MS-E2122 - Nonlinear Optimization
Lecture I

Fernando Dias

Department of Mathematics and Systems Analysis

Aalto University
School of Science

Outline of this lecture

Context

- Mathematical programming and optimisation
- Types of mathematical optimisation models

Example of Applications

- Resource allocation
- The pooling problem: refinery operations planning
- Robust optimisation
- Combinatorial optimisation

Outline of this lecture

Definition

NonLinear (or Non-Linear):

- *adjective*
- not arranged in a straight line.
- not sequential or straightforward.

Optimization (or Optimisation):

- *noun*
- the action of making the best or **most effective** use of a **situation** or **resource**.

Definition

Field of applied mathematics. The goal is to search values for variables in a given domain that maximise/minimise function values.

Can be achieved by:

- Analysing/Visualizing properties of functions / extreme points or

Definition

Field of applied mathematics. The goal is to search values for variables in a given domain that maximise/minimise function values.

Can be achieved by:

- Analysing/Visualizing properties of functions / extreme points or
- Applying numerical methods

Definition

Field of applied mathematics. The goal is to search values for variables in a given domain that maximise/minimise function values.

Can be achieved by:

- Analysing/Visualizing properties of functions / extreme points or
- Applying numerical methods

Optimisation has important applications in fields such as

- **operations research (OR)**;
- economics;
- statistics;
- bioinformatics;
- machine learning and artificial intelligence.

What is optimisation?

In this course, optimisation is viewed as the core element of mathematical programming.

Math. programming is a central OR modelling paradigm:

- **variables** $\rightarrow$ decisions/point of interest: business decisions, parameter definitions, settings, geometries, ...;

What is optimisation?

In this course, optimisation is viewed as the core element of mathematical programming.

Math. programming is a central OR modelling paradigm:

- **variables** $\rightarrow$ decisions/point of interest: business decisions, parameter definitions, settings, geometries, ...;
- **domain** $\rightarrow$ constraints/limitations: logic, design, engineering, ...;

What is optimisation?

In this course, optimisation is viewed as the core element of mathematical programming.

Math. programming is a central OR modelling paradigm:

- **variables** $\rightarrow$ decisions/point of interest: business decisions, parameter definitions, settings, geometries, ...;
- **domain** $\rightarrow$ constraints/limitations: logic, design, engineering, ...;
- **function** $\rightarrow$ objective function/profit: measurement of (decision) quality.

What is optimisation?

In this course, optimisation is viewed as the core element of mathematical programming.

Math. programming is a central OR modelling paradigm:

- **variables** $\rightarrow$ decisions/point of interest: business decisions, parameter definitions, settings, geometries, ...;
- **domain** $\rightarrow$ constraints/limitations: logic, design, engineering, ...;
- **function** $\rightarrow$ objective function/profit: measurement of (decision) quality.

However, math. programming has many applications in fields other than OR, which causes some confusion;

We will study math. programming in its most general form: both constraints and objectives are nonlinear functions.

Types of programming

Rule of Thumb:

The simpler are the assumptions which define a type of problems, the better are the methods to solve such problems.

Types of programming

Rule of Thumb:

The simpler are the assumptions which define a type of problems, the better are the methods to solve such problems.

Some useful notation:

- $x \in \mathbb{R}^n$ - vector of (decision) variables x_j, $j = 1, \ldots, n$;

Types of programming

Rule of Thumb:

The simpler are the assumptions which define a type of problems, the better are the methods to solve such problems.

Some useful notation:

- $x \in \mathbb{R}^n$ - vector of (decision) variables x_j, $j = 1, \ldots, n$;
- $f : \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ - objective function;

Types of programming

Rule of Thumb:

The simpler are the assumptions which define a type of problems, the better are the methods to solve such problems.

Some useful notation:

- $x \in \mathbb{R}^n$ - vector of (decision) variables x_j, $j = 1, \ldots, n$;
- $f : \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ - objective function;
- $X \subseteq \mathbb{R}^n$ - ground set (physical constraints);

Types of programming

Rule of Thumb:

The simpler are the assumptions which define a type of problems, the better are the methods to solve such problems.

Some useful notation:

- $x \in \mathbb{R}^n$ - vector of (decision) variables x_j, $j = 1, \ldots, n$;
- $f : \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ - objective function;
- $X \subseteq \mathbb{R}^n$ - ground set (physical constraints);
- $g_i, h_i : \mathbb{R}^n \to \mathbb{R}$ - constraint functions;

Types of programming

Rule of Thumb:

The simpler are the assumptions which define a type of problems, the better are the methods to solve such problems.

Some useful notation:

- $x \in \mathbb{R}^n$ - vector of (decision) variables x_j, $j = 1, \ldots, n$;
- $f : \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ - objective function;
- $X \subseteq \mathbb{R}^n$ - ground set (physical constraints);
- $g_i, h_i : \mathbb{R}^n \to \mathbb{R}$ - constraint functions;
- $g_i(x) \leq 0$ for $i = 1, \ldots, m$ - inequality constraints;

Types of programming

Rule of Thumb:

The simpler are the assumptions which define a type of problems, the better are the methods to solve such problems.

Some useful notation:

- $x \in \mathbb{R}^n$ - vector of (decision) variables x_j, $j = 1, \ldots, n$;
- $f : \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ - objective function;
- $X \subseteq \mathbb{R}^n$ - ground set (physical constraints);
- $g_i, h_i : \mathbb{R}^n \to \mathbb{R}$ - constraint functions;
- $g_i(x) \leq 0$ for $i = 1, \ldots, m$ - inequality constraints;
- $h_i(x) = 0$ for $i = 1, \ldots, l$ - equality constraints.

Types of programming

Our goal will be to solve variations of the general problem P:

$$
\begin{aligned}
(P):\quad \text{min.}\quad & f(x) \\
\text{subject to:}\quad & g_i(x) \leq 0, i = 1, \dots, m \\
& h_i(x) = 0, i = 1, \dots, l \\
& x \in X.
\end{aligned}
$$

Types of programming

Our goal will be to solve variations of the general problem P:

$$
\begin{aligned}
(P):\quad \text{min.}\quad & f(x) \\
\text{subject to:}\quad & g_i(x) \leq 0, i = 1, \dots, m \\
& h_i(x) = 0, i = 1, \dots, l \\
& x \in X.
\end{aligned}
$$

- **Linear programming (LP):** linear $f(x) = c^\top x$ with $c \in \mathbb{R}^n$; constraint functions $g_i(x)$ and $h_i(x)$ are affine ($a_i^\top x - b_i$, with $a_i \in \mathbb{R}^n$, $b \in \mathbb{R}$); $X = \{x \in \mathbb{R}^n : x_j \geq 0, j = 1, \dots, n\}$.

Types of programming

Our goal will be to solve variations of the general problem P:

$$\begin{aligned}(P):\quad \text{min.}\quad & f(x) \\ \text{subject to:}\quad & g_i(x) \leq 0, i = 1, \dots, m \\ & h_i(x) = 0, i = 1, \dots, l \\ & x \in X.\end{aligned}$$

- **Linear programming (LP):** linear $f(x) = c^\top x$ with $c \in \mathbb{R}^n$; constraint functions $g_i(x)$ and $h_i(x)$ are affine ($a_i^\top x - b_i$, with $a_i \in \mathbb{R}^n$, $b \in \mathbb{R}$); $X = \{x \in \mathbb{R}^n : x_j \geq 0, j = 1, \dots, n\}$.
- **Nonlinear programming (NLP):** some (or all) of the functions f, g_i or h_i are nonlinear;

Types of programming

Our goal will be to solve variations of the general problem P:

$$
\begin{aligned}
(P): \quad \text{min.} \quad & f(x) \\
\text{subject to:} \quad & g_i(x) \leq 0, i = 1, \dots, m \\
& h_i(x) = 0, i = 1, \dots, l \\
& x \in X.
\end{aligned}
$$

- **Linear programming (LP):** linear $f(x) = c^\top x$ with $c \in \mathbb{R}^n$; constraint functions $g_i(x)$ and $h_i(x)$ are affine ($a_i^\top x - b_i$, with $a_i \in \mathbb{R}^n$, $b \in \mathbb{R}$); $X = \{x \in \mathbb{R}^n : x_j \geq 0, j = 1, \dots, n\}$.
- **Nonlinear programming (NLP):** some (or all) of the functions f, g_i or h_i are nonlinear;
- **(Mixed-)integer programming ((M)IP):** LP where (some of the) variables are binary (or integer). $X \subseteq \mathbb{R}^k \times \{0, 1\}^{n-k}$

Types of programming

Our goal will be to solve variations of the general problem P:

$$
\begin{aligned}
(P): \quad \text{min.} \quad & f(x) \\
\text{subject to:} \quad & g_i(x) \leq 0, i = 1, \dots, m \\
& h_i(x) = 0, i = 1, \dots, l \\
& x \in X.
\end{aligned}
$$

- **Linear programming (LP):** linear $f(x) = c^\top x$ with $c \in \mathbb{R}^n$; constraint functions $g_i(x)$ and $h_i(x)$ are affine ($a_i^\top x - b_i$, with $a_i \in \mathbb{R}^n$, $b \in \mathbb{R}$); $X = \{x \in \mathbb{R}^n : x_j \geq 0, j = 1, \dots, n\}$.
- **Nonlinear programming (NLP):** some (or all) of the functions f, g_i or h_i are nonlinear;
- **(Mixed-)integer programming ((M)IP):** LP where (some of the) variables are binary (or integer). $X \subseteq \mathbb{R}^k \times \{0,1\}^{n-k}$
- **Mixed-integer nonlinear programming (MINLP):** MIP+NLP.

Types of programming

Outline of this lecture

- Context
  - Mathematical programming and optimisation
  - Types of mathematical optimisation models
- Example of Applications
  - Resource allocation
  - The pooling problem: refinery operations planning
  - Robust optimisation
  - Combinatorial optimisation

Resource allocation and portfolio optimisation

Problem statement. Plan production that maximises return. Let

- $I = \{1, \ldots, i, \ldots, M\}$ resources;
- $J = \{1, \ldots, j, \ldots, N\}$ products;
- c_j - return per unit of product $j \in J$;
- a_{ij} - resource $i \in I$ requirement for making product $j \in J$;
- b_i - availability of resource $i \in I$;
- x_j - production of $j \in J$.

Resource allocation and portfolio optimisation

Problem statement. Plan production that maximises return. Let

- $I = \{1, \dots, i, \dots, M\}$ resources;
- $J = \{1, \dots, j, \dots, N\}$ products;
- c_j - return per unit of product $j \in J$;
- a_{ij} - resource $i \in I$ requirement for making product $j \in J$;
- b_i - availability of resource $i \in I$;
- x_j - production of $j \in J$.

$$
\begin{aligned}
\text{max.} \quad & \sum_{j \in J} c_j x_j \\
\text{subject to:} \quad & \sum_{j \in J} a_{ij} x_j \leq b_i, \forall i \in I \\
& x_j \geq 0, \forall j \in J
\end{aligned}
$$

Resource allocation and portfolio optimisation

Problem statement. Plan production that maximises return. Let

- $I = \{1, \dots, i, \dots, M\}$ resources;
- $J = \{1, \dots, j, \dots, N\}$ products;
- c_j - return per unit of product $j \in J$;
- a_{ij} - resource $i \in I$ requirement for making product $j \in J$;
- b_i - availability of resource $i \in I$;
- x_j - production of $j \in J$.

$$\begin{aligned} \text{max.} \quad & \sum_{j \in J} c_j x_j \\ \text{subject to:} \quad & \sum_{j \in J} a_{ij} x_j \leq b_i, \forall i \in I \\ & x_j \geq 0, \forall j \in J \end{aligned}$$

Remark:

- notice that max. $f(x) =$ min. $-f(x)$;

Resource allocation and portfolio optimisation

Problem statement. Plan production that maximises return. Let

- $I = \{1, \dots, i, \dots, M\}$ resources;
- $J = \{1, \dots, j, \dots, N\}$ products;
- c_j - return per unit of product $j \in J$;
- a_{ij} - resource $i \in I$ requirement for making product $j \in J$;
- b_i - availability of resource $i \in I$;
- x_j - production of $j \in J$.

$$\begin{aligned} \text{max.} \quad & \sum_{j \in J} c_j x_j \\ \text{subject to:} \quad & \sum_{j \in J} a_{ij} x_j \leq b_i, \forall i \in I \\ & x_j \geq 0, \forall j \in J \end{aligned}$$

Remark:

- notice that max. $f(x) =$ min. $-f(x)$;
- the base of most practical optimisation problems; exploits mature LP technology.

Portfolio optimization

Problem statement. Plan portfolio of assets to minimise exposition to risk. Let

- $J = \{1, \ldots, j, \ldots, N\}$ assets;
- μ_j - expected relative return of asset $j \in J$;
- Σ - covariance matrix;
- ϵ - minimum expected return;
- x_j - position of asset $j \in J$

Portfolio optimization

Problem statement. Plan portfolio of assets to minimise exposition to risk. Let

- $J = \{1, \dots, j, \dots, N\}$ assets;
- μ_j - expected relative return of asset $j \in J$;
- Σ - covariance matrix;
- ϵ - minimum expected return;
- x_j - position of asset $j \in J$

$$
\begin{aligned}
\text{min. } \; & x^\top \Sigma x \\
\text{subject to: } \; & \mu^\top x \geq \epsilon \\
& 0 \leq x_j \leq 1, \forall j \in J
\end{aligned}
$$

Portfolio optimization

Problem statement. Plan portfolio of assets to minimise exposition to risk. Let

- $J = \{1, \dots, j, \dots, N\}$ assets;
- μ_j - expected relative return of asset $j \in J$;
- Σ - covariance matrix;
- ϵ - minimum expected return;
- x_j - position of asset $j \in J$

$$\begin{aligned} \text{min.} \quad & x^\top \Sigma x \\ \text{subject to:} \quad & \mu^\top x \geq \epsilon \\ & 0 \leq x_j \leq 1, \forall j \in J \end{aligned}$$

Remarks:

- The term $x^\top \Sigma x$ measures exposition to risk. It is credited to Harry Markowitz (1952).
- Another important class: quadratic programming (nonlinear).

Refinery Operations Planning Problem

Oil refinery operational planning

- Goal is to maximize profit;
- Several possible configurations;
- Product property specifications must be met;

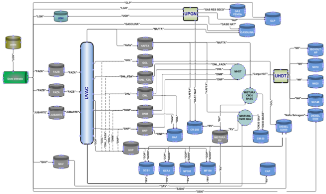
Refinery Operations Planning Problem

Oil refinery operational planning

- Goal is to maximize profit;
- Several possible configurations;
- Product property specifications must be met;

Model characteristics:

- Bilinear (nonconvex) and mixed-integer;
- Large number of flows;
- Several nonlinear constraints.

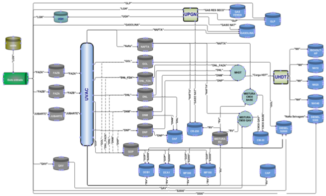


Refinery Operations Planning Problem

Objective: maximize profit

Variables:

- Stream Flows (crude, intermediate and final products);
- Storage;
- Stream properties.



Refinery Operations Planning Problem

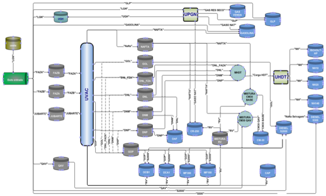
Objective: maximize profit

Variables:

- Stream Flows (crude, intermediate and final products);
- Storage;
- Stream properties.

Constraints

- Mass balance;
- Market features (supply and demand);
- Unit capacities;
- Stream property limits;
- Calculation of mix properties (nonlinear).

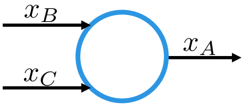


Refinery Operations Planning Problem

The challenging aspect is how to model the calculation of product properties in a mix. Let:

- x_p be the volume of product $p \in P$ and
- q_p the value of a given chemical property (sulphur content, octane content, viscosity...).

In a given mix, mass and property balances are calculated as:



$$x_A = x_B + x_C$$

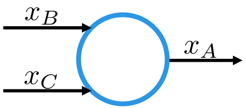
$$q_A = \frac{q_B x_B + q_C x_C}{x_A}$$

Refinery Operations Planning Problem

The challenging aspect is how to model the calculation of product properties in a mix. Let:

- x_p be the volume of product $p \in P$ and
- q_p the value of a given chemical property (sulphur content, octane content, viscosity...).

In a given mix, mass and property balances are calculated as:



$$x_A = x_B + x_C$$

$$q_A = \frac{q_B x_B + q_C x_C}{x_A}$$

Remarks:

- More complex mixes (such as nonlinear balances) might need to be considered.
- These are bilinear programming problems (nonlinear).

Robust optimisation

Is a subarea of mathematical programming concerned with uncertainty in the input data.

It's a risk-averse perspective that seeks protection against variability.

Robust optimisation

Is a subarea of mathematical programming concerned with uncertainty in the input data.

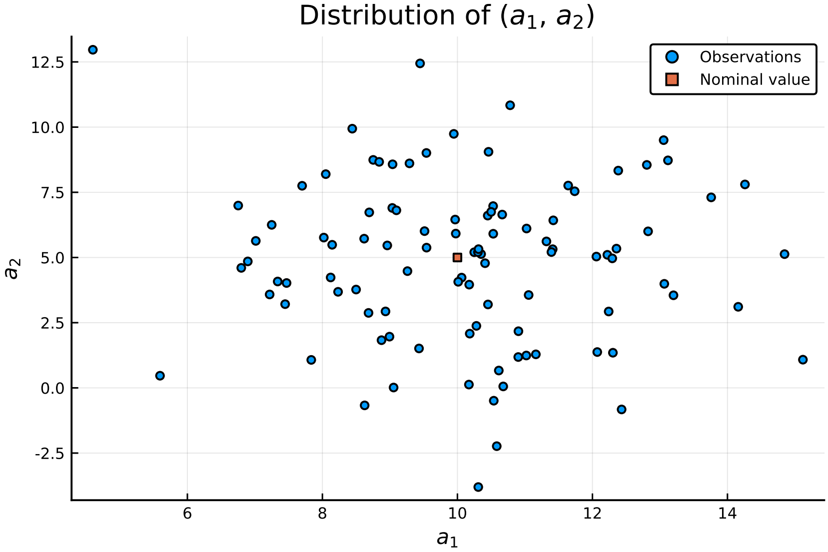
It's a risk-averse perspective that seeks protection against variability.

Consider the resource allocation problem under uncertainty:

$$\begin{aligned} \text{max.} \quad & c^\top x \\ \text{subject to:} \quad & \tilde{a}_i^\top x \leq b_i, \forall i \in I \\ & x_j \geq 0, \forall j \in J, \end{aligned}$$

where $\tilde{a}_i$ is a random variable.

Robust optimisation



Robust optimisation

Assume that, for any $i \in I$, $\tilde{a}_i \in \epsilon_i = \{\overline{a}_i + P_i u : ||u||_2 \leq \Gamma_i\}$, where

- $\overline{a}_i$ is the nominal (average) value;
- P_i is the characteristic matrix of the ellipsoid ϵ;
- Γ_i is risk-aversion control parameter.

Robust optimisation

Assume that, for any $i \in I$, $\tilde{a}_i \in \epsilon_i = \{\overline{a}_i + P_i u : ||u||_2 \leq \Gamma_i\}$, where

- $\overline{a}_i$ is the nominal (average) value;
- P_i is the characteristic matrix of the ellipsoid ϵ;
- Γ_i is risk-aversion control parameter.

Then, the robust counterpart can be stated as

$$\begin{aligned} \text{max.} \quad & c^\top x \\ \text{subject to:} \quad & \max_{a_i \in \epsilon_i} \left\{ a_i^\top x \right\} \leq b_i, \forall i \in I \\ & x_j \geq 0, \forall j \in J. \end{aligned}$$

Robust optimisation

Assume that, for any $i \in I$, $\tilde{a}_i \in \epsilon_i = \{\overline{a}_i + P_i u : ||u||_2 \leq \Gamma_i\}$, where

- $\overline{a}_i$ is the nominal (average) value;
- P_i is the characteristic matrix of the ellipsoid ϵ;
- Γ_i is risk-aversion control parameter.

Then, the robust counterpart can be stated as

$$\begin{aligned} \text{max.} \quad & c^\top x \\ \text{subject to:} \quad & \max_{a_i \in \epsilon_i} \left\{ a_i^\top x \right\} \leq b_i, \forall i \in I \\ & x_j \geq 0, \forall j \in J. \end{aligned}$$

Notice that

$$\max_{a_i \in \epsilon_i} \left\{ a_i^\top x \right\} = \overline{a}_i^\top x + \max_{u} \left\{ u^\top P_i x : ||u||_2 \leq \Gamma_i \right\} = \overline{a}_i^\top x + \Gamma_i ||P_i x||_2$$

Robust optimisation

The robust counterpart can be equivalently stated as:

$$\begin{aligned}
\text{max.}\quad & c^\top x \\
\text{subject to:}\quad & \overline{a}_i^\top x + \Gamma_i ||P_i x||_2 \leq b_i, \forall i \in I \\
& x_j \geq 0, \forall j \in J.
\end{aligned}$$

Remarks:

- In case data is available, P_i can be obtained from the empirical covariance matrix;

Robust optimisation

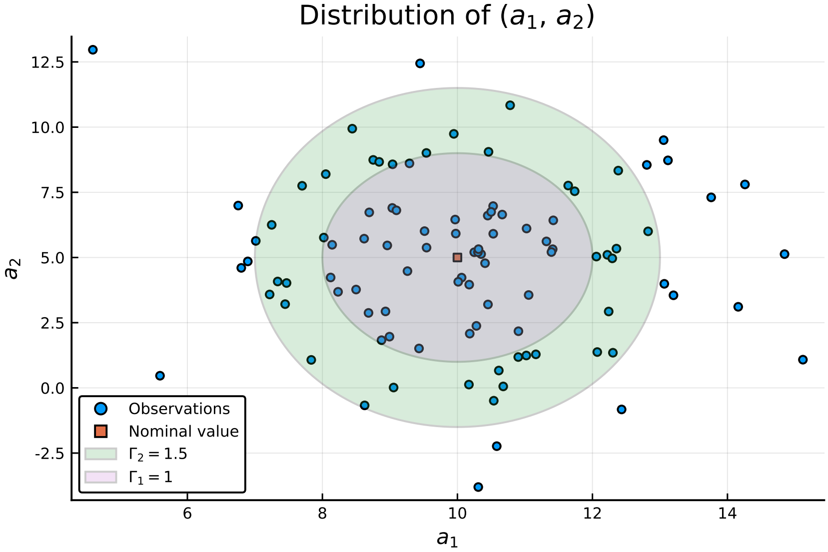
The robust counterpart can be equivalently stated as:

$$\begin{aligned} \text{max. } & c^\top x \\ \text{subject to: } & \overline{a}_i^\top x + \Gamma_i ||P_i x||_2 \leq b_i, \forall i \in I \\ & x_j \geq 0, \forall j \in J. \end{aligned}$$

Remarks:

- In case data is available, P_i can be obtained from the empirical covariance matrix;
- Values of Γ_i can be drawn, for example, from a Chi-squared distribution. Γ_i is sometimes called the budget of uncertainty.

Robust optimisation



Combinatorial optimisation

Is a subfield of mathematical programming regarding finding optimal solutions from a inite set of possible solutions. The finite set is discrete or can be reduced to a discrete set.

Can be also refered as discrete optimization, integer programming

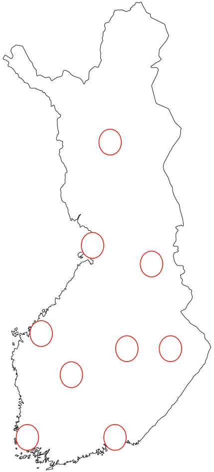
Combinatorial optimisation

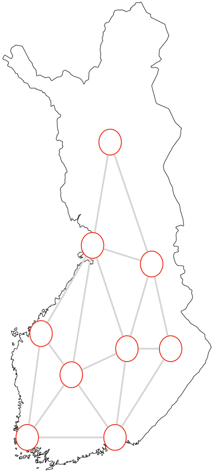
Is a subfield of mathematical programming regarding finding optimal solutions from a inite set of possible solutions. The finite set is discrete or can be reduced to a discrete set.

Can be also refered as discrete optimization, integer programming

Combinatorial optimisation

One of most classic problems is Travelling Salesman Problem (TSP)





Combinatorial optimisation

Assume a number of cities labeled with numbers $1, \cdots, n$, and the distance (or any cost) between two cities i and j is defined as c_{ij}.

The main variable is x_{ij} such that:

- 1: if there is a path between city i and city j;
- 0: otherwise

Combinatorial optimisation

Assume a number of cities labeled with numbers $1, \cdots, n$, and the distance (or any cost) between two cities i and j is defined as c_{ij}.

The main variable is x_{ij} such that:

- 1: if there is a path between city i and city j;
- 0: otherwise

x_{ij} is binary variable $\leftarrow$ integer programming

Combinatorial optimisation

The goal can be stated as:

Find a path that goes through every city using the least amount of resources.

Combinatorial optimisation

The goal can be stated as:

> *Find a path that goes through every city using the least amount of resources.*

The objective function is:

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

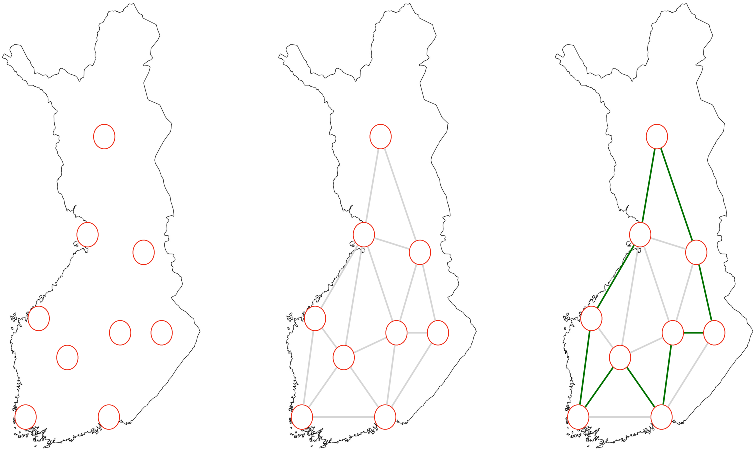
Combinatorial optimisation

The constraints can be written as:

$$\sum_{i \neq j, i=1}^{n} x_{ij} = 1 \qquad \forall j = 1, \cdots, n \qquad (2)$$

$$\sum_{j \neq i, j=1}^{n} x_{ij} = 1 \qquad \forall j = 1, \cdots, n \qquad (3)$$

Combinatorial optimisation




THANK YOU

Reference

Sahinidis, Nikolaos V. "Mixed-integer nonlinear programming 2018." Optimization and Engineering 20 (2019): 301-306.

Floudas, Christodoulos A. Nonlinear and mixed-integer optimization: fundamentals and applications. Oxford University Press, 1995.

Tawarmalani, Mohit, and Nikolaos V. Sahinidis. Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications. Vol. 65. Springer Science & Business Media, 2013.