01

MASTER2 _ IAII

Introduction to

OPTIMIZATION PROBLEMS

BOUABD CHAIMAA
TAKATART YASMINE
AIT ELBAZ AICHA

Contents

→

Introduction

A. Presentation of the topic and its importance.

Optimization is essentially everywhere, from **engineering design** to **economics** and from holiday planning to Internet routing. As **money**, **resources** and **time** are always **limited**, the optimal utility of these available resources is crucially important.

Most real-world optimizations are highly **nonlinear** and **multimodal**, under various complex constraints. Different objectives are often conflicting. Even for a single objective, sometimes, optimal solutions may not exist at all. In general, finding an optimal solution or even sub-optimal solutions is not an easy task.

B. Course objectives.

Upon completion of this course, you will be able to:

- Grasp the fundamental **concepts of optimization,** including the **types of optimization problems** and their real-world applications.
- Differentiate between **continuous and combinatorial optimization** problems, understanding their unique **characteristics** and challenges.
- Formulate real-world problems into **mathematical optimization model**s.
- Analyze continuous optimization problems, including **linear regression,** and apply appropriate mathematical methods to find optimal solutions.

- Explore combinatorial problems like the **Traveling Salesman Problem** (TSP).
- Gain hands-on experience using Python libraries such as SciPy, CVXPY, NetworkX, and PuLP for solving both continuous and combinatorial optimization problems.
- Apply optimization techniques in practical, real-world scenarios, including industry-related problems and research applications.

C. Application Areas

Optimization problems represent a very powerful tool for decision support in economic, social, and industrial institutions. They find applications in various fields, including but not limited to:

- Public Services: Hospitals, public transportation, information technology.
- Industry: Automotive, aviation, energy, telecommunications, production.
- Finance: Portfolio management.
- Military: Resource management, logistics.
- Economics: Project planning, activity organization, task allocation, transportation problems.
- Computer Science: Program design, implementation of computer systems.
- Sociology: Modeling and study of social phenomena.

D. Definition of optimization.

Optimization refers to the process of **making something as effective or functional** as possible. In mathematics, computer science, and related fields, optimization specifically refers to the selection of **the best element** or outcome from a set of **feasible options.**

In the context of mathematical optimization, it involves finding the **best solution** to a problem from **a set of possible solutions**. The goal is to either **minimize or maximize** a certain **objective function** while satisfying a set of **constraints**. These constraints can include limitations on resources, physical or logical constraints, or other conditions that the solution must adhere to.

An optimization problem can be represented in the following way:

Maximization Problem:

Maximize $f(x)$
subject to $g_i(x) \leq 0$,i=1,2,...,m
and $h_j(x) = 0$, j=1,2,...,p

Minimization Problem:

Minimize $f(x)$
subject to $g_i(x) \leq 0$,i=1,2,...,m
and $h_j(x) = 0$,j=1,2,...,p

Given: a function $f : A \to \mathbb{R}$ from some set A to the real numbers
Sought: an element $\mathbf{x}_0 \in A$ such that $f(\mathbf{x}_0) \leq f(\mathbf{x})$ for all $\mathbf{x} \in A$ ("minimization")
or
such that $f(\mathbf{x}_0) \geq f(\mathbf{x})$ for all $\mathbf{x} \in A$ ("maximization").

Specific forms of f(x), gi(x), and hj(x) will depend on the nature of the problem: whether it's **linear, nonlinear, convex, non-convex,** etc. Constraints and the objective function can involve a combination of linear and nonlinear equations or functions, depending on the complexity of the problem.

For example, in a linear optimization problem, both the objective function and the constraints are linear functions of the variables. In a quadratic optimization problem, the objective function is quadratic, but the constraints can still be linear.

Steps to solve an optimization problem

1. Analysis and formulation of the problem

- Identify the different variables, their natures and their areas of study
- Define the objective of the problem
- Define possible constraints

2. Mathematical modeling of the problem

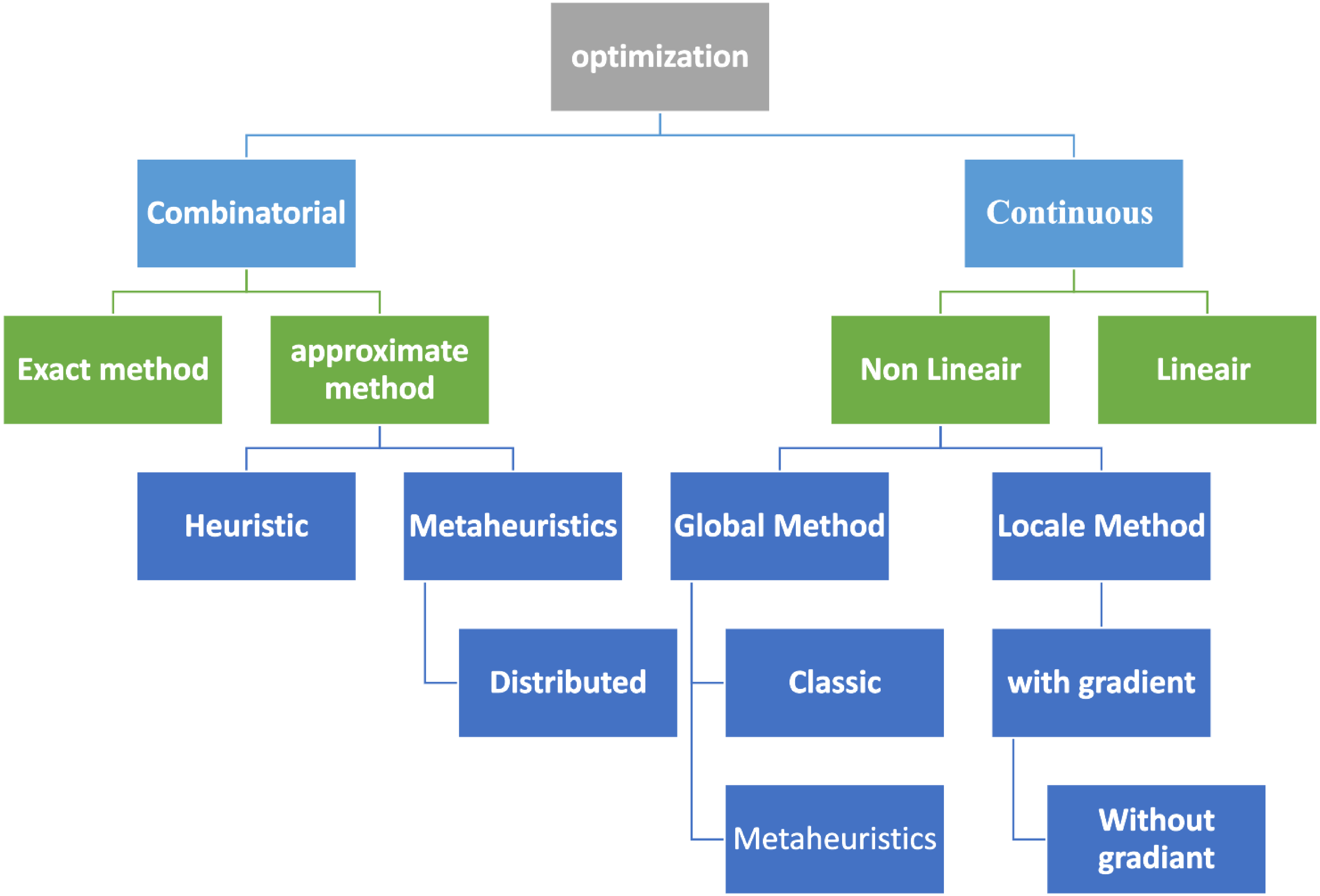
- Formulate a mathematical function, called an objective function, which describes the problem
- Write the constraints in mathematical form

3**. Choice of a method for solving the modeled problem**

- Choose an optimization method appropriate to the type of problem
- Apply the optimization method to solve the modeled problem

II. Optimization Problems

A. Understanding the types of optimization problems


optimization
Combinatorial
Continuous
Exact method
approximate method
Non Lineair
Lineair
Heuristic
Metaheuristics
Global Method
Locale Method
Distributed
Classic
with gradient
Metaheuristics
Without gradiant

Optimization problems can be divided into two categories, depending on whether the variables are continuous or discrete:

- An optimization problem with discrete variables is indeed known as discrete optimization or combinatorial optimization. In these types of problems, the goal is to find the best solution from a countable set of possible solutions, where the variables can only take on distinct, separate values (such as integers, permutations, or graph structures

- A problem with continuous variables is known as a continuous optimization, in which an optimal value from a continuous function must be found. They can include constrained problems and multimodal problems.

Continuous Optimization

Continuous optimization involves variables that can take on a **continuous range of values,** usually r**eal numbers**, distinguishing it from discrete or combinatorial optimization where variables are binary, integer, or drawn from sets with finitely many elements.

Continuous optimization problems are solved using **algorithms that generate a sequence of variable values** (iterates) converging to a solution. These algorithms use knowledge from previous iterates and information about the current model, including sensitivities defined in terms of first and second derivatives of the defining functions, to determine the next step.

Combinatorial optimization

Combinatorial optimization problems involve selecting **the best solution** from a **finite set of discrete choices**, prevalent in domains like **logistics, manufacturing, and scheduling.** The challenge is to explore this finite solution space efficiently while **meeting specific constraints.**

Solving these problems is vital for optimizing operations and making informed decisions in complex, real-world situations. Researchers use advanced algorithms and **heuristics** to **enhance efficiency,** cost-effectiveness, and overall performance in various applications.

B. Examples of real problems

1. Cost minimization problem

The problem of cost minimization is a common issue in many fields of industry, commerce, and engineering. Here are some specific examples of situations where this problem can arise:

- **Supply Chain Management:** Companies aim to minimize costs related to storage, production, and distribution of their products. By optimizing these processes, total costs can be reduced, leading to significant savings.
- **Transportation and Logistics:** Transportation companies seek to minimize costs related to vehicle fleets, routes, and warehouse management. Optimizing routes, scheduling, and efficient warehouse management are crucial to reducing operational costs.

- **Manufacturing:** Manufacturing companies aim to minimize production costs by optimizing the use of resources such as labor, raw materials, and energy. Improving production processes can reduce waste and increase efficiency, contributing to cost minimization.
- **Financial Planning:** Investors and portfolio managers aim to minimize costs related to transactions, management fees, and taxes. Effective financial planning can help minimize these costs and maximize returns on investments.
- **Human Resaources Management:** Companies aim to minimize costs related to personnel management, including recruitment, training, compensation, and benefits. Optimizing human resources management can help reduce these costs while maintaining a qualified and motivated workforce.

2. Profit maximization problem

1. **Pricing and Sales Strategy:** Businesses aim to maximize profits by setting optimal prices for their products or services. This involves balancing market demand and production costs while maximizing profit margins.
2. **Investments and Financial Portfolios:** Investors seek to maximize profits by selecting investments with the best returns. This can involve decisions about stocks, bonds, real estate, or other investments, as well as diversifying the portfolio to maximize returns while minimizing risks.
3. **Sales and Marketing Management:** Companies aim to maximize profits by optimizing their sales and marketing strategies. This includes identifying the most profitable market segments, optimizing advertising campaigns, and maximizing the return on investment in marketing initiatives.

4. Production Optimization: Manufacturing companies aim to maximize profits by optimizing production processes. This involves supply chain optimization, reducing production costs, improving labor efficiency, and adopting modern technologies to increase output.

5. Service Operations Optimization: Service-based businesses such as airlines, hotels, and restaurants aim to maximize profits by optimizing their operations. This includes managing reservations, optimizing capacities, ensuring customer satisfaction, and cost management to maximize profit margins.

Continuous Optimization problems

Continuous optimization problems involve finding the best solution within a **continuous domain**, typically represented by real numbers, rather than discrete values. These problems can be described mathematically, and there are various algorithms and techniques to solve them.

The standard form of a continuous optimization problem is:

Minimize or (Maximize) a real-valued objective function

$$f(x) : D \to \mathbb{R}$$

Subject to constraints :

$$g(x) \leq 0, \qquad for\ i = 1, \dots, m$$

$$h(x) = 0, \qquad for\ j = 1, \dots, p$$

Where :

$f: \mathbb{R}^n \to \mathbb{R}$: is the objective function to be minimized over the n-variable vector x

$g(x) \leq 0$ are called inequality constraints

$h(x) = 0$ are called equality constraints and,

$m \geq 0$ and $p \geq 0$

If $m = p = 0$ the problem is an unconstrained optimization problem. By convention, the standard form defines a minimization problem. A maximization problem can be treated by negating the objective function.

Characteristics:

Continuous variable space.
Objective function and constraints can be linear or nonlinear.
Numerous real-world applications.

Modelization

1- LINEAR PROGRAMMING (LP) :

Example : Ressource Allocation

Description: In a manufacturing company, you want to allocate resources to different production processes to maximize profit or minimize costs, considering resource availability and demand constraints.

Decision variables :

- Let *xi* be the quantity of resource allocated to process *i*
- *i* is an index that can range from *1* to *n* where *n* is the number of processes

Objective Function :

The goal in resource allocation is to either maximize profit or minimize cost.

1- LINEAR PROGRAMMING (LP) :

Example : Ressource Allocation

- To maximise profit : We have a profit vector **c**, The objective function is to Maximize the total profit, which is the sum of profits from each process i multiplied by the quantity of the ressources allcated xi

Max Z = c1x1 + c2x2 + …… cnxn

- To minimize cost : We have a cost vector c, The objective function is to Minimize the total cost, which is the sum of costs for each process i multiplied by the quantity of the ressources allocated xi

Min Z = c1x1 + c2x2 + …… cnxn

1- LINEAR PROGRAMMING (LP) :

Example : Ressource Allocation

- Constraints :

The ressource allocation problem typically involves constraints, including ressource availability and demand requirements.

A_ressource x <= b

Where : A_ressource is a matrix representing the ressource constraints

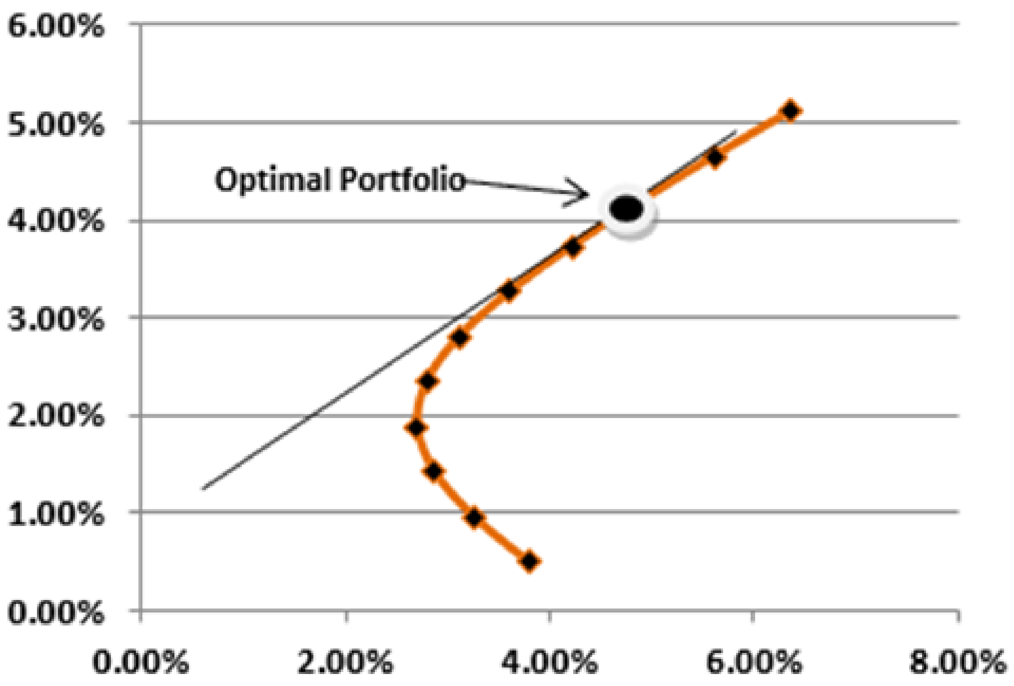
A_demand >=d

Where : A_demand is a matrix representing the demand constraints, and d is the vector of demand requirements

2-Quadratic Programming (QP) :

Example : Portfolio Optimization

Description: In finance , we want to optimize a portfolio of assets to maximize returns while minimizing risks



Objective Function :

Maximize the expected return subject to a quadratic risk measure

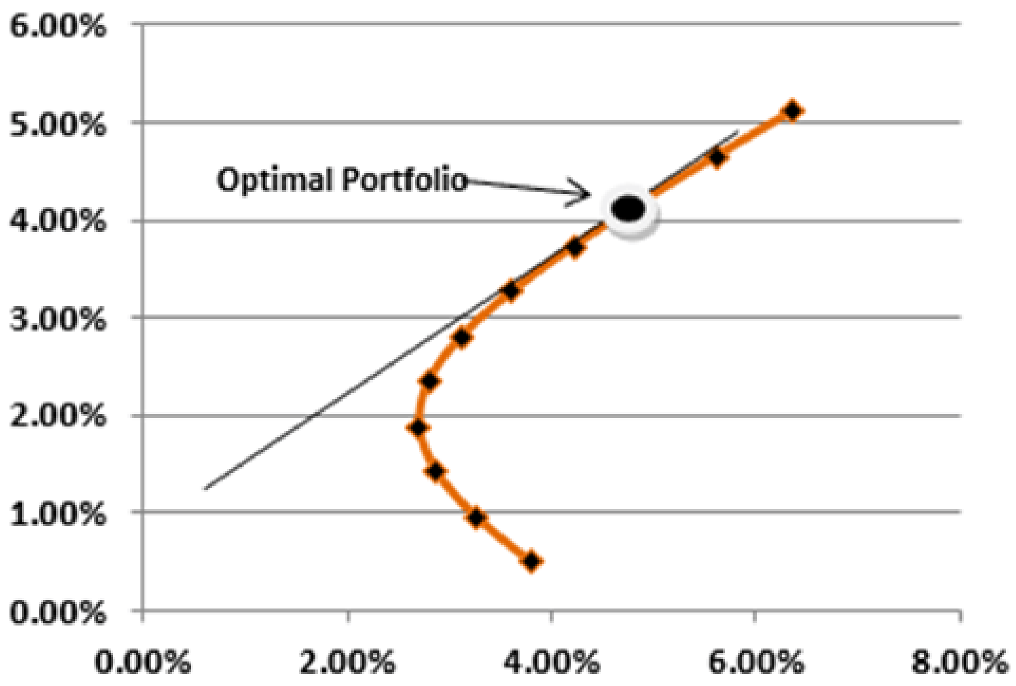
$$\max \ \mu^T w - \lambda \, w^T \Sigma w$$

Where :

- μ is a vector of expected returns for the assets in the portfolio.
- w is a vector representing the allocation (weights) of each asset in the portfolio.
- λ is a risk-aversion parameter that controls the trade-off between risk and return.
- Σ is the covariance matrix representing the risk (variance and covariance) of asset returns.

2-Quadratic Programming (QP) :

Example : Portfolio Optimization



Constraints :

In portfolio optimization, there are typically constraints on the portfolio, such as:

- Budget Constraint: The sum of the asset weights should equal 1, indicating that the entire budget is invested.

$$\sum_{i=1}^{n} w_i = 1$$

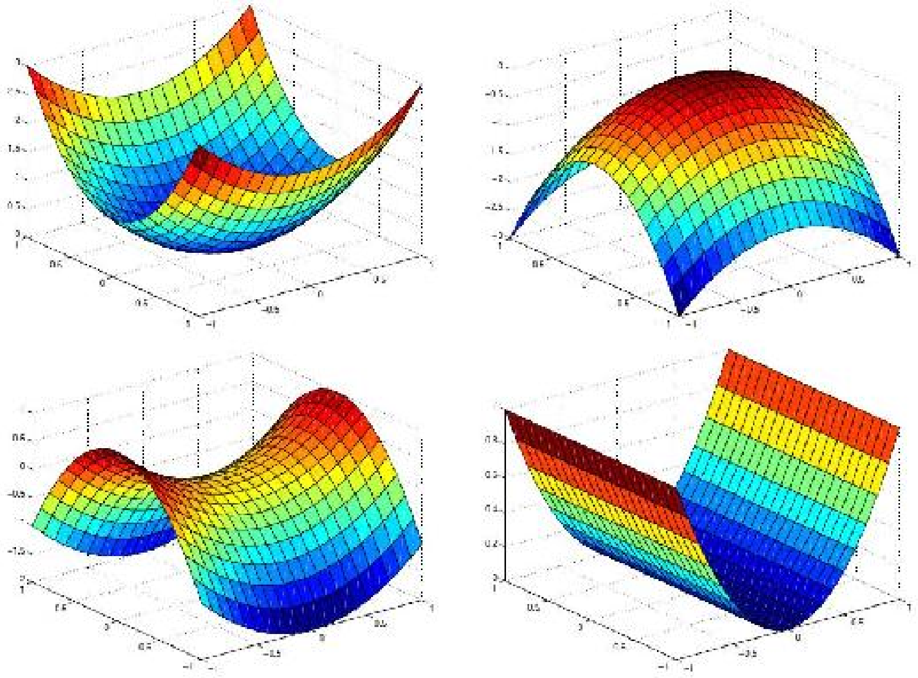
- Minimum/Maximum Weights: imposing minimum and maximum weight limits on individual assets,

$$0 \leq w_i \leq 0.2.$$

3- NONLINEAR PROGRAMMING (NLP) :

Example : Engineering Design

Description: Nonlinear programming (NLP) is a subset of continuous optimization that deals with optimization problems where the objective function and/or constraints are nonlinear in nature. This means that the relationships between variables are not linear, and they may involve non-integer powers, exponentials, trigonometric functions, or other nonlinear expressions.

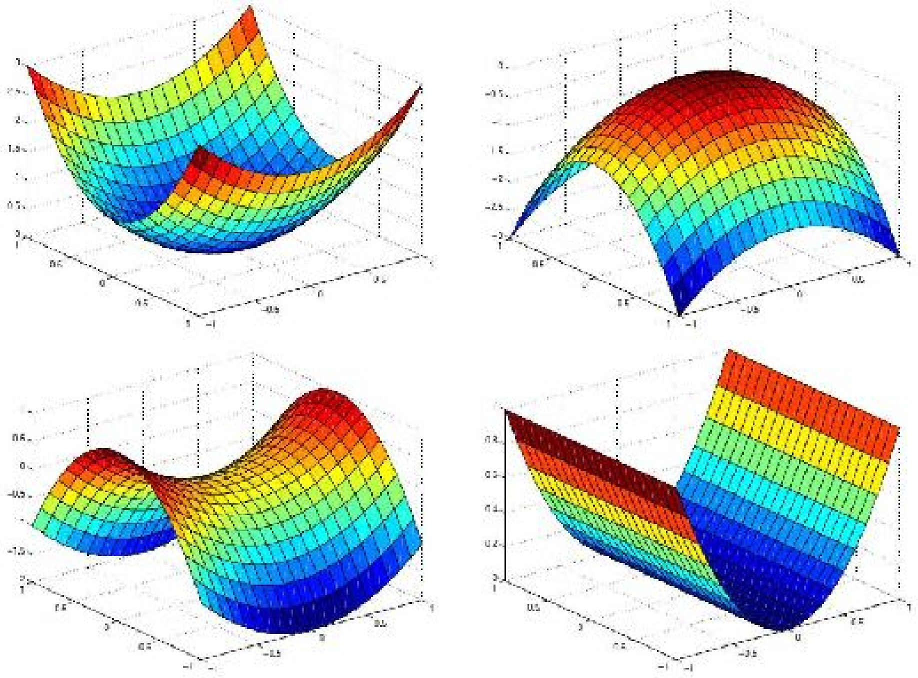


Decision variables :

- Let x be a vector of design parameters

3- NONLINEAR PROGRAMMING (NLP) :

Example : Engineering Design



Objective Function :

This is the function that we want to either maximize or minimize. It is often nonlinear and is expressed as f(x), where x represents the vector of continuous variables we're trying to optimize.

- **Minimize f(x) or Maximize –f(x)**

Constraints:

They can be equality constraints or inequality constraints. These can also be nonlinear in terms of x

$$g(x) \leq 0, \qquad for\ i = 1, \dots, m$$

$$h(x) = 0, \qquad for\ j = 1, \dots, p$$

4-CONVEX PROGRAMMING

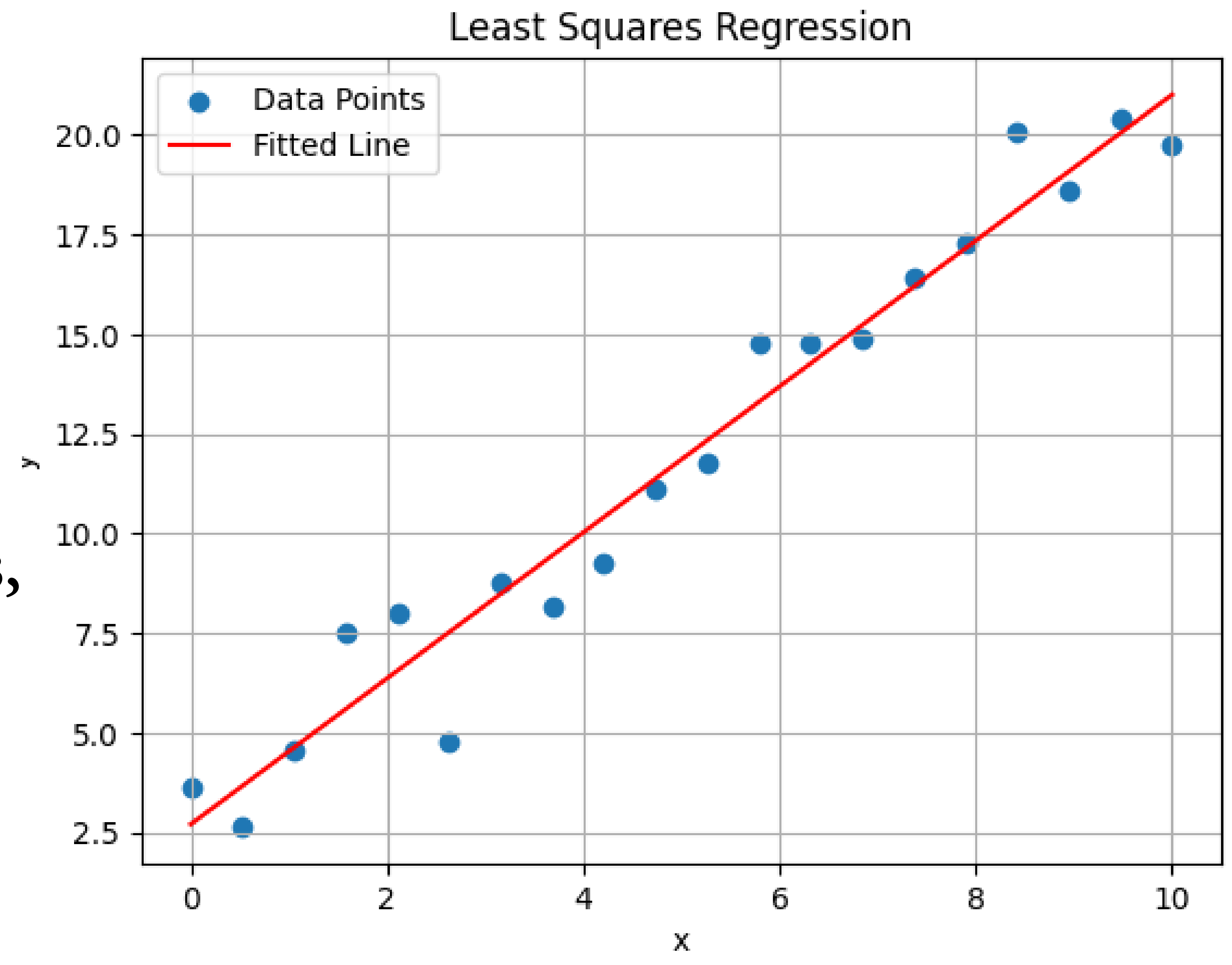
Description :

Convex programming is a subset of mathematical optimization where the objective function and constraints satisfy a specific property known as convexity.

Convex optimization problems have well-defined solutions and are widely used in various applications, including least squares regression.

In statistics , we want to find parameters that minimizes the sum of squared differences between observed and predicted values.

Example : Least square Regression



4-CONVEX PROGRAMMING

Objective Function:

The objective function for the least squares regression problem is to minimize the sum of squared residuals

$$\text{Minimize } f(a,b) = \sum_{i=1}^{n} (y_i - (a \cdot x_i + b))^2$$

This function is convex because it is a quadratic function of *a* and *b*.

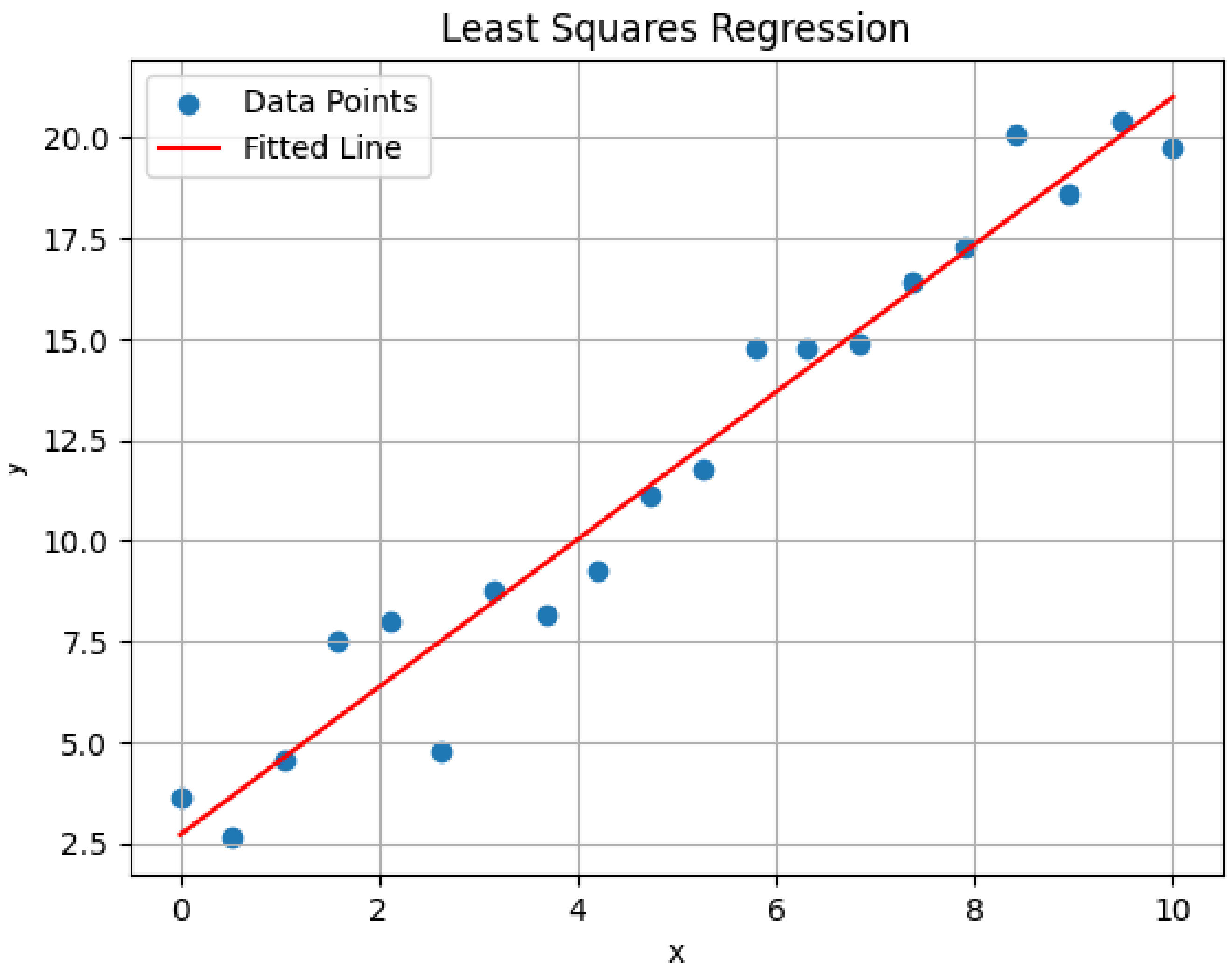
Decision variables :

In least square Regression, the decision variables typically represents the parameters to be estimated

Constraints :

In Least square Regression, there is no specific constraints

Example : Least square Regression



Combinatorial Optimization problems

Combinatorial optimization is a field of study that focuses on finding the best solution among a finite set of possibilities. This field is particularly useful in situations where there are many possible solutions, but only a few are optimal.

Modelization

SOME CLASSICAL COMBINATORIAL OPTIMIZATION PROBLEMS

KNAPSACK PROBLEM

SET COVERING PROBLEM

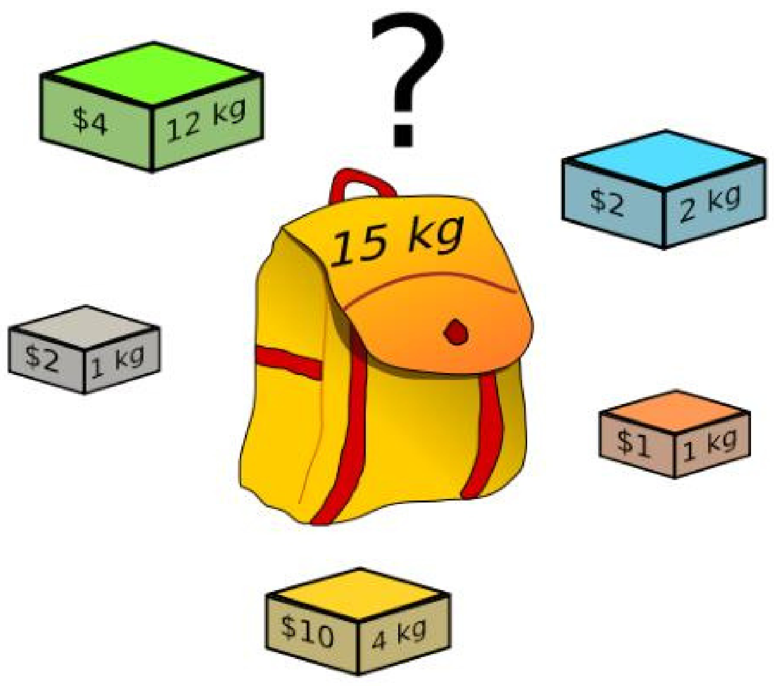
TRAVELLING SALESMAN PROBLEM

PARTITIONNING PROBLEM

FACILITY LOCATION PROBLEM

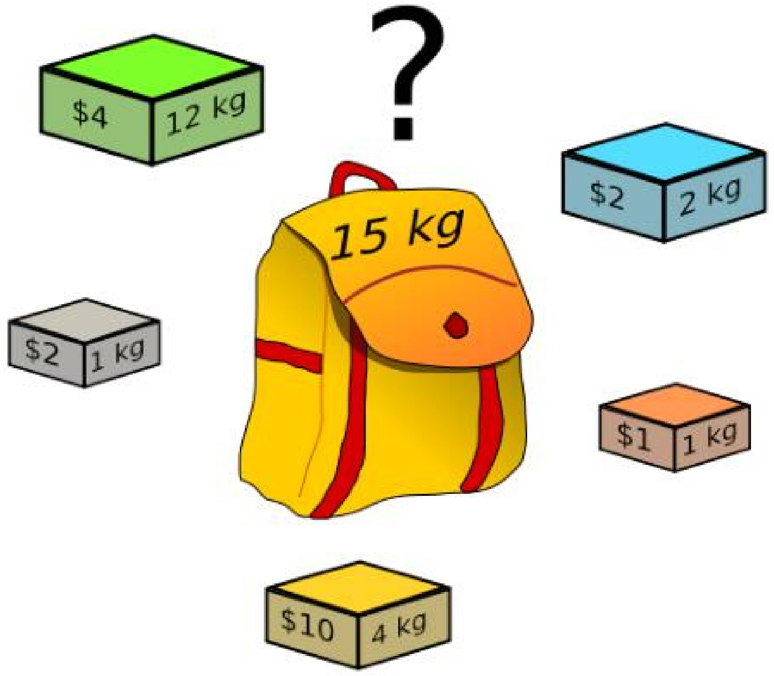
PACKING PROBLEM

KNAPSACK PROBLEM



Given a set of items, each with a weight and a value, determine which items to include in the collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.

KNAPSACK PROBLEM



$$\text{Max} \quad Z = \sum_{i=1}^{n} v_i \cdot x_i$$

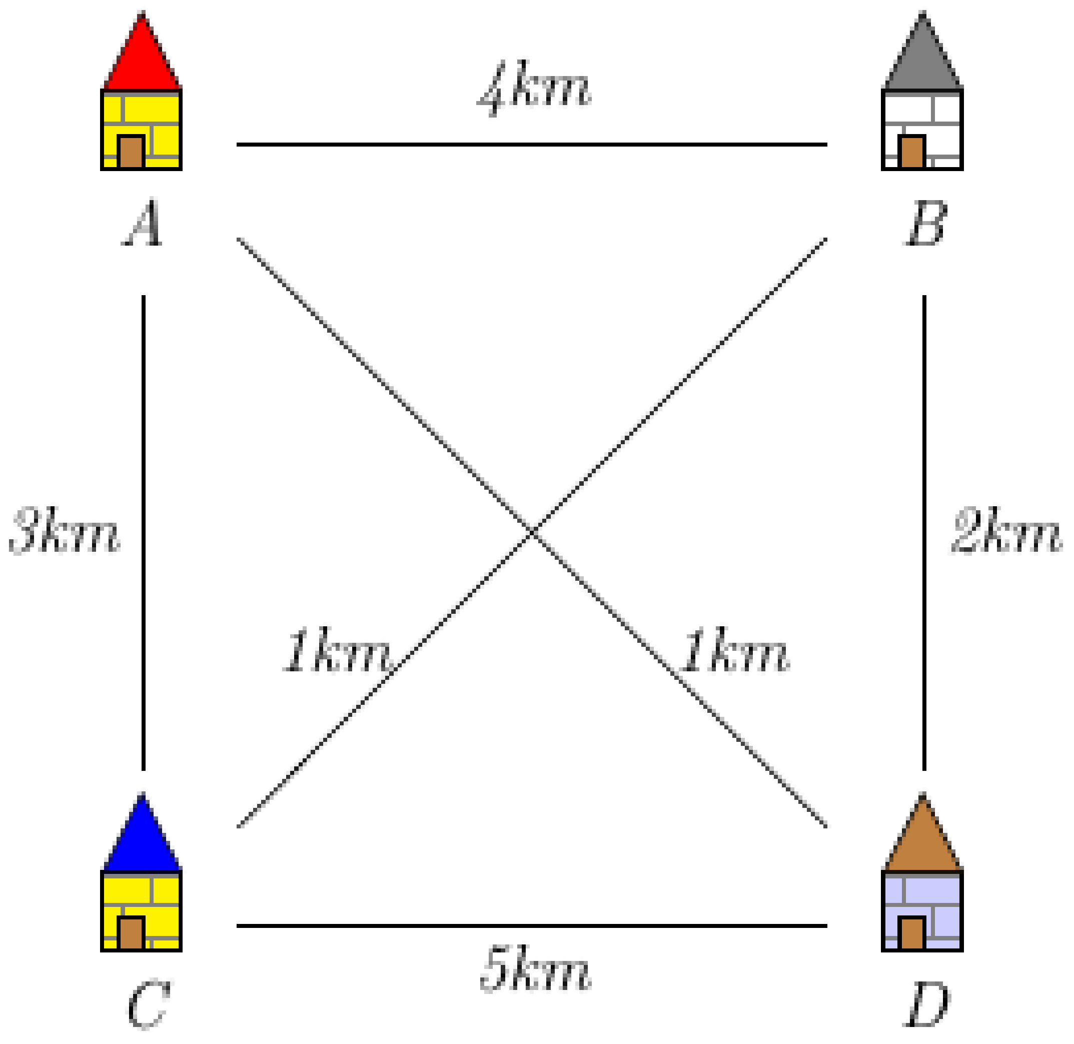
Constraints:

$$\sum_{i=1}^{n} w_i \cdot x_i <= W$$

Decision variables :

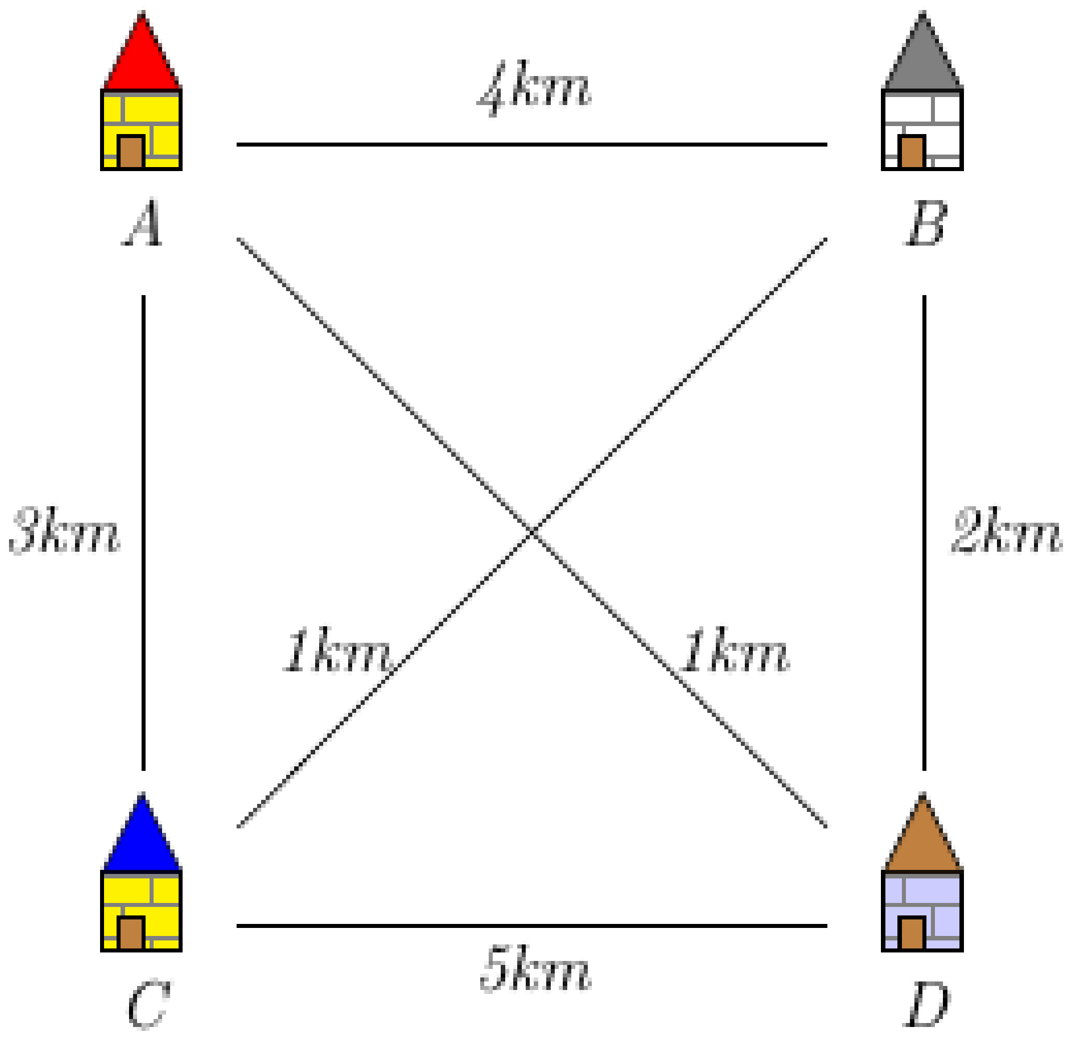
$$x_i \in \{0, 1\}$$

<u>TRAVELLING SALESMAN PROBLEM</u>



The travelling salesman problem (TSP) asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?"

TRAVELLING SALESMAN PROBLEM



$$Min\ Z = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} \cdot x_{ij}$$

Constraints:

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall\, j \in E$$

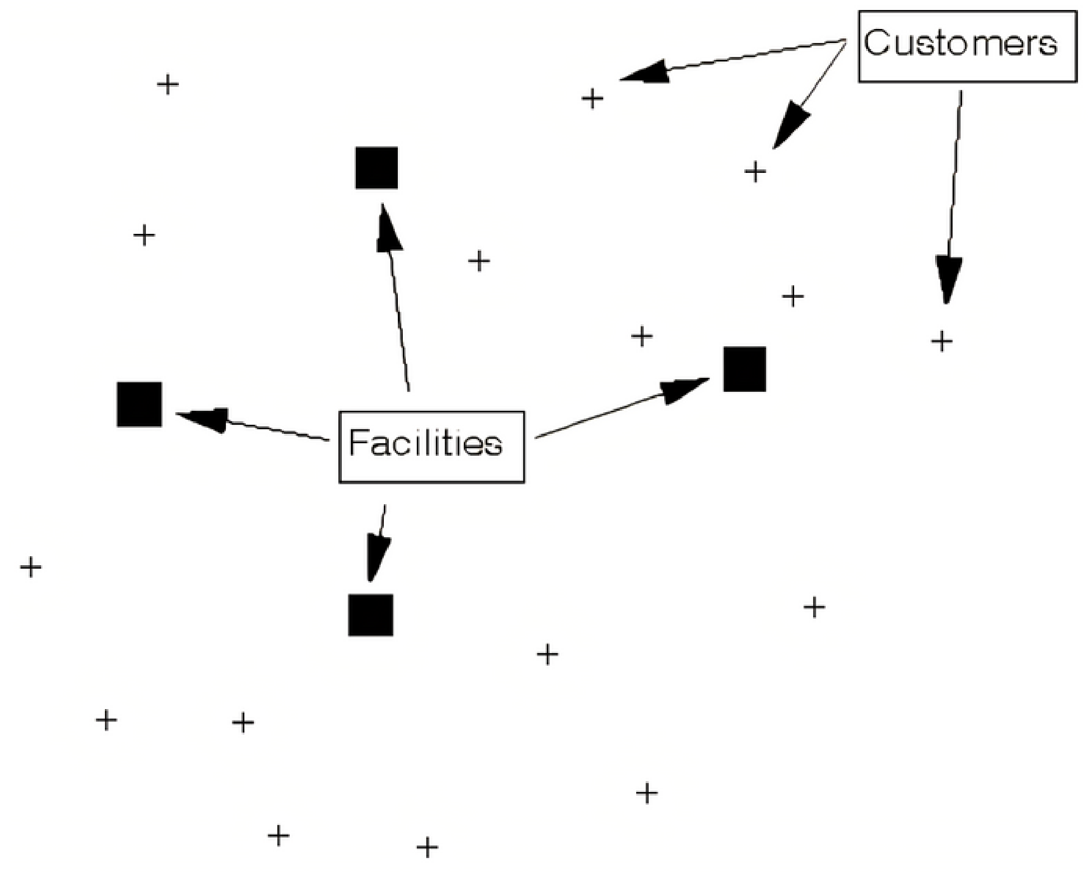
$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall\, i \in E$$

$$\sum_{i \in Q}\sum_{j \in \bar{Q}} x_{ij} \geq 1 \quad for\ Q \subseteq E$$

Decision variables :

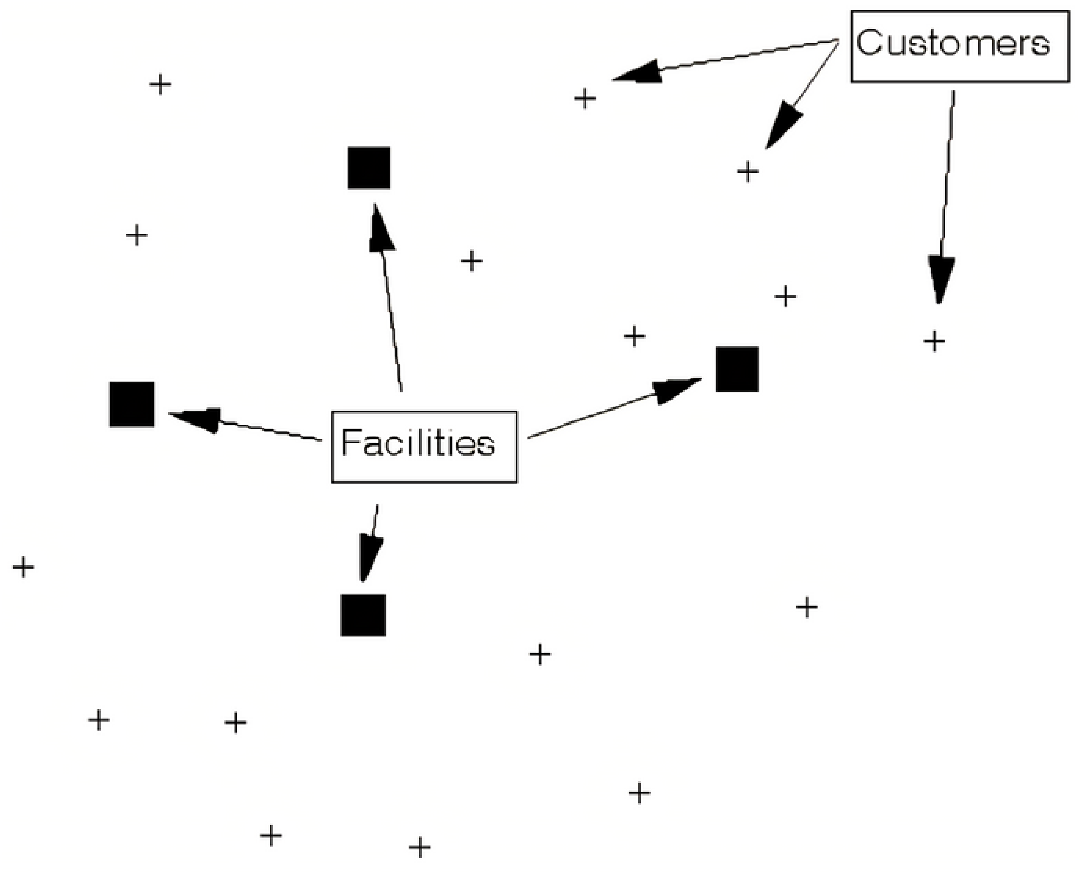
$$x_{ij} \in \{0, 1\}$$

FACILITY LOCATION PROBLEM



The objective of the problem is to select facility sites in order to minimize costs; these typically include a part which is proportional to the sum of the distances from the demand points to the servicing facilities, in addition to costs of opening them at the chosen sites.

CAPACITATED FACILITY LOCATION PROBLEM



$$\text{Min} \quad Z = \sum_{j=1}^{m} f_i \cdot y_j + \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} \cdot x_{ij}$$

Constraints:

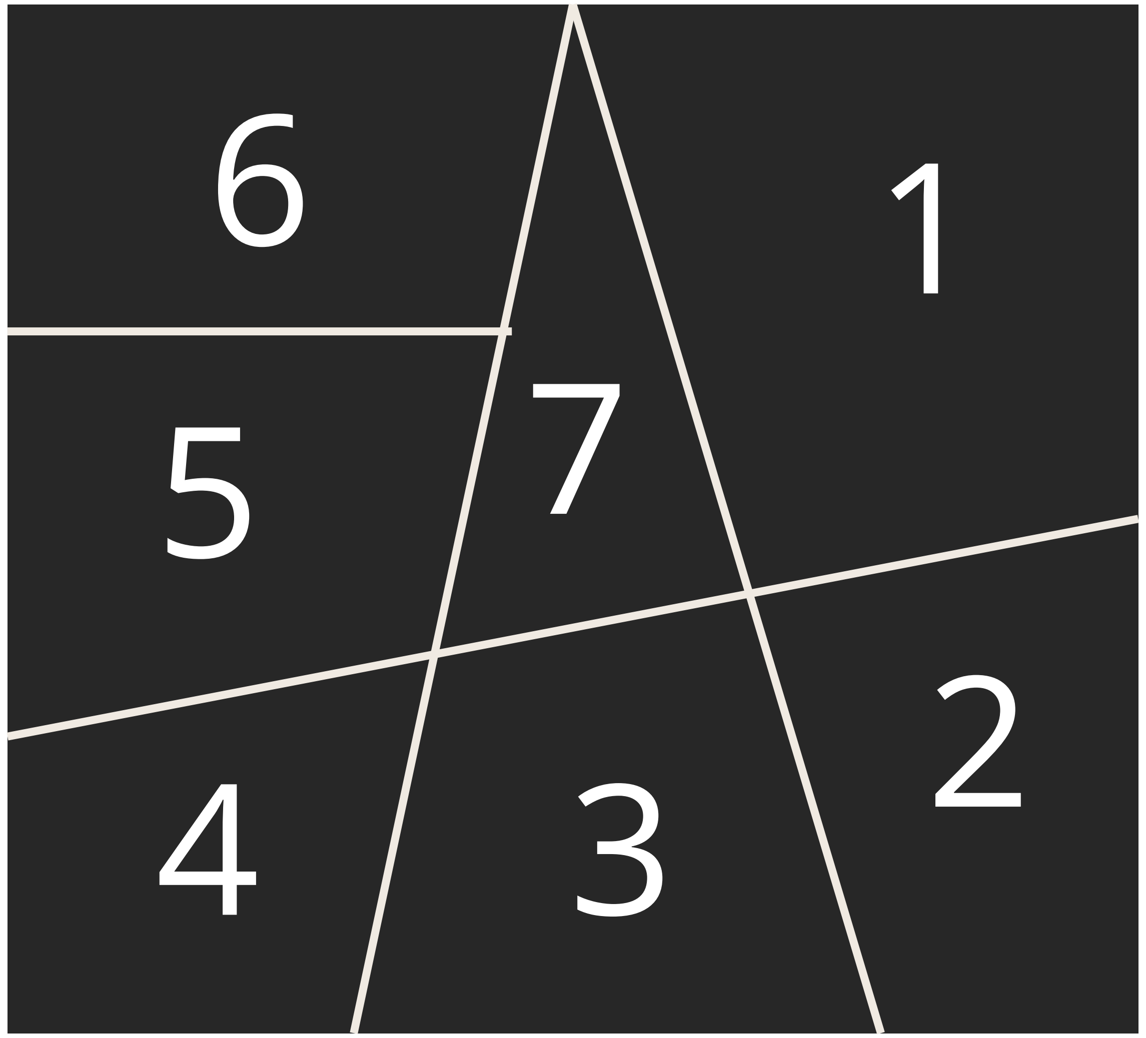
$$\sum_{j=1}^{m} x_{ij} = 1 \qquad for\ i = \ 1, ..., n$$

$$x_{ij} <= y_j \qquad for\ i = \ 1, ..., n,\ j = \ 1, ..., m$$

Decision variables :

$$y_i \in \{0, 1\}\ for\ i = \ 1, ..., n$$

$$x_{ij} \in \{0, 1\}\ for\ i = \ 1, ..., n,\ ; j = \ 1, ..., m$$

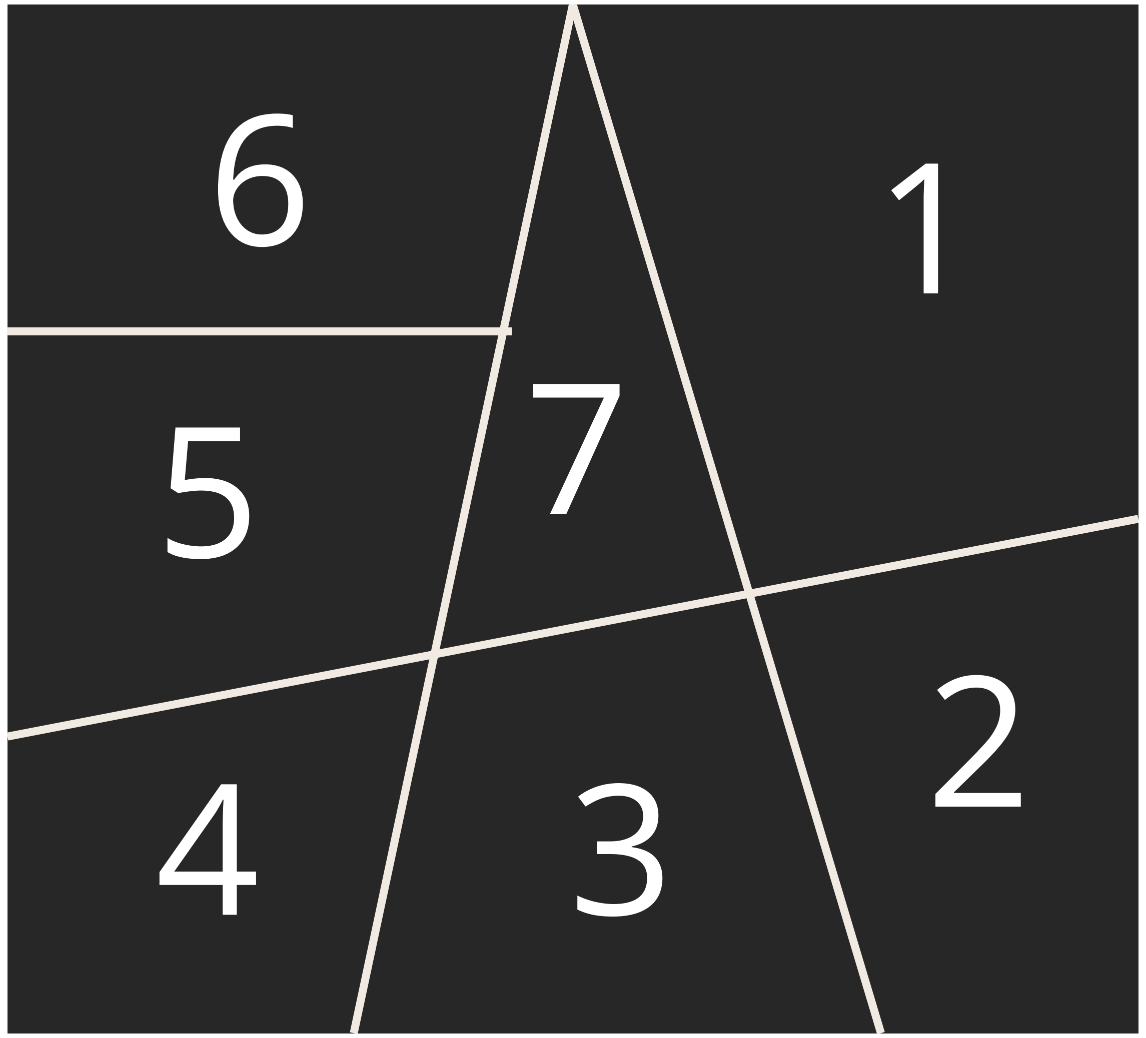


SET COVERING / PARTITIONNING/PACKING PROBLEMS

$$\begin{aligned} &\min CX \\ &AX \geq 1 \\ &X \in \{0,1\}^n \end{aligned}$$

$$\begin{aligned} &\min CX \\ &AX = 1 \\ &X \in \{0,1\}^n \end{aligned}$$

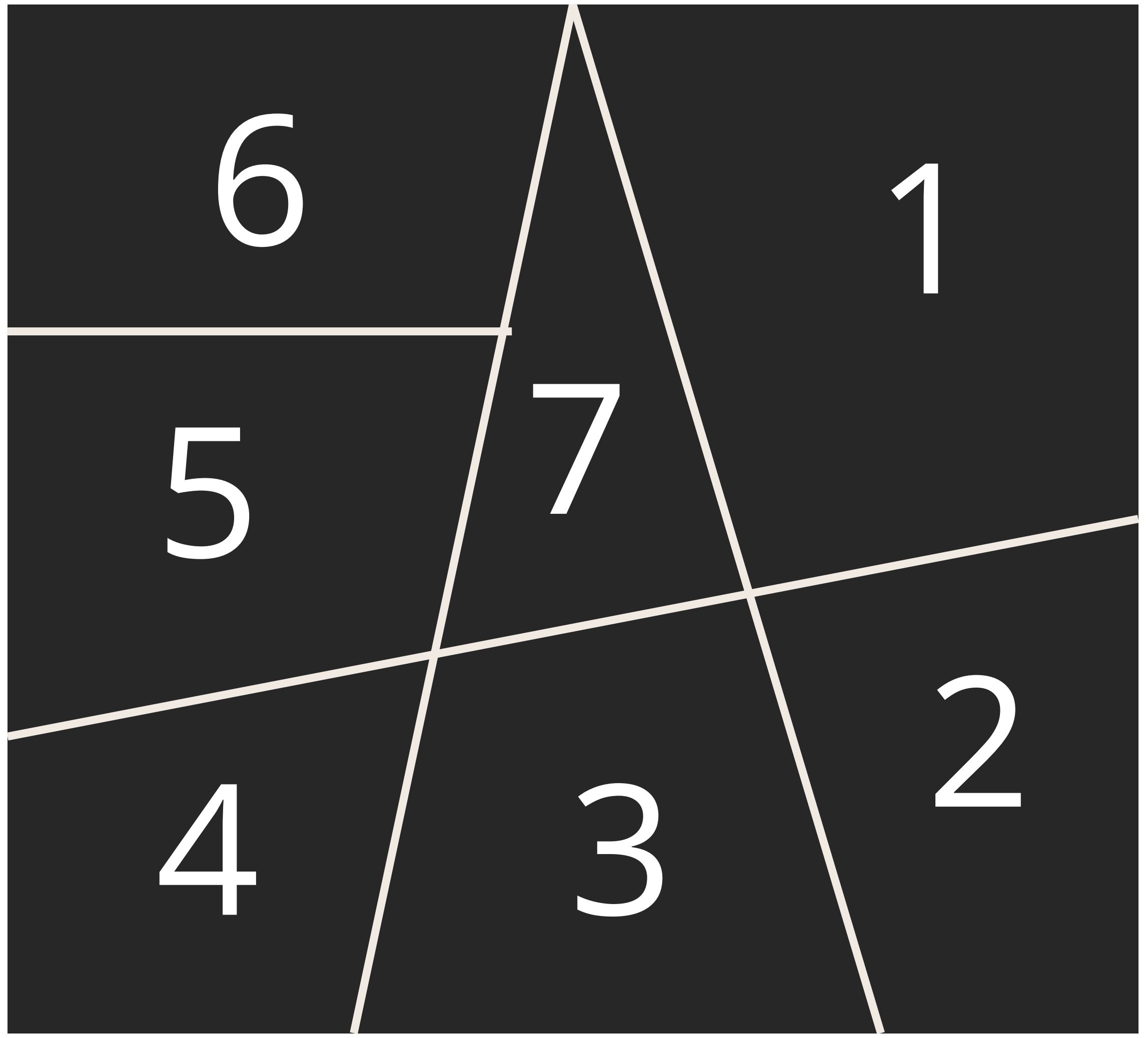
$$\begin{aligned} &\max CX \\ &AX \leq 1 \\ &X \in \{0,1\}^n \end{aligned}$$



SET COVERING PROBLEM

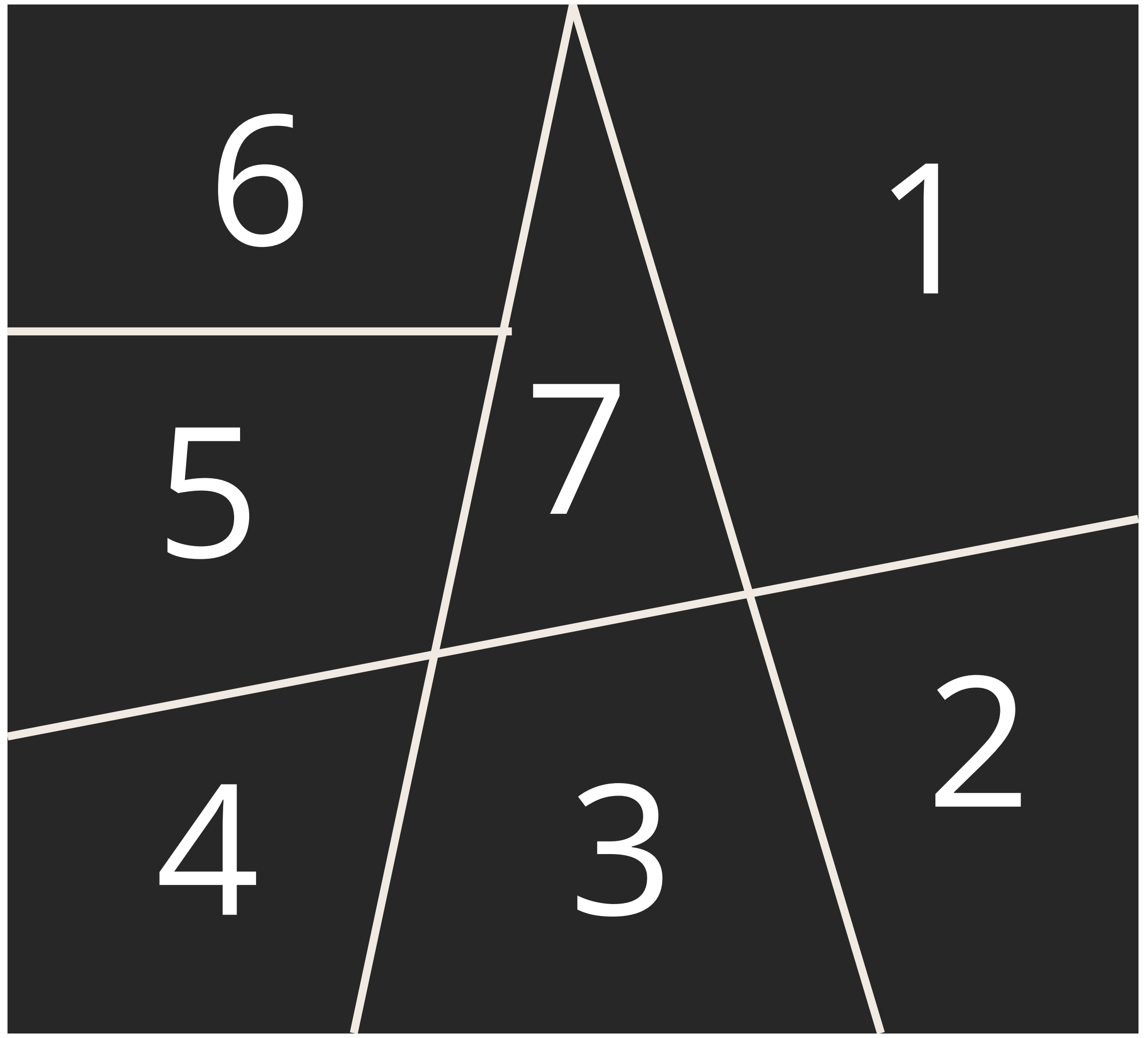
$$\min CX$$
$$AX \geq 1$$
$$X \in \{0,1\}^n$$

SET COVERING PROBLEM



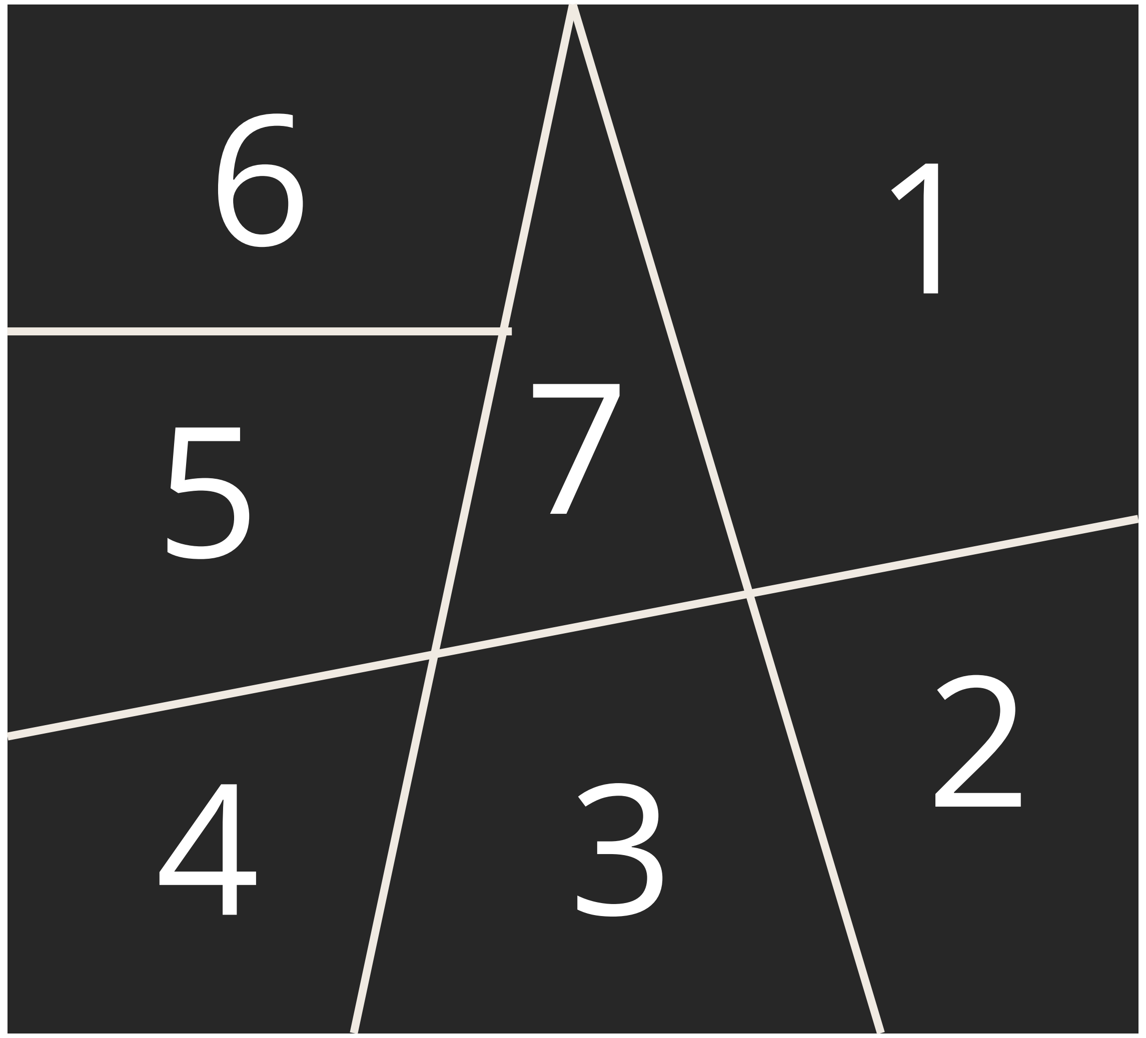
P1 = {1,2,7}	5
P2 = {2,4,6,7}	65
P3 = {3,5}	30
P4 = {4,6}	60
P5 = {1,3,5,6}	25
P6 = {7}	15

SET COVERING PROBLEM

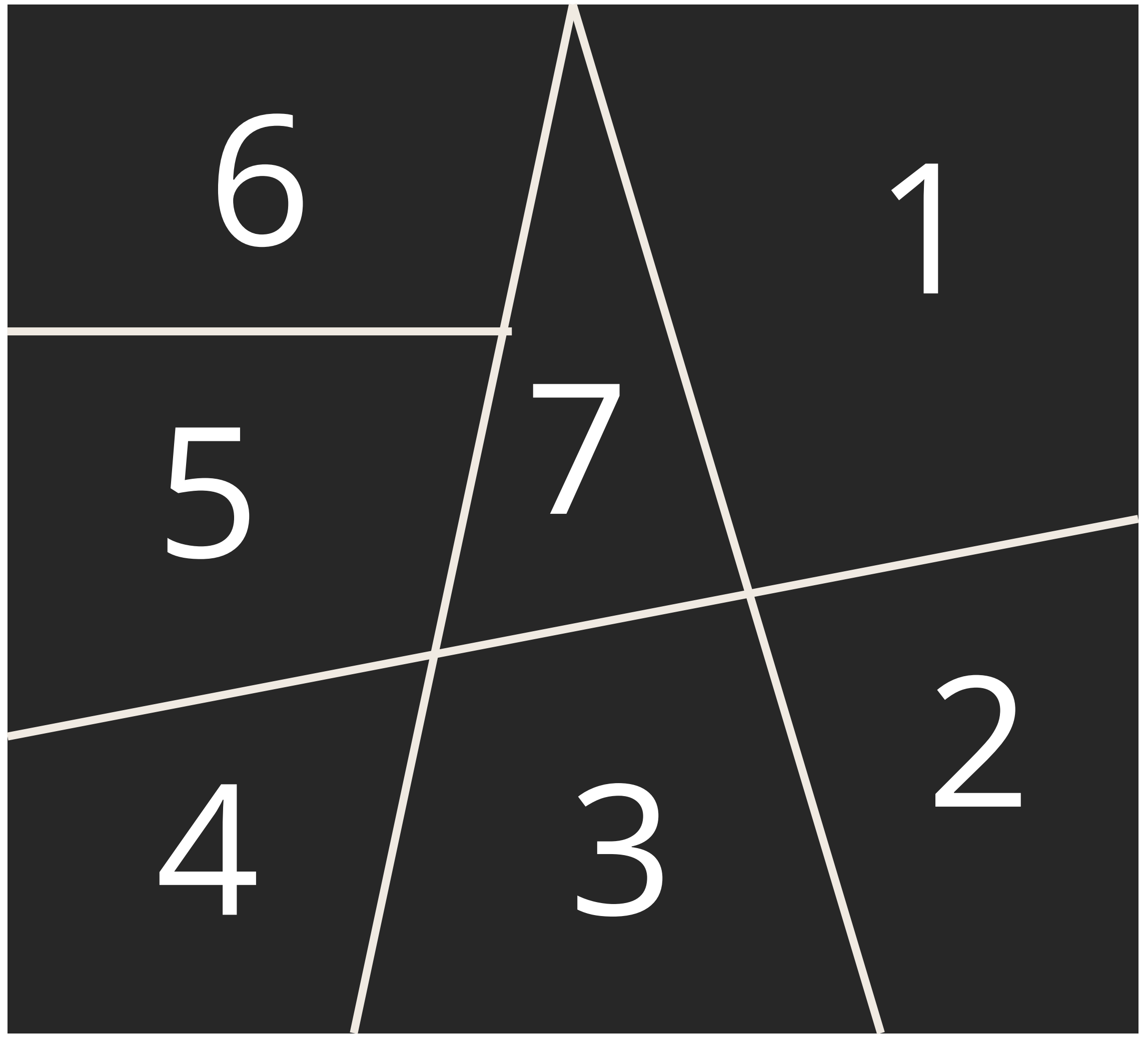


$P1 = \{1,2,7\}$	5
$P2 = \{2,4,6,7\}$	65
$P3 = \{3,5\}$	30
$P4 = \{4,6\}$	60
$P5 = \{1,3,5,6\}$	25
$P6 = \{7\}$	15

SET COVERING PROBLEM



P1 = {1,2,7}	5
P2 = {2,4,6,7}	65
P3 = {3,5}	30
P4 = {4,6}	60
P5 = {1,3,5,6}	25
P6 = {7}	15



SET COVERING / PARTITIONNING/PACKING PROBLEMS

$$\min CX \quad AX \geq 1 \quad X \in \{0,1\}^n$$

$$\min CX \quad AX = 1 \quad X \in \{0,1\}^n$$

$$\max CX \quad AX \leq 1 \quad X \in \{0,1\}^n$$

Python Libraries for Optimization

For Continuous Optimization

SciPy

SciPy is a scientific computation library that uses NumPy underneath. SciPy stands for Scientific Python. It provides more utility functions for optimization, stats and signal processing.

CVXPY

CVXPY is an open source Python-embedded modeling language for convex optimization problems. It lets you express your problem in a natural way that follows the math, rather than in the restrictive standard form required by solvers.

For Combinatory Optimization

PuLP

PuLP is one of many libraries in Python ecosystem for solving optimization problems, used for Linear programming optimization problems

NetworkX
Network Analysis in Python

NetworkX is a Python package for the creation, manipulation and the study of the structure, dynamics and functions of complex networks. It offers data structures for graphs, digraphs, multigraphs, and multidigraphs as well as standard graph algorithms.

Example 1 : SciPy Library

A company manufactures two products (G and H) and has two resources (X and Y) available.
Each unit of product G requires 3 units of resource X and 8 units of resource Y
Each unit of product H requires 6 units of resource X and 4 units of resource Y
The company has a maximum of 30 units of resource X and 44 units of resource Y available.
The company wants to maximize profits:
$100 per unit of product G
$125 per unit of product H

Example 1 : SciPy Library

Linear programming is an optimization method for solving systems of linear constraints and objectives. This problem is mathematically expressed as:

Maximize 100G + 125H

Subject to:

3G+6H<=30

8G+4H<=44

G,H>=0

where G and H are the number of units of products to be produced, respectively.

Example 1 : SciPy Library

The following code shows how to use linear programming to solve this problem in scipy.optimize with the linprog function.

```
from scipy.optimize import linprog
c = [-100, -125]
A = [[3, 8], [6, 4]]
b = [30, 44]
bound = (0, None)
res = linprog(c, A_ub=A, b_ub=b, bounds=[bound, bound], method='highs')

#print solution
print(f'Optimal solution: G = {res.x[0]:.2f}, H = {res.x[1]:.2f}')
print(f'Maximum profit = $ {-res.fun:.2f}')
```

Example 2 : CVXPY Library

We can also use CVXPY library for convex optimization problems

```
import cvxpy as cp
# Define the variable
x = cp.Variable()
# Define the objective
objective = cp.Minimize(x**2)
# Define the problem
problem = cp.Problem(objective)
# Solve the problem
problem.solve()
```

In this example, we used the CVXPY library to set up and solve a convex optimization problem. CVXPY simplifies the process of specifying the objective and constraints while ensuring that the problem is convex, which is essential for optimality. The library abstracts many of the complexities involved in convex optimization, making it easier to formulate and solve a wide range of convex problems.

Example 3 : PuLP Library

```python
# Example: Using PuLP for linear programming
from pulp import LpProblem, LpVariable, lpSum, LpMaximize

# Create a linear programming problem
prob = LpProblem("Example_LP", LpMaximize)
# Define decision variables
x = LpVariable("x", lowBound=0)
y = LpVariable("y", lowBound=0)
# Define the objective function
prob += 2 * x + 3 * y, "Objective"
# Add a constraint
prob += x + y <= 4, "Constraint 1"
# Solve the linear programming problem
prob.solve()
```

In this example, we used the PuLP library to create and solve a simple linear programming problem. We defined decision variables, specified an objective function to maximize, and added a constraint. PuLP abstracts many of the complexities involved in linear programming, making it user-friendly for formulating and solving various linear programming problems.

Example 4 : NetworkXLibrary

```
# Example: Using NetworkX for network flow problem
import networkx as nx

# Create a directed graph
G = nx.DiGraph()
# Add edges with capacities
G.add_edge("source", "A", capacity=3)
G.add_edge("source", "B", capacity=2)
G.add_edge("A", "C", capacity=2)
G.add_edge("B", "C", capacity=1)
G.add_edge("C", "sink", capacity=3)
# Compute maximum flow
flow_value, flow_dict = nx.maximum_flow(G, "source", "sink")
```

In this example, we used the NetworkX library to model and solve a network flow problem. We created a directed graph, added edges with capacities, and computed the maximum flow from the source node to the sink node. NetworkX is a valuable tool for solving various network-related problems, including network flow and connectivity analysis.

CONCLUSION