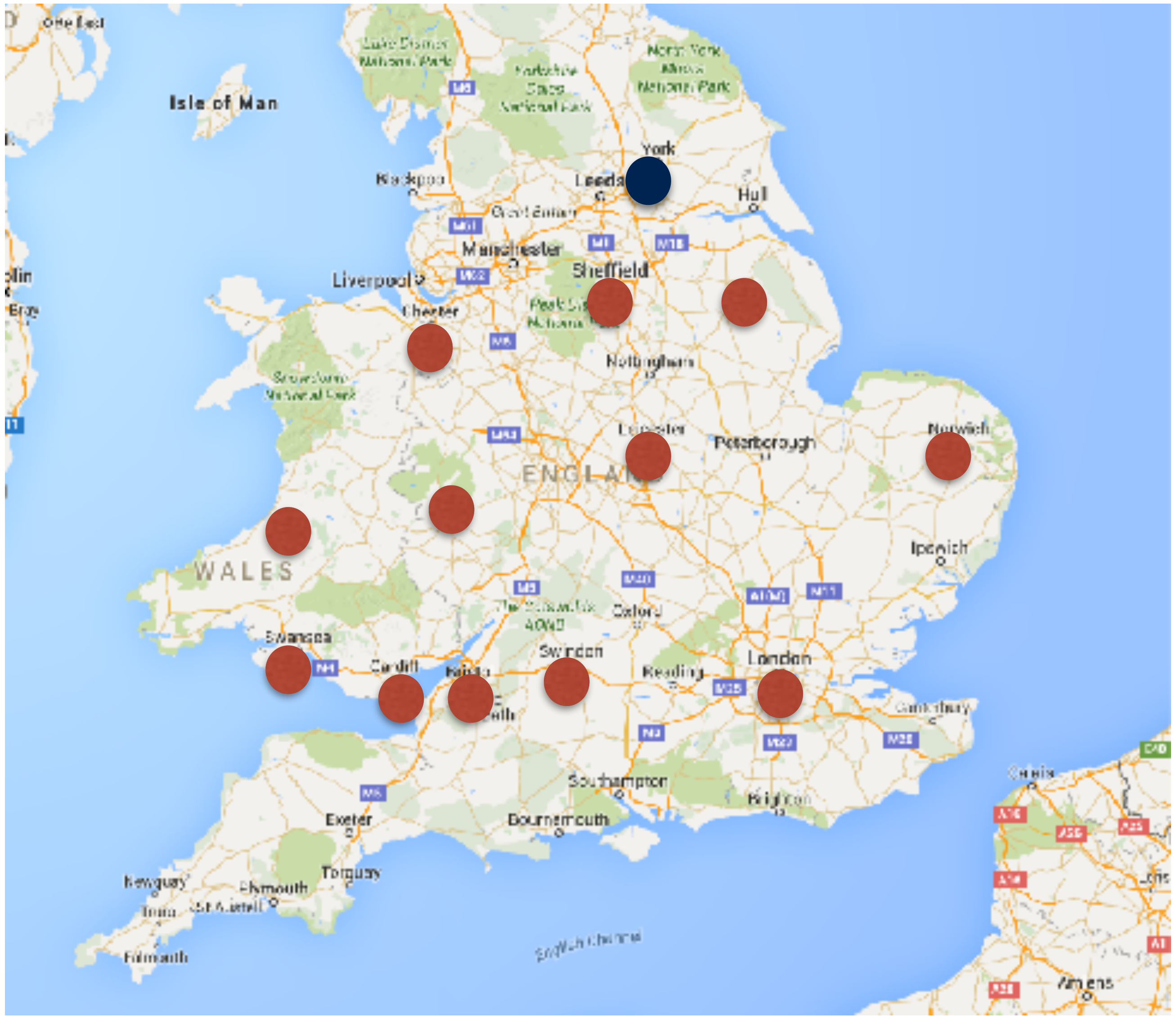
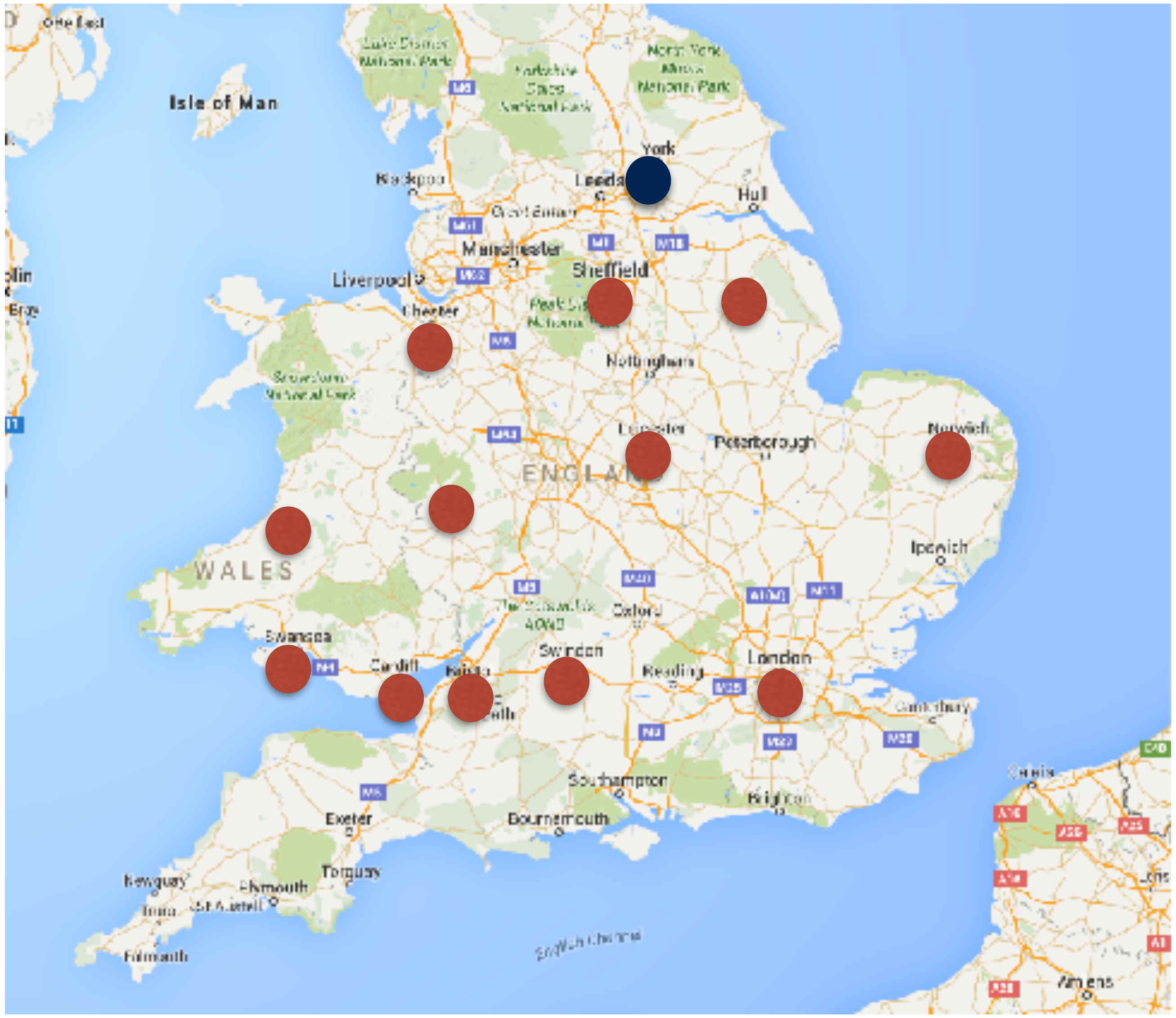


Traveling Salesman Problem Formulation

Leandro L. Minku

Examples of Optimisation Problems

- Traveling Salesman Problem:
  - A salesman must travel passing through *N* cities.
  - Each city must be visited once and only once.
  - He/she must finish where he/she was at first.
  - The path between each pair of cities has a distance (or cost).



Problem: find a sequence of cities that minimises traveling distance (or cost), where each city appears once and only once.

Traveling Salesman Problem Formulation

- Design variables represent a candidate solution.
  - Sequence **x** of *N* cities to be visited, where cities are in *C*.
  - *C* is a set containing the *N* cities to be visited.
  - The search space is all possible sequences of cities.

- Objective function defines the cost of a solution.
  - Total_distance(**x**) =
    sum of distances between consecutive cities in **x** + distance from last city to the origin.
  - To be minimised.

$\overline{}\ \overline{}\ \overline{}\ \overline{}\ \overline{}$

$x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5$

- [Optional] Solutions must satisfy certain constraints.
  - Each city must appear once and only once in **x** (explicit constraint).
  - Salesman must return to the city of origin (implicit constraint).
  - Only cities in C must appear in **x** (implicit constraint).

Traveling Salesman Problem Formulation

- Design variables represent a candidate solution.
  - The design variable is a sequence **x** of *N* cities, where $x_i \in \{1,\cdots,N\}$, $\forall i \in \{1,\cdots,N\}$.
  - The *N* cities to be visited are represented by values {1,…,*N*}.
  - The search space is all possible sequences of *N* cities, where cities are in {1,…,*N*}.

- Objective function defines the cost of a solution.

$$\text{minimise totalDistance}(\mathbf{x}) = \left(\sum_{i=1}^{N-1} D_{x_i,x_{i+1}}\right) + D_{x_N,x_1}$$

1	3	2	4	5
x_1	x_2	x_3	x_4	x_5

where $D_{j,k}$ is the distance of the path between cities j and k.

- [Optional] Solutions must satisfy certain constraints.
  - Each city must appear once and only once in **x** (explicit constraint).

"For each city i in {1,…,N}",

$$\forall i \in \{1,\cdots,N\}, \left(\sum_{j=1}^{N} 1(x_j = i)\right) = 1$$

$$1(x_j = i) = \begin{cases} 1, \text{ if } x_j = i \\ 0, \text{ if } x_j \neq i \end{cases}$$

Traveling Salesman Problem Formulation

$$\forall i \in \{1, \cdots, N\}, \left(\sum_{j=1}^{N} 1(x_j = i) \right) = 1 \qquad 1(x_j = i) = \begin{cases} 1, \text{ if } & x_j = i \\ 0, \text{ if } & x_j \neq i \end{cases}$$

$$\forall i \in \{1, \cdots, N\}, h_i(\mathbf{x}) = \left(\sum_{j=1}^{N} 1(x_j = i) \right) - 1 = 0$$

$\{1,2,3,4,5\}$

$\uparrow$

i

4	2	1	3	3
x_1	x_2	x_3	x_4	x_5

$\uparrow$

j

Sum_1: 0 +0 +1 +0 +0 = 1

Sum_2: 0 +1 +0 +0 +0 = 1

Sum_3: 0 +0 +0 +1 +1 = 2

Traveling Salesman Problem Formulation

- Design variables represent a candidate solution.
  - The design variable is a sequence $\mathbf{x}$ of N cities, where $x_i \in \{1, \cdots, N\}$, $\forall i \in \{1, \cdots, N\}$.
  - The N cities to be visited are represented by values $\{1,\ldots,N\}$.
  - The search space is all possible sequences of N cities, where cities are in $\{1,\ldots,N\}$.

- Objective function defines the cost of a solution.

$$\text{minimise totalDistance}(\mathbf{x}) = \left(\sum_{i=1}^{N-1} D_{x_i, x_{i+1}} \right) + D_{x_N, x_1}$$

where $D_{j,k}$ is the distance of the path between cities j and k.

- [Optional] Solutions must satisfy certain constraints.

For each city i, $h_i(\mathbf{x}) = 0$

Summary

- Traveling salesman problem formulation, including constraints.

Next

- How to deal with constraints?