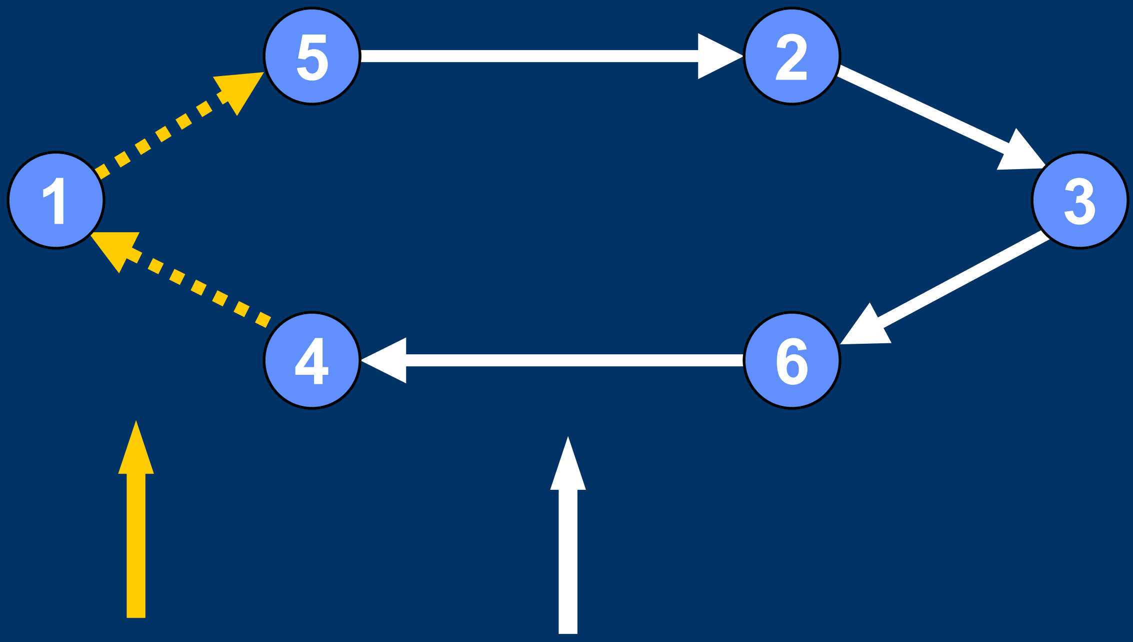
$$x_{ij} \geq 0 \text{ and integer}$$

Redundant Constraint

Minimum spanning tree on nodes 2 to n

Underlying Structure

Decomposed Tour



Arc in and out of node 1

Path (hence tree) on nodes 2 to n

Lagrangian Relaxation

$$L(u,v) \;=\; \text{minimize} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=2}^{n} u_i \left[\sum_{j=1}^{n} x_{ij} - 1 \right]$$

$$+ \sum_{j=2}^{n} v_j \left[\sum_{i=1}^{n} x_{ij} - 1 \right]$$

subject to $\sum_{j=1}^{n} x_{1j} = 1$

$$\sum_{i=1}^{n} x_{i1} = 1$$

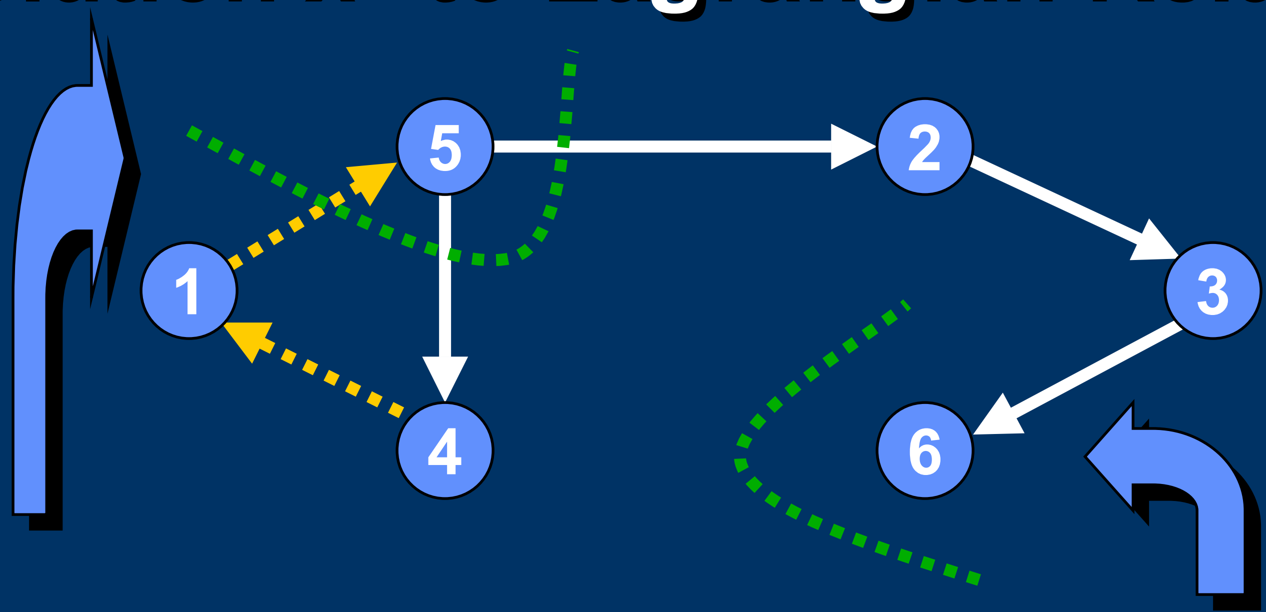
$$\sum_{i=2}^{n} \sum_{j=2}^{n} x_{ij} = n-2$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \le |S| - 1 \text{ for all } S \subseteq \{2, 3, \ldots, n\}$$

$$x_{ij} \ge 0 \text{ and integer}$$

Improving Lagrangian Lower Bound

Solution *x** to Lagrangian Relaxation



Too many arcs out
Increase u_5
$c_{5j} + u_5 + v_j$ more expensive

Too few arcs out
Decrease u_6
$c_{6j} + u_6 + v_j$ less expensive

Solution Approach (Dual Ascent)

- Solve $L(u,v)$
  - Use costs $c_{ij} + v_i + u_j$
  - Select least cost arc out of and into node 1
  - Find minimal spanning tree on nodes 2 to n (easy)
- Let x^* be optimal solution to $L(u,v)$
- Using x^* alter u and v to increase lower bound $L(u,v)$
- Iterate to solve Lagrangian dual

$$d = \max_{u,v} L(u,v)$$

Dual Ascent in General

$$v^* = \text{minimize } \; cx$$

$$\text{subject to } Ax = b$$

Complicating Constraints

$$x \in X$$

e.g., $X = \{x : Dx = d, x \geq 0\}$

Lagrangian Dual

$$L(u) = \text{minimize } \; cx + u[Ax - b]$$

$$\text{subject to } \; x \in X$$

Dual Ascent in General

$$\text{If } L(u^k) = cx(u^k) + u^k\left[Ax(u^k) - b\right]$$

$$\nabla L(u^k) \approx Ax(u^k) - b$$

$$u^{k+1} = u^k - \theta_k\left[Ax(u^k) - b\right]$$

$$\theta_k = \frac{\lambda_k\left[cx(u^k) - v^*\right]}{\left\|Ax(u^k) - b\right\|^2}$$

Dual Ascent Convergence

Theorem

If $0 < \varepsilon_1 \le \lambda_k \le 2 - \varepsilon_2$

and $\left\| Ax(u^k) - b \right\|$ **are bounded,**

then $c(x^k)$ **converges to** v^*.

In practice, continue with fixed λ_k except half λ_k after some number (50?) of iterations if $c(x^k)$ doesn't decrease

Solving Minimum Spanning Tree

Greedy (Kruskal Algorithm)

- Order arcs from smallest to highest costs
- Choose arcs in order
  - If arc does not form an undirected circuit with arcs already chosen, then choose arc;
  - Otherwise eliminate arc from consideration

Amazing Facts!

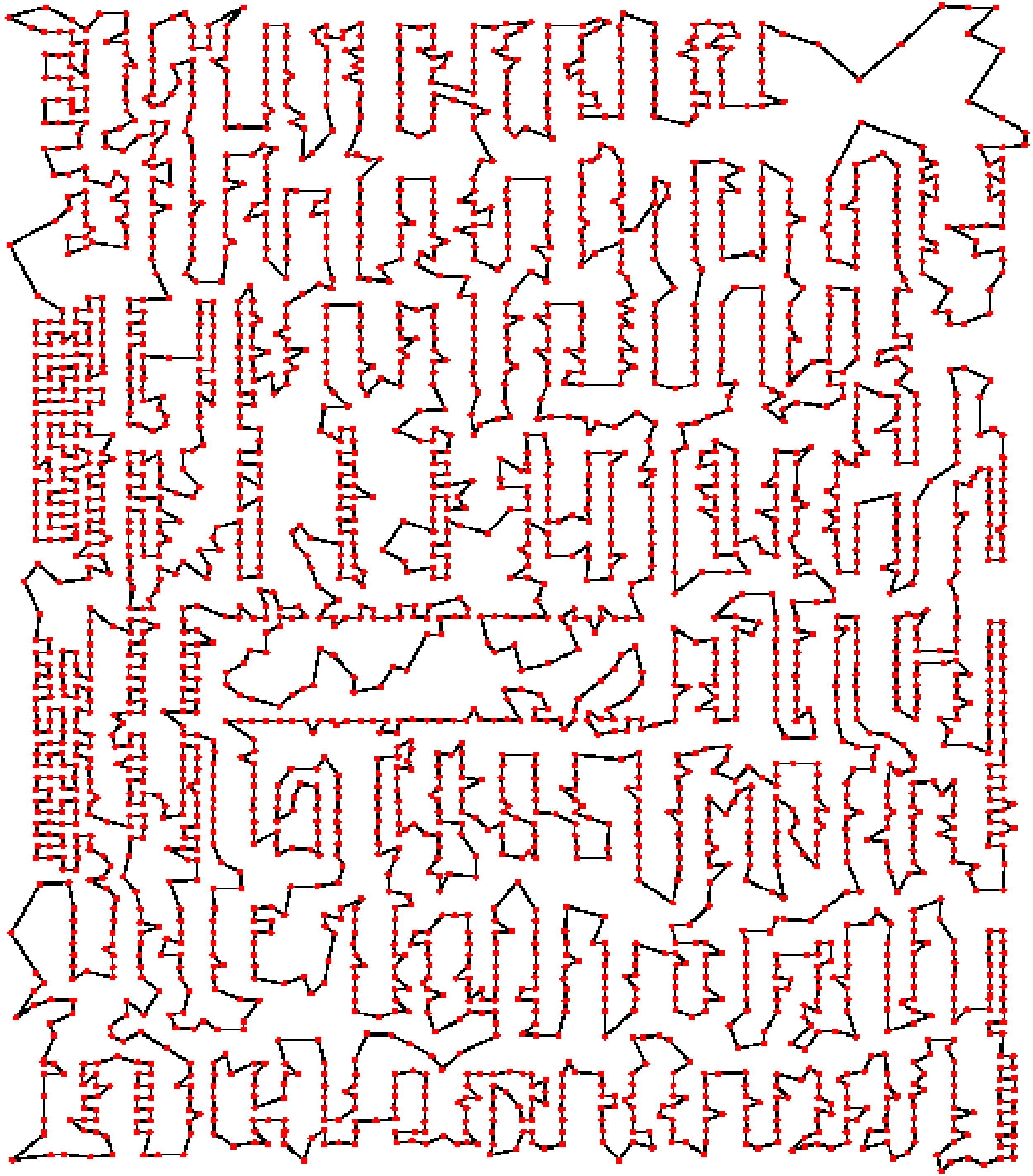
- Greedy algorithm (and several variants) solves the minimum spanning problem
- The linear programming relaxation of the formulation we have given for the minimum spanning problem always has an integer solution (the underlying polyhedron has integer extreme points)

Solving Network Flow Problems

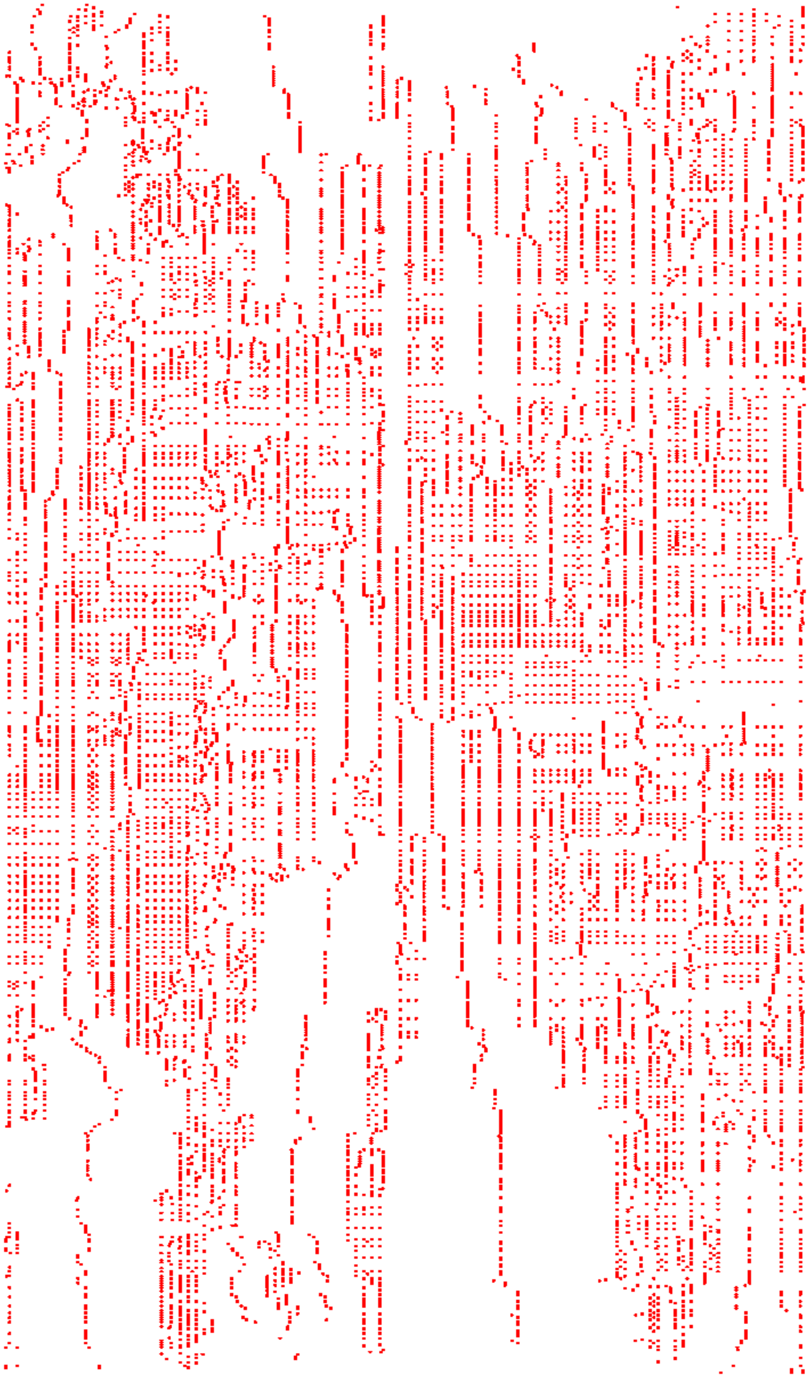
- Giden (Graphical Environment for Network Optimization)
  - Demonstration of
    - Minimum spanning trees
    - Maximum flows
    - Minimum cost flows

(Based on Ahuja, Magnanti, Orlin's book *Network Flows*)

2103 hole printed circuit board example

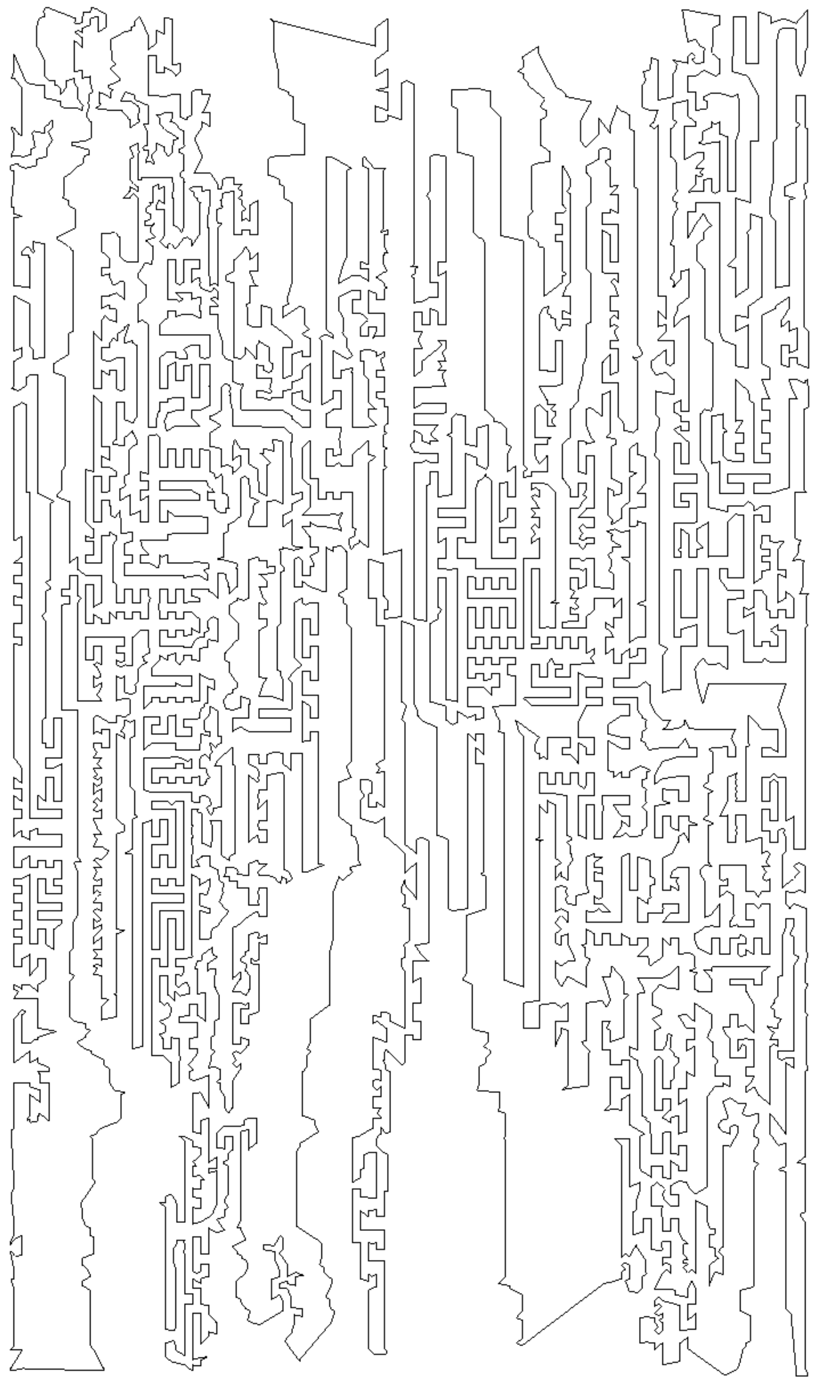


11,849 hole printed circuit board example



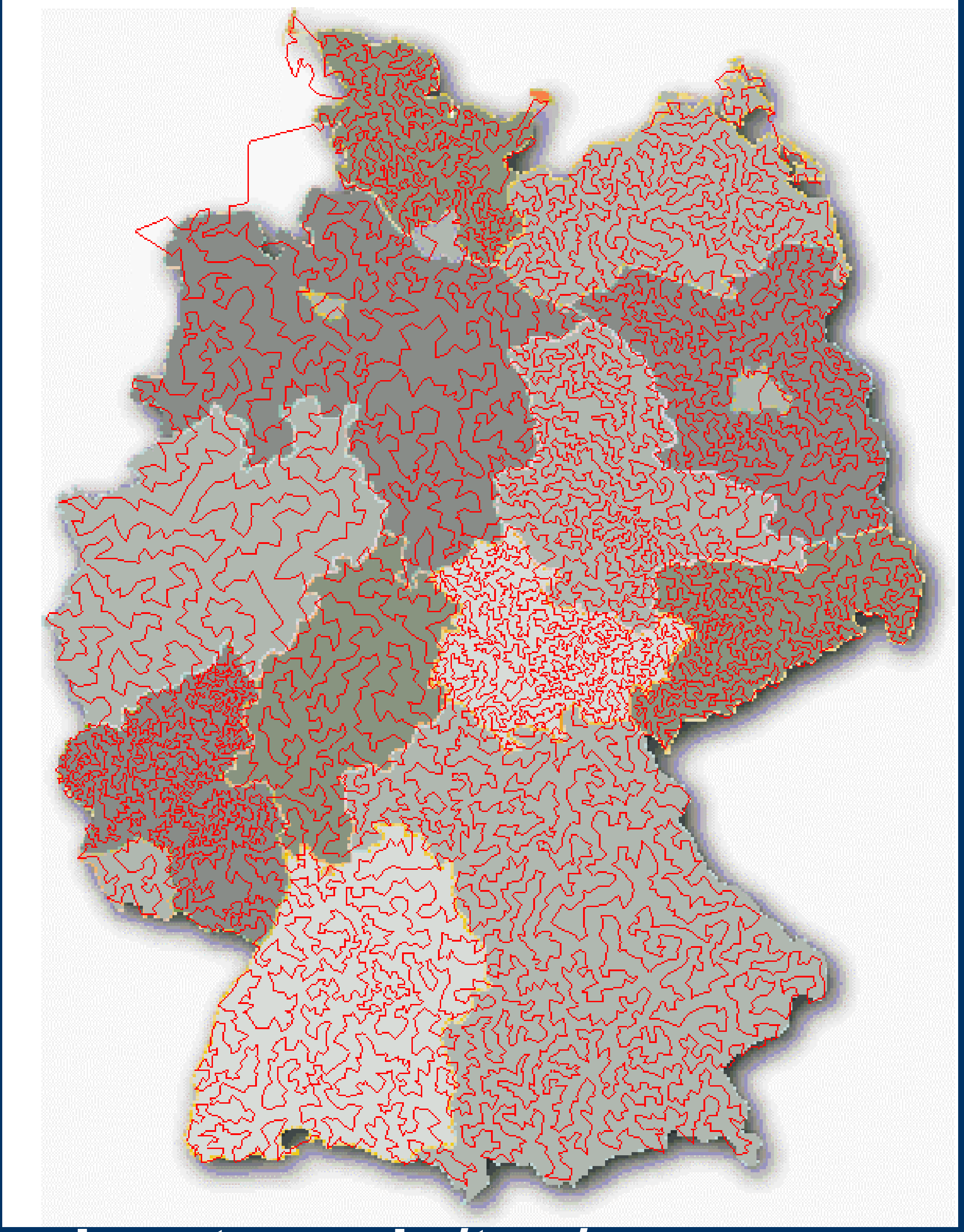
11,849 hole printed circuit board solution

13,094,345 seconds total on 55 CPUs



Traveling salesman problem through 15,112 cities in Germany

22.6 years of computation on network of 110 Processors



See http://www.math.princeton.edu/tsp/

Today's Lessons

- Traveling salesman problem arises in numerous applications
- Problem is a large-scale integer program
- Many heuristic methods: often find good solutions
- Lagrangian dual (bounding) exploits special problem structure (embedded minimal spanning tree)
- MST is easy to solve

We did NOT examine polyhedral (cutting plane) methods