Branch and Cut for TSP

Jesper Larsen

jesla@man.dtu.dk

Department of Management Engineering
Technical University of Denmark

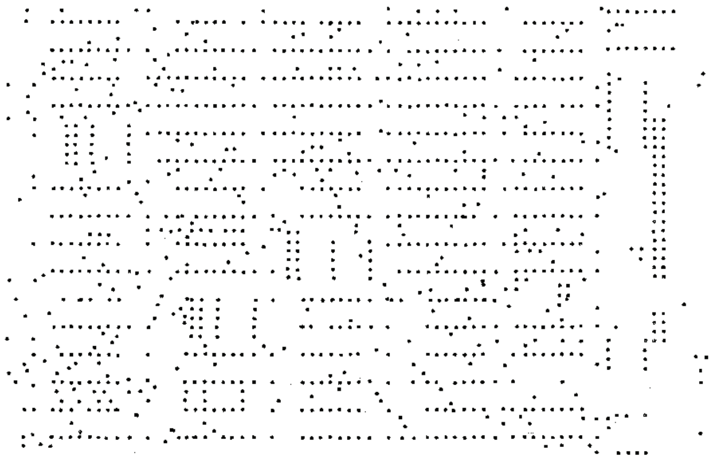
Branch-and-Cut for TSP

- 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 cut-set elimination constraints.

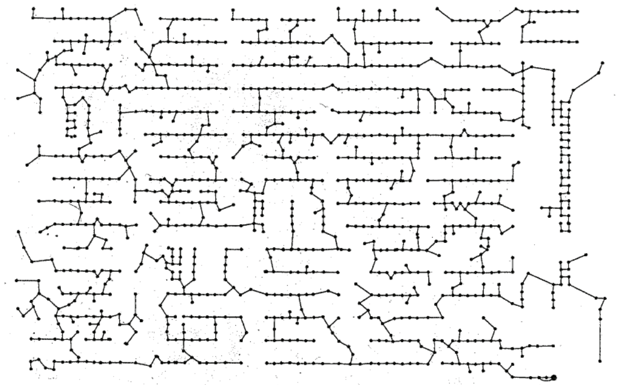
The symmetric TSP

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in E} c_{ij}x_{ij} \\
\text{s.t.} \quad & \sum_{j:(i,j)\in E} x_{ij} = 2, \quad i = 1,\dots,n \\
& \sum_{(i,j)\in\delta(S)} x_{ij} \geq 2, \quad \emptyset \subset S \subset V \\
& x_{ij} \in \{0,1\}, \qquad (i,j) \in E
\end{aligned}
$$

A Sample TSP



Lower bound: 1-tree bound



Lower bound: 51488

A solution approach

1. The number of cut-set elimination constraints is huge ($2^{|V|}$) and even though we can remove half of those due to symmetry there are still exponentially many.
2. Therefore, in the relaxed version we remove the integrality constraints and the exponentially many cut-set elimination constraints.
3. Even though there are "only" $O(n^2)$ variables we need to be able to solve the LP relaxation efficiently.

Challenges

For the cutting plane approach to work we need to

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

Problem 2: Solving the LP "intelligently"

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

$$\begin{aligned}
\text{LP}(E') \quad \min \quad & \sum_{(i,j)\in E'} c_{ij}x_{ij} \\
\text{s.t.} \quad & \sum_{j:(i,j)\in E'} x_{ij} = 2, \quad i = 1,\ldots,n \\
& \sum_{(i,j)\in\delta(S)} x_{ij} \geq 2, \quad \emptyset \subset S \subset V \\
& 0 \leq x_{ij} \leq 1, \qquad (i,j) \in E'
\end{aligned}$$

Generating a feasible solution

- An **optimal** solution x' for $LP(E')$ 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.

Checking optimality

- Idea: Look at the dual problem.
- Let y_i be the dual variable for the i'th constraint

$$\sum_{j:(i,j)\in E'} x_{ij} = 2,$$

- Let Y_S be the dual variable for for "S'th" constraint

$$\sum_{(i,j)\in\delta(S)} x_{ij} \geq 2,$$

Dual of the STSP I

$$\begin{array}{llll} \text{DP}(E') & \max & \sum_{i\in V} 2y_i + \sum_{S\subset V} 2Y_S & \\ & \text{s.t.} & y_i + y_j + \sum_{\text{"}(i,j)\in\delta(S)\text{"}} Y_S \leq c_{ij}, & (i,j) \in E' \\ & & Y_S \geq 0, & S \subset V \end{array}$$

Dual of the STSP II

- 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: Separation algorithm

- Start of by removing the cut-set elimination constraints. Then we get:

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e \\ \text{s.t.} \quad & \sum_{j:(i,j) \in E} x_{ij} = 2, \quad i = 1, \ldots, n \\ & 0 \leq x_e \leq 1, \qquad e \in E \end{aligned}$$

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

Structure of a solution

- If the solution is split into several “independent” components then the node set S of each component violates a cut-set 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 cut-set elimination constraints that are violated. How do we detect those?

The separation algorithm: using max-flow

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

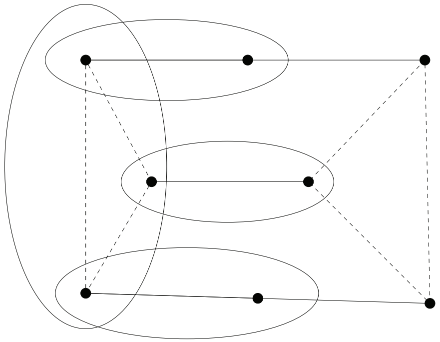
- Max-flow works on directed graphs – this is a non-directed graph.
- We need a sink and a source to run the max-flow algorithm.
- Max-flow needs capacities on the edges.

Is that enough?

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

Fractional solution

Consider the following part of a graph (dash edges have a flow of 0.5, the remaining ones a flow of 1).



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:

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

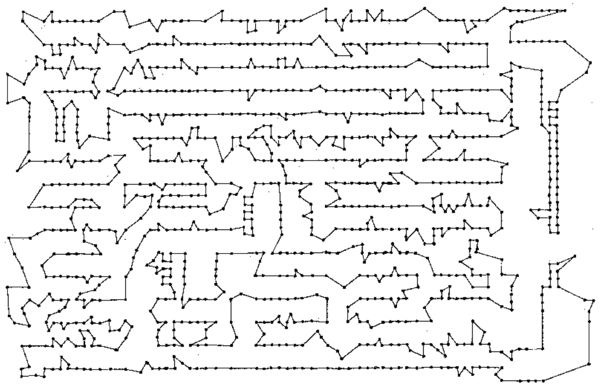
Cuts for the TSP I

- It is known that the cut-set elimination cuts and comb inequalities are facet-defining (Grötschel and Padberg 1979).
- These cuts are generally still not enough but there are more cuts we could add:
  - Blossom (Padberg and Rao 1982)
  - Path inequalities (Naddef and Rinaldi 1998)
  - 2-handled clique tree (Padberg and Rinaldi 1991)
  - Star inequalities (Fleischmann 1988)

Cuts for the TSP II

- 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.

An upper bound: Chained Lin Kernighan

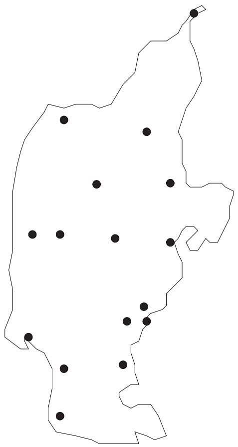


Upper bound: 56892

Effective methods give a small gap

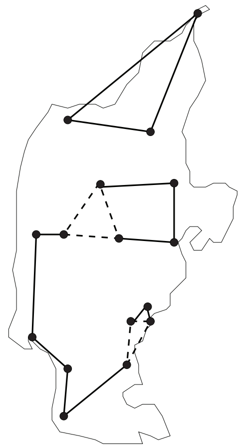
- THEREFORE branch after having added all the "simple" valid inequalities.
- **Example:**
  - Upper bound:
    - 72337: Nearest insertion
    - 65980: Farthest insertion
    - 56892: Chained Lin-Kernigan heuristic
  - Lower bound:
    - 56785: LP-relaxation, cut-set and simple comb-ineq
  - Gap: 0.2% !!

Example: Tour of Jutland I



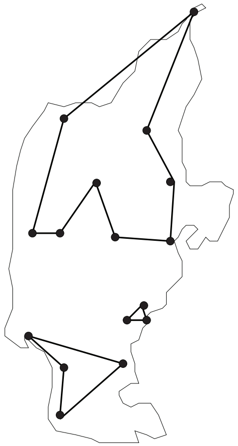
- 16 cities – $15 + 14 + 13 + \ldots + 2 + 1 = \frac{16 \times 15}{2} = 120$ variables.
- Let us keep the constraint that $\sum_j x_{ij} = 2, \; i = 1, \ldots, N.$
- Relax integrality constraints on variables to $0 \leq x_{ij} \leq 1$

Example: Tour of Jutland II



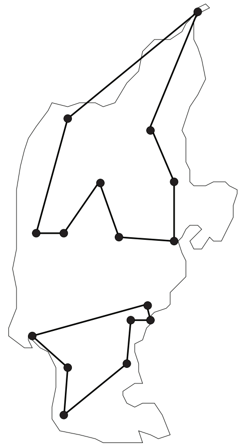
- Objective value: 920
- Forbid the subtour Skagen-Thisted-Aalborg and resolve

Example: Tour of Jutland III



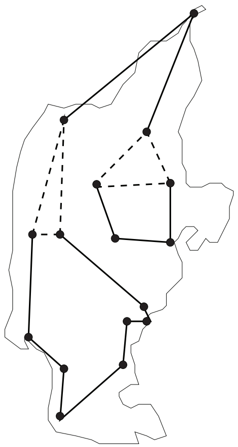
- Objective value: 960
- Forbid the subtour Fredericia-Kolding-Vejle and resolve

Example: Tour of Jutland IV



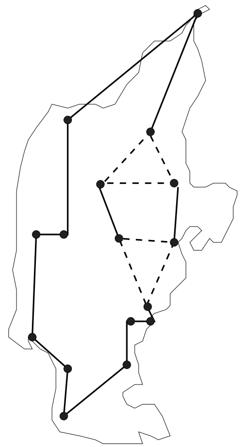
- Objective value: 982
- Forbid the subtour Kolding-Fredericia-Vejle-Esbjerg-Aabenraa-Tønder-Ribe and resolve

Example: Tour of Jutland V



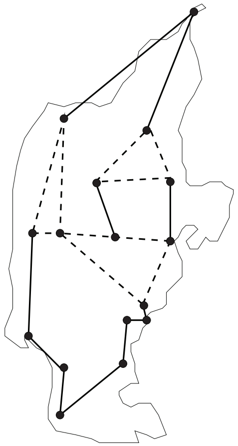
- Objective value: 992.5
- Identify a comb inequality: Handle being Thisted, Ringkøbing and Herning; teeth being (Thisted, Skagen), (Ringkøbing, Esbjerg) and (Herning, Vejle).

Example: Tour of Jutland VI



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

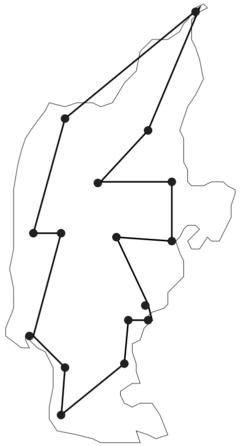
Example: Tour of Jutland VII



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

Department of Management Engineering / Operations Research

Example: Tour of Jutland VIII



- Objective value: 996
- Integer solution!!!