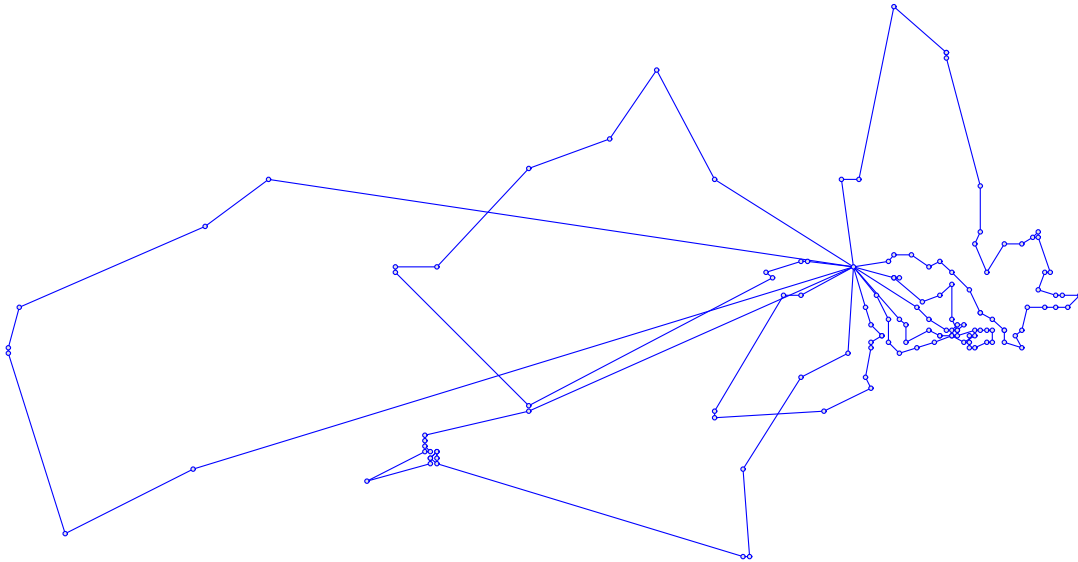


Branch, Cut, and Price for Routing and Packing Problems



Ted Ralphs

Joseph Hartman

Department of Industrial and Systems Engineering

Lehigh University

Bethlehem, PA

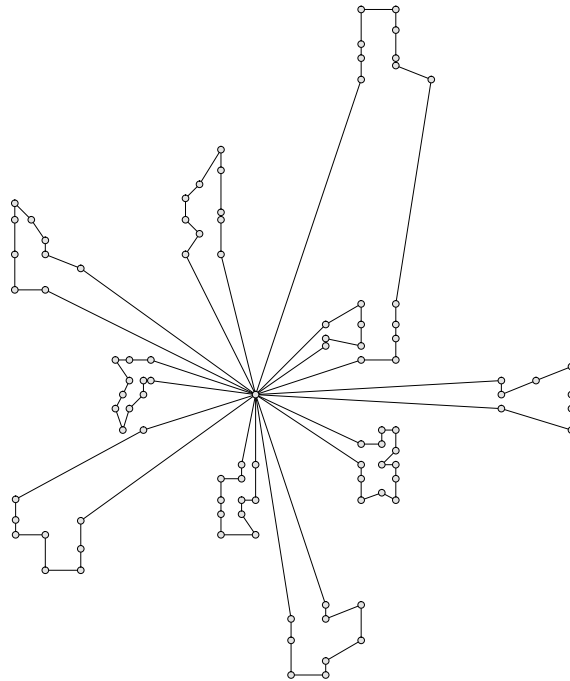
tkralphs@lehigh.edu

www.lehigh.edu/~tkr2

www.BranchAndCut.org

Outline of Talk

- Introduction and Motivation
- Modeling
- Complexity and Special Cases
- Polyhedral Structure
- Implementation
- Computational Issues and Results
- Future Directions



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** $\{R_1, \dots, R_k\}$ 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 .

Standard IP Formulation for the VRP

IP Formulation:

$$\sum_{j=1}^n x_{0j} = 2k$$

$$\sum_{j=1}^n x_{ij} = 2 \quad \forall i \in V^-$$

$$\sum_{\substack{i \in S \\ j \notin S}} x_{ij} \geq 2b(S) \quad \forall S \subset V^-, |S| > 1.$$

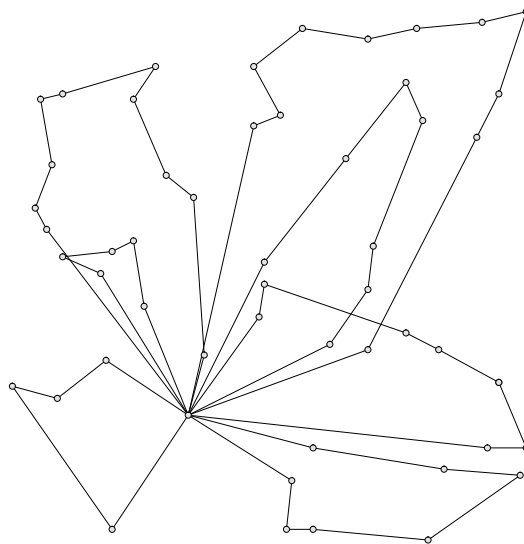
$b(S)$ = lower bound on the number of trucks required to service S (normally $\lceil (\sum_{i \in S} d_i) / C \rceil$).

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.

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

17 21

22

14

12
21 24 26

69

18

20 21

14 12

16 10 16 5 23

16

15

19 4 26 10

8

10 18 3 7 12 10

13

23

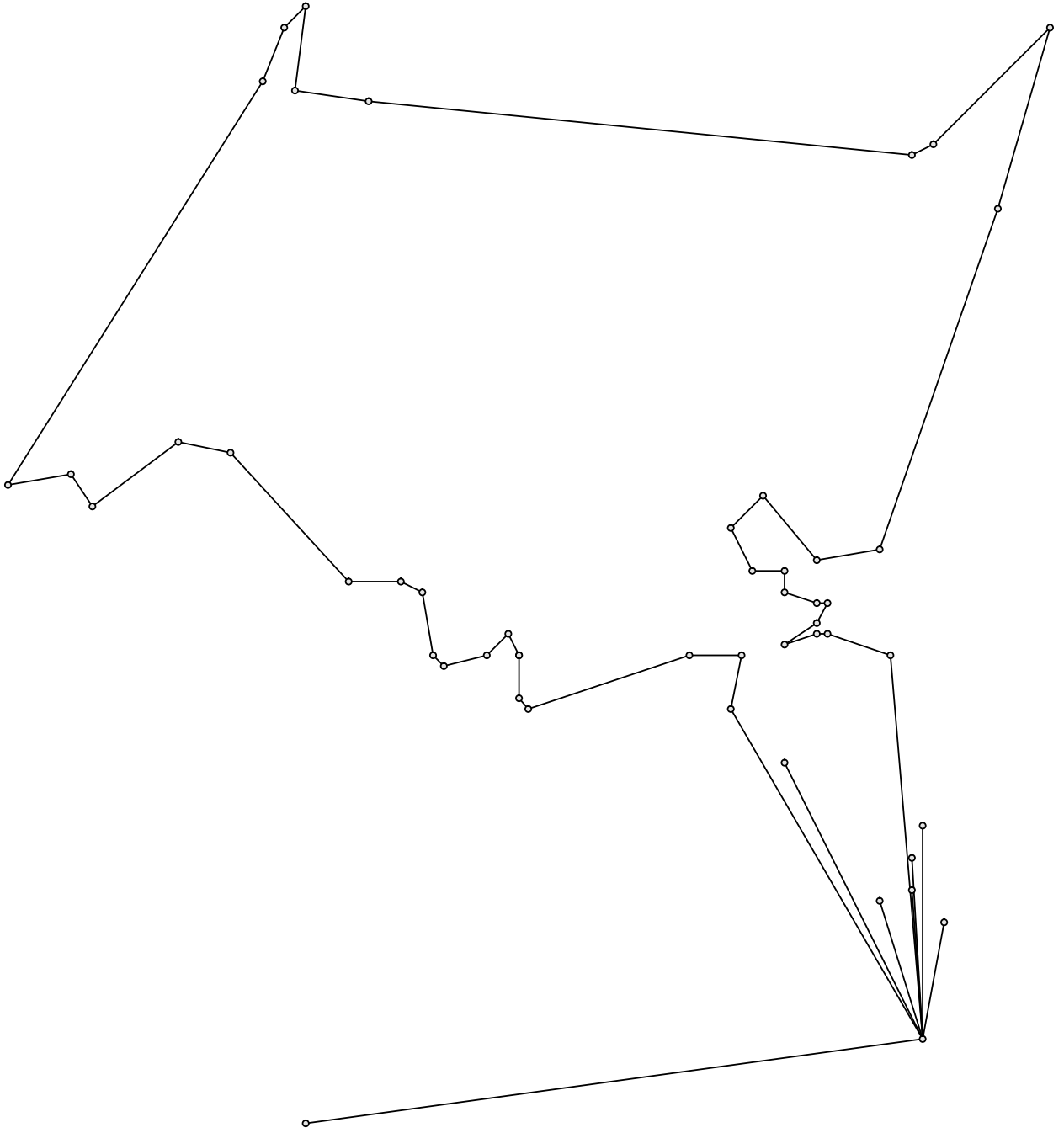
25

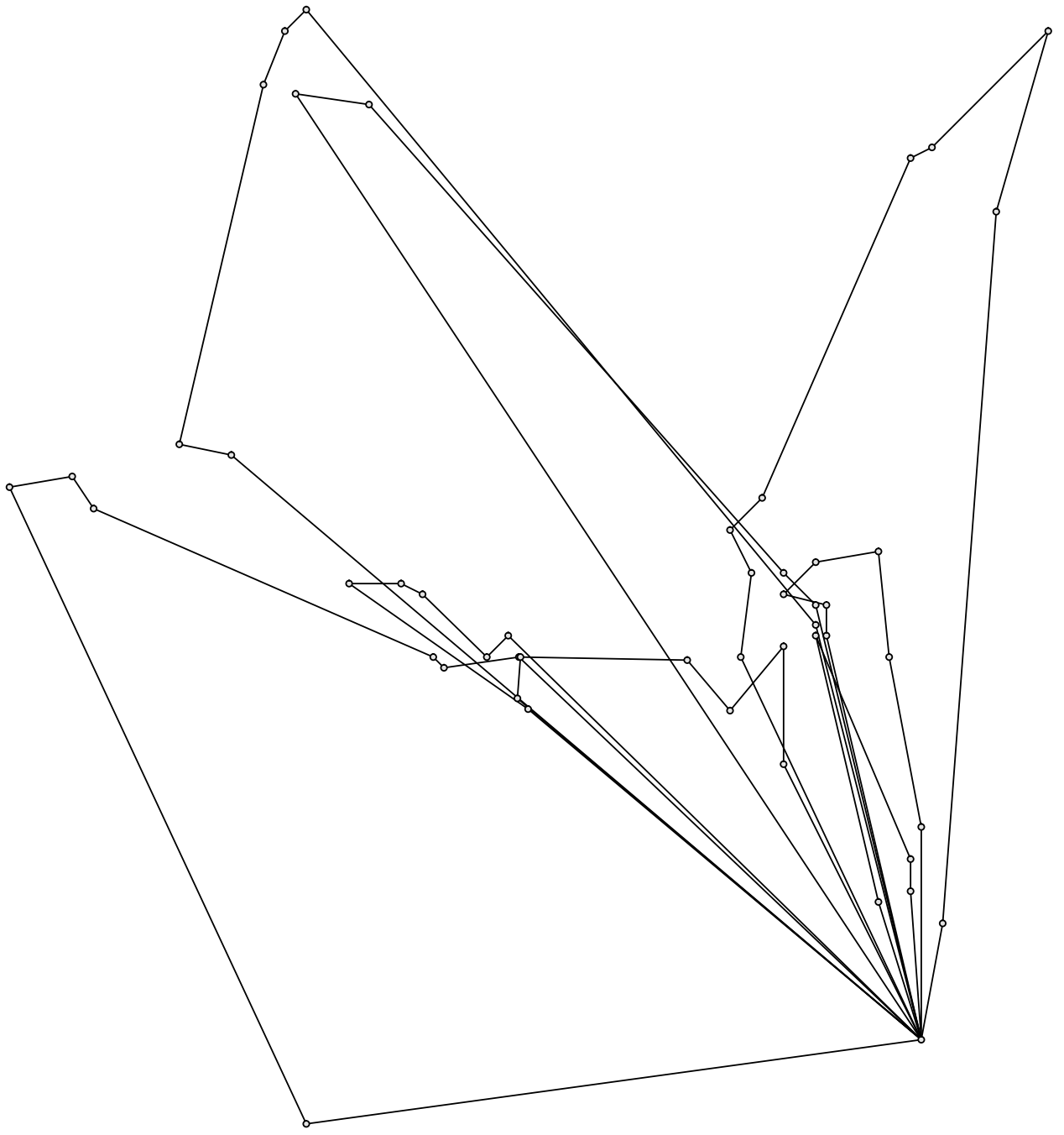
5 15 3 2 3

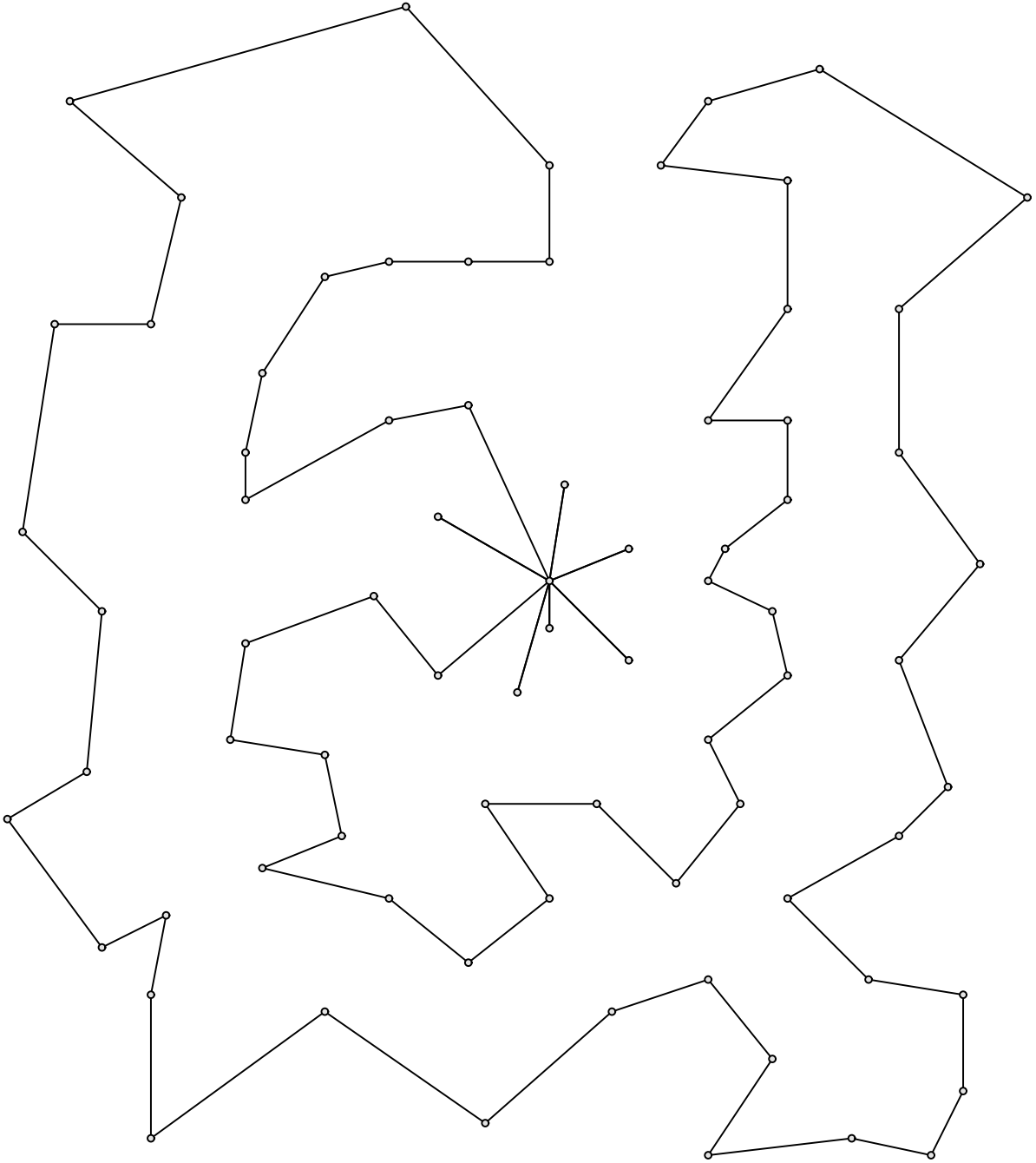
25 2

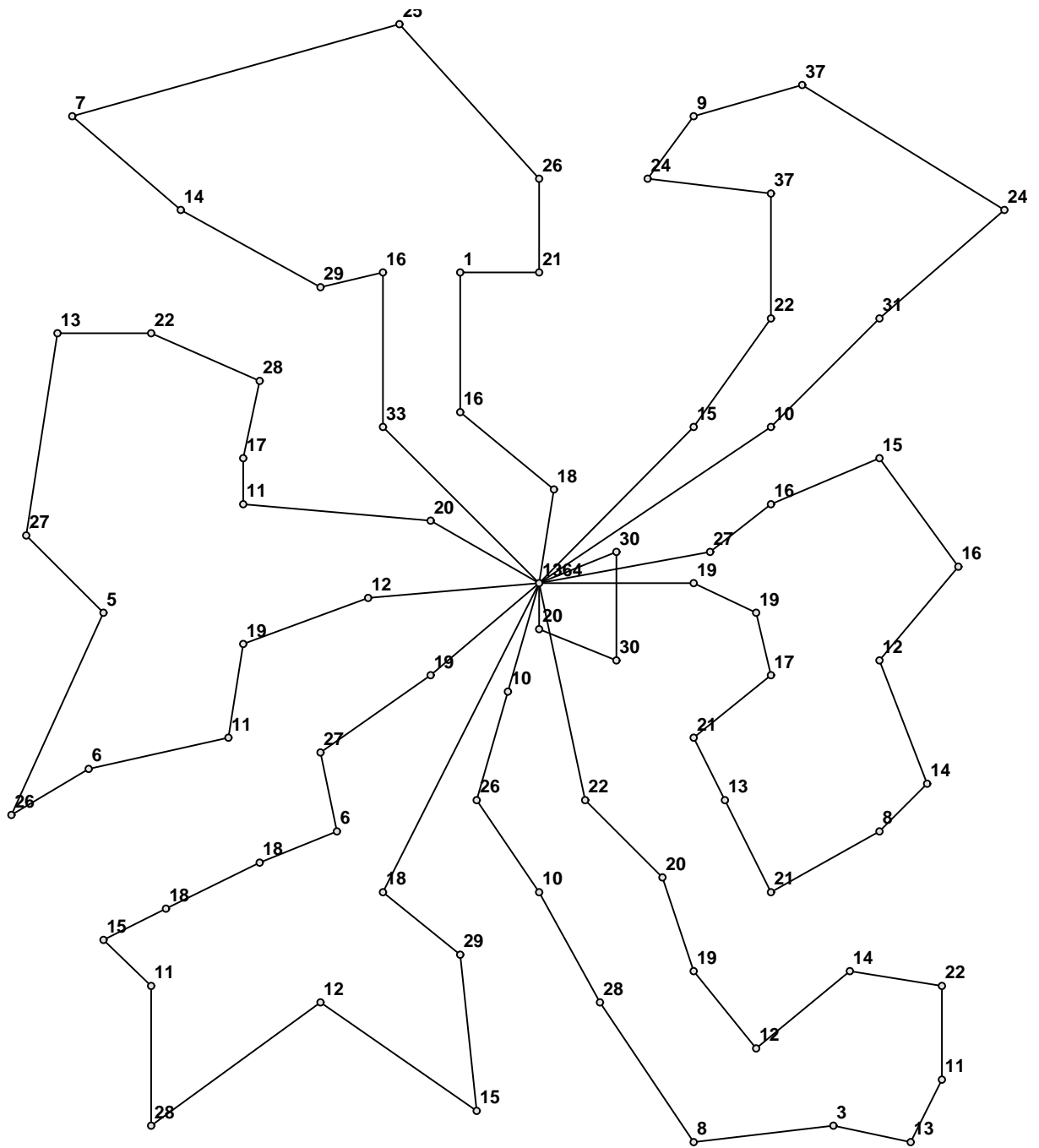
9

12



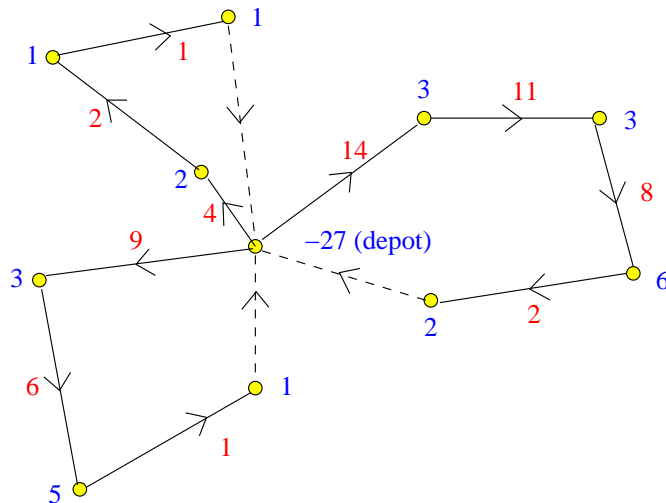


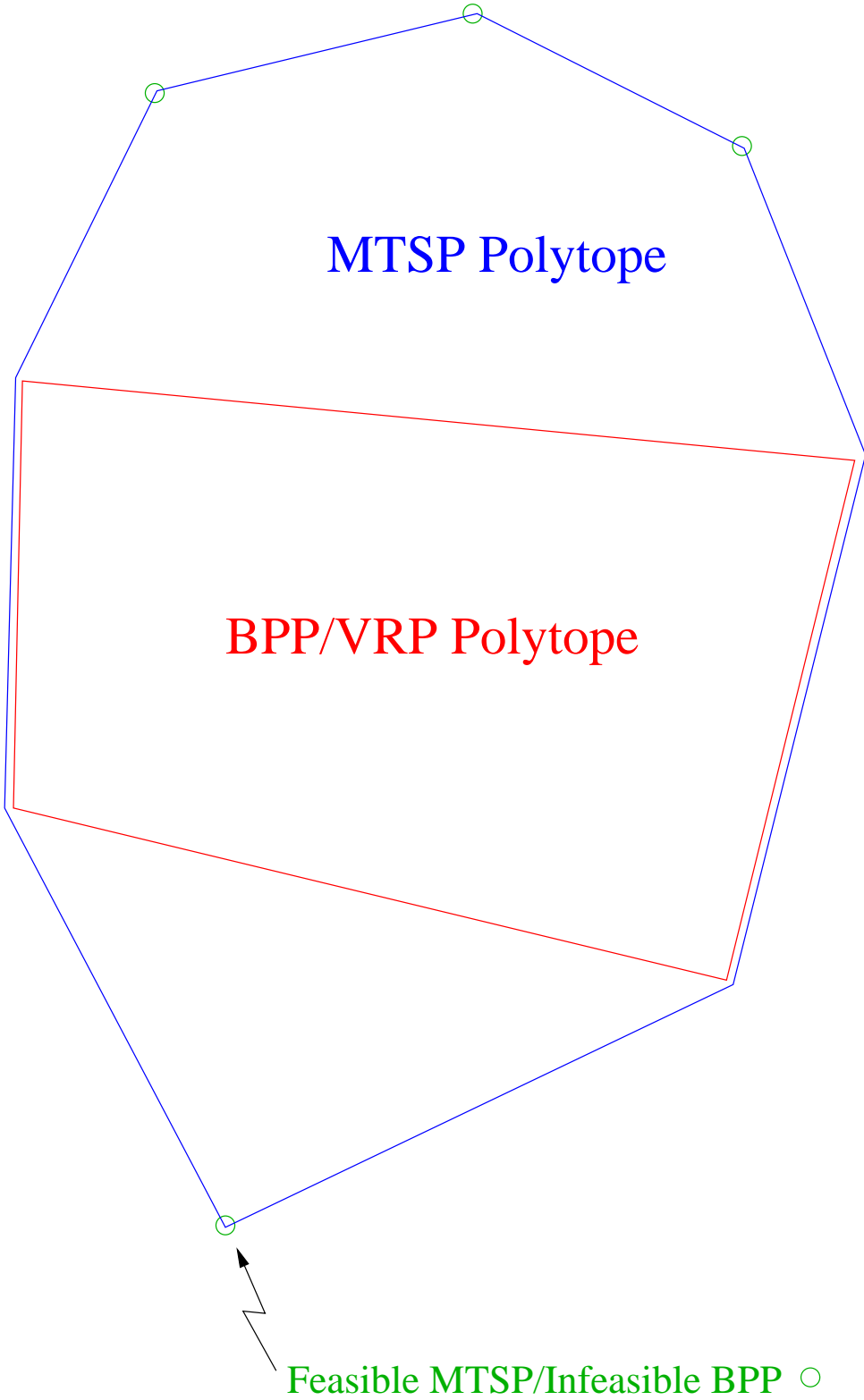




Motivation

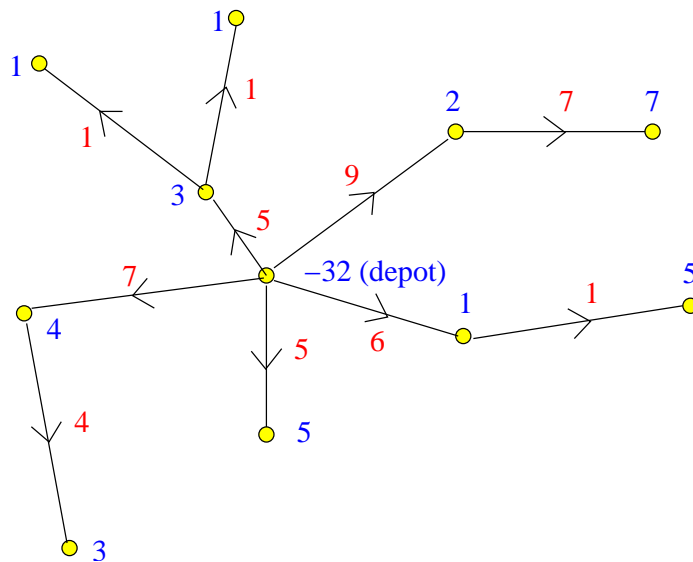
- Why is the **Vehicle Routing Problem** difficult?
- It is the intersection of two difficult problems.
 - **Traveling Salesman Problem** (**Routing**)
 - **Bin Packing Problem** (**Packing**)
- We don't have an effective, polynomially sized relaxation.
- Current approaches treat it as a **routing** problem.
- We know very little about the **packing** aspect.
- **Idea**: Consider flow-based formulations.





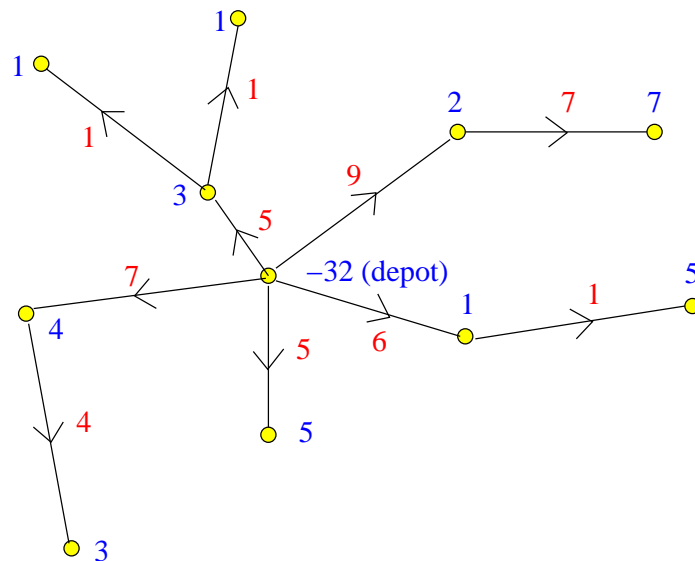
Node Routing

- We are given an undirected graph $G = (V, E)$.
 - The edges represent **transportation arteries** or **communication links**.
 - Each edge has an associated *cost* or *length*.
 - The nodes represent **supply/demand** points.
- Assume one supply point (the *depot*).
- A *node routing* is a directed subgraph G' of G satisfying the following properties:
 - G' is (weakly) connected.
 - The **in-degree** of each non-depot node is 1.



Optimal Node Routing

- Properties of a **node routing**.
 - It is a **spanning arborescence** plus (possibly) some edges returning to the depot.
 - There is a **unique path** from the depot to each demand point.
- We wish to construct a **least cost routing**.
- Cost Measures
 - Sum the lengths of all edges in G' .
 - Sum the length of all paths from the depot.
 - Some linear combination of these two.



IP Formulation

IP formulation for this routing problem:

$$\text{Min} \quad \sum_{(i,j) \in A} \gamma c_{ij} x_{ij} + \tau c_{ij} f_{ij}$$

$$\text{s.t.} \quad x(\delta(V \setminus \{i\})) = 1 \quad \forall i \in V \setminus \{0\}$$

$$f(\delta(V \setminus \{i\})) - f(\delta(\{i\})) = d_i \quad \forall i \in V \setminus \{0\}$$

$$f_{ij} \leq M x_{ij} \quad \forall (i, j) \in A$$

$$f_{ij} \geq 0 \quad \forall (i, j) \in A$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A$$

where:

- $V^- = V \setminus \{0\}$.
- x_{ij}, x_{ji} (*fixed-charge variables*) indicate whether $\{i, j\}$ is in the routing *and its orientation*.
- f_{ij} (*flow variable*) represents demand flow.

Complexity

- This node routing problem is **NP-complete** in general.
- Polynomially solvable special cases.
 - $\tau = 0 \Rightarrow$ **Minimum Spanning Tree Problem**.
 - $\gamma = 0 \Rightarrow$ **Shortest Paths Tree Problem**.
 - Note that **demands are irrelevant**.
- Other special cases.
 - $\tau, \gamma > 0 \Rightarrow$ **Cable-Trench Problem (CTP)**.
 - $\tau = 0$ and $x(\delta(\{i\})) = 1 \Rightarrow$ **Traveling Salesman Problem (TSP)**.
 - $\tau > 0$ and $x(\delta(\{i\})) = 1 \Rightarrow$ **Variable Cost TSP (VCTSP)**.
 - $x(\delta(V \setminus \{0\})) = x(\delta(\{0\})) = k \Rightarrow$ **k -TSP**.

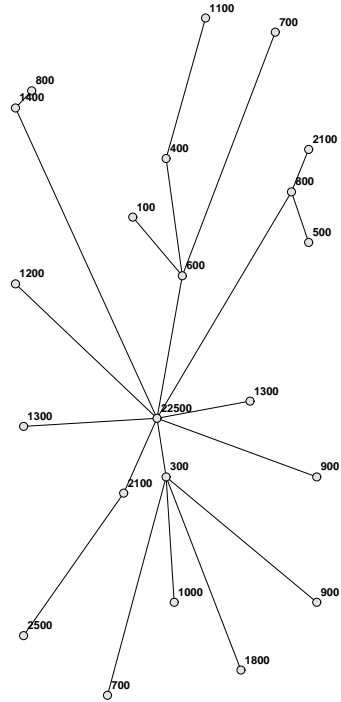
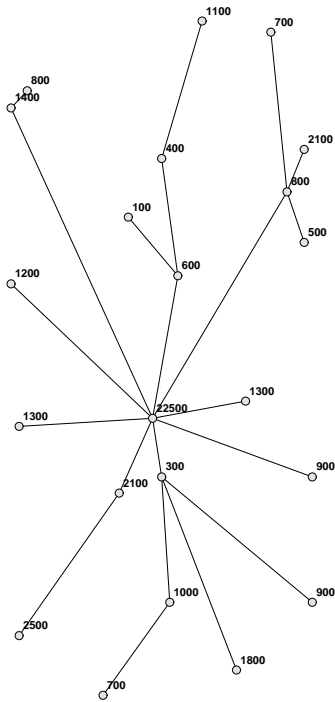
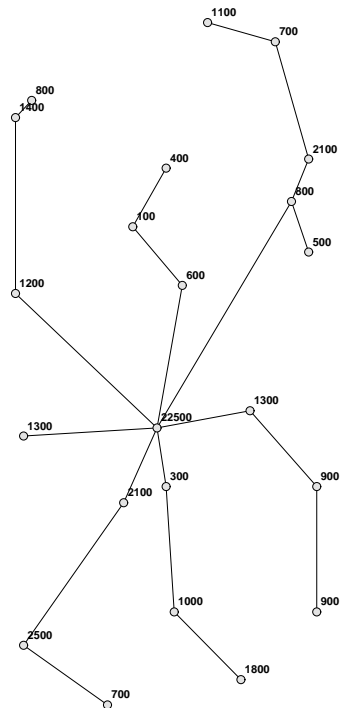
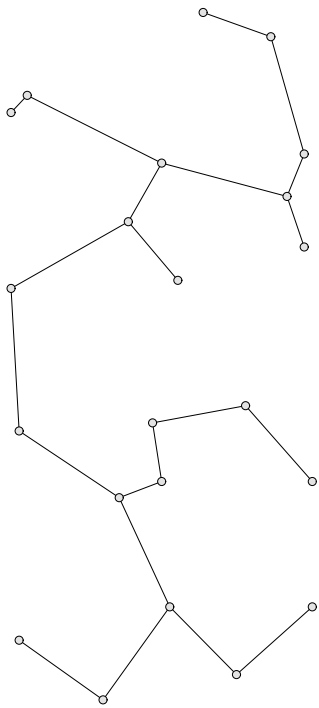
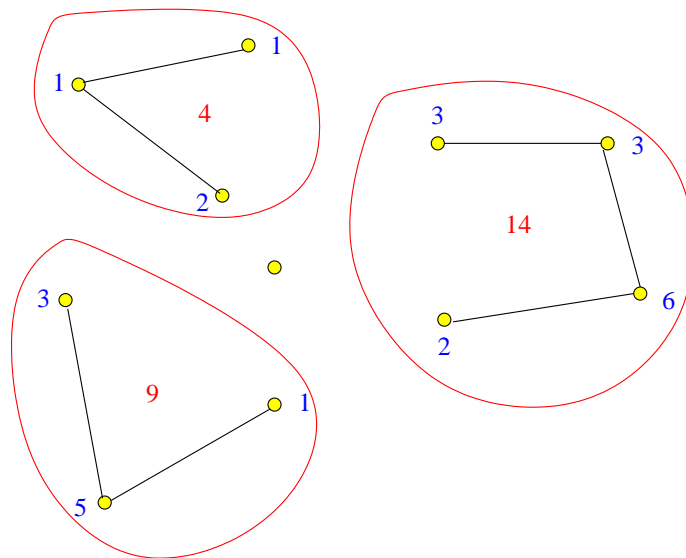


Figure 1: Sample Spanning Trees

Packing

- We now want to *capacitate* the model.
- A *capacitated routing* is one in which the demand in each component of $G' \setminus \{0\}$ is $\leq C$.
- Even the feasibility problem is NP-complete.
- This case is easily modeled by setting $M = C$.
- We can now model the **VRP** and its variable cost generalization.
- We can also model the **Capacitated Spanning Tree Problem (CSTP)**.



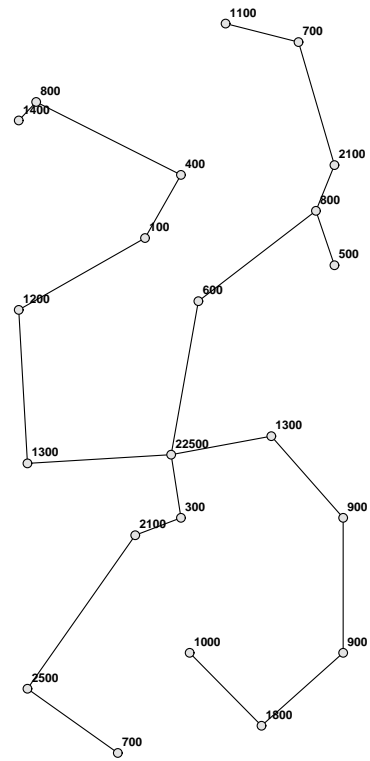
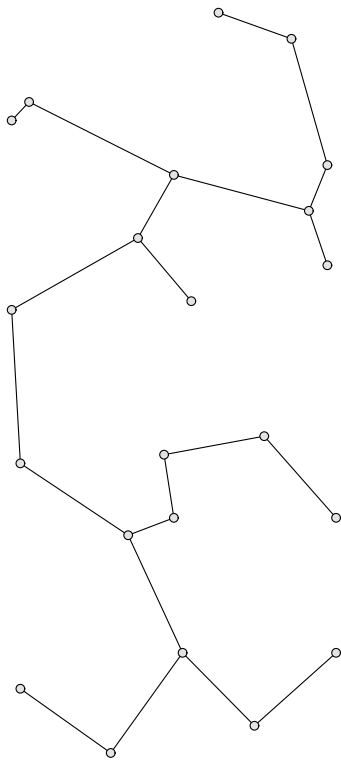


Figure 2: Sample Capacitated Spanning Trees

Valid Inequalities

- Note that any inequalities valid for the **TSP**, **VRP**, or **CSTP** have counterparts here.
- Many can be strengthened by taking advantage of the directed formulation.
- Fractional Capacity Constraints

$$\sum_{i \notin S, j \in S} x_{ij} \geq d(S)/C, 0 \notin S$$

- Multi-star Inequalities

$$\sum_{i \notin S, j \in S} x_{ij} \geq d(S)/C + \frac{\sum_{i \notin S, j \in S} x_{ji} d_i}{C}, 0 \notin S$$

- Rounded Capacity Constraints

$$\sum_{i \notin S, j \in S} x_{ij} \geq \lceil d(S)/C \rceil$$

- Generalized, framed capacity constraints
- Combs, Hypo-tours, Clique Clusters
- Path-bin inequalities

Flow Linking

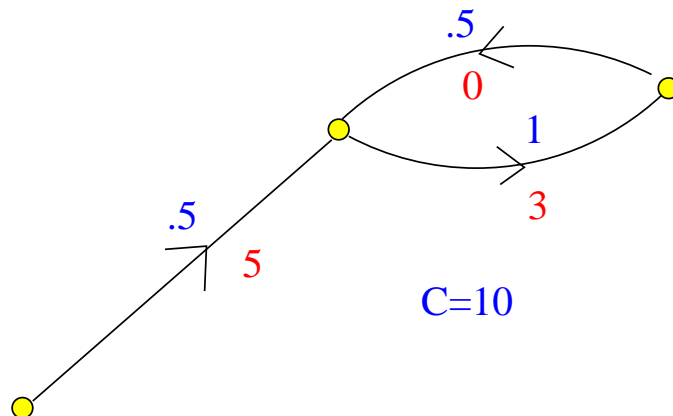
- Note that only the edge variables are required to be integral (Benders' decomposition??).
- We use the flow variables to force integrality of the edge variables through *flow linking constraints*.
- Flow Linking Constraints

$$f_{ij} \leq (C - d_i)x_{ij} \Leftrightarrow x_{ij} \geq \frac{f_{ij}}{C - d_i}$$

$$f_{ij} - \sum_{k \neq j} f_{jk} \leq x_{ij}d_j$$

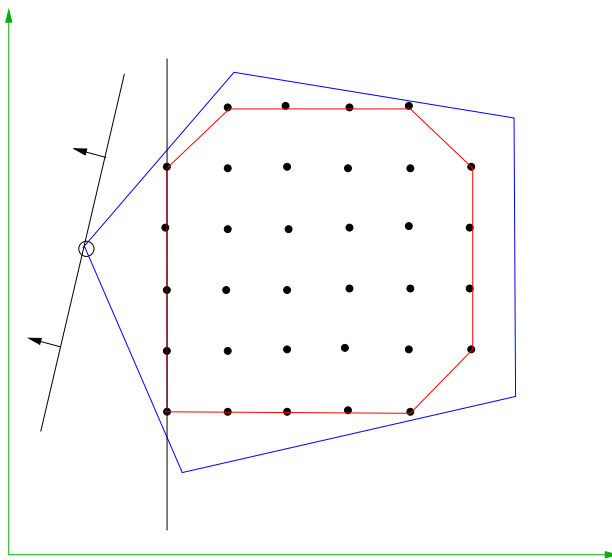
- Edge Cuts

$$x_{ij} + x_{ji} \leq 1$$



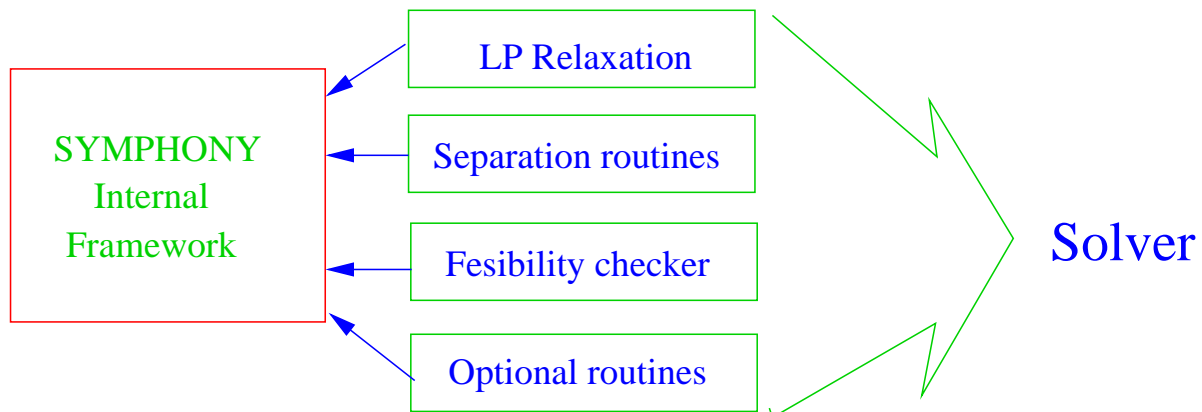
Separation

- The **fractional capacity constraints** and **multi-star inequalities** are automatically satisfied.
- **Flow linking constraints** and **edge cuts** can be included explicitly or separated in **polynomial time**.
- Separating **rounded capacity constraints** is **NP-complete**, but can be done effectively.
- Heuristic procedures for other classes have not yet been implemented.



Implementation

- The implementation uses **SYMPHONY**, a parallel framework for branch, cut, and price (relative of COIN/BCP).
- In **SYMPHONY**, the 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



Formulation Issues

- The new formulation is **polynomial** and yields **stronger relaxations**, but there are drawbacks.
- For the **VRP**, the formulation creates **symmetry** and is not yet effective.
- It also seems to make branching less effective.
- There is a related “undirected” formulation which uses one variable per edge.
 - This formulation is smaller and performs much better for the **VRP**.
 - For the **CSTP** and **CTP**, the undirected formulation is extremely weak.
 - We cannot model the **VCTSP** this way.

Other Computational Issues

- The **packing constraints** increase difficulty.
- Current approaches do not account for this.
- Consider relaxations of the **VRP**.
 - The **TSP** is very easy relative to the **VRP**.
 - The **CSTP** is not much easier than the **VRP**.
- The **VCTSP** is extremely difficult for our methods.
 - On a 22 node instance, the gap was **25%** after several hours.
 - Is this due to the **upper bound** or **lower bound**?
 - Our cutting planes and branching rules are ineffective.

problem	Tree Size	Tree Depth	CPU sec	Wallclock
eil13	1	0	0.00	1.42
eil22	1	0	0.02	1.33
eil33	2	1	0.44	2.06
bayg29	4	2	0.32	2.31
bays29	5	5	0.55	2.51
ulysses16.tsp	1	0	0.01	1.49
ulysses22.tsp	1	0	0.03	1.66
gr17	1	0	0.01	1.61
gr21	1	0	0.03	1.09
gr24	4	3	0.40	2.02
fri26	8	8	0.39	1.72
swiss42	10	4	2.45	4.33
att48	193	15	30.10	32.97
gr48	16	5	4.17	6.10
hk48	45	8	21.19	23.79
eil51	11	5	10.79	12.28
A – n32 – k5	2	1	0.20	1.57
A – n33 – k5	7	4	0.90	2.17
A – n34 – k5	9	5	2.63	4.35
A – n36 – k5	51	14	7.95	9.73
A – n37 – k5	11	6	0.97	2.45
A – n38 – k5	111	14	21.80	23.91
A – n39 – k5	480	24	310.92	320.90
A – n44 – k6	1185	21	1525.78	1559.00
A – n45 – k6	133	12	145.59	150.63
A – n46 – k7	2	1	1.95	3.54
A – n48 – k7	1949	33	1620.57	1660.62
A – n53 – k7	619	18	881.05	905.39
B – n31 – k5	1	0	0.08	1.52
B – n38 – k6	14	6	1.73	3.95
B – n39 – k5	1	0	0.05	1.50
B – n41 – k6	20	6	2.89	4.24
B – n43 – k6	138	13	34.92	37.69
B – n45 – k5	18	8	5.81	7.86
B – n51 – k7	129	11	32.48	34.92
B – n52 – k7	26	11	0.76	2.30
B – n56 – k7	1	0	0.29	2.16
Total	5211		4670.22	4839.07

Figure 3: Vehicle Routing Problem

problem	Tree Size	Tree Depth	CPU sec	Wallclock
eil13	13	6	0.09	1.61
eil22	2	2	0.10	1.70
eil33	69	12	3.97	6.25
bayg29	1	0	0.04	2.32
bays29	15	5	1.12	2.80
ulysses16.tsp	1	0	0.03	1.82
ulysses22.tsp	1	0	0.06	1.97
gr17	5	3	0.05	2.27
gr21	1	0	0.02	2.24
gr24	5	3	0.27	1.93
fri26	1	0	0.07	2.26
swiss42	35	9	3.66	6.72
att48	92	10	5.04	8.18
gr48	1	0	0.07	2.37
hk48	209	12	22.88	25.24
eil51	77	14	15.11	17.59
A – n32 – k5	1	0	0.07	2.22
A – n33 – k5	3	2	0.21	2.25
A – n34 – k5	4	2	0.40	2.75
A – n36 – k5	52	8	5.17	7.21
A – n37 – k5	5	4	0.22	2.63
A – n38 – k5	1	1	0.13	1.61
A – n39 – k5	11	5	0.99	3.54
A – n44 – k6	586	17	84.08	89.45
A – n45 – k6	47	9	6.19	8.42
A – n46 – k7	3	1	0.20	2.65
A – n48 – k7	775	44	507.41	518.15
A – n53 – k7	115	10	19.99	22.27
B – n31 – k5	3	1	0.63	3.40
B – n38 – k6	5	4	0.56	2.90
B – n39 – k5	188	14	9.67	12.02
B – n41 – k6	216	14	18.96	21.98
B – n43 – k6	1	0	0.36	2.67
B – n45 – k5	22	9	1.13	2.60
B – n51 – k7	1	0	0.13	2.34
B – n52 – k7	1	1	0.20	2.27
B – n56 – k7	38	10	2.20	3.77
Total	2606		711.48	804.38

Figure 4: Capacitated Spanning Tree Problem

problem	Tree Size	Tree Depth	CPU sec	Wallclock
eil13	1	0	0.00	1.98
eil22	1	0	0.11	2.37
eil33	1	0	0.02	1.76
bayg29	1	0	0.12	2.04
bays29	1	0	0.17	2.58
ulysses16.tsp	1	0	0.00	1.82
ulysses22.tsp	1	0	0.00	0.99
gr17	1	0	0.01	1.39
gr21	1	0	0.00	1.72
gr24	1	0	0.02	2.20
fri26	1	0	0.02	1.50
swiss42	1	0	0.02	2.37
att48	2	1	0.30	2.40
gr48	2	1	1.38	3.49
hk48	1	0	0.19	2.31
eil51	1	1	0.16	2.46
A – n32 – k5	1	0	0.02	2.24
A – n33 – k5	3	2	0.81	3.19
A – n34 – k5	6	3	2.06	4.43
A – n36 – k5	1	0	0.03	1.31
A – n37 – k5	1	0	0.03	2.12
A – n38 – k5	1	0	0.10	2.24
A – n39 – k5	1	1	0.30	2.52
A – n44 – k6	3	2	1.72	3.51
A – n45 – k6	2	1	0.27	2.28
A – n46 – k7	1	1	1.25	3.58
A – n48 – k7	2	1	2.01	4.37
A – n53 – k7	1	0	0.62	1.90
B – n31 – k5	1	0	0.01	1.31
B – n38 – k6	1	0	0.04	2.24
B – n39 – k5	1	0	0.03	2.21
B – n41 – k6	1	0	0.08	2.14
B – n43 – k6	1	0	0.09	1.34
B – n45 – k5	1	0	0.09	1.71
B – n51 – k7	1	0	0.36	2.31
B – n52 – k7	1	0	0.19	3.42
B – n56 – k7	1	0	0.10	2.25
Total	50		12.73	86.01

Figure 5: Traveling Salesman Problem

Conclusions and Future Directions

- This is a rich class of problems with interesting properties.
- We hope this approach will yield **new insights**.
- There are some interesting **methodological questions**.
- We need to know much more about the **polyhedral structure**.
- We need some **general methods** for dealing with problem perturbations.
- **Better flow linking** seems to be crucial.
- We also need some **new branching rules**.
- It is easy to **generalize** the model even further.
 - Pickup and delivery problems.
 - General degree constraints.