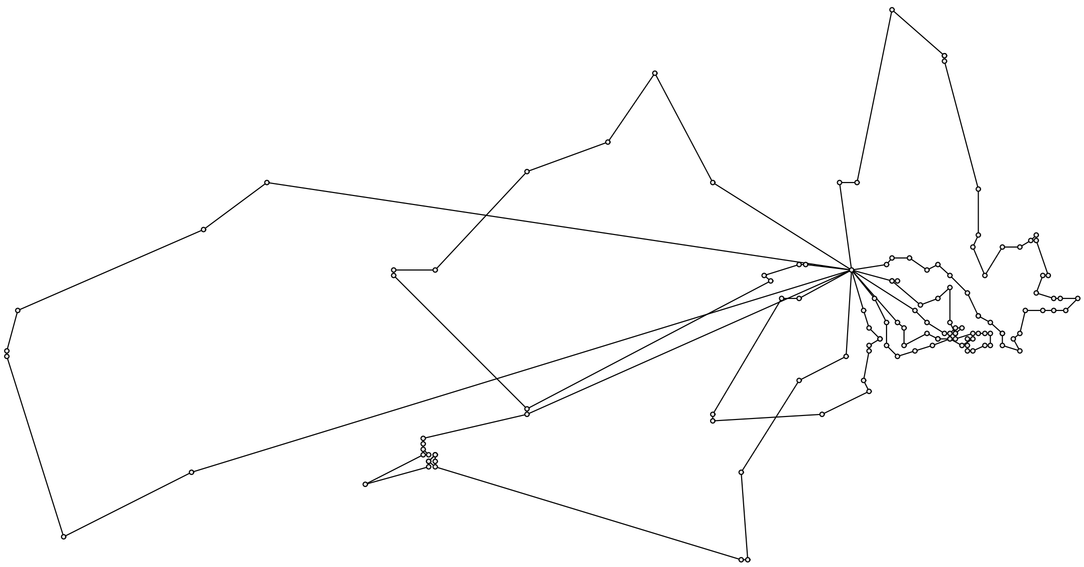
.6in=0in.6in=0.69444.6in .6in=0in.6in=0.69444.6in .6in=0in.6in=0.69444.6in .6in=0in.6in=0.69444.6in

Branch and Cut for Vehicle Routing

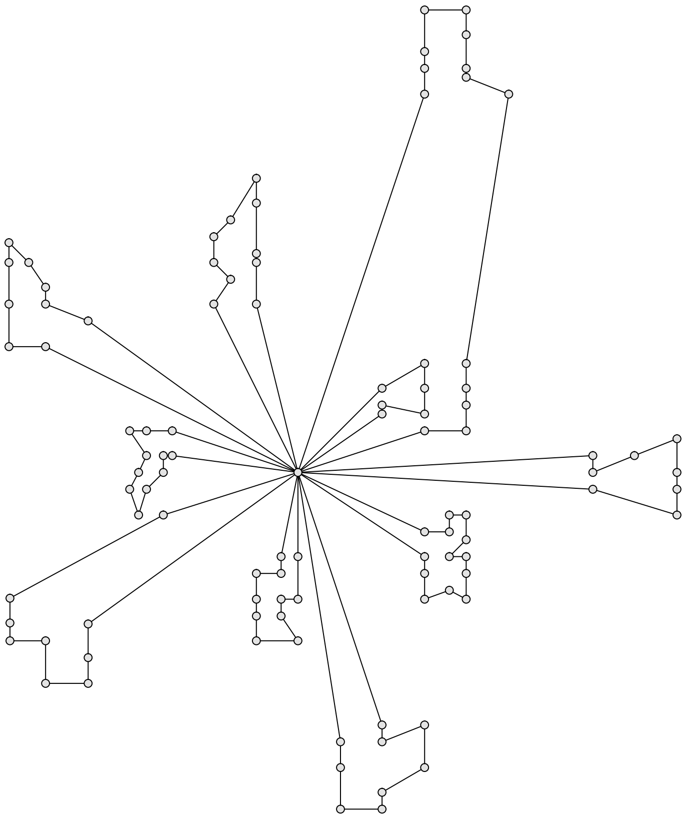


Ted Ralphs
Department of Industrial and
Manufacturing Systems Engineering
Lehigh University
Bethlehem, PA

tkralphs@lehigh.edu
www.lehigh.edu/~tkr2
www.BranchAndCut.org

Outline of Talk

- Description of the Problem
- Relationship to Other Combinatorial Models
- Cutting
- Branching
- SYMPHONY
- Future work



The Vehicle Routing Problem

The VRP is a combinatorial problem whose *ground set* is the edges of a graph $G(N, E)$. Notation:

- N is the set of customers and the depot (0).
- d is a vector of the customer demands.
- $N^- = N \setminus \{0\}$.
- 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

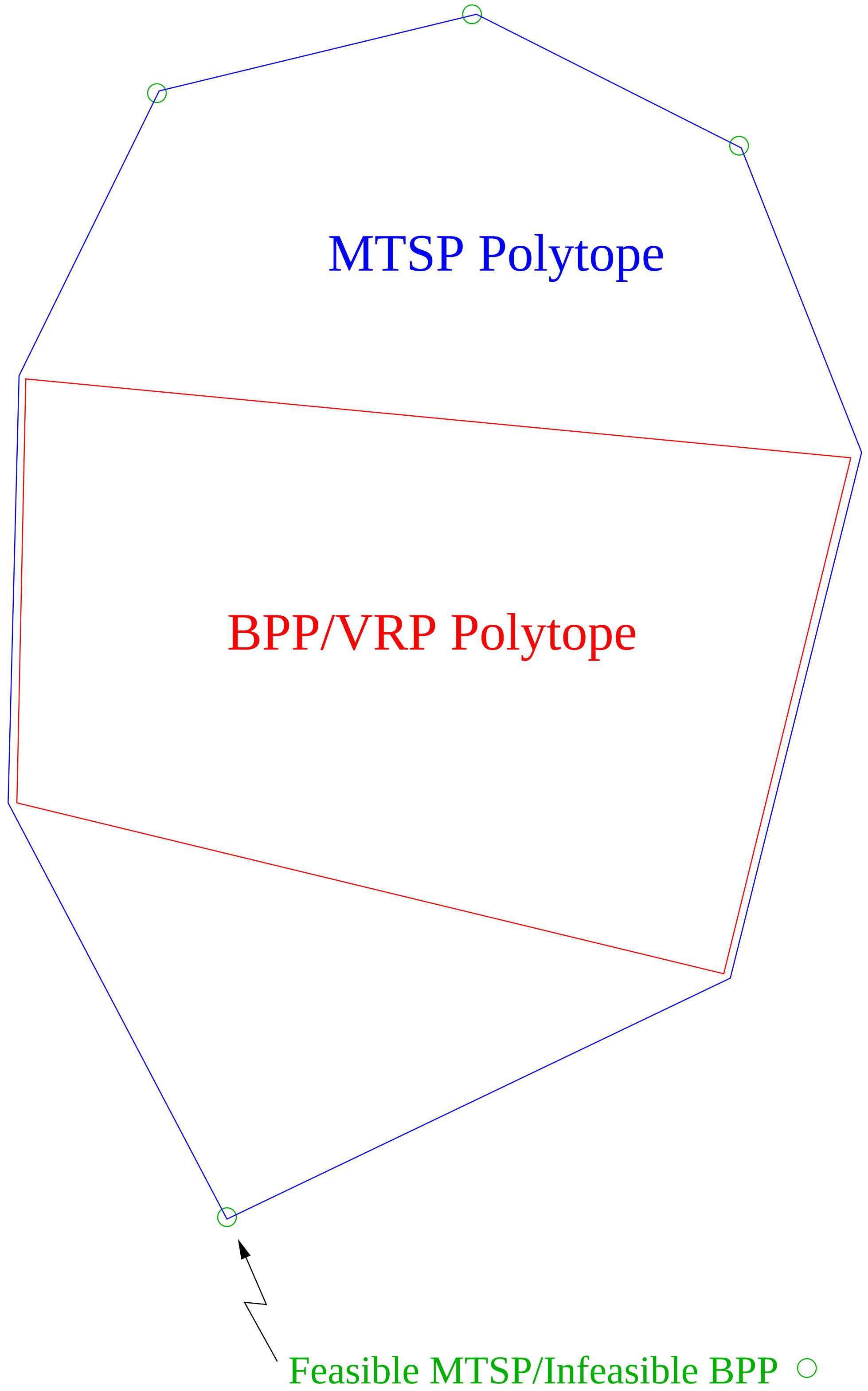
IP Formulation:

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

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

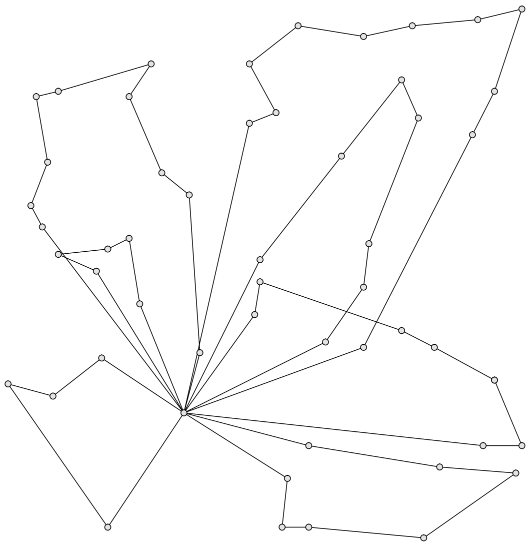
If $C = \sum_{i \in S} d_i$, then we have the Multiple Traveling Salesman Problem.

Alternatively, if the edge costs are all zero, then we have the Bin Packing Problem.

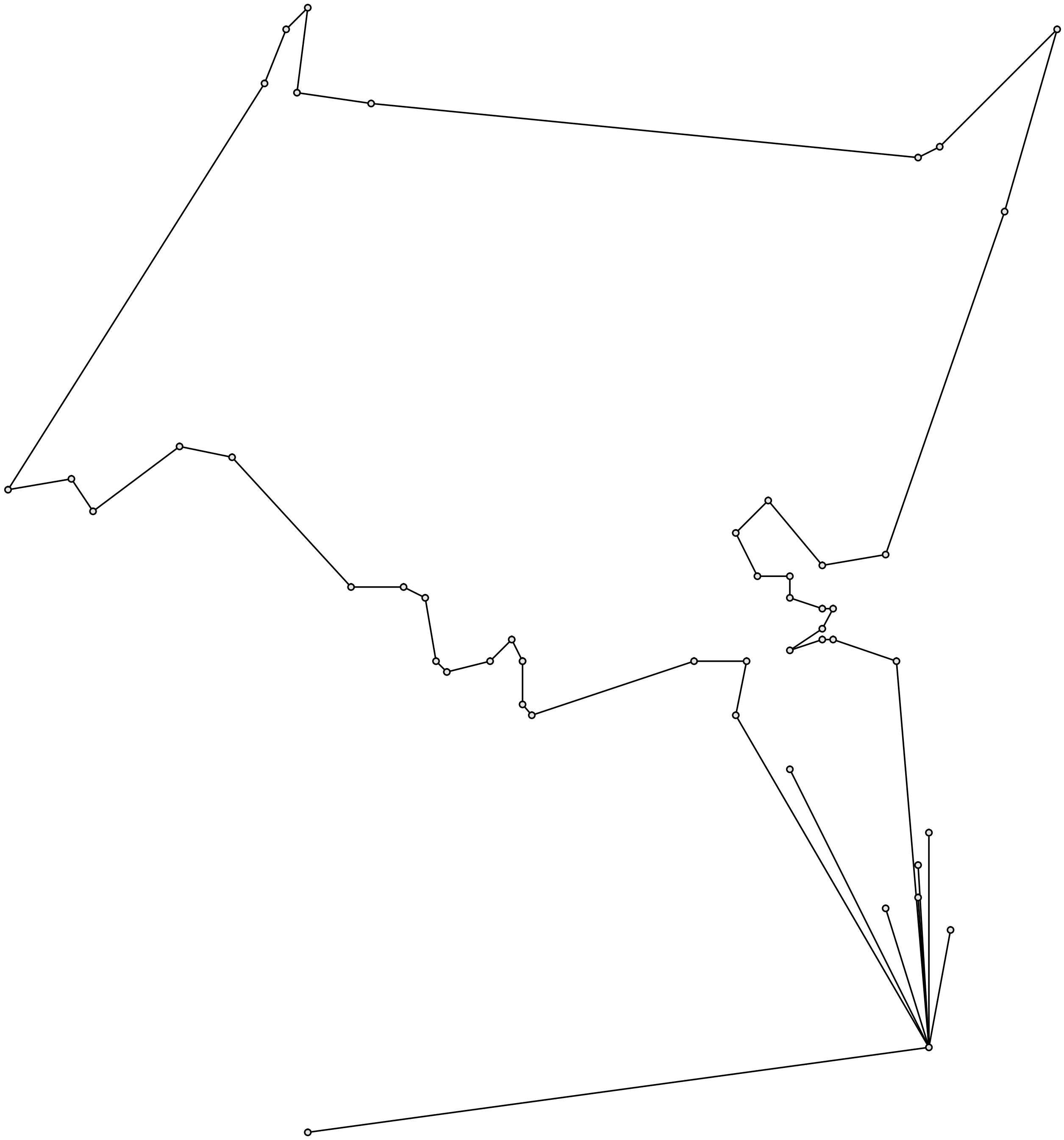

MTSP Polytope
BPP/VRP Polytope
Feasible MTSP/Infeasible BPP

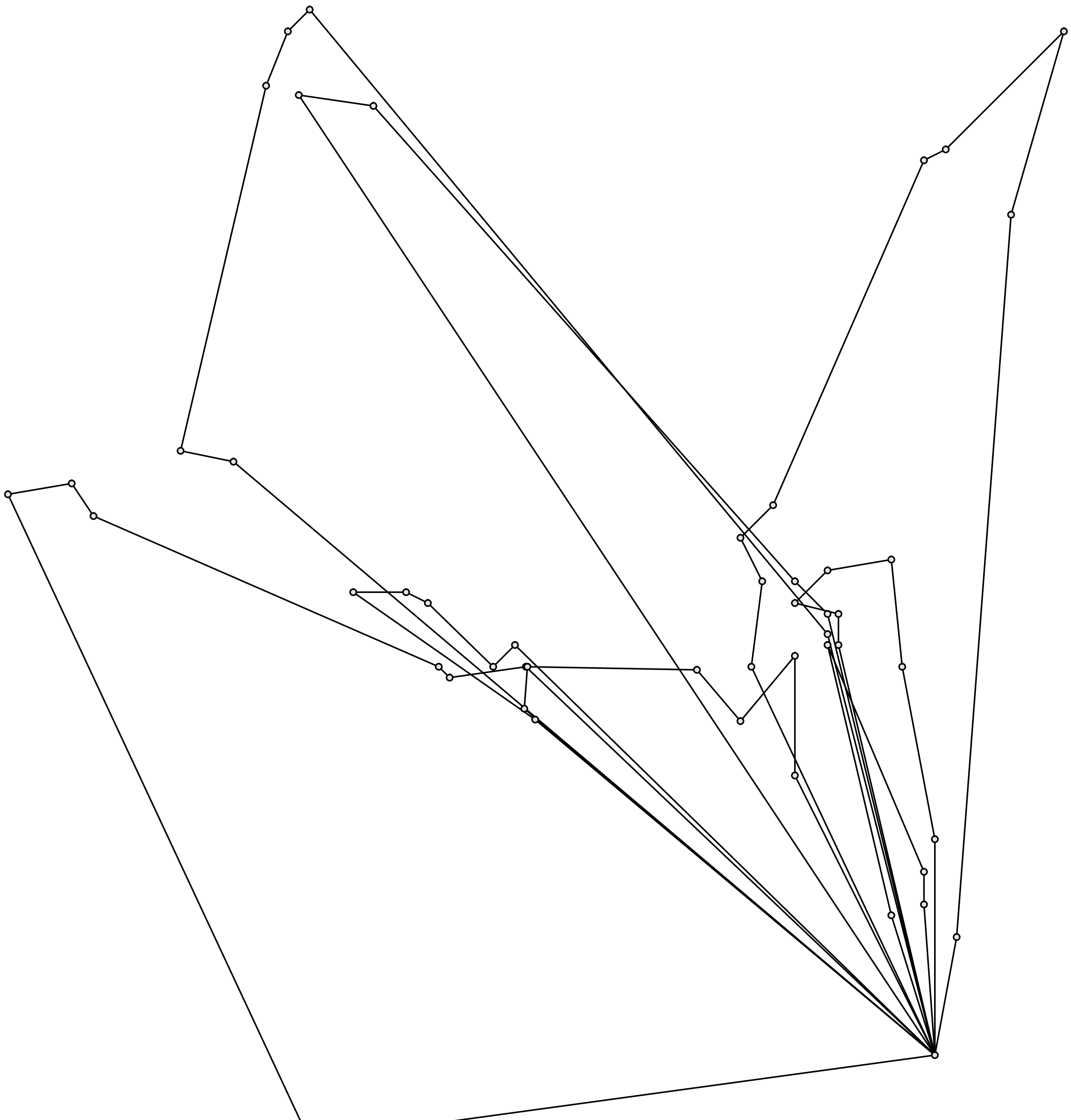
How hard is the VRP?

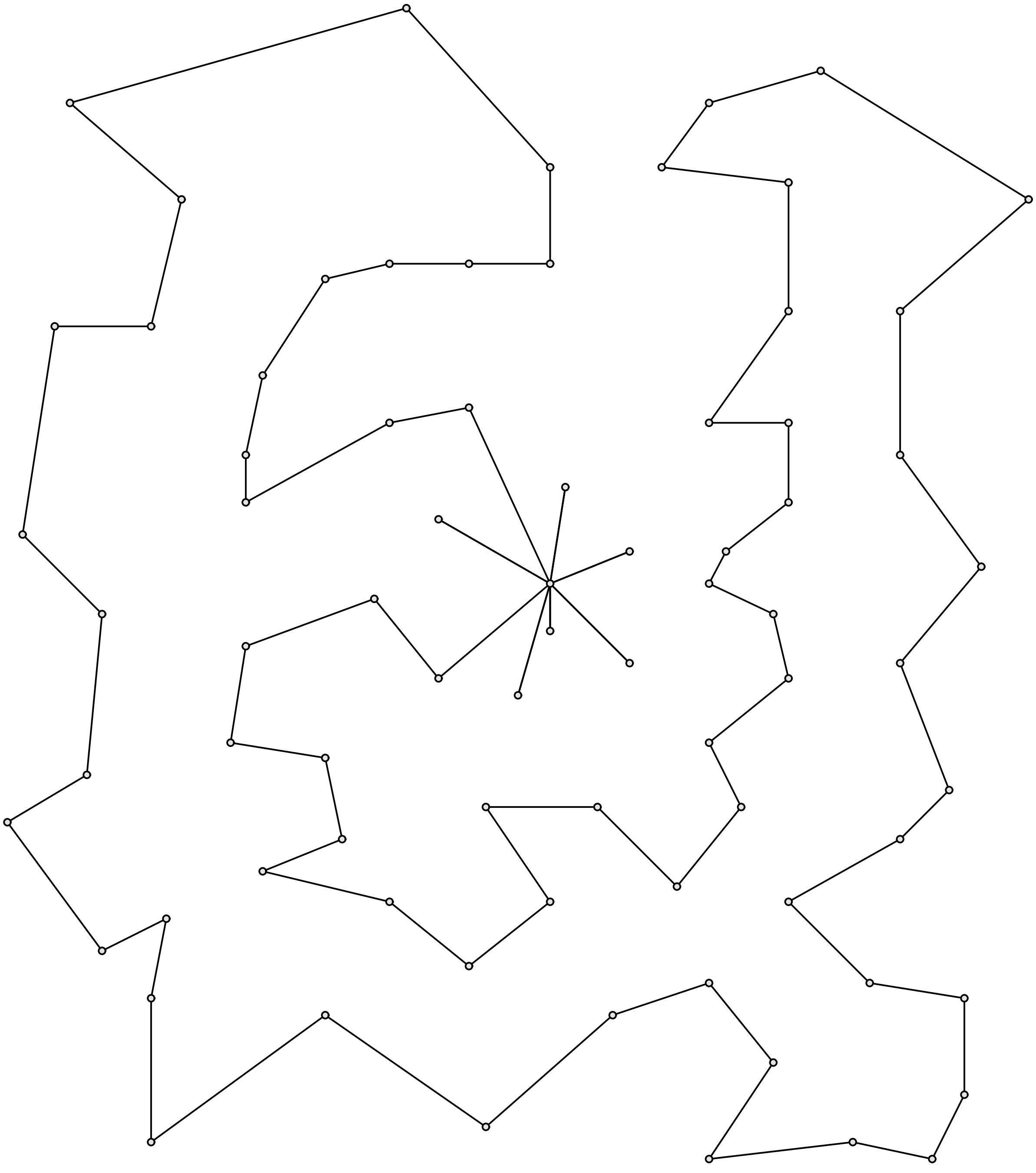
- Test Set
  - TSPLIB/VRPLIB
  - Augerat's repository
  - Available at `BranchAndCut.org/VRP`
- Largest VRP instance solved: F-n135-k7
- Smallest VRP instance unsolved: B-n50-k8
- Largest TSP instance solved: usa13509
- Time to solve B-n50-k8 as an MTSP: .1 sec
- Why the gap?

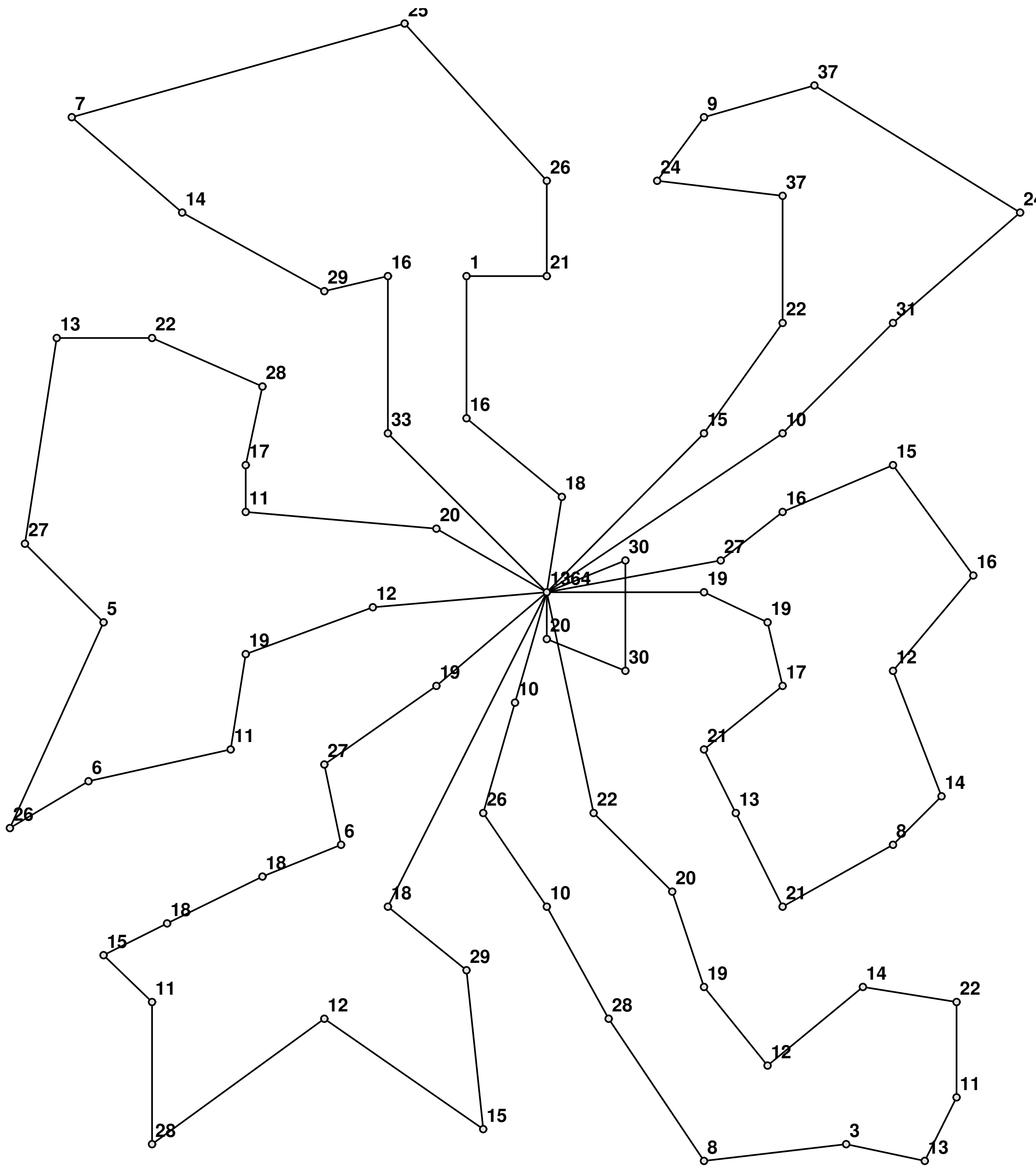



12
21
17
22
14
12
24
16
21
26
18
69
20
21
14
16
23
16
15
19
26
4
10
8
13
10
18
3
7
10
12
5
23
25
15
3
25
735
2
3
2
9
12





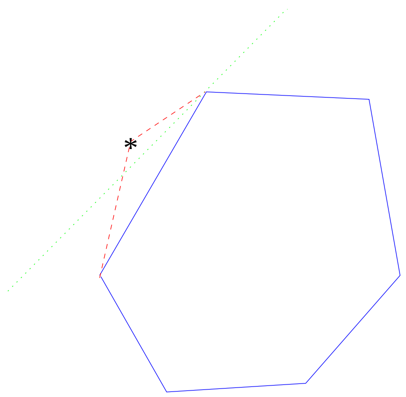




25
7
9
37
26
24
37
14
24
16
1
21
29
13
22
22
31
28
16
33
15
10
17
15
18
11
16
20
30
27
27
16
1364
19
12
5
19
20
19
30
12
17
10
11
21
6
27
14
26
26
22
13
6
8
18
20
21
18
10
18
15
29
19
14
22
11
12
12
28
11
15
3
28
8
13

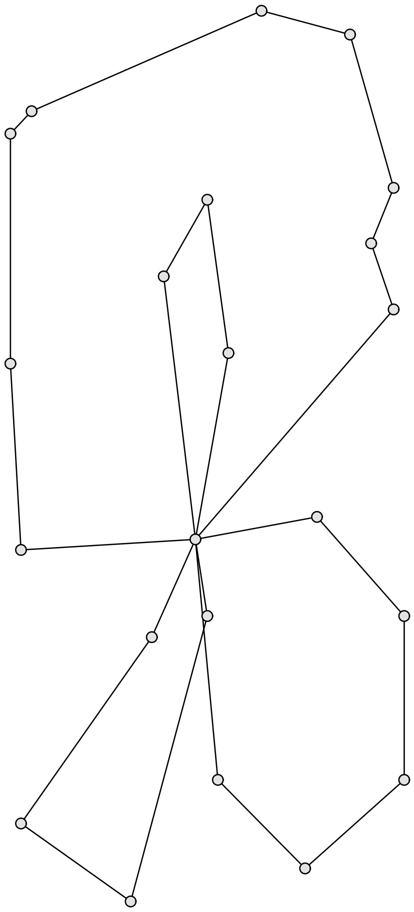
Known Classes of Valid Inequalities

- Fractional capacity constraints
- Capacity constraints
- Generalized capacity constraints
- Framed capacity constraints
- Multi-star inequalities
- Cuts from the TSP
  - Combs
  - Hypo-tours
- Cuts from the Stable Set Problem
- Path-bin inequalities

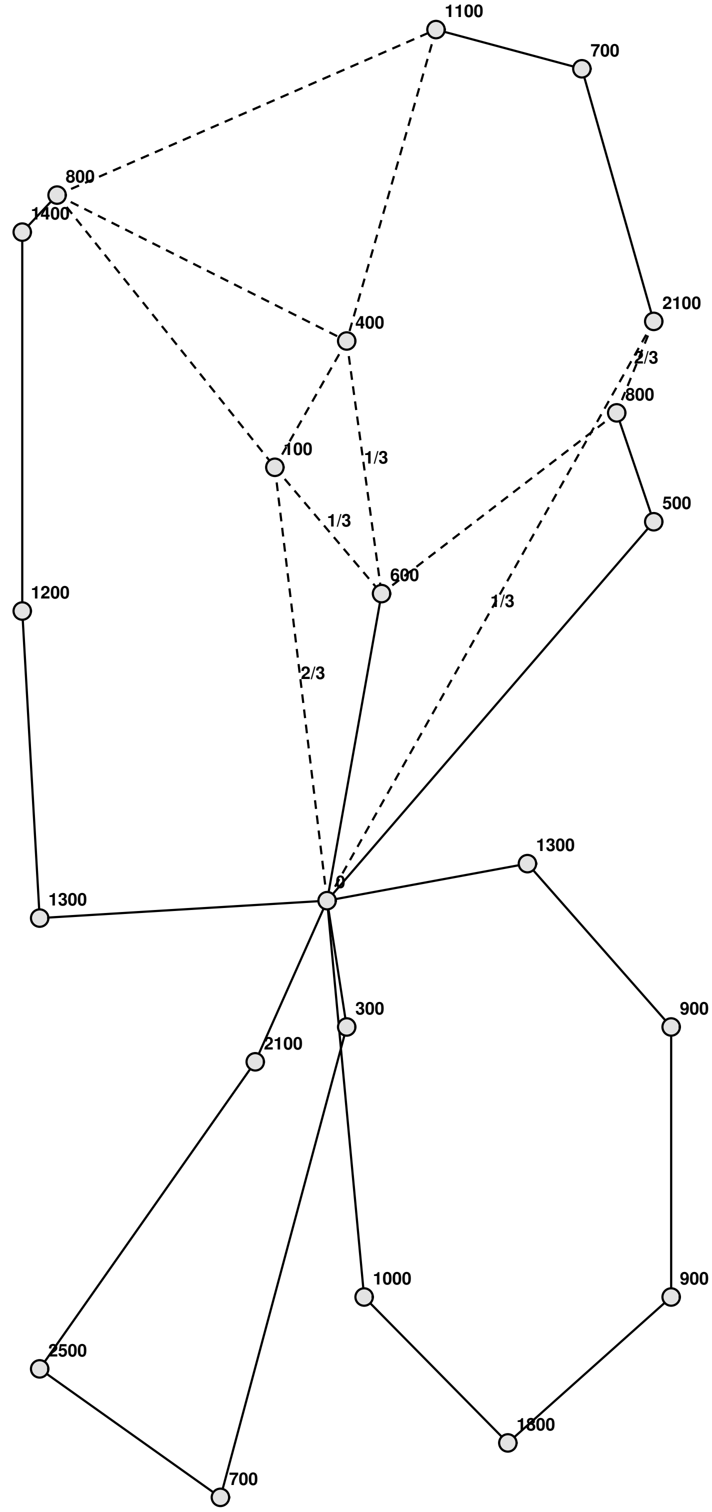


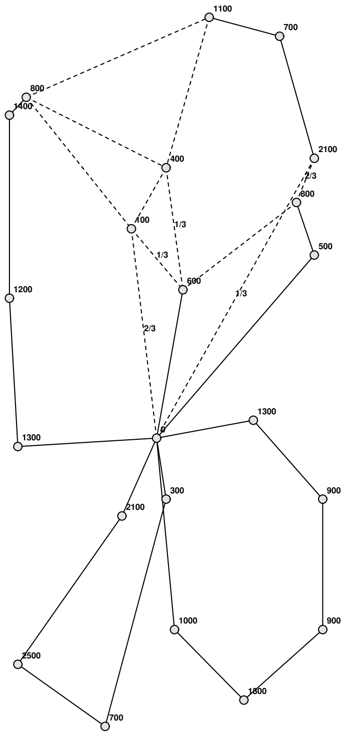
A Separation Algorithm for Side Constraints

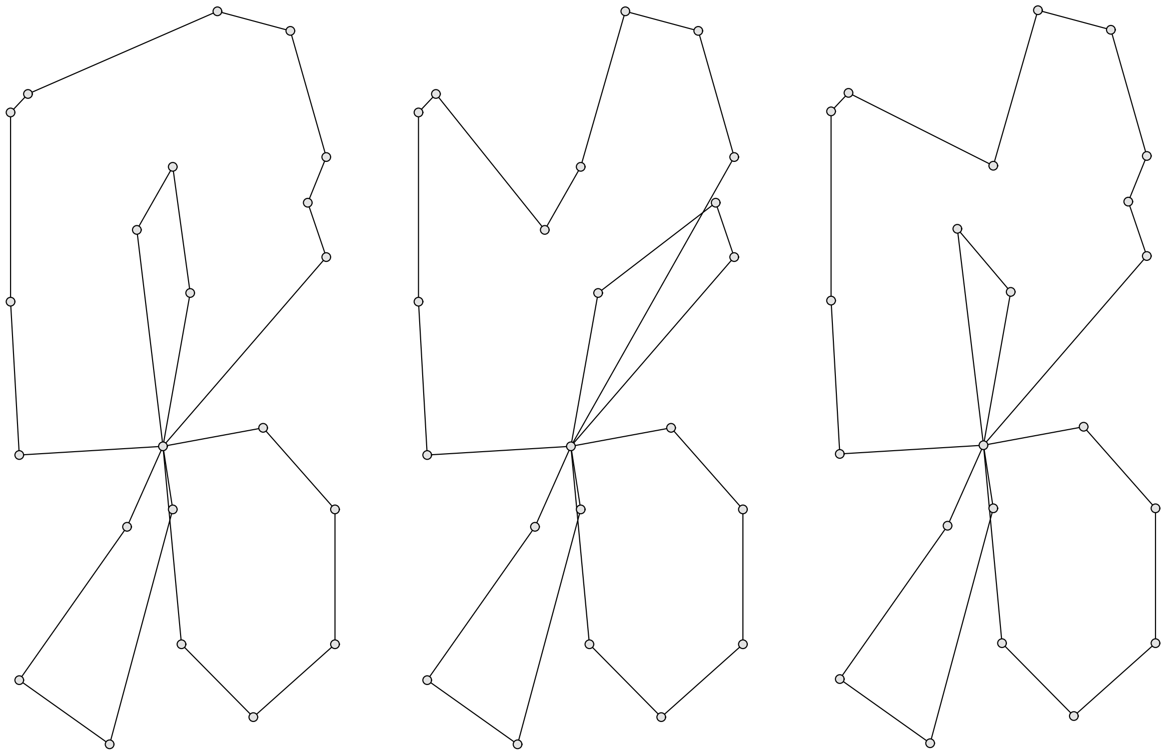
- The VRP can be thought of as a side-constrained M-TSP.

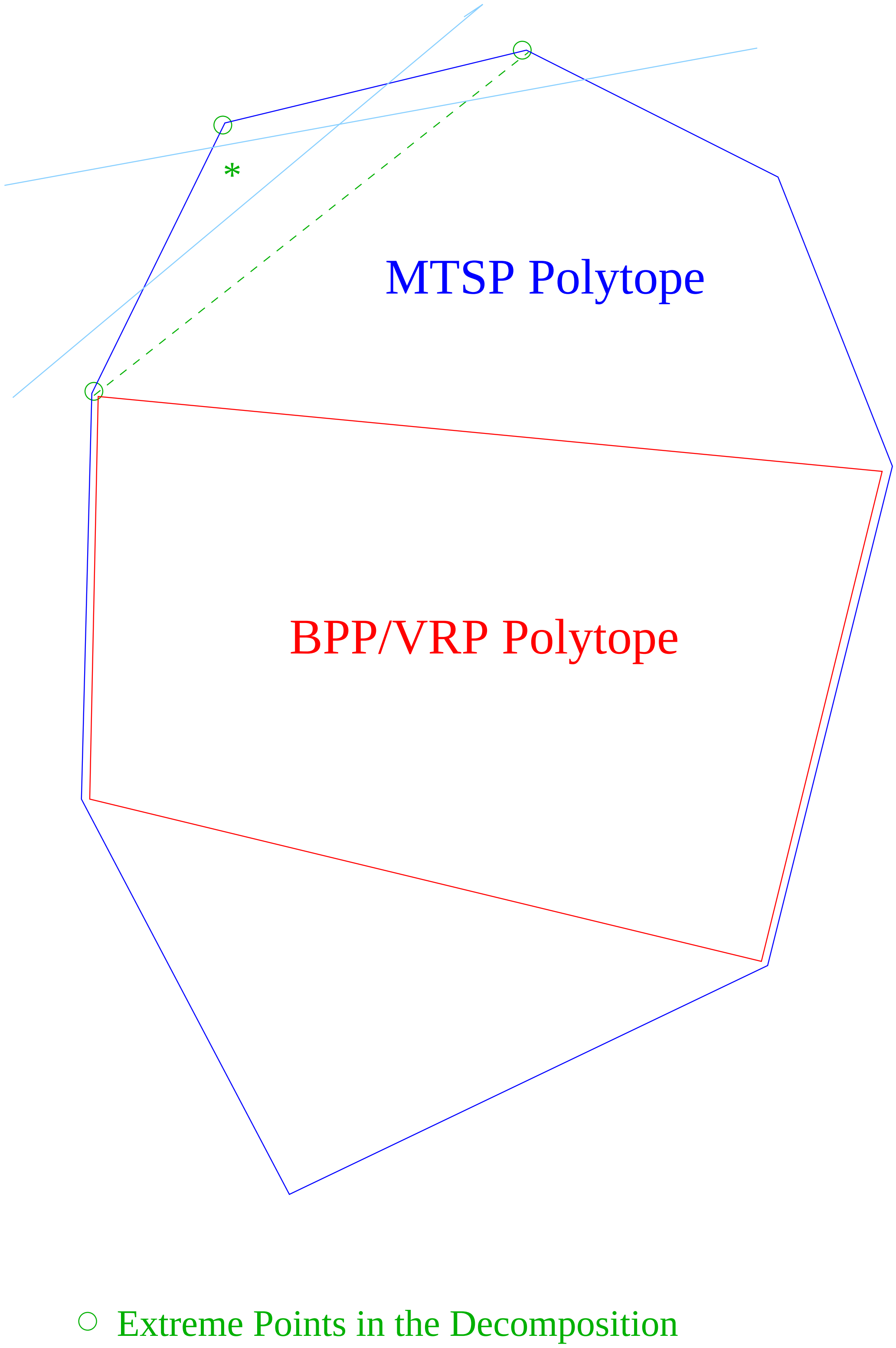


- Key observation: We can determine in $O(n)$ time whether a particular TSP tour satisfies all the capacity constraints.


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




*
MTSP Polytope
BPP/VRP Polytope
Extreme Points in the Decomposition

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 $\mathcal{P} = conv\left(\{x^S : S \in \mathcal{F}\}\right)$

- 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.

Conclusions from the Decomposition Algorithm

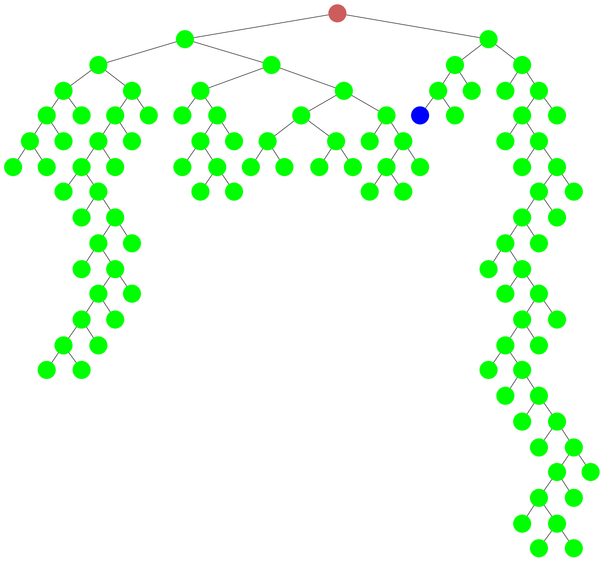
- Our heuristic separation routines for the capacity constraints are effective.
- Decomposition only finds new capacity constraints about 20% of the time.
- There are small examples (13 nodes) that cannot be solved with just capacity constraints.
- About 20% of the time, we are outside the TSP polytope when separation fails $\Rightarrow$ apply TSP cuts.
  - This differs greatly from problem to problem.
- The Farkas inequalities are not that effective.
- Typically, the computational effort doesn't pay off.

Branching

- Branching on Edges
- Branching on Capacity constraints
- Importance of Strong Branching

	1 Candidate	7 Candidates
Nodes	102303	9628
Running Time	17535	1997
SB Time	609	363

- Graduated Strong Branching
- Branching based on Bin Packing



SYMPHONY

SYMPHONY is an object-oriented framework for implementing parallel BCP algorithms.

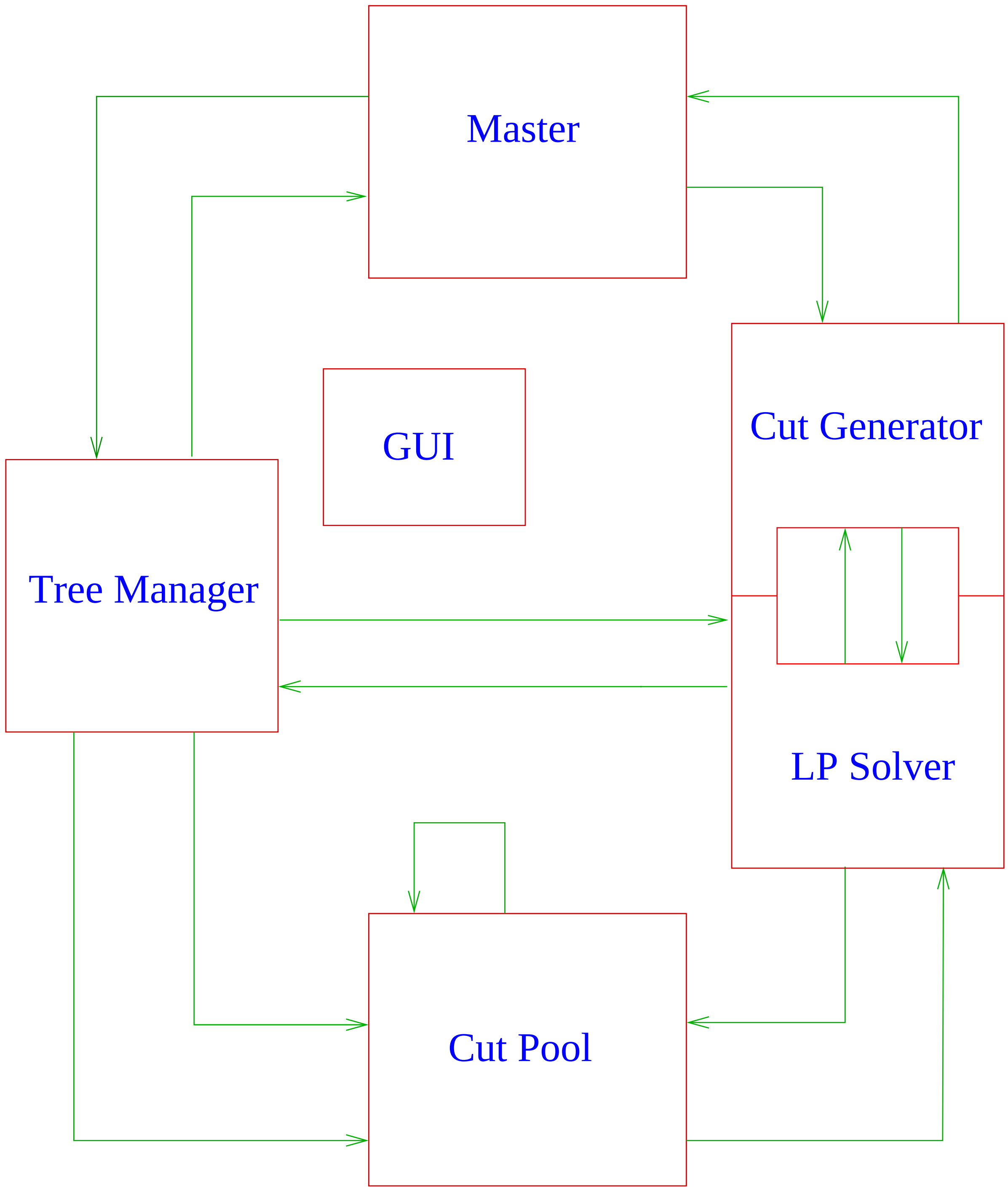
- It was designed specifically to run in a parallel environment.

- 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



	With Decomposition			Without Decomposition		
problem	Tree Size	Tree Depth	Wallclock	Tree Size	Tree Depth	Wallclock
E − n30 − k3	123	17	4.01	206	25	3.53
E − n31 − k7	7	3	1.08	36	31	3.32
E − n33 − k4	8	4	1.67	5	3	0.35
E − n51 − k5	24	7	37.74	42	2	22.86
gr − n21 − k3	1	0	0.05	4	3	0.14
gr − n24 − k4	4	3	1.06	4	7	0.32
fri − n26 − k3	23	9	0.95	16	10	.35
swiss − n42 − k5	10	4	4.69	23	21	1.52
att − n48 − k4	346	24	112.85	440	9	30.58
gr − n48 − k3	31	11	5.86	36	12	2.92
hk − n48 − k4	115	11	87.28	122	7	17.02
A − n32 − k5	8	3	0.79	12	6	0.70
A − n33 − k5	13	4	4.53	9	5	0.78
A − n33 − k6	17	6	8.84	36	7	2.41
A − n34 − k5	15	6	4.80	15	7	1.51
A − n36 − k5	45	10	14.83	91	10	7.92
A − n37 − k5	12	6	6.40	18	5	2.31
A − n38 − k5	195	16	46.45	351	20	21.55
A − n45 − k6	315	18	253.28	422	20	120.59
A − n46 − k7	6	3	6.27	8	3	5.83
A − n44 − k6	1403	23	935.67	1966	24	621.22
A − n53 − k7	788	19	1379.20	1222	18	733.20
B − n31 − k5	8	7	0.80	7	3	0.20
B − n34 − k5	562	16	27.34	1120	21	17.76
B − n38 − k6	28	7	13.25	88	12	6.43
B − n39 − k5	10	7	0.61	5	4	0.20
B − n41 − k6	66	11	32.51	43	9	3.40
B − n43 − k6	551	23	341.53	509	22	35.09
B − n44 − k7	1	0	0.88	4	3	0.55
B − n45 − k5	58	12	31.04	52	8	5.04
B − n50 − k7	3	3	0.50	8	5	0.52
B − n51 − k7	294	41	102.94	340	36	23.44
B − n52 − k7	10	8	2.59	21	11	0.85
B − n56 − k7	97	15	27.41	91	13	8.63
B − n64 − k9	18	7	24.06	18	7	8.51
Total	5215	–	3523.76	7390	–	1711.55
E − n76 − k7	121,811	–	288,640	115,991	–	278,613
E − n76 − k8	484,245	–	1,927,422	–	–	–
E − n101 − k8	244,968	–	1,900,671	–	–	–

Future Work

- Add known cutting planes to our implementation
- Learn how to solve the **Bin Packing Problem**
- New formulation??
- New branching rules
- New inequalities from Bin Packing
- Use decomposition sparingly
- Incorporate TSP cuts when “outside” the TSP polytope