Modelling and Optimization

INF170

#12:Travelling Salesman Problem

AHMAD HEMMATI

Optimization Group
Dept. of Informatics
University of Bergen

Fall Semester
2018

Agenda

- Travelling Salesman Problem (TSP)
- Models
  - Symmetric
  - Assignment + tight subtours (DFJ)
  - Assignment + loose subtours (MTZ)
  - Flow (Svestka)
  - Steps (Dantzig)

Travelling Salesman Problem (TSP)

- A saleswoman located in Indianapolis wants to visit all 48 state capitals of the continental United States to sell her wares
- What is shortest way of visiting all the capitals and then returning to Indianapolis?

TRAVELLING SALESMAN PROBLEM (TSP)

- A saleswoman located in Indianapolis wants to visit all 48 state capitals of the continental United States to sell her wares
- What is shortest way of visiting all the capitals and then returning to Indianapolis?

Travelling Salesman Problem (TSP)

- Proctor and Gamble Contest
  - 1962

Travelling Salesman Problem (TSP)

- One of the most studied problems in the area of optimization.
- "travelling salesman problem" ~ 700k hits on Google
- The TSP has several applications even in its purest formulation, such as
  - ✓ Planning
  - ✓ Logistics
  - ✓ Manufacture of microchips
  - ✓ DNA sequencing.
  - ✓ …
- In these applications, the concept city represents, for example, customers, soldering points, or DNA fragments, and the concept distance represents travelling times or cost, or a similarity measure between DNA fragments.
- In many applications, additional constraints such as limited resources or time windows may be imposed.

Travelling Salesman Problem (TSP)

- 1971 *Cook*, 1972 *Karp*, 1973 *Levin*: *computational complexity*.
  - How hard is a problem to solve, as a function of the input data?
- If Problem 1 converts easily to Problem 2 & vice versa, then Problem 1 is as easy or hard as Problem 2.
- Some problems are
  - easy: shortest path, min spanning tree, assignment.
  - hard: TSP – because no "good" algorithms have been found.
- Many hard problems, such as job shop scheduling, can be converted algebraically to & from the TSP.
  - A good TSP algorithm will be good for other hard problems.
- The TSP is easy to state, takes no math background to understand, and no great talent to find feasible solutions.

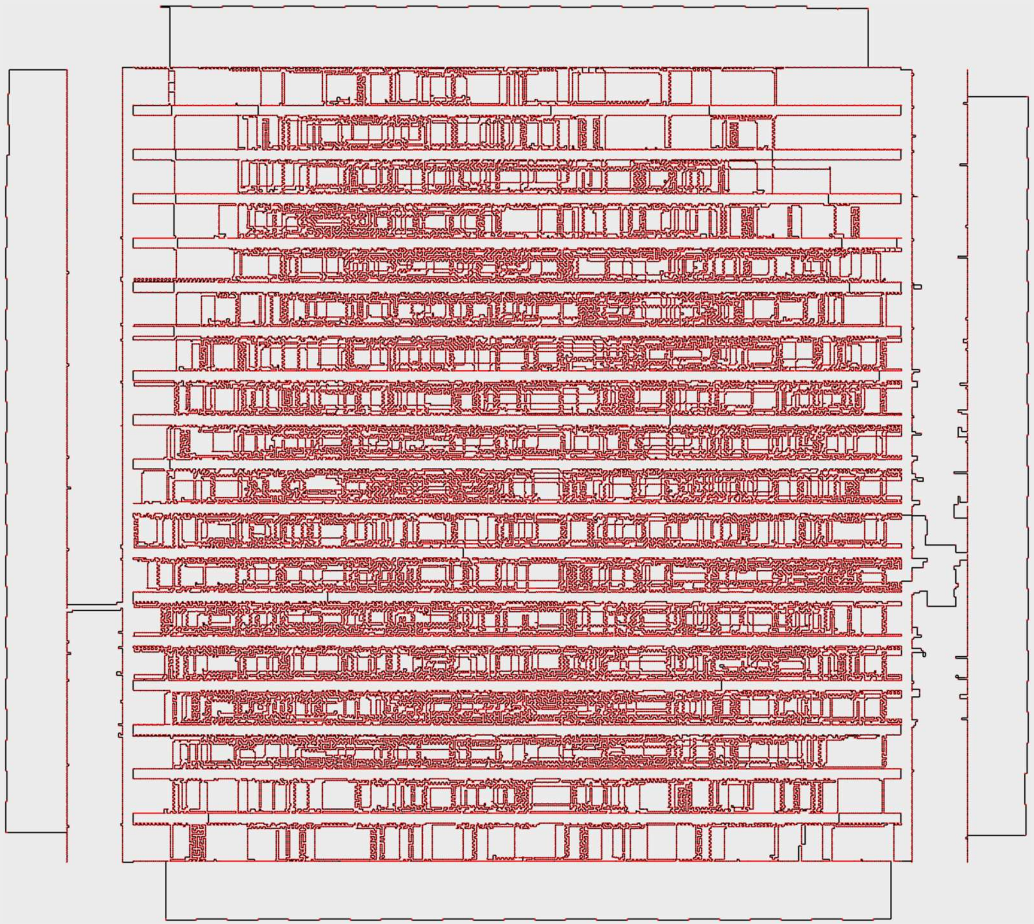
Travelling Salesman Problem (TSP)

- The TSP is a NP-hard problem
  - No polynomial time algorithm for solving it to optimality.
  - Exponential in the number of cities.
  - $(N-1)!$ different tours.
  - $\frac{1}{2}(N-1)!$ if we do not care about the direction of the tour

- Large problems solved to optimality!!

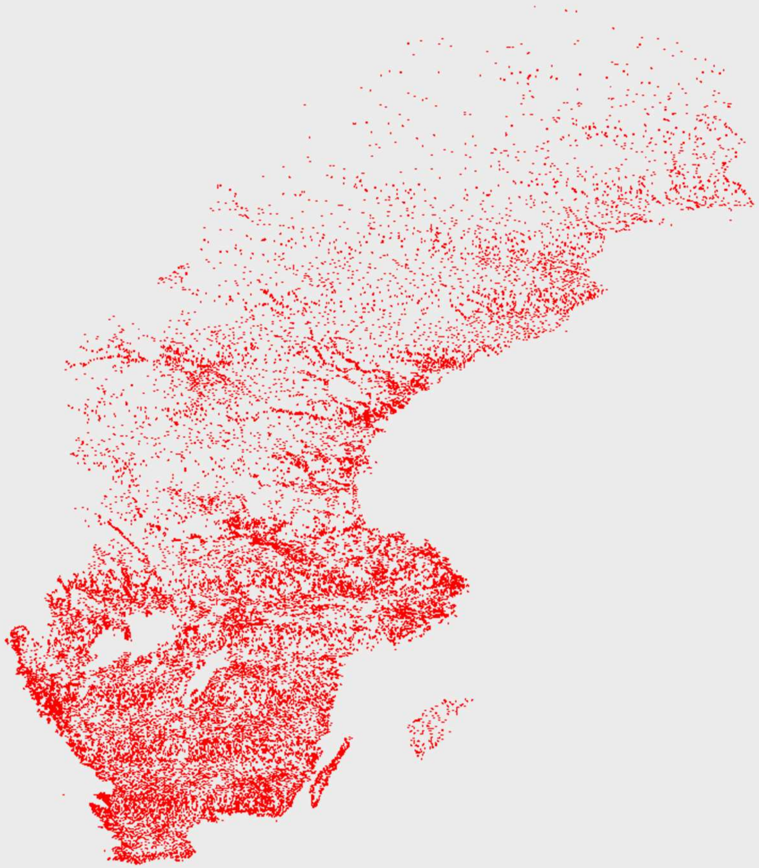
Travelling Salesman Problem (TSP)

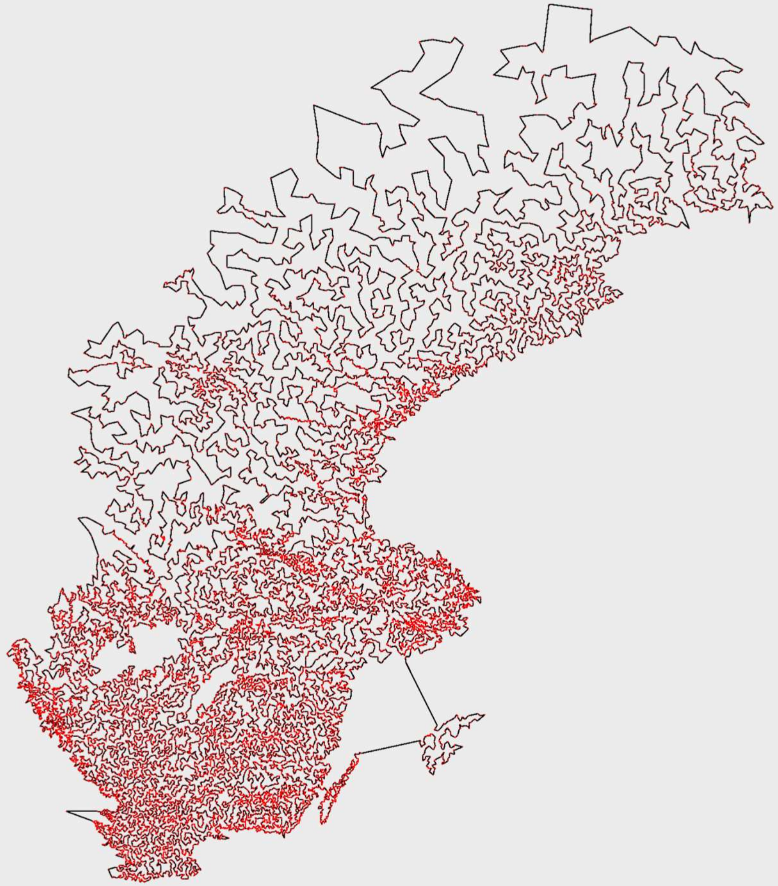
- VLSI problem (85,900 nodes). Solved 2004, first studied 1991.



Travelling Salesman Problem (TSP)

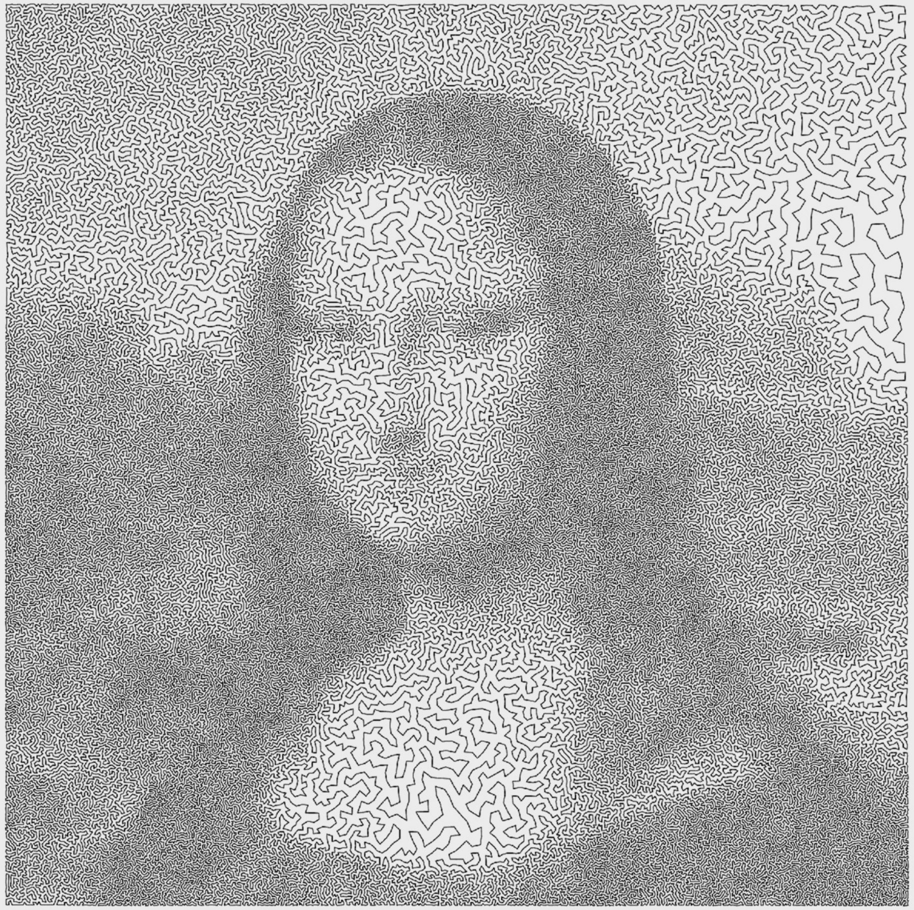
- Shortest tour between all the cities in Sweden (24,978 cities) found in 2001. Length ~72,500 km





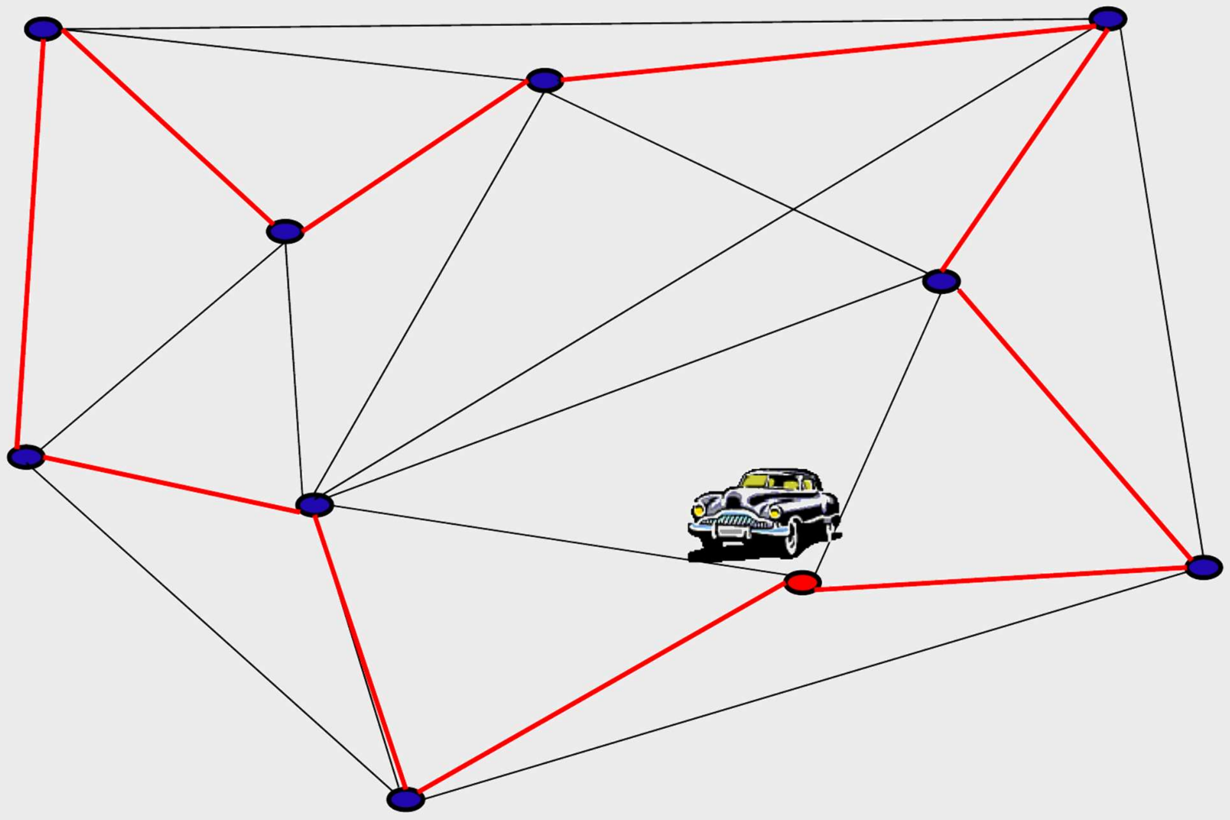
Travelling Salesman Problem (TSP)

- Mona Lisa TSP Challenge! (100,000 nodes)
- $1000 Prize Offered!

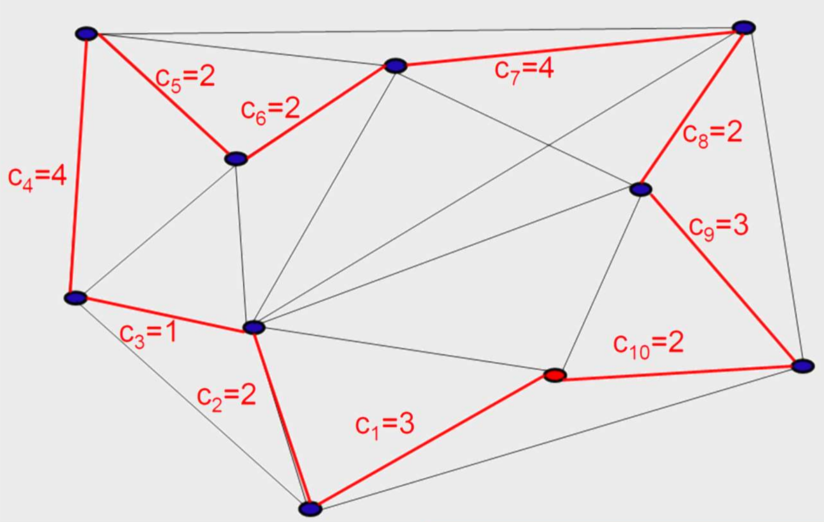


TSP – Formulation

- 10 cities



- K : Set of all links in the network
- c_k : Distance in link k, $\forall\, k \in K$



Total distance = 25

Decision variables

$$x_k = \begin{cases} 1 & \text{if link } k \text{ is included in the TSP tour} \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, k \in K$$

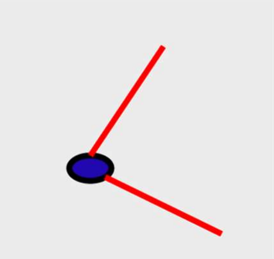
Objective function

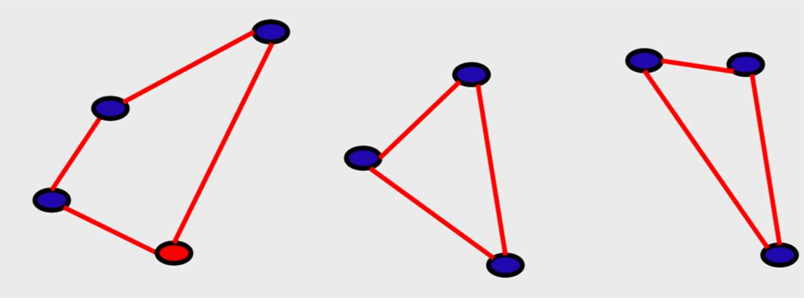
$$\min z = \sum_{k \in K} c_k x_k$$

TSP – Formulation

- A travelling salesman has to visit n cities. Each city is to be visited exactly once…
- N : Set of cities
- L_j : Set of all links connecting to city j, $\forall\, j \in N$

$$\sum_{k \in L_j} x_k = 2 \quad \forall\, j \in N$$





We need to avoid these "sub-tours"

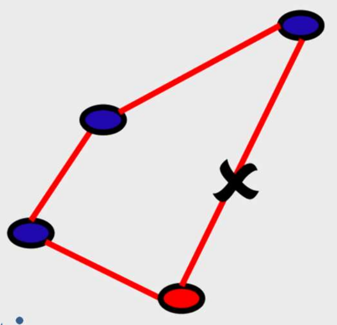
- $B(S)$: Set of all links between the cities in the subset S

$$\sum_{k \in B(S)} x_k = |S| - 1 \quad \forall\, S \subset N$$

|S| is the cardinality of set S (number of cities)

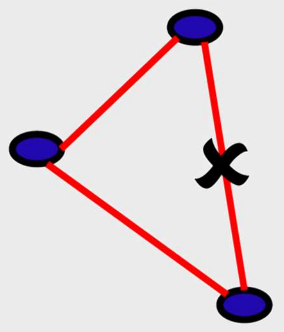
Constraint formulated for all subsets S containing more than 1 and less than n cities

S: subset of these 4 cities



$$\sum_{k \in B(S)} x_k = 4 - 1 \ = 3$$

S: subset of these 3 cities



$$\sum_{k \in B(S)} x_k = 3 - 1 \ = 2$$

TSP – Formulation (Symmetric TSP)

$$min\, z = \sum_{k \in K} c_k x_k$$

s.t.

$$\sum_{k \in L_j} x_k = 2 \quad \forall\, j \in N$$

$$\sum_{k \in B(S)} x_k = |S| - 1 \quad \forall\, S \subset N, 2 \leq |S| \leq |N| - 2$$

$$x_k \in \{0,1\} \qquad \forall\, k \in K$$

TSP – Formulation (Symmetric TSP)

$$x_{ij} = \begin{cases} 1 & \text{if link from city } i \text{ to city } j \text{ is included in the TSP tour} \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, i,j \in N$$

Defined only for $i < j$. Half as many variables as the asymmetric!

$$min\, z = \sum_{i=1}^{n} \sum_{j=i+1}^{n} c_{ij} x_{ij}$$

s.t.

$$\sum_{i=1}^{j-1} x_{ij} + \sum_{i=j+1}^{n} x_{ji} = 2 \quad \forall\, j \in N$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} = |S| - 1 \quad \forall\, S \subset N, 2 \le |S| \le |N| - 2$$

$$x_{ij} \in \{0,1\} \qquad \forall\, i,j \in N$$

TSP – Formulation (Symmetric TSP)

$$min\, z = \sum_{i=1}^{n} \sum_{j=i+1}^{n} c_{ij} x_{ij}$$

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

The variables into city 5 are: $x_{15}, x_{25}, x_{35}, x_{45}, x_{65}, x_{75}, x_{85}, x_{95}$.

The variables out of city 5 are: $x_{51}, x_{52}, x_{53}, x_{54}, x_{56}, x_{57}, x_{58}, x_{59}$.

Since costs are symmetric, $c_{ij} = c_{ji}$, let's drop half the variables.
For x_{ij}, require $i < j$. Allow only the variables going out.
We need only variables $x_{15}, x_{25}, x_{35}, x_{45}, x_{56}, x_{57}, x_{58}, x_{59}$.
The meaning is not "Go in" or "come out", but "use this arc".

The summation makes sure that we cover only the variables we need.
$x_{15} + x_{25} + x_{35} + x_{45} + x_{56} + x_{57} + x_{58} + x_{59} = 2$.

TSP – Formulation (Symmetric TSP)

```
set S ordered;
param n := card {S};
set SS := 0 .. (2**n - 1);
set POW{k in SS}:= {i in S: (k div 2**(ord(i)-1)) mod 2 = 1};
set LINKS := {i in S, j in S: ord(i) < ord(j)};
param cost {LINKS} >= 0;

var X {LINKS} binary;

minimize TotCost:
sum {(i,j) in LINKS} cost[i,j] * X[i,j];

subj to Tour {i in S}:
sum {(i,j) in LINKS} X[i,j] + sum {(j,i) in LINKS} X[j,i]=2;

subj to SubtourElim {k in SS diff {0,2**n-1}}:
sum{i in POW[k],j in S diff POW[k]:(i,j) in LINKS}X[i,j]+
sum{i in POW[k],j in S diff POW[k]:(j,i) in LINKS}X[j,i]>= 2;
```

Travelling Salesman Problem (TSP)

- For a large TSP, we may need many subtour breaking constraints.
- In the worst case, we may need 2^n subtour breaking constraints
- From binomial theorem:

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}$$

$$2^n = (1+1)^n = \sum_{k=0}^{n} \binom{n}{k} 1^k 1^{n-k} = \sum_{k=0}^{n} \binom{n}{k}$$

The Dantzig, Fulkerson & Johnson (DFJ) model

$$x_{ij} = \begin{cases} 1 & \text{if link from city } i \text{ to city } j \text{ is included in the TSP tour} \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, i,j \in N$$

$$min\, z = \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}$$

s.t.

$$\sum_{j \in N} x_{ij} = 1 \quad \forall\, i \in N$$

$$\sum_{i \in N} x_{ij} = 1 \quad \forall\, j \in N$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} = |S| - 1 \quad \forall\, S \subset N, 2 \leq |S| \leq |N| - 2$$

$$x_{ij} \in \{0,1\} \quad \forall\, i,j \in N$$

TSP – Formulation (Asymmetric TSP)

The Miller, Tucker & Zemlin (MTZ) model

$$x_{ij} = \begin{cases} 1 & \text{if link from city } i \text{ to city } j \text{ is included in the TSP tour} \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i,j \in N$$

u_i = number of cities visited at city *i*.

$$min\, z = \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}$$

s.t.

$$\sum_{j \in N} x_{ij} = 1 \quad \forall\, i \in N$$

$$\sum_{i \in N} x_{ij} = 1 \quad \forall\, j \in N$$

$$u_i + 1 \le u_j + |N|\,(1 - x_{ij}), \quad \text{for } i\,, j \ge 2 \in N\,, i \ne j$$

$$x_{ij} \in \{0,1\} \ \forall\, i,j \in N, \quad u_i \ge 0 \quad \forall i \in N$$

TSP – Formulation (Asymmetric TSP)

The Miller, Tucker & Zemlin (MTZ) model

```
param N integer > 2; # Number of nodes
set Nodes ordered := {1..N};
set Arcs := {i in Nodes, j in Nodes: i <> j};
param C{(i,j) in Arcs};
var x {(i,j) in Arcs} binary;
var u {Nodes} >= 0;

minimize Tourlength: sum {(i,j) in Arcs} C[i,j]*x[i,j];
subject to Degree1 {i in Nodes}: sum{(i,j) in Arcs} x[i,j]=1;
subject to Degree2 {i in Nodes}: sum{(j,i) in Arcs} x[j,i]=1;
subject to NoSubtour1 {(i,j) in Arcs: i<>j and i>=2 and
j>=2}: u[i] - u[j] + N*x[i,j] <= N - 1;
# These rows tighten the model:
subject to NoSubtour2 {i in Nodes: i >= 2}:
u[i] <= N - 1 - (N - 2)*x[1,i];
subject to NoSubtour3 {i in Nodes: i >= 2}:
u[i] >= 1  +  (N - 2)*x[i,1];
```

TSP – Formulation (Asymmetric TSP)

The gain (Svestka) model

Parameter: f = gain in flow from city i to city j

Variables: $x_{ij} = \begin{cases} 1 & \text{if link from city } i \text{ to city } j \text{ is included in the TSP tour} \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i,j \in N$

y_{ij} = flow from city i city j.

$$min\, z = \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}$$

s.t.

$$\sum_{j \in N, j \neq i} y_{ji} \geq 1 \quad \forall\, i \geq 2 \in N$$

$$\sum_{j \in N, j \neq i} y_{ij} - \sum_{j \in N, j \neq i} y_{ji} = f \quad \forall\, i \in N$$

$$\sum_{i \in N} \sum_{j \in N} x_{ij} \leq |N|$$

$$y_{ij} \leq (1 + |N| f)\, x_{ij}, \quad \text{for } i, j \in N$$

$$x_{ij} \in \{0,1\} \quad \forall\, i,j \in N,\; y_{ij} \geq 0\; \forall\, i,j \in N$$

TSP – Formulation (Asymmetric TSP)

The gain (Svestka) model

```
param N integer > 2; # Number of nodes
set Nodes ordered := {1..N};
set Arcs := {i in Nodes, j in Nodes: i <> j};
param length{(i,j) in Arcs};
var x {(i,j) in Arcs} binary;
var y {(i,j) in Arcs} >= 0;
param f := 0.01;

minimize Tourlength: sum {(i,j) in Arcs} length[i,j] * x[i,j];
subject to Demandy {i in Nodes: i >= 2}: sum {(i,j) in Arcs} y[j,i] >= 1;
subject to Flowy {i in Nodes: i >= 2}:
           sum {(i,j) in Arcs} y[i,j] - sum {(i,j) in Arcs} y[j,i] = f;
subject to Totalx: sum {(i,j) in Arcs} x[i,j] <= N;
subject to Arcgain {(i,j) in Arcs}: y[i,j] <= (1 + N*f)*x[i,j];
# These rows tighten the model:
subject to Demandx {i in Nodes: i >= 2}: sum {(i,j) in Arcs} x[j,i] = 1;
subject to Flowx {i in Nodes: i >= 2}:
           sum {(i,j) in Arcs} x[i,j] = sum {(i,j) in Arcs} x[j,i];
subject to Startx: sum {i in Nodes: i >= 2} x[1,i] = 1;
subject to Starty: sum {i in Nodes: i >= 2} y[1,i] = 1;
```

TSP – Formulation (Asymmetric TSP)

The steps (Dantzig) model

$$x_{ijt} = \begin{cases} 1 & \text{if we drive from city } i \text{ to city } j \text{ at step } t \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, i,j,t \in N$$

$$min\, z = \sum_{i \in N} \sum_{j \in N} \sum_{t \in N} c_{ij} x_{ijt}$$

s.t.

$$\sum_{i \in N} x_{ijt} - \sum_{k \in N} x_{j,k,t+1} = 0 \quad \forall\, j,t \in N$$

$$\sum_{j \in N} \sum_{t \in N} x_{ijt} = 1 \quad \forall\, i \in N$$

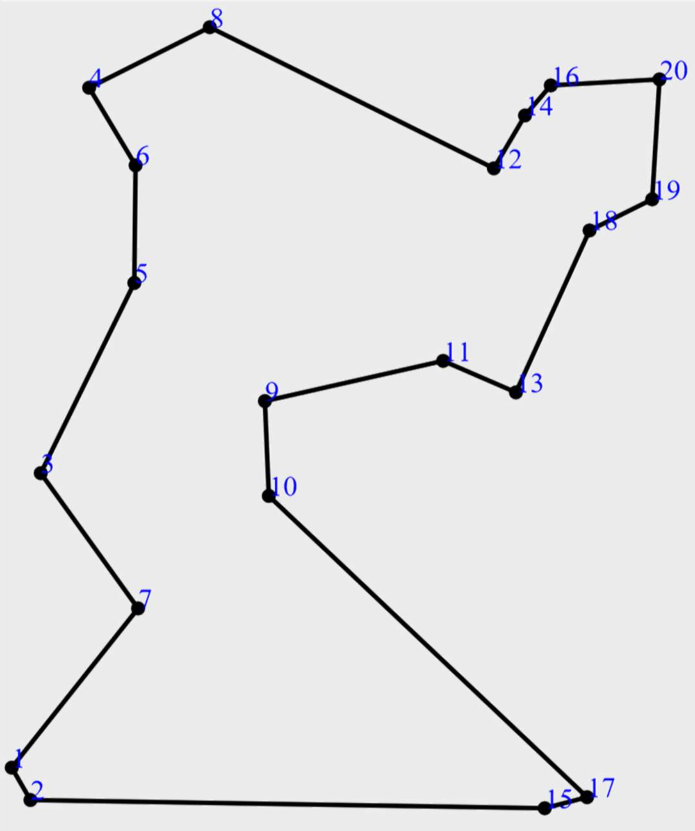
$$x_{ijt} \in \{0,1\} \qquad \forall\, i,j,t \in N$$

TSP – Formulation (Asymmetric TSP)

The steps (Dantzig) model

```
param N integer > 2; # Number of nodes
set Nodes ordered := {1..N};
set Steps := {i in Nodes, j in Nodes, k in Nodes: i <> j};
param length {i in Nodes, j in Nodes: i <> j};

var z {(i,j,k) in Steps} binary;

minimize Tourlength:
sum {(i,j,k) in Steps: i <> j} length[i,j] * z[i,j,k];

subject to Demand {i in Nodes}: sum {(i,j,k) in Steps} z[i,j,k] = 1;
subject to End {i in Nodes}:
sum {j in Nodes: j <> i} z[j,i,N] = sum {k in Nodes: k <> i} z[i,k,1];

subject to Step {t in Nodes, j in Nodes: t <= N - 1}:
sum {i in Nodes: i <> j} z[i,j,t] = sum {k in Nodes: k <> j} z[j,k,t+1];
```

20 nodes

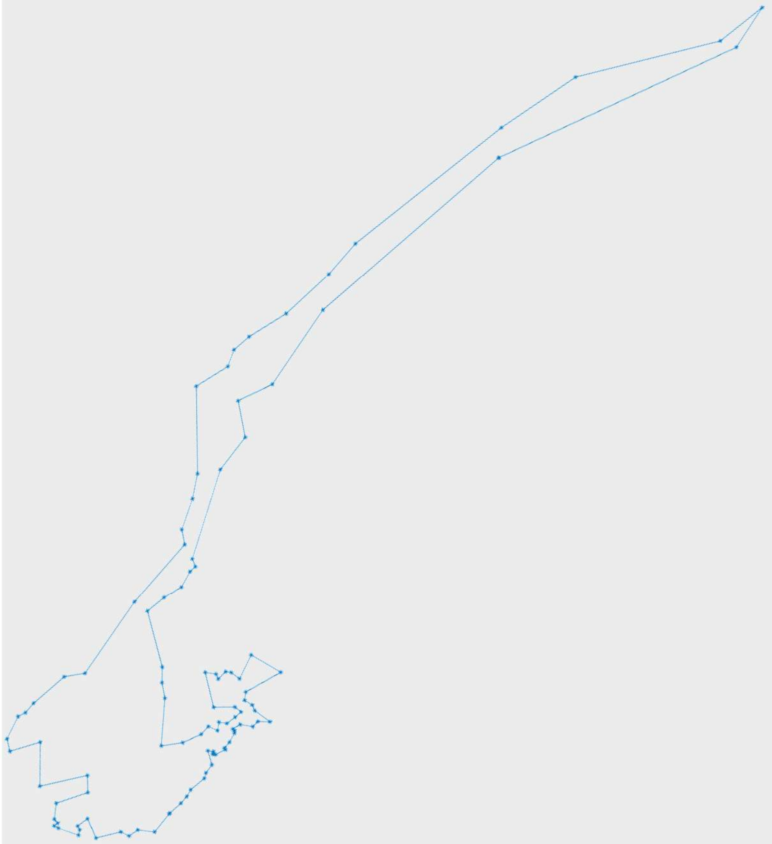


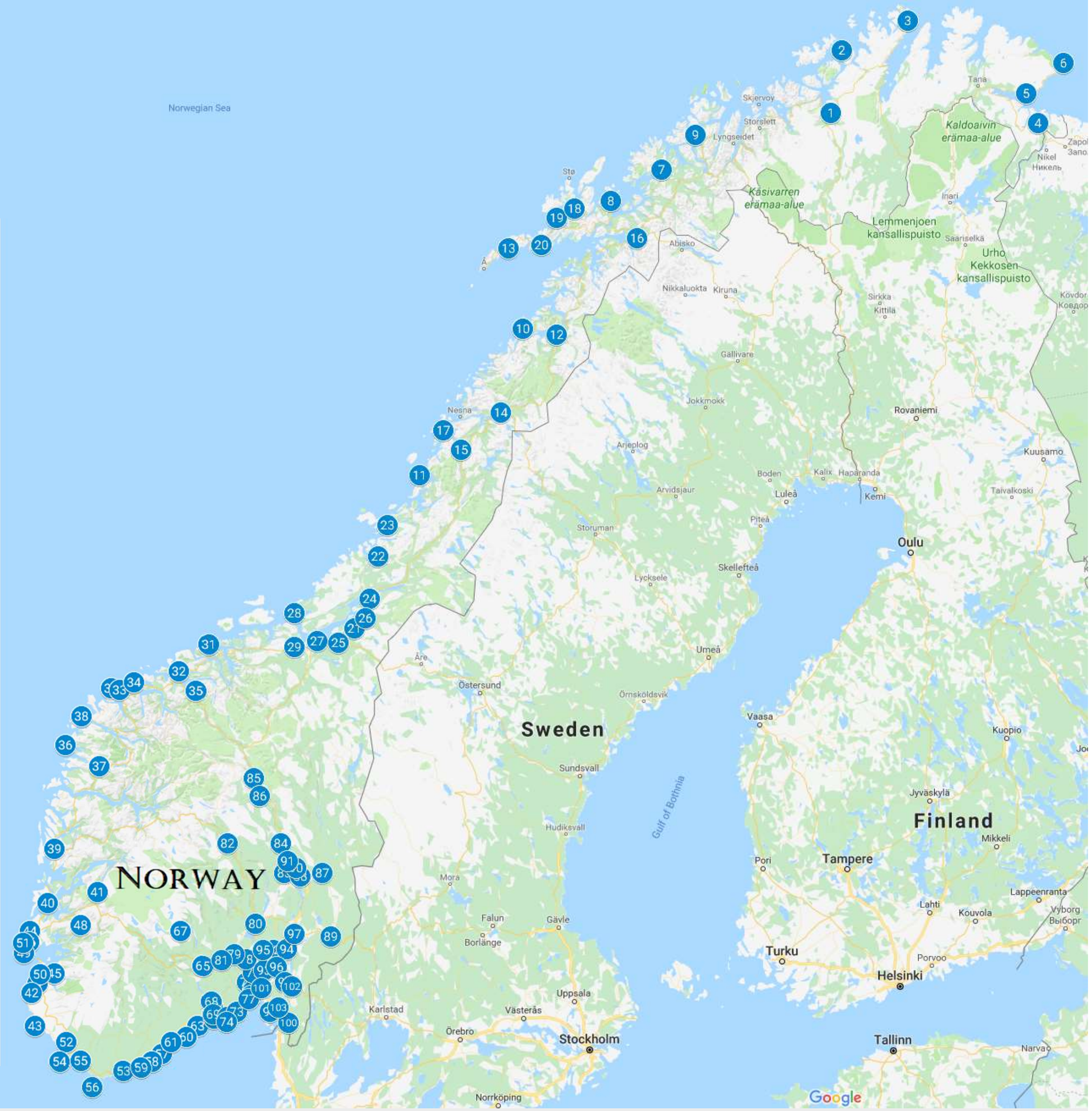
TSP – Implementation

	MTZ	gain(Svestka)	Steps(Dantzig)
_nvars	400	760	7600
_ncons	420	459	420
Time	0.14	0.23	7.4

TSP – Implementation

103 cities in Norway





ASSIGNMENT #7:

- MTZ, Svestka, and Step model
- 48_Caps_US.txt
- display _nvars;
- display _ncons;
- option solver gurobi;

- option gurobi_options "timelim=3600 outlev=1";
- param length {(i,j) in Arcs} := sqrt((xcoord[i] - xcoord[j])^2 + (ycoord[i] - ycoord[j])^2);

Lecture #13:

Heuristics for

Travelling Salesman Problem