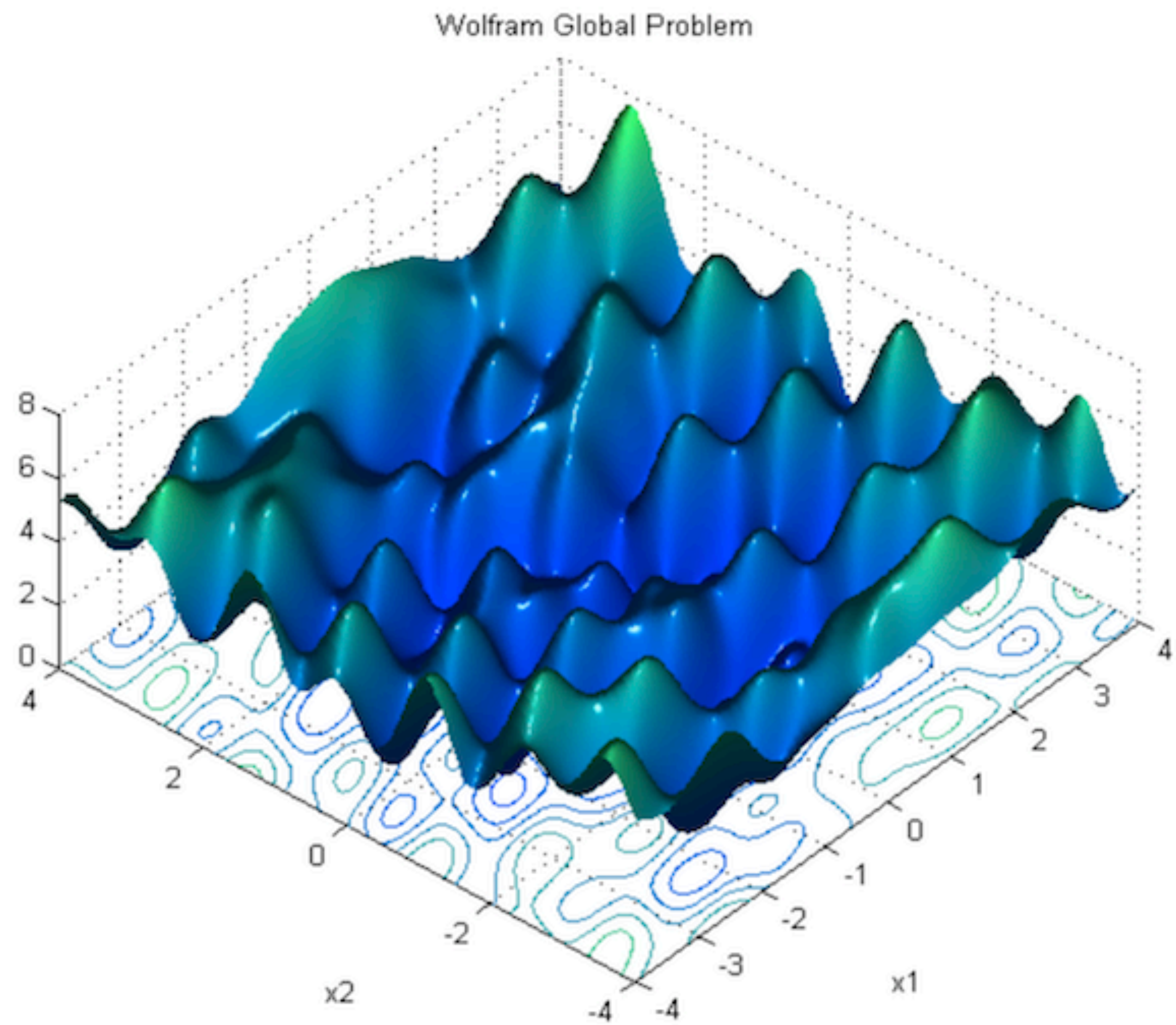


Optimisation Problem Formulation

Leandro L. Minku

Optimisation Problems

- Optimisation problems: to find a solution that minimises/maximises one or more pre-defined objective functions.
- Maximisation / minimisation problems.
- What constitutes a solution depends on the problem in hands.

Optimisation Problems

Objective function

Design variables

Minimise?
Maximise?

Minimise $\quad f(\mathbf{x})$

Subject to $\quad g_i(\mathbf{x}) \leq 0, \quad i = 1, \cdots, m$

$\qquad\qquad\quad h_j(\mathbf{x}) = 0, \quad j = 1, \cdots, n$

Constraints

Search space: space of all possible $\mathbf{x}$ values.

Multi-Objective Optimisation Problems

Minimise $f_k(\mathbf{x}), \quad k = 1 \cdots, p$

Subject to $g_i(\mathbf{x}) \leq 0, \quad i = 1, \cdots, m$

$h_j(\mathbf{x}) = 0, \quad j = 1, \cdots, n$

Formulating Optimisation Problems

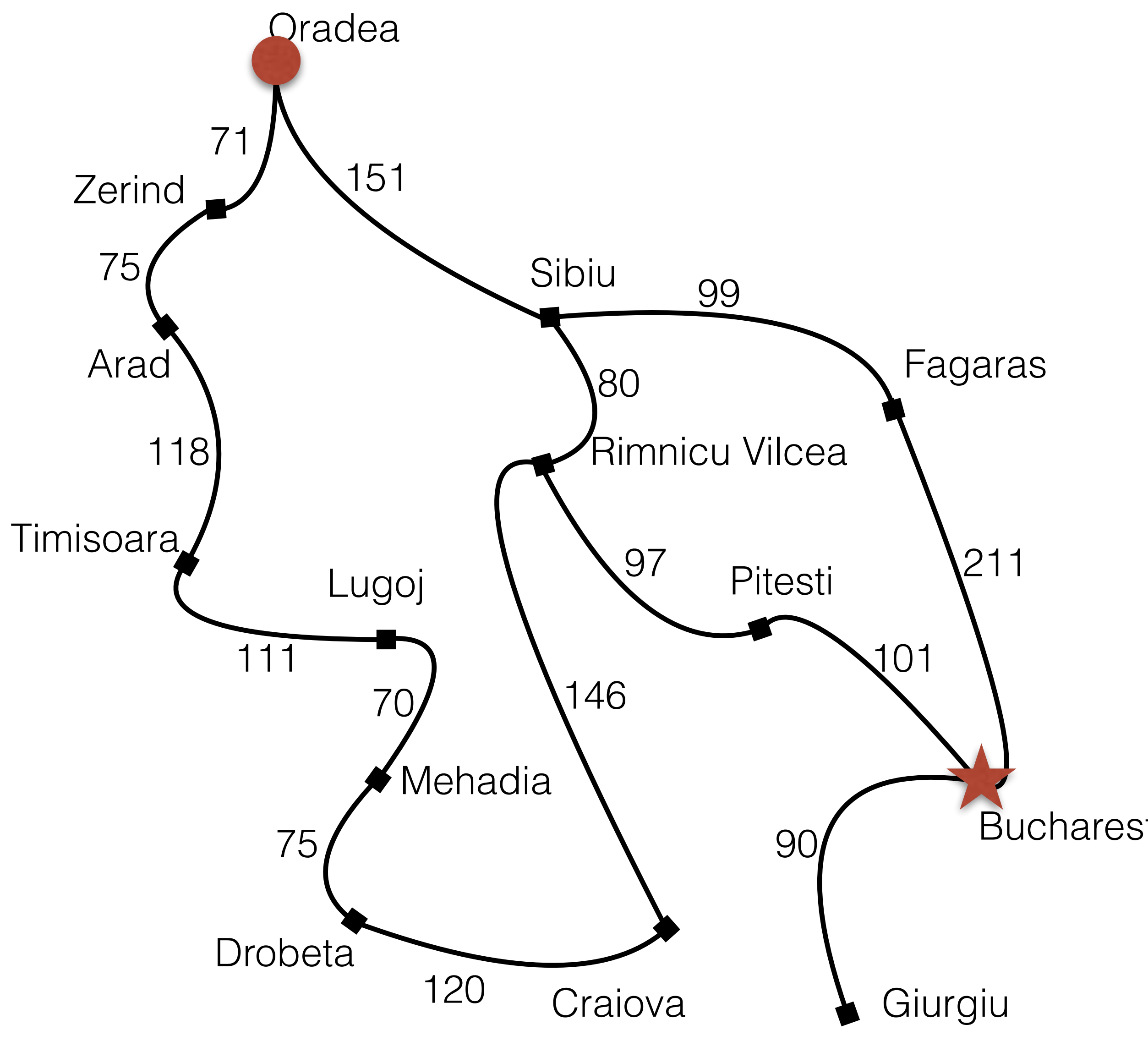
- Design variables represent a candidate solution.
  - Design variables define the search space of candidate solutions.

- Objective function defines the quality (or cost) of a solution.
  - Function to be optimised (maximised or minimised).

- [Optional] Solutions must satisfy certain constraints, which define solution feasibility.
  - Candidate solutions may be feasible or infeasible.

Examples of Optimisation Problems

- Routing problem:
  - Given a motorway map containing N cities.
  - The map shows the distance between connected cities.
  - We have a city of origin and a city of destination.



Problem: find a path from the origin to the destination that minimises the distance travelled, while ensuring that direct paths between non-neighbouring cities are not used.

Routing Problem: Formulation as an Optimisation Problem

- Design variables represent a candidate solution.
  - Sequence $\mathbf{x}$ containing the cities to be visited, where $x_i \in C$, C is the set of available cities, and $\mathbf{x}$ can be of any size.
  - The search space consists of all possible sequences of cities.

Oradea	Sibiu	Fagaras	Bucharest
x_1	x_2	x_3	x_4

Routing Problem: Formulation as an Optimisation Problem

- Objective function defines the quality (or cost) of a solution.

  Minimise the sum of the distances between consecutive cities in $\mathbf{x}$.

Oradea	Sibiu	Fagaras	Bucharest
x_1	x_2	x_3	x_4

Routing Problem: Formulation as an Optimisation Problem

- [Optional] Solutions must satisfy certain constraints, which define solution feasibility.
  - (Inexistent) direct paths between non-neighbouring cities must not be used (explicit constraint).
  - We must start at the city of origin and end at the city of destination (explicit constraint).
  - Only cities in C can be used (implicit constraint).

Oradea	Sibiu	Fagaras	Bucharest
x_1	x_2	x_3	x_4

Routing Problem: Formulation as an Optimisation Problem

- Design variables represent a candidate solution.
  - Sequence $\mathbf{x}$ containing the cities to be visited, where $x_i \in C$, C is the set of available cities, and $\mathbf{x}$ can be of any size.
  - The search space consists of all possible sequences of cities.

- Objective function defines the quality (or cost) of a solution.

  Minimise the sum of the distances between consecutive cities in $\mathbf{x}$.

- [Optional] Solutions must satisfy certain constraints, which define solution feasibility.
  - (Inexistent) direct paths between non-neighbouring cities must not be used (explicit constraint).
  - We must start at the city of origin and end at the city of destination (explicit constraint).
  - Only cities in C can be used (implicit constraint).

Routing Problem: Making it More Formal

- Design variables represent a candidate solution.
  - Sequence $\mathbf{x}$ containing the cities to be visited, where $x_i \in \{1, \cdots, N\}$ and $\mathbf{x}$ can be of any size.
  - The search space consists of all possible sequences of cities.

Oradea	Sibiu	Fagaras	Bucharest
x_1	x_2	x_3	x_4

1	10	2	14
x_1	x_2	x_3	x_4

Routing Problem: Making it More Formal

- Design variables represent a candidate solution.
  - Sequence $\mathbf{x}$ containing the cities to be visited, where $x_i \in \{1, \cdots, N\}$ and $\mathbf{x}$ can be of any size.
  - The search space consists of all possible sequences of cities.

- Objective function defines the quality (or cost) of a solution.

$$\text{Minimise } f(\mathbf{x}) = \sum_{i=1}^{\text{size}(\mathbf{x})-1} D_{x_i, x_{i+1}}$$

where D is a matrix of distances, with each position $D_{i,j}$ containing:

- the distance in km to travel directly between city i and j, or
- -1 if such direct path does not exist.

Note that this objective function doesn't work well when an inexistent direct path is used, but this is ok because constraints will be defined next.

Routing Problem: Making it More Formal

- Design variables represent a candidate solution.
  - Sequence $\mathbf{x}$ containing the cities to be visited, where $x_i \in \{1, \cdots, N\}$ and $\mathbf{x}$ can be of any size.
  - The search space consists of all possible sequences of cities.

- Objective function defines the quality (or cost) of a solution.

$$\text{Minimise } f(\mathbf{x}) = \sum_{i=1}^{\text{size}(\mathbf{x})-1} D_{x_i, x_{i+1}}$$

- [Optional] Solutions must satisfy certain constraints, which define solution feasibility.
  - (Inexistent) direct paths between non-neighbouring cities must not be used (explicit constraint).
  - We must start at the city of origin and end at the city of destination (explicit constraint).
  - Only cities in $\{1, \cdots, N\}$ can be used (implicit constraint).

Routing Problem: Making it More Formal

Minimise $f(\mathbf{x})$

Subject to $g_i(\mathbf{x}) \leq 0, \quad i = 1, \cdots, m$

$h_j(\mathbf{x}) = 0, \quad j = 1, \cdots, n$

- (Inexistent) direct paths between non-neighbouring cities must not be used (explicit constraint).

Assume that we have a matrix D where each position $D_{i,j}$ contains

- the distance to travel directly between city i and j, or
- -1 if such direct path does not exist.

$$h_1 : \mathbf{x} \rightarrow \{0,1\} \qquad h_1(\mathbf{x}) = \begin{cases} 0 & \text{if } D_{x_i, x_{i+1}} \neq -1, \quad \forall i \in \{1, \ldots, \text{size}(\mathbf{x}) - 1\} \\ 1 & \text{otherwise} \end{cases}$$

Routing Problem: Making it More Formal

- Design variables represent a candidate solution.
  - Sequence $\mathbf{x}$ containing the cities to be visited, where $x_i \in \{1, \cdots, N\}$ and $\mathbf{x}$ can be of any size.
  - The search space consists of all possible sequences of cities.

- Objective function defines the quality (or cost) of a solution.

$$\text{Minimise } f(\mathbf{x}) = \sum_{i=1}^{\text{size}(\mathbf{x})-1} D_{x_i, x_{i+1}}$$

- [Optional] Solutions must satisfy certain constraints, which define solution feasibility.
  - $h_1(\mathbf{x}) = 0$
  - We must start at the city of origin and end at the city of destination (explicit constraint).
  - Only cities in $\{1, \cdots, N\}$ can be used (implicit constraint).

Routing Problem: Making it More Formal

$$\text{Minimise } f(\mathbf{x})$$

$$\text{Subject to } g_i(\mathbf{x}) \leq 0, \quad i = 1, \cdots, m$$

$$h_j(\mathbf{x}) = 0, \quad j = 1, \cdots, n$$

- We must start at the city of origin and end at the city of destination (explicit constraint).

$$h_2 : \mathbf{x} \rightarrow \{0,1\}$$

$$h_2(\mathbf{x}) = \begin{cases} 0 & \text{if } x_1 = \text{OriginCity and } x_{\text{size}(\mathbf{x})} = \text{DestinationCity} \\ 1 & \text{otherwise} \end{cases}$$

Routing Problem: Making it More Formal

- Design variables represent a candidate solution.
  - Sequence $\mathbf{x}$ containing the cities to be visited, where $x_i \in \{1, \cdots, N\}$ and $\mathbf{x}$ can be of any size.
  - The search space consists of all possible sequences of cities.

- Objective function defines the quality (or cost) of a solution.

$$\text{Minimise } f(\mathbf{x}) = \sum_{i=1}^{\text{size}(\mathbf{x})-1} D_{x_i, x_{i+1}}$$

- [Optional] Solutions must satisfy certain constraints, which define solution feasibility.
  - $h_1(\mathbf{x}) = 0$
  - $h_2(\mathbf{x}) = 0$
  - Only cities in $\{1, \cdots, N\}$ can be used (implicit constraint).

Routing Problem: Making it More Formal

- Design variables represent a candidate solution.
  - Sequence $\mathbf{x}$ containing the cities to be visited, where $x_i \in \{1, \cdots, N\}$ and $\mathbf{x}$ can be of any size.
  - The search space consists of all possible sequences of cities.

- Objective function defines the quality (or cost) of a solution.

$$\text{Minimise } f(\mathbf{x}) = \sum_{i=1}^{\text{size}(\mathbf{x})-1} D_{x_i, x_{i+1}}$$

- [Optional] Solutions must satisfy certain constraints, which define solution feasibility.
  - $h_1(\mathbf{x}) = 0$
  - $h_2(\mathbf{x}) = 0$

Routing Problem: Making it More Formal

Minimise $f(\mathbf{x}) = \sum_{i=1}^{\text{size}(\mathbf{x})-1} D_{x_i,x_{i+1}}$

Subject to $h_1(\mathbf{x}) = 0$ and $h_2(\mathbf{x}) = 0$

Where $x_i \in \{1,\cdots,N\}$; $\{1,\cdots,N\}$ are the cities in the map; $\mathbf{x}$ has any size;

D is a matrix of distances, with each position $D_{i,j}$ containing:

- the distance in km to travel directly between city i and j, or
- -1 if such direct path does not exist.

$$h_1(\mathbf{x}) = \begin{cases} 0 & \text{if } D_{x_i,x_{i+1}} \neq -1, \quad \forall i \in \{1,\ldots,\text{size}(\mathbf{x}) - 1\} \\ 1 & \text{otherwise} \end{cases}$$

$$h_2(\mathbf{x}) = \begin{cases} 0 & \text{if } x_1 = \text{OriginCity and } x_{\text{size}(\mathbf{x})} = \text{DestinationCity} \\ 1 & \text{otherwise} \end{cases}$$

Summary

- We can formulate an optimisation problem by specifying:
  - Design variables.
  - Objective functions.
  - Constraints.

Next

- How to solve optimisation problems?