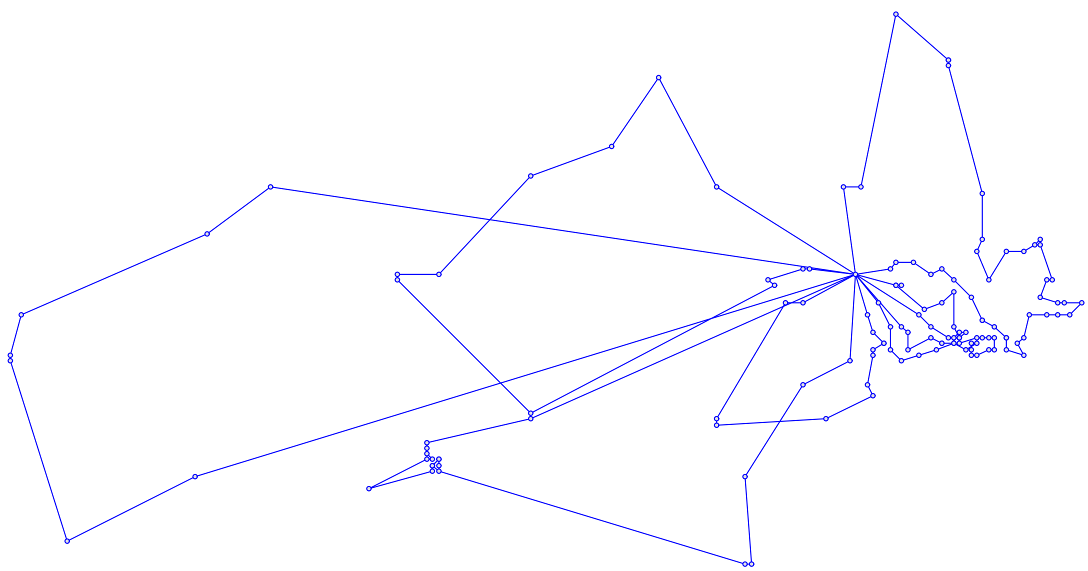
.6in=0in.6in=0.69444.6in .6in=0in.6in=0.69444.6in .6in=0in.6in=0.69444.6in .6in=0in.6in=0.69444.6in

Solving Hard Combinatorial Problems: A Research Overview



Ted Ralphs
Department of Industrial and
Manufacturing Systems Engineering
Lehigh University
Bethlehem, PA

Outline of Talk

- Introduction to combinatorial optimization
- Cutting plane methods
- Branch and cut methods
- SYMPHONY
- Future work

Combinatorial Optimization

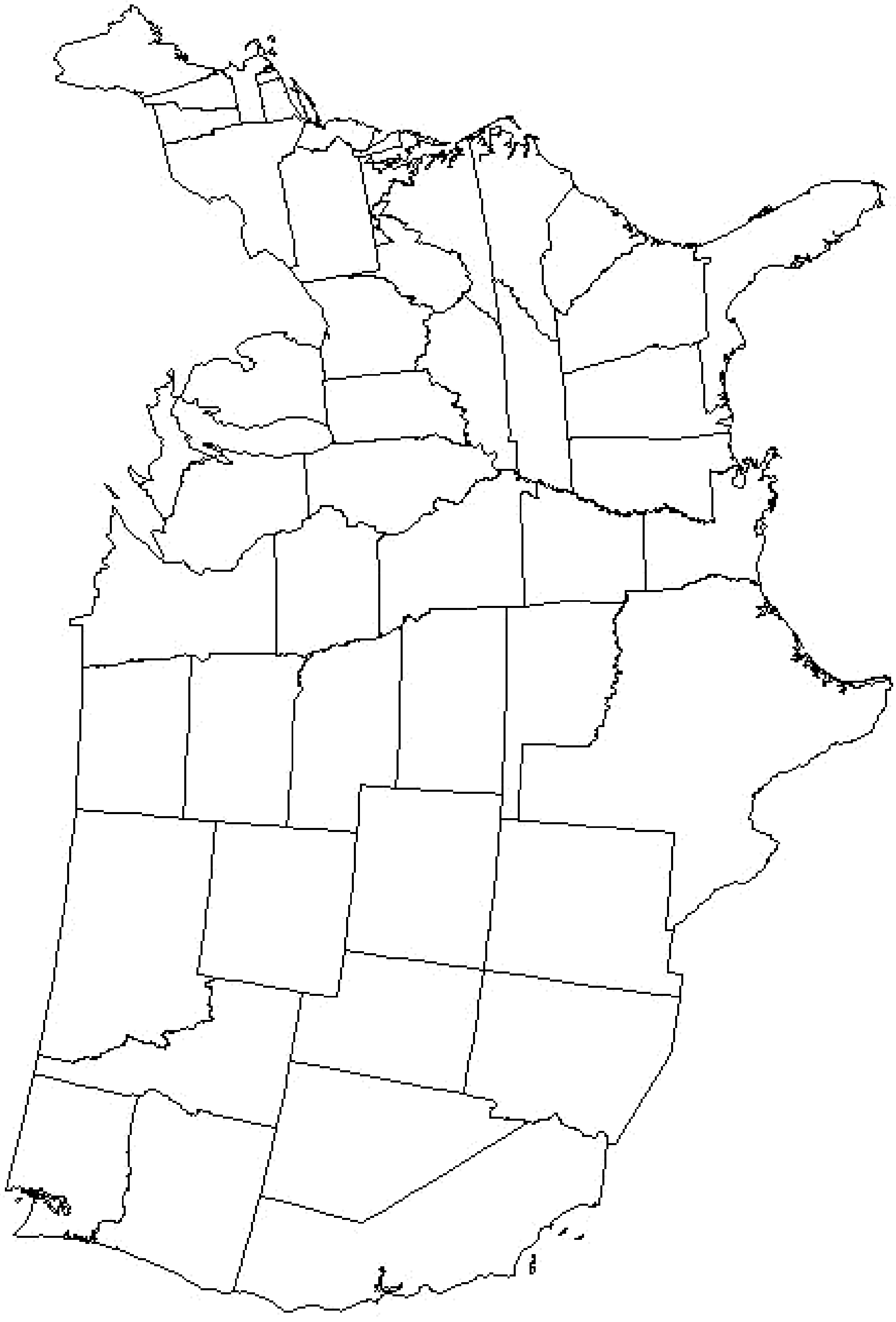
- A *combinatorial optimization problem* $CP = (E, \mathcal{F})$ consists of

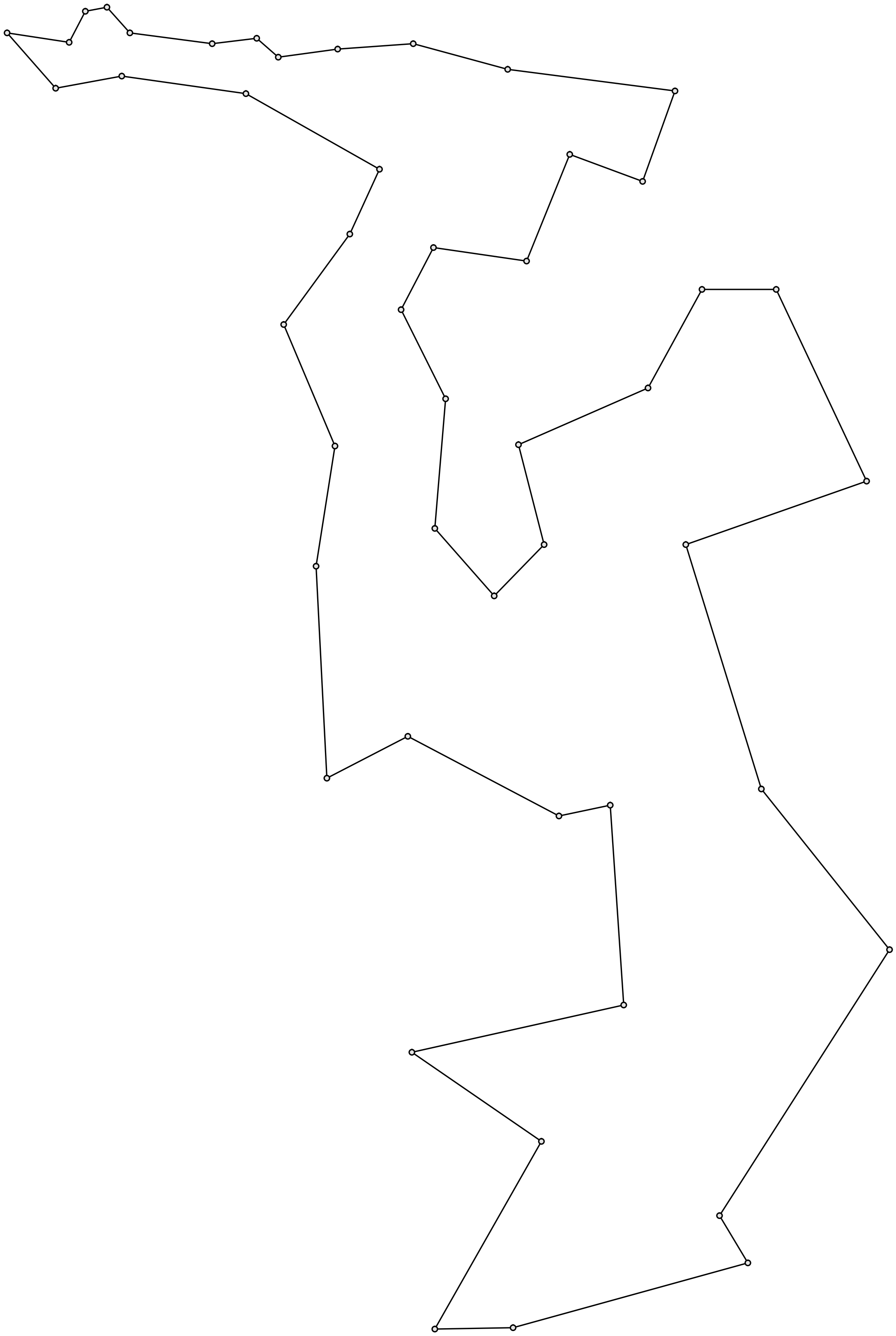
  - A *ground set* E,
  - A set $\mathcal{F} \subseteq 2^E$ of *feasible solutions*, and
  - A *cost function* $c \in \mathbf{Z^E}$ (optional).

- The *cost* of $S \in \mathcal{F}$ is $c(S) = \sum_{e \in S} c_e$.

- A *subproblem* is defined by $\mathcal{S} \subseteq \mathcal{F}$.

- Problem: Find a least cost member of $\mathcal{F}$.





The Traveling Salesman Problem

The TSP is a combinatorial problem $(E, \mathcal{F})$ whose ground set is the edge set of a graph $G = (V, E)$.

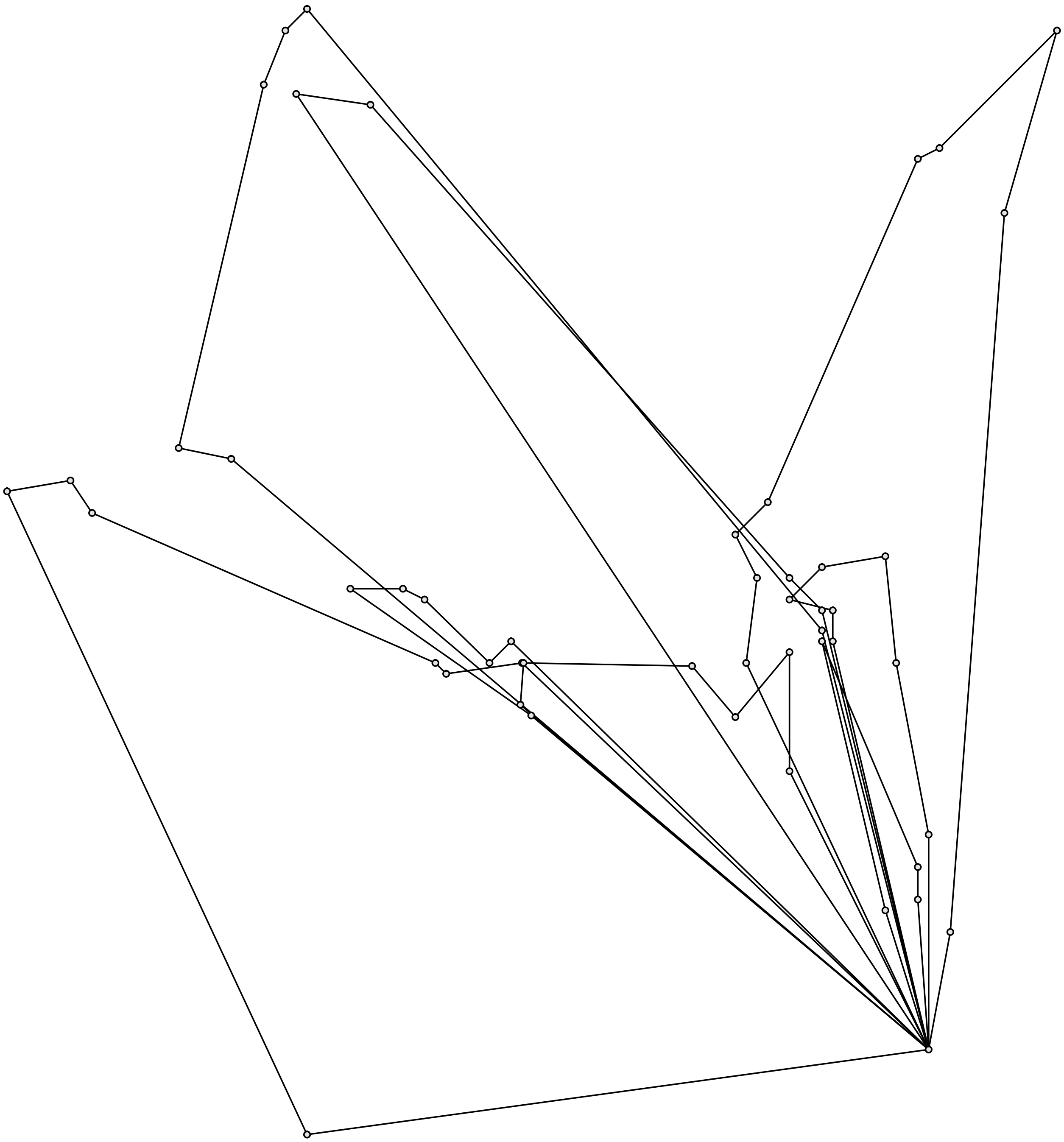
- V is the set of customers.
- E is the set of travel links between the customers.

A feasible solution is a permutation σ of V specifying the order of the customers.

IP Formulation:

$$\begin{aligned} \sum_{j=1}^{n} x_{ij} &= 2 \quad \forall i \in N^{-} \\ \sum_{\substack{i \in S \\ j \notin S}} x_{ij} &\geq 2 \quad \forall S \subset V, \ |S| > 1. \end{aligned}$$

where x_{ij} is a binary variable indicating $\sigma(i) = j$.



The Vehicle Routing Problem

In the VRP, we have additional constraints.

- There is a designated depot node (0).
- d is a vector of the customer demands.
- $V^- = V \setminus \{0\}$.
- k is the number of routes.
- C is the capacity of a truck.

A feasible solution is composed of:

- a partition $\{R_1, \ldots, R_k\}$ of V such that $\sum_{j \in R_i} d_j \leq C,\ 1 \leq i \leq k$;
- a permutation σ_i of $R_i \cup \{0\}$ specifying the order of the customers on route i.

IP Formulation for the VRP

IP Formulation:

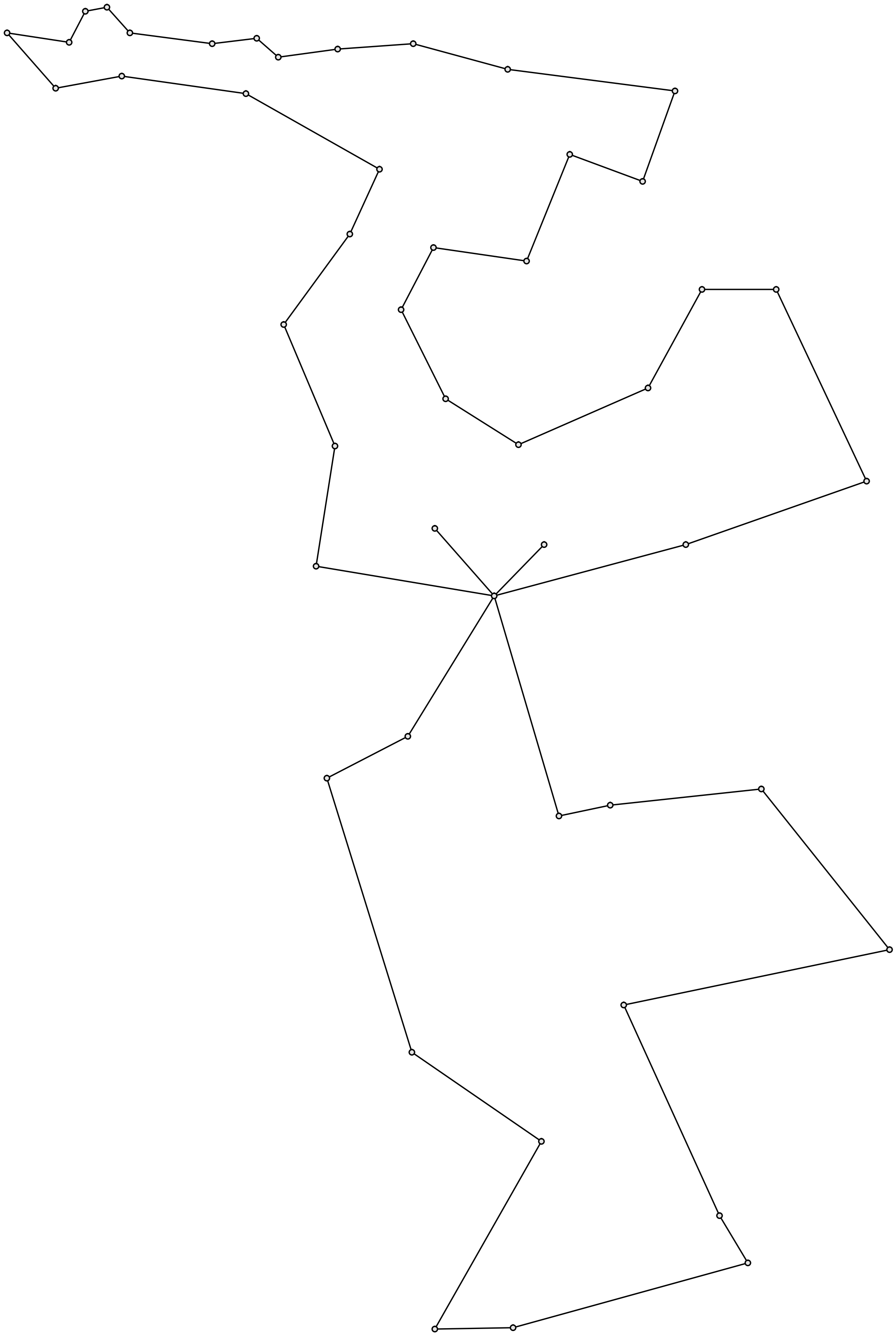
$$
\begin{aligned}
\textstyle\sum_{j=1}^{n} x_{0j} &= 2k \\
\textstyle\sum_{j=1}^{n} x_{ij} &= 2 \qquad \forall i \in V^- \\
\textstyle\sum_{\substack{i \in S \\ j \notin S}} x_{ij} &\geq 2b(S) \quad \forall S \subset V^-,\ |S| > 1.
\end{aligned}
$$

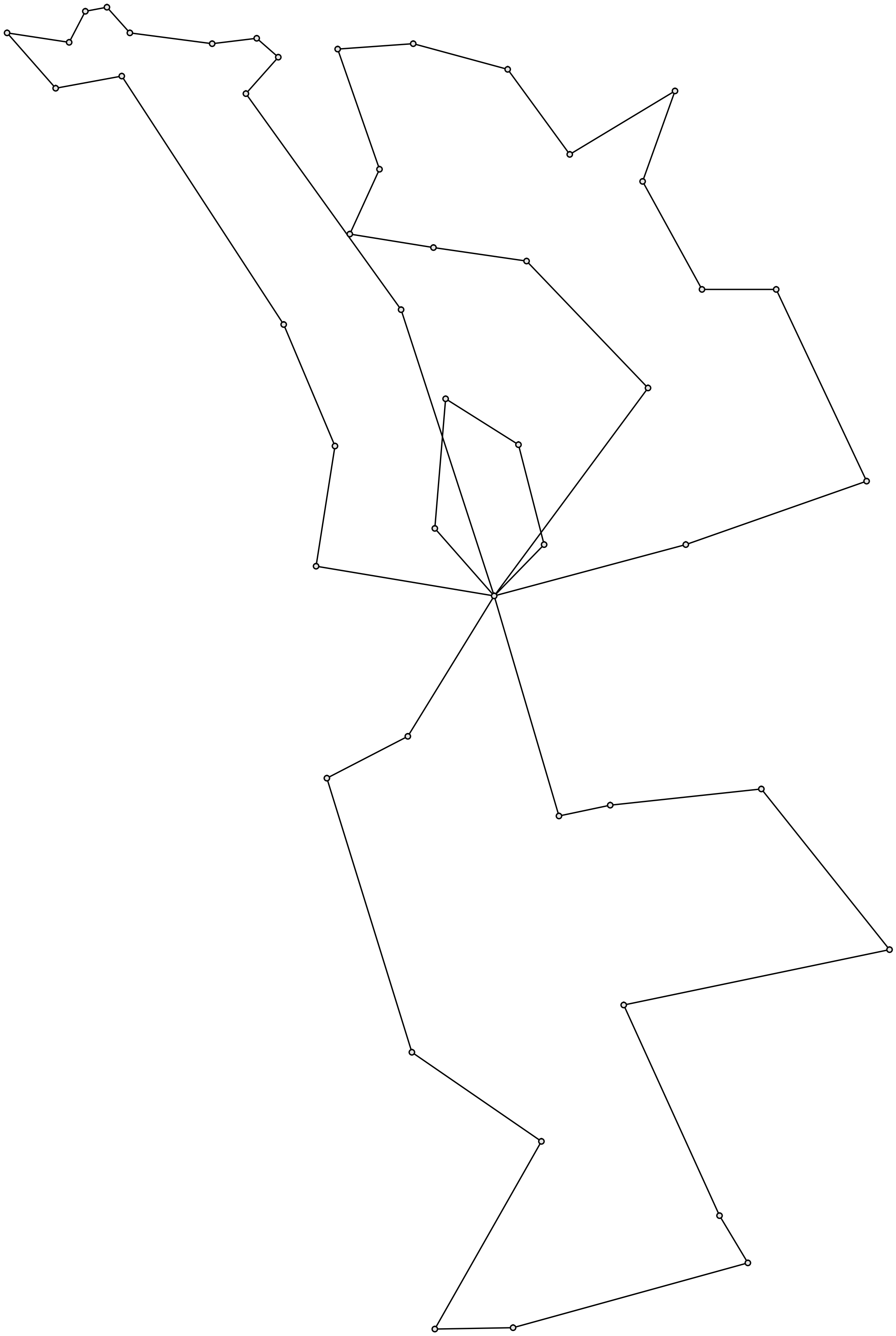
$b(S)$ = lower bound on the number of trucks required to service S (nominally $\lceil (\sum_{i \in S} d_i)/C \rceil$).

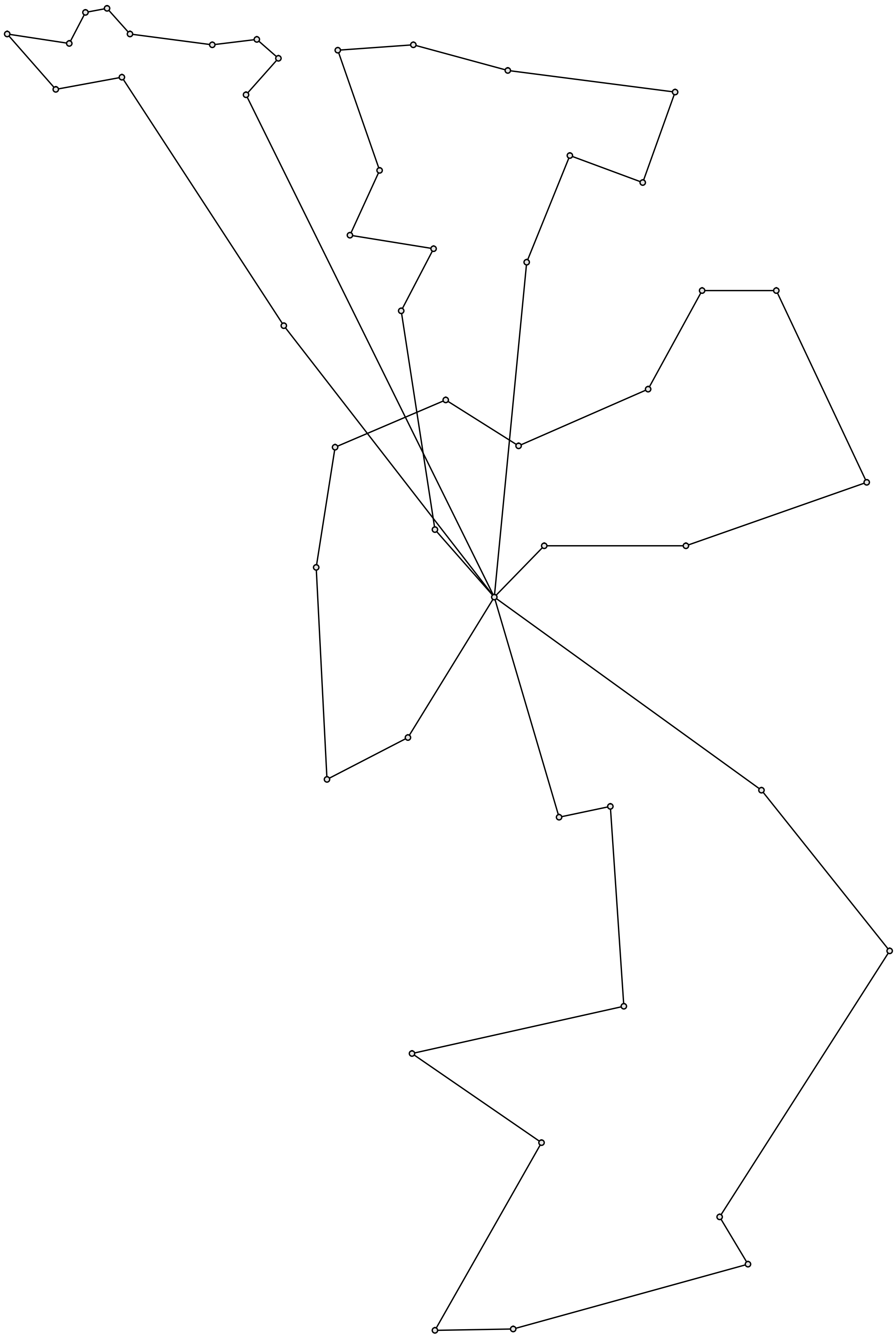
If $C = \infty$, then we have the Multiple Traveling Salesman Problem.

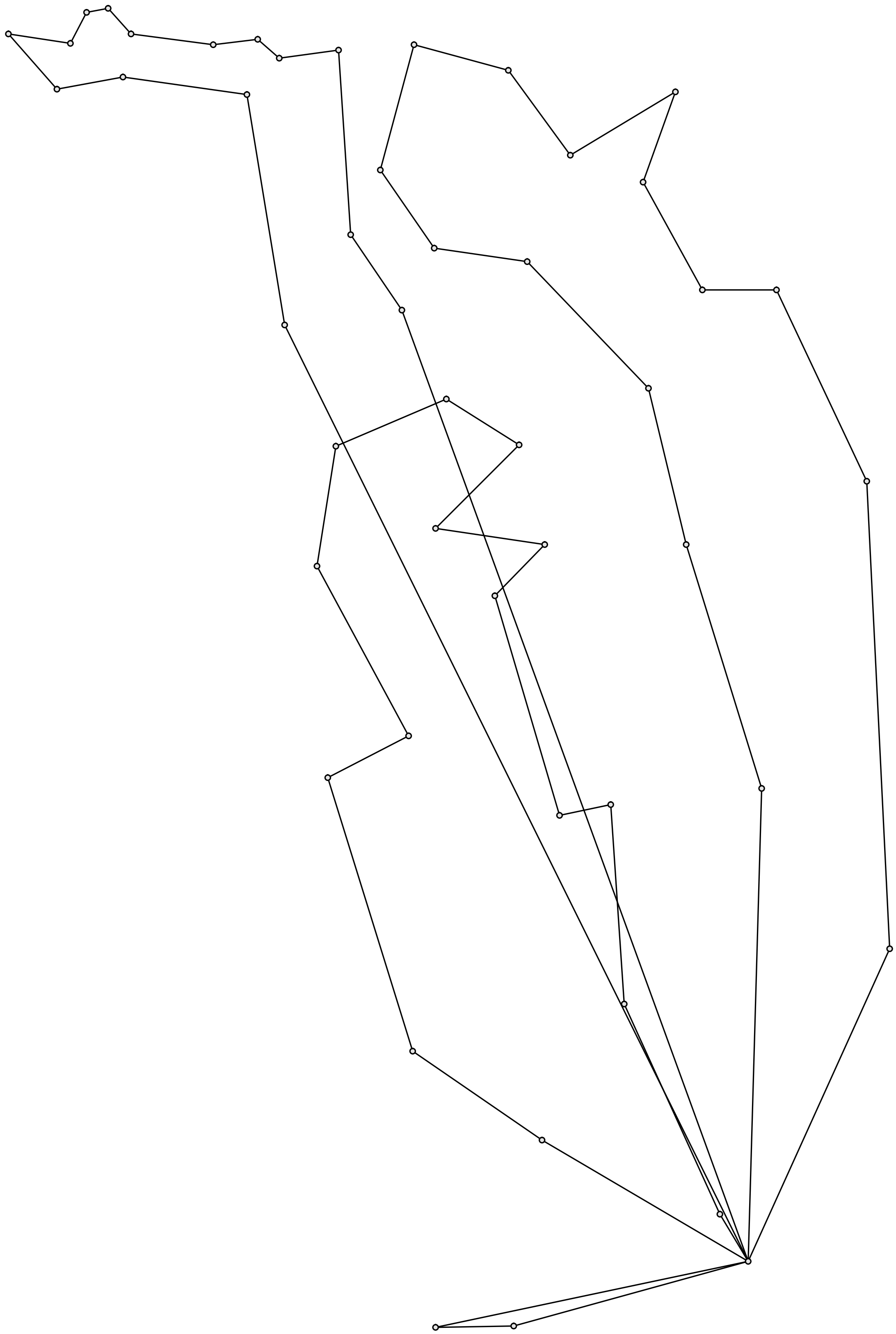
How hard are these problems?

I don't know.







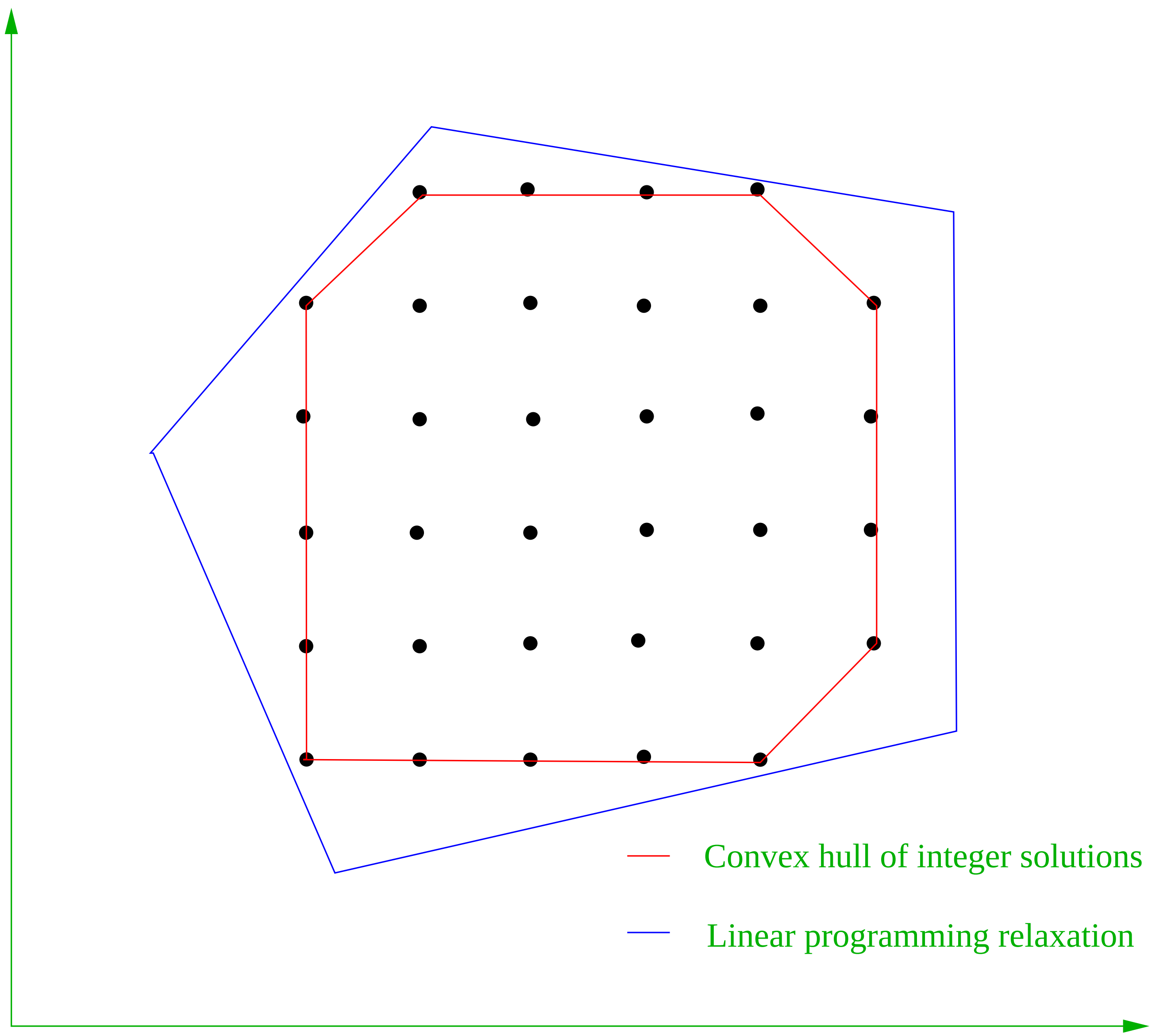


How do we solve these hard problems?

- Try to reduce it to something easier
  - Integer Program ⇒ Linear Program
  - Divide and conquer

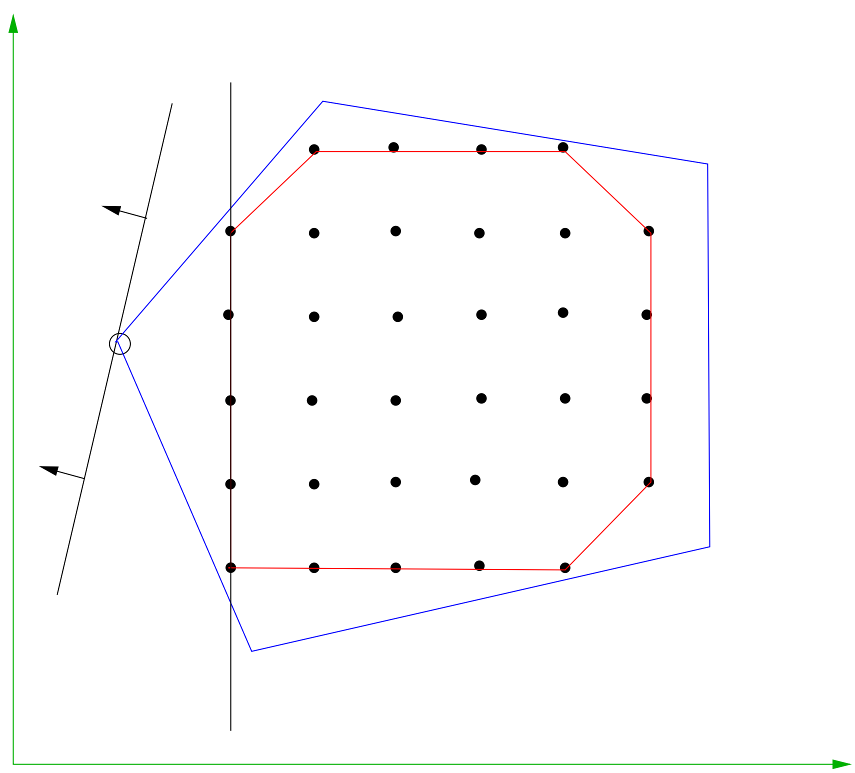
- Use a bigger hammer
  - Faster processors
  - More memory
  - Parallelism

Integer Programming



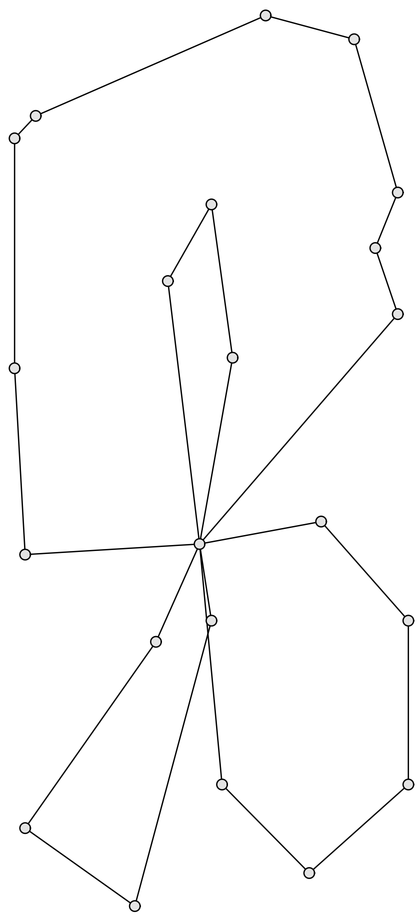
Cutting Plane Method

- Basic cutting plane algorithm
  - Relax the integrality constraints.
  - Solve the relaxation. Infeasible $\Rightarrow$ STOP.
  - If $\hat{x}$ integral $\Rightarrow$ STOP.
  - Separate $\hat{x}$ from $\mathcal{P}$.
  - No cutting planes $\Rightarrow$ algorithm fails.
- The key is good separation algorithms.

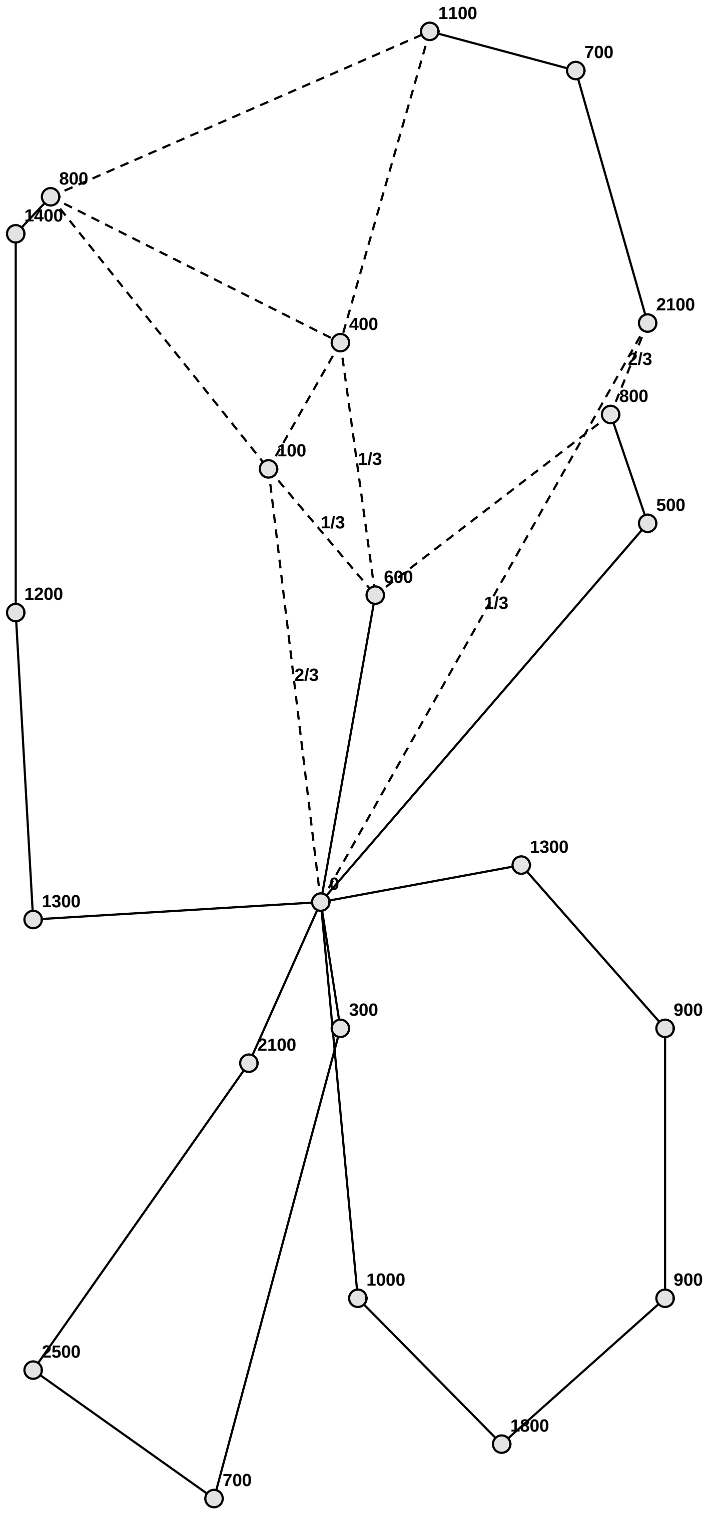


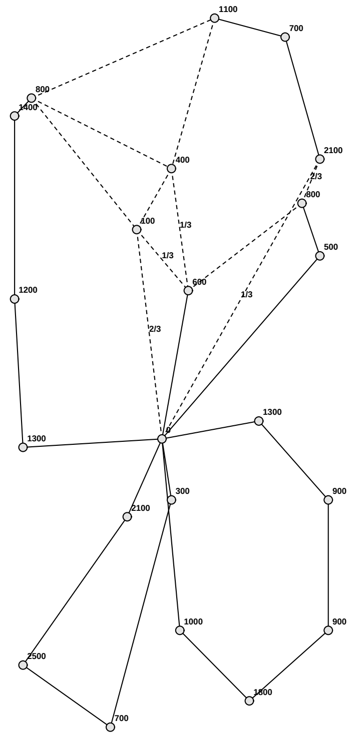
A Separation Algorithm for Side Constraints

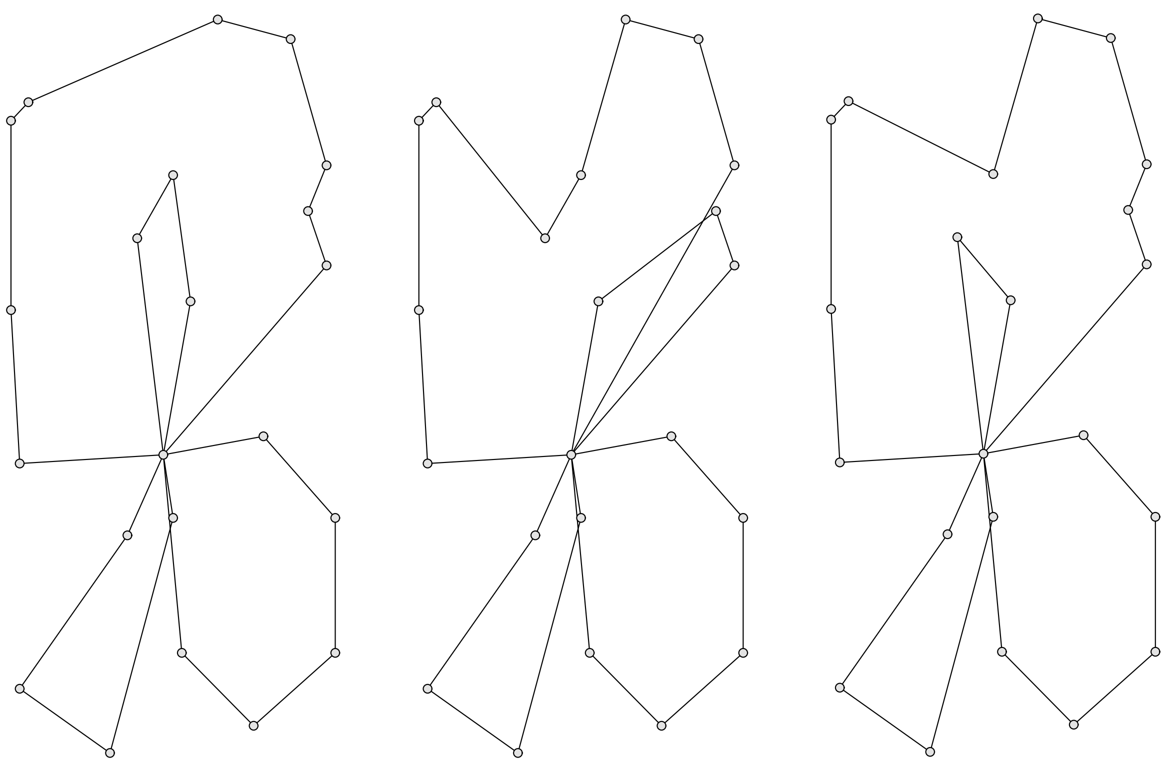
- The VRP can be thought of as a side-constrained M-TSP.

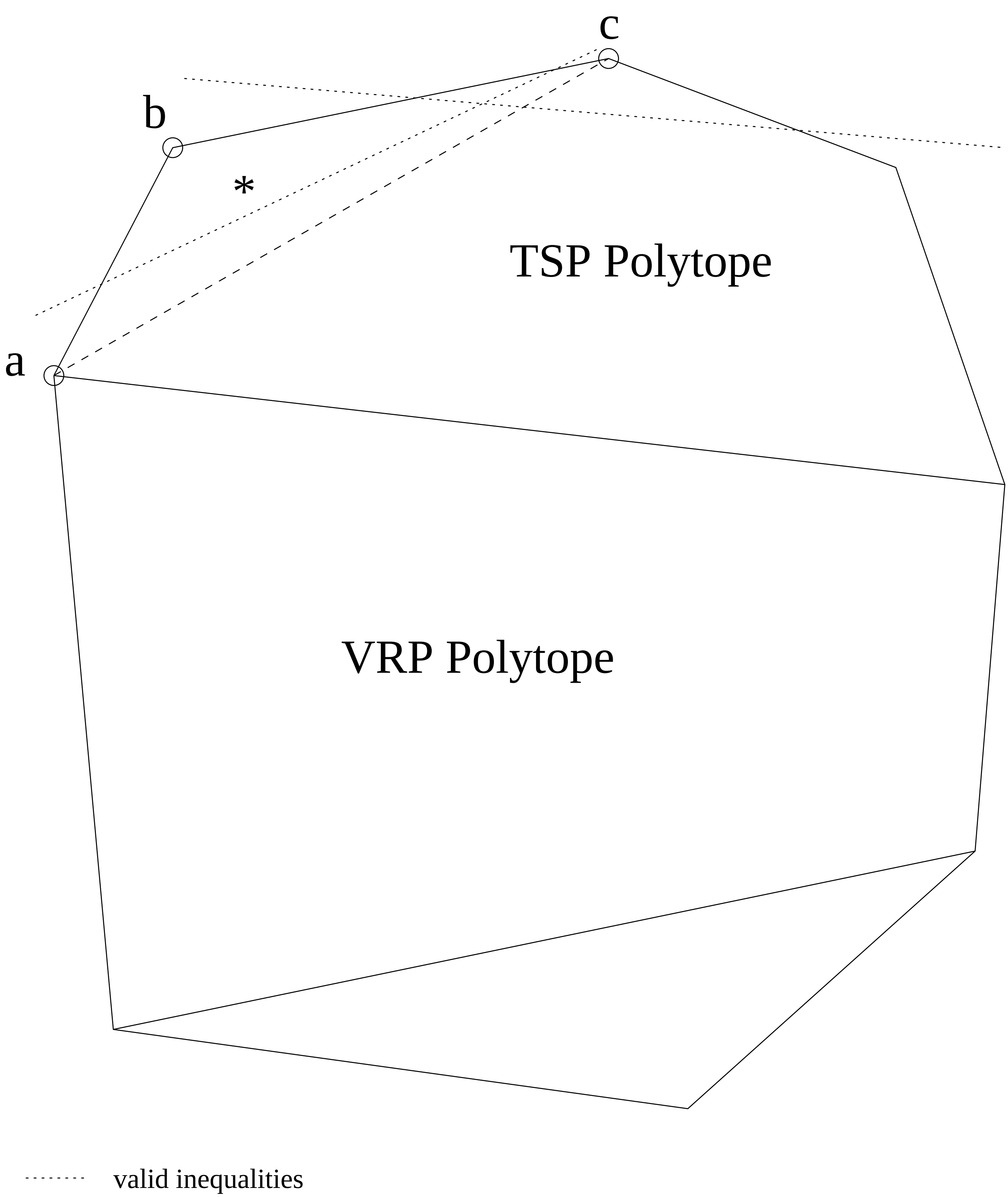


- Key observation: We can determine in $O(n)$ time whether a particular TSP tour satisfies all the capacity constraints.


1100
700
800
1400
400
2100
2/3
800
100
1/3
1/3
500
600
1200
1/3
2/3
1300
0
1300
300
900
2100
1000
900
2500
1800
700


1100
700
800
1400
400
2100
2/3
800
100
1/3
500
1/3
600
1/3
1200
2/3
1300
0
1300
300
900
2100
1000
900
2500
1800
700




c
b
*
TSP Polytope
a
VRP Polytope
valid inequalities

The Decomposition Algorithm

- Suppose we have two combinatorial problems

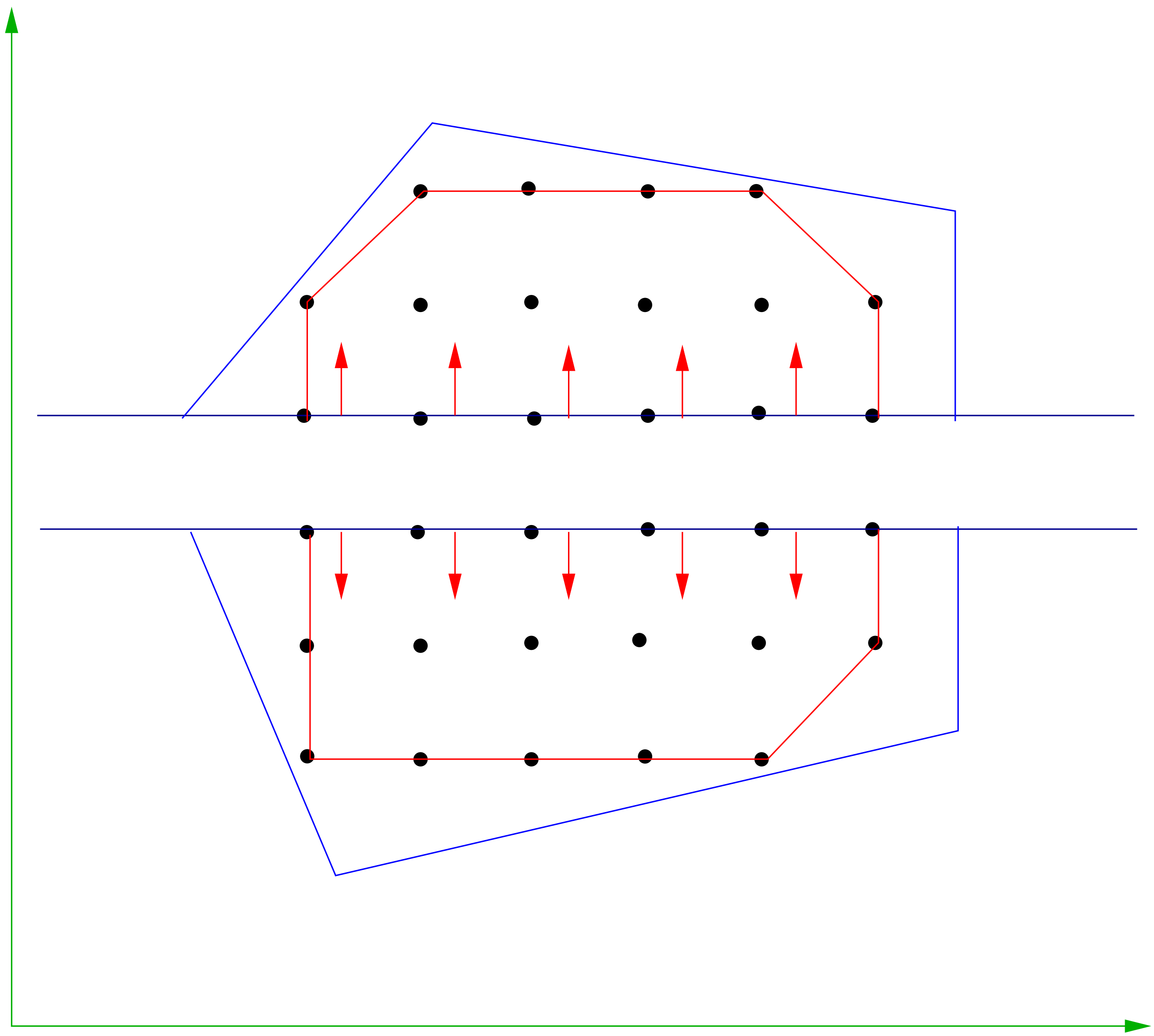
$$\begin{aligned} CP &= (E, \mathcal{F}) \\ CP' &= (E, \mathcal{H}) \end{aligned}$$

  such that $\mathcal{H} \subseteq \mathcal{F}$.

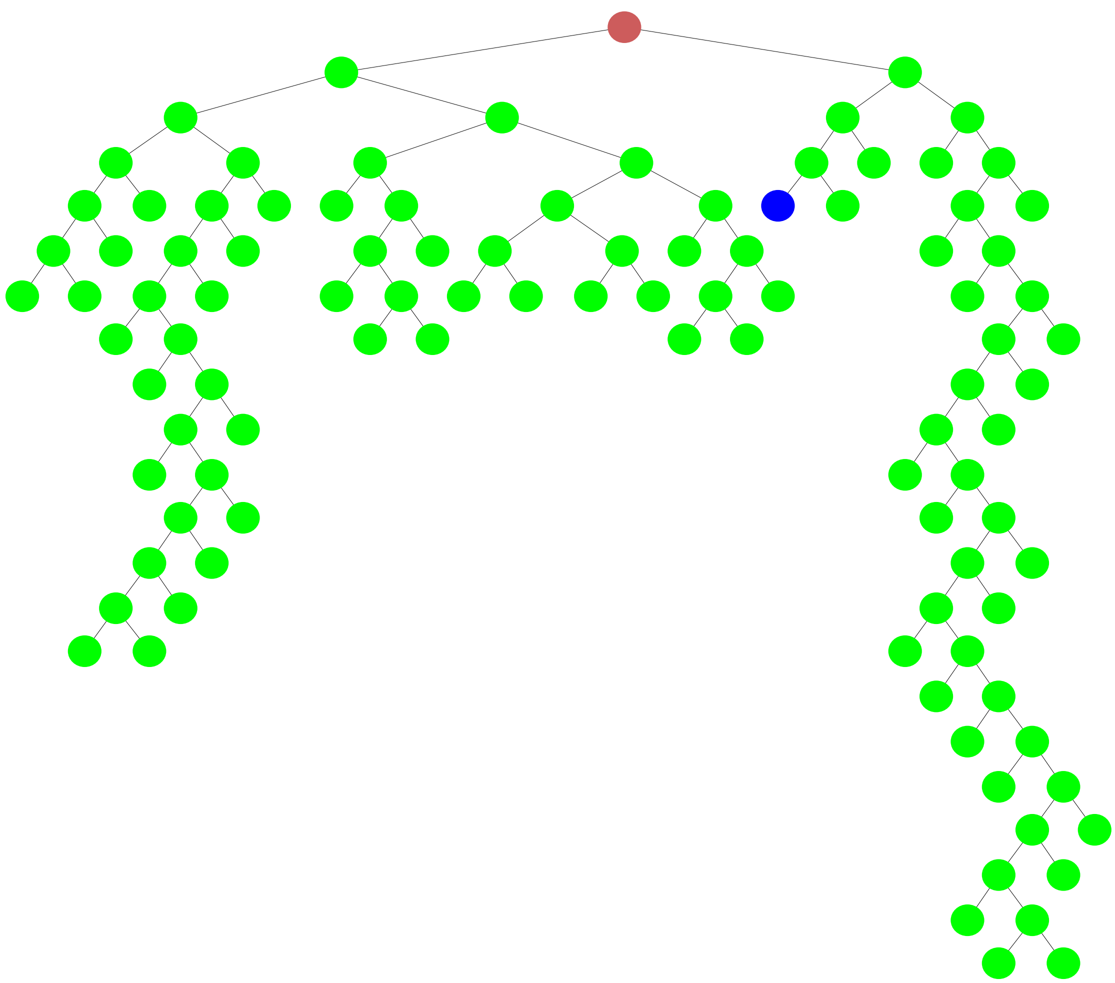
- Attempt to decompose a given fractional point $\hat{x}$ into a convex combination of extreme points of $\mathcal{P} = conv\big(\{x^S : S \in \mathcal{F}\}\big)$
- If successful, then examine the extreme points in the linear combination to obtain a set of possibly violated inequalities.
- Otherwise, derive a Farkas inequality that separates $\hat{x}$ from $\mathcal{P}$.
- This can be implemented using linear programming with column generation.

Branch and Cut Methods

If the cutting plane approach fails, then we divide and conquer (branch).



Branch and Cut Tree



SYMPHONY

SYMPHONY is a framework for implementing parallel BCP algorithms.

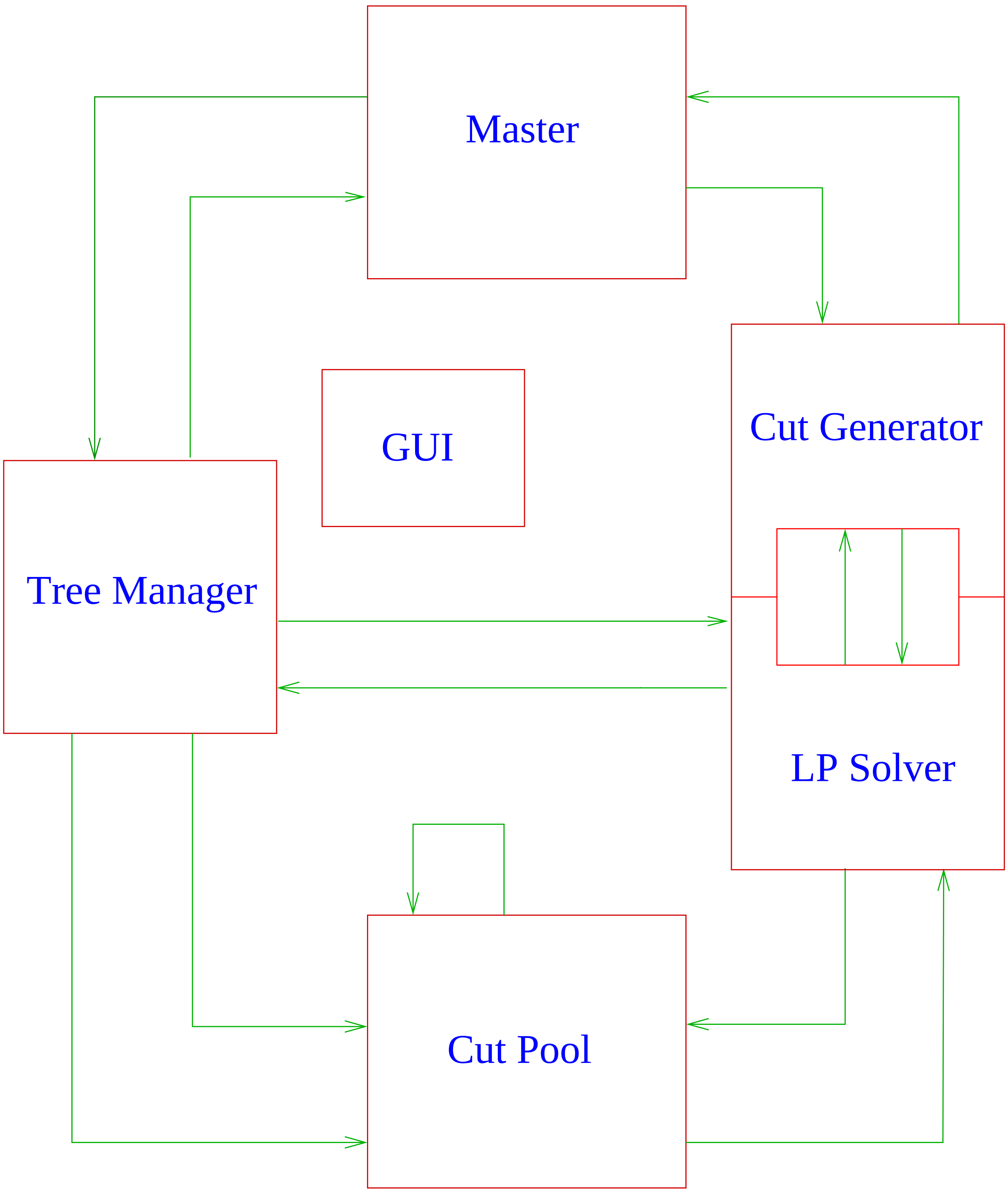
- It was designed specifically to run in a parallel environment.

- User supplies:
  - the initial LP relaxation,
  - separation subroutines,
  - feasibility checker, and
  - other optional subroutines.

- SYMPHONY handles everything else.

- The source code and documentation are available from `www.BranchAndCut.org`

The Processes of Parallel Branch and Cut



Current Status and Future Goals

- SYMPHONY
  - Development began in 1994.
  - First public release early this year.
  - Registered users at 20 universities.
  - The software is mature and contains advanced features available in specialty codes.
  - Research emphasis is on scalability.
    - Parallel scalability (more processors)
    - Data scalability (solving bigger problems)
- Applications
  - Work on the VRP and the decomposition algorithm are continuing.
  - New techniques from the literature to be incorporated into the VRP software.
  - Need out of the box thinking for the next big breakthrough.
  - Looking for other interesting application areas.