Branch and Cut for TSP

Jesper Larsen & Jens Clausen

jla,jc@imm.dtu.dk

Informatics and Mathematical Modelling

Technical University of Denmark

Branch-and-Cut for TSP

- Branch-and-Cut is a general technique applicable e.g. to solve symmetric TSP problem.
- TSP is $\mathcal{NP}$-hard – no one believes that there exists a polynomial algorithm for the problem.
- TSP can be formulated as an integer programming problem – for an n-vertex graph the number of binary variables becomes $\frac{n(n-1)}{2}$, and the problem has an exponential number of subtour elimination constraints.

The symmetric TSP

$$
\begin{array}{lll}
\min & \sum_{e \in E} d_e x_e & \\
\text{s.t.} & x(\delta(v)) = 2, & v \in \{1, \ldots, n\} \\
& x(\delta(S)) \geq 2, & \emptyset \subset S \subset V \\
& x_e \in \{0, 1\}, & e \in E
\end{array}
$$

- The number of subtour elimination constraints is huge ($2^{|V|}$) and even though we can remove half of those due to symmetry there are still exponentially many.
- therefore, in the relaxed version we remove the integrality constraints and the exponentially many subtour elimination constraints.

Challenges

For the cutting plane approach to work we need to

1. be able to check whether any subtour elimination constraints are violated (efficiently) and
2. we must be able to solve the LP relaxation efficiently.

Problem 2

- Start by solving a smaller variant of the original problem. Let $E' \subseteq E$ and solve:

$$\begin{array}{ll} \min & \sum_{e \in E'} d_e x_e \\ \text{s.t.} & x(\delta(v)) = 2, \quad v \in \{1, \ldots, n\} \\ & x(\delta(S)) \geq 2, \quad \emptyset \subset S \subset V \\ & 0 \leq x_e \leq 1, \quad e \in E' \end{array}$$

- An **optimal** solution x' for this problem can be extended to a **feasible** solution for the original problem by $x^*_e = x'_e, e \in E'$ and $x^*_e = 0, e \in E \setminus E'$.

- BUT this solution might not be optimal in the original relaxed problem.
- Idea: Look at the dual problem.

Dual of the STSP

$$\begin{array}{lll} \max & \sum_{v\in V} 2y_v + \sum_{S\subset V} 2Y_S & \\ \text{s.t.} & y_u + y_v + \sum_{(u,v)\in\delta(S)} Y_S \le d_{uv}, & (u,v) \in E' \\ & Y_S \ge 0, & S \subset V \end{array}$$

- If (y', Y') is also feasible for the dual linear programming problem of the original problem then we know that x^* is optimal
- Otherwise add variables to E' that violated the constraint of the dual linear programming problem and resolve.

Problem 1

- Here we use branch-and-cut.
- Start of by removing the subtour elimination constraints. Then we get:

$$\begin{array}{ll} \min & \sum_{e \in E} d_e x_e \\ \text{s.t.} & x(\delta(v)) = 2, \quad v \in \{1, \ldots, n\} \\ & 0 \leq x_e \leq 1, \quad e \in E \end{array}$$

- Let x^* be a feasible solution to the initial linear programming problem.

- If the solution falls apart into several components then the node set S of each component violates a subtour elimination constraint. This situation is very easy to detect.
- We might end in a situation where the graph is not disconnected but there are actually subtour elimination constraints that are violated. How do we detect those?

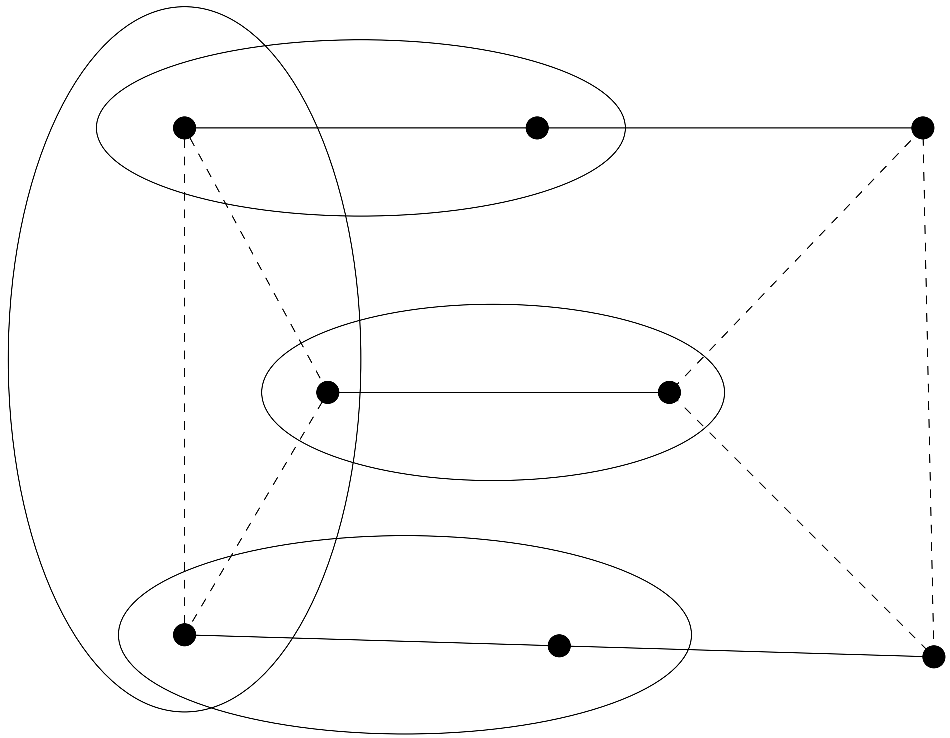
The separation algorithm

Use max-flow to find cuts that are violated in the present situation. Here we have two problems:

- Max-flow works on directed graphs – this is a non-directed graph.
- We need a sink and a source to run the max-flow algorithm.

Now we are in a good position. We are now able to detect all possible subtour elimination constraints, but is that enough to solve the problem?

Consider the following part of a graph (dash is a flow of 0.5).



Comb inequalities

Let C be a comb with a handle H and teeth $T_1, T_2, \ldots, T_{2k+1}$ for $k \geq 1$. Then the solution x for a feasible solution must satisfy:

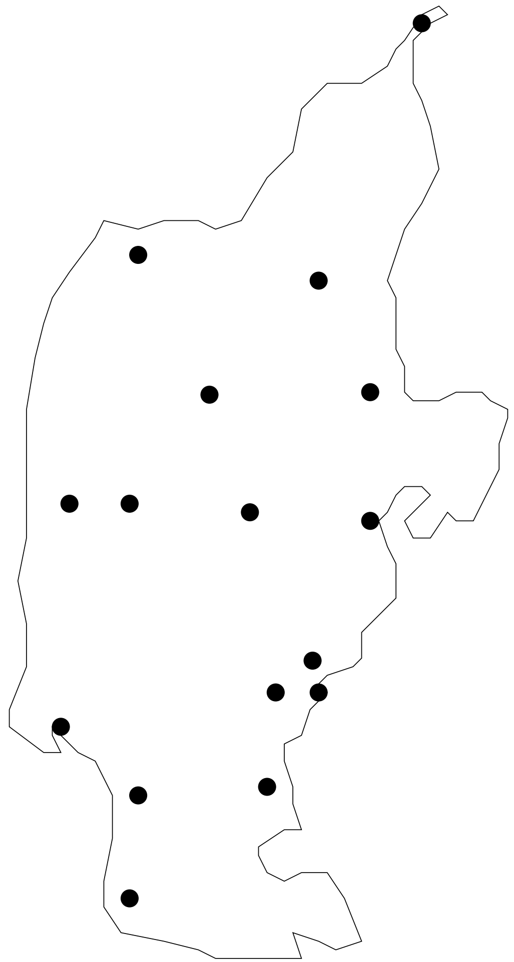
$$x(E(H)) + \sum_{i=1}^{2k+1} x(E(T_i)) \leq |H| + \sum_{i=1}^{2k+1} (|T_i| - 1) - (k+1)$$

These cuts are generally still not enough but there are more cuts we could add:

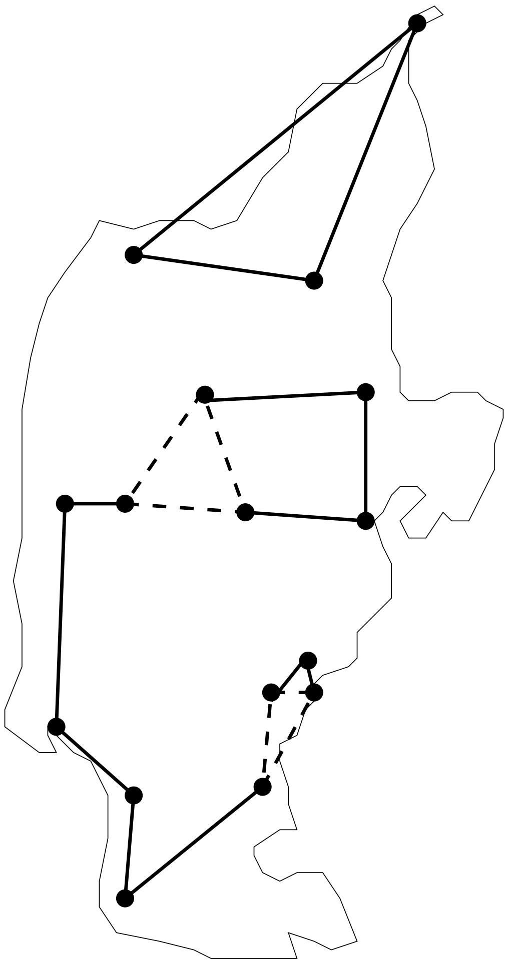
- Grötschel and Padberg (1985)
- Jünger, Reinelt and Rinaldi (1995)
- Naddef (1990)

Even these are not enough. There is today no full description of the convex hull for the TSP. Furthermore for some of the valid inequalities the exists no efficient (polynomial) separation algorithm.

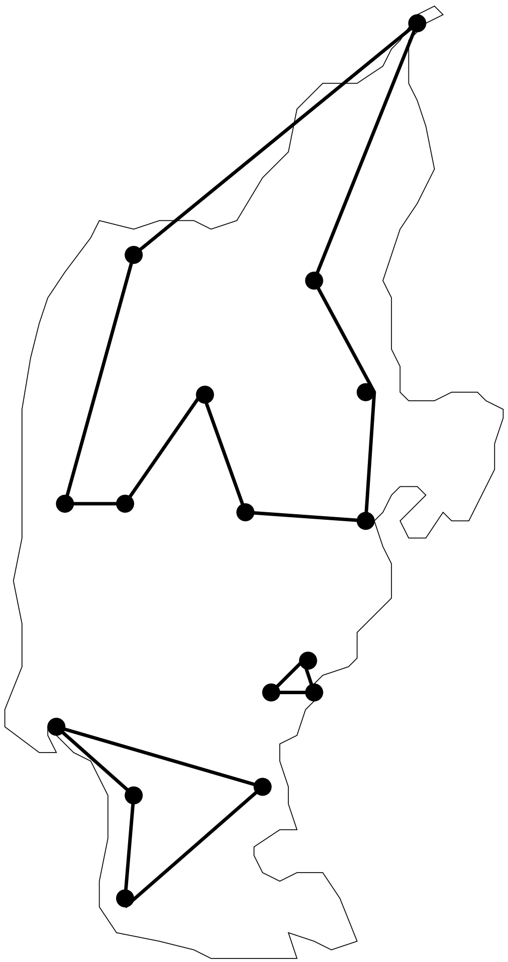
- THEREFORE branch after having added all the “simple” valid inequalities.
- **Example:**
  - Upper bound: 56892 (simple heuristic)
  - Lower bound: 56785 (LP-relaxation, subtour and simple comb-ineq)
  - Gap: 0.2% !!



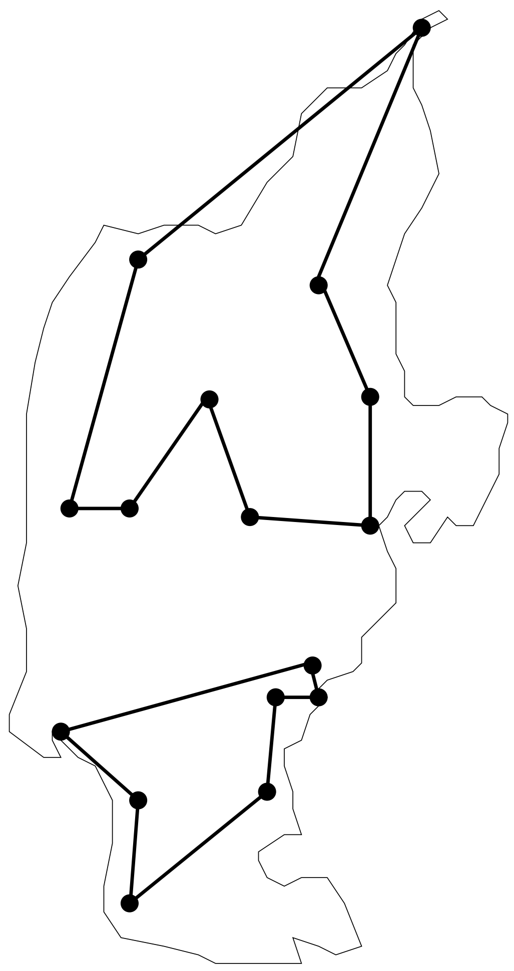
- 16 cities – $15 + 14 + 13 + \ldots + 2 + 1 = \frac{16 \times 15}{2} = 120$ variables.
- Let us keep the constraint that $\sum_j x_{ij} = 2,\ i = 1, \ldots, N$.
- Relax integrality constraints on variables to $0 \leq x_{ij} \leq 1$



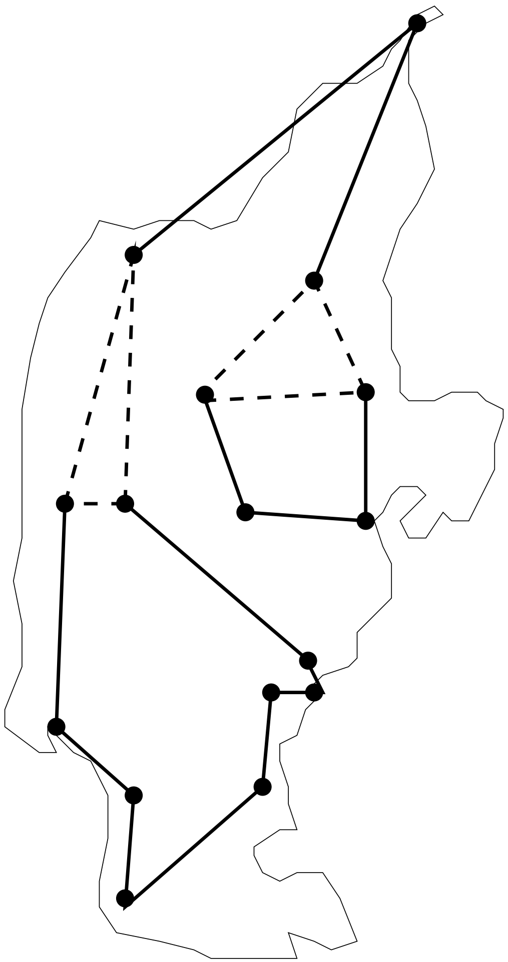
- Objective value: 920
- Forbid the subtour Skagen-Thisted-Aalborg and resolve



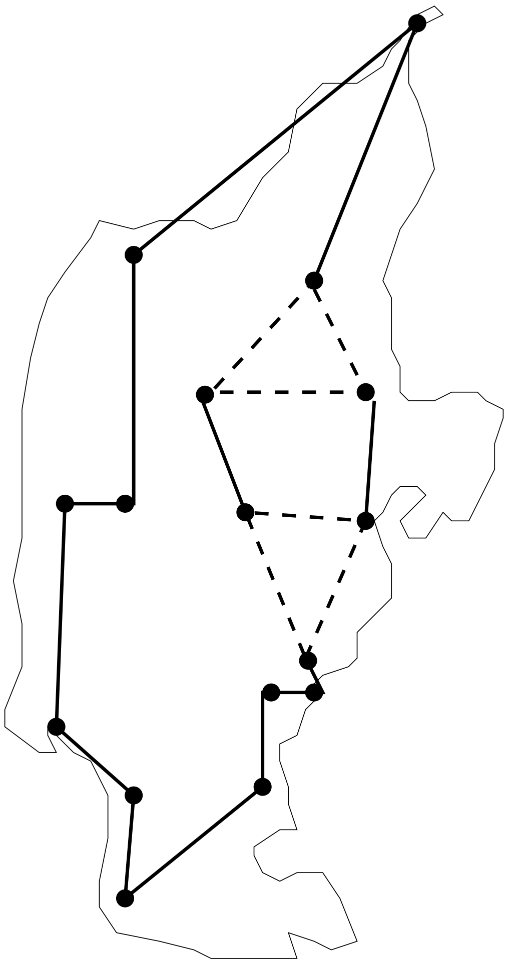
- Objective value: 960
- Forbid the subtour Fredericia-Kolding-Vejle and resolve



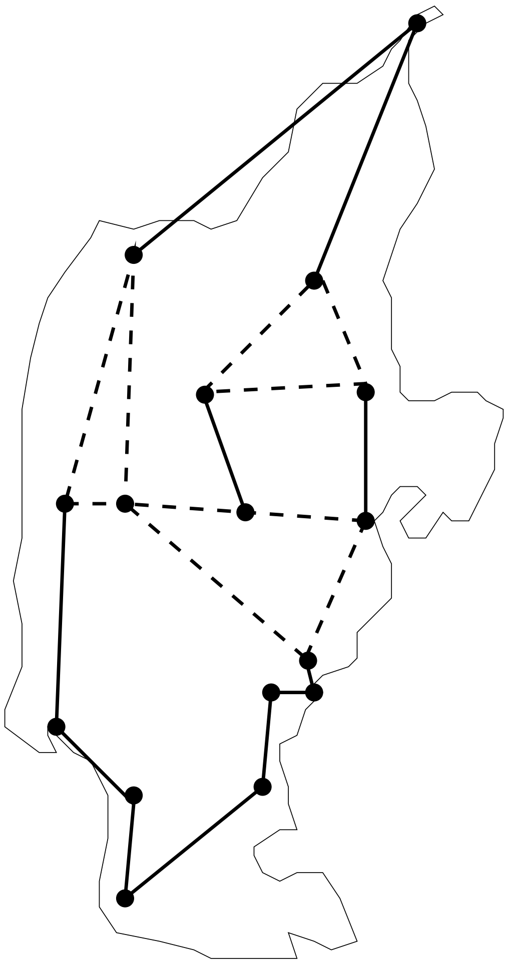
- Objective value: 982
- Forbid the subtour Kolding-Fredericia-Vejle-Esbjerg-Aabenraa-Tønder-Ribe and resolve



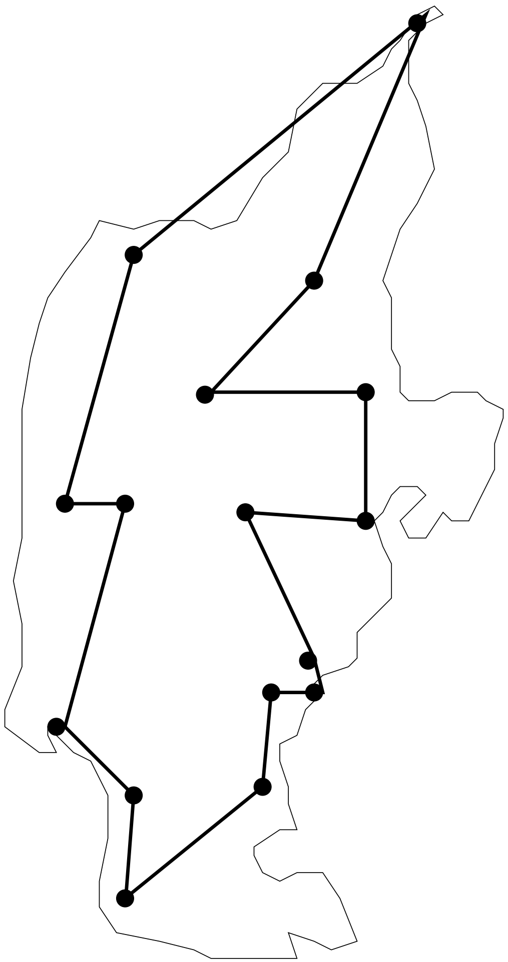
- Objective value: 992.5
- Identify a comb inequality: Handle being Thisted, Ringkøbing and Herning; teeth being (Thisted, Skagen), (Ringkøbing, Esbjerg) and (Herning, Vejle).



- Objective value: 992.5
- Identify a comb inequality: Handle being Vejle, Silkeborg and Aarhus; teeth being (Vejle, Fredericia), (Silkeborg, Viborg) and (Aarhus, Randers).



- Objective value: 994.5
- Identify a comb inequality: Handle being Viborg, Randers and Aalborg; teeth being (Viborg, Silkeborg), (Randers, Aarhus) and (Aalborg, Skagen).



- Objective value: 996
- Integer solution!!!