# Course: MSc DS

## Optimisation

**Module**: **4**

**Learning Objectives:**

1. Understand Linear Programming Basics
2. Master LP Problem Formulation
3. Interpret Solution Spaces
4. Implement LP in Software
5. Conduct Sensitivity Analysis
6. Apply LP to Real-world Scenarios

**Structure:**

## 4.1 Introduction to Linear Programming: Concepts & Applications

Linear Programming (LP) is a mathematical methodology used to determine the best possible outcome or solution from a given set of resources or inputs. Essentially, it's about optimising (either maximising or minimising) a linear objective function subject to linear equality and inequality constraints.

Historically, LP's foundations trace back to the work of Leonid Kantorovich during World War II. He aimed to find the most efficient methods to allocate resources in the Leningrad region during the war. However, the real momentum for LP came in 1947 with George Dantzig's development of the simplex method, which provided a systematic way to solve linear programming problems.

### 4.1.1 Importance in Optimisation and Decision-making:

Linear Programming is paramount in optimisation and decision-making because it provides a structured framework to make decisions in a world of limited resources. Whether you're trying to maximise profit, minimise costs, or allocate resources in the most efficient manner, LP can be applied to a wide range of strategic, tactical, and operational problems.

### 4.1.2 Basic Terminologies in LP:

1. **Decision Variables**: These are the variables that decide the output. For example, in a business model, they could be the amount of each product to produce.
2. **Objective Function**: This is the function that needs to be optimised. In most cases, this could be profit maximisation or cost minimisation.
3. **Constraints**: These are the restrictions or limitations on the decision variables. They could be in the form of limited resources like manpower, material, money, and time.

### 4.1.3 Real-world Applications of Linear Programming:

1. **Supply Chain Optimisation**: LP helps in determining the optimal way to fulfil customer demands given constraints like storage capacity, manpower, and transportation.
2. **Transportation and Logistics**: Determines the best routes for delivery to minimise costs or time. This could involve selecting the most efficient routes for a fleet of delivery

trucks or determining the best shipment method for goods.

3. **Manufacturing Processes**: Determines the optimal production mix, considering the constraints in raw materials, labour hours, machine hours, etc. It can also be used for job scheduling in a production house.
4. **Finance and Investment Strategies**:LP can be used to create an ideal portfolio by choosing a combination of investments that either maximises return for a given level of risk or minimises risk for a given level of return.

**4.2 Problem Formulation in Linear Programming**

**1. Steps in LP Problem Formulation**

**a. Understanding the Problem** Before diving into mathematics, it's crucial to have a clear and comprehensive understanding of the problem at hand. This step requires:

- Recognising the objectives and goals of the problem.
- Identifying the resources and constraints.
- Ascertaining potential challenges or variables that can influence the problem.

**b. Defining the Decision Variables** These are the unknowns or the quantities we want to determine. For instance, in a production problem, the decision variables might be the quantity of each product to produce.

- Ensure each variable has a clear and unique definition.
- The number of decision variables will often shape the complexity of the problem.

**c. Establishing the Objective Function** This is the function we aim to optimise (either maximise or minimise). In a business context, examples include profit maximisation or cost minimisation.

- The function should be linear in terms of the decision variables.
- It combines the variables in a weighted manner, reflecting their impact on the overall objective.

**d. Laying out the Constraints** Constraints are the restrictions or limitations on the decision variables. They can arise from limited resources, time constraints, or other real-world restrictions.

- Constraints should also be linear in nature.

- They define the feasible region within which solutions can exist.

**2. Types of Linear Programming Problems**

**a. Pure LP versus Mixed LP**

- **Pure LP**: All decision variables are continuous, meaning they can take on any real number values within a feasible range.
- **Mixed LP (MILP)**: This involves both continuous and integer decision variables. MILP problems can be more complex due to the integer constraints.

**b. Use of Fractional Linear Programming**- The objective function in these issues is the ratio of two linear functions. Despite the presence of a ratio, with certain transformations, these problems can still be tackled using linear programming techniques.

**3. Common Mistakes in Problem Formulation and How to Avoid Them**

**a. Overlooking Important Constraints**

- Ensure a comprehensive understanding of the real-world scenario the model represents.

- Collaboration with experts in the domain can assist in identifying overlooked constraints.

**b. Not Properly Defining Decision Variables**

- Each decision variable should have a clear, unambiguous definition.
- Avoid using a variable for multiple purposes.

**c. Incorrectly Formulating the Objective Function**

- Ensure that all relevant components are included.
- Verify that the weights or coefficients represent the actual contribution of each decision variable to the objective.

**d. Forgetting Non-Negativity Constraints**

- LP decision variables are typically non-negative. This should be explicitly stated unless the context suggests otherwise.

**e. Not Validating the Model with Real-World Data**

- After formulating, use real-world data to test and validate the model.
- Adjustments might be needed if predictions or solutions do not align with actual outcomes.

## 4.3 Solution Spaces in Linear Programming: Feasibility and Optimality

Linear Programming (LP) is a powerful optimisation technique for solving problems where resources are limited, and there is a linear relationship between variables. At its core, LP issues frequently aim to optimise or decrease a linear objective function while considering several linear constraints.

- **Graphical Representation of LP Problems**:
    - o When dealing with two variables, the coordinate plane can represent LP problems graphically.
    - o The objective function can be represented as a line on this plane, and its movement (while remaining parallel to its original position) corresponds to different objective values.
    - o Constraints are represented as half-planes (or lines, in the case of equality constraints). These divide the coordinate plane into regions where the constraint is satisfied or violated.

- **Identifying Feasible Regions**:
  - The feasible region is the intersection of the half-planes (or lines) that represent the constraints. It contains all possible solutions that satisfy all the constraints simultaneously.
  - In a two-variable LP, this feasible region can take the form of a polygon, a single point, a line segment, or it may not exist at all.
  - The vertices or corner points of this feasible region (if it's a polygon) are of special interest because, for a linear objective function, the optimal solution will always lie at one of these vertices.
- **The Concept of Feasibility**:
  - A solution is termed 'feasible' if it satisfies all the given constraints of the LP problem.
  - It's crucial to differentiate between a solution that's merely feasible and one that's optimal. While the former just satisfies the constraints, the latter not only

satisfies the constraints but also provides the best possible objective value.

- **Infeasible Solutions and Their Implications**:
  - If no solution satisfies all the constraints simultaneously, the problem is termed 'infeasible'.
  - Infeasibility can arise from contradictory constraints or from constraints that are too restrictive.
  - Identifying infeasibility is critical, as it indicates that there's an inherent conflict in the problem's constraints or that the problem itself might need a re-evaluation.
- **Boundary and Interior Points**:

  Within the context of the feasible region, there are boundary and interior points.
  - Boundary Points: These lie on the edge or boundary of the feasible region. For a two-variable LP, the optimal solution often lies on these boundary points, specifically at the vertices.

    - Interior Points: These are the points that lie inside the feasible region but not on its boundary. For LP problems, while these points are feasible, they are not optimal.

**4.3.1 The Concept of Optimality**

In optimisation and, specifically, Linear Programming (LP), 'optimality' denotes the best possible solution among a set of feasible solutions. This "best" solution yields the maximum (or minimum) value of the objective function. The concept is fundamental as it forms the endpoint for most optimisation problems.

- **The Role of Extreme Points (Vertices) in LP**:
    - The fascinating property of LP problems is that, given a polygonal feasible region, the optimal solution will always lie at an extreme point or vertex.
    - The objective function and restrictions are linear, which is the cause of this. The best result can be found by going from one vertex to another along the boundaries

of the viable region.

  - This characteristic simplifies the solution process, especially for algorithms like the Simplex method which work by traversing from one vertex to another in search of the optimal solution.

- **Multiple Optimal Solutions and Degeneracy**:
  - In some LP scenarios, more than one solution can yield the same optimal value for the objective function. When this happens, the objective function is parallel to one of the edges of the feasible region.
  - Degeneracy occurs when a solution, based on the constraints, lies exactly on a vertex. However, due to the algorithm's design or due to precision issues, the method may not immediately recognise it as such. Degeneracy can lead to potential computational challenges or cycling in certain algorithms like the Simplex.

- **Sensitivity Analysis**:

  - Once an optimal solution is found, it's essential to understand how changes in the problem's parameters will affect this solution. This is the crux of sensitivity analysis.
  - For LP, sensitivity analysis can reveal how changes in the coefficients of the objective function or the right-hand side values of the constraints impact the optimal solution's value and position.
- **Understanding the Impact of Changing Coefficients**:
  - Altering coefficients can shift the position and slope of constraints or the objective function. This can, in turn, move the optimal solution to a different vertex or change the feasible region altogether.
  - For instance, if a constraint becomes less stringent, the feasible region might expand, possibly leading to a different optimal solution.
- **Practical Implications for Business Decisions**:
  - LP's geometric interpretation and sensitivity analysis

play a pivotal role in real-world decision-making.

- By understanding the robustness of an optimal solution (how much parameter changes it can tolerate before it's no longer optimal), businesses can make informed decisions under uncertainty.
- Additionally, sensitivity analysis can help firms anticipate the potential impacts of changing conditions, allowing for better strategic planning.

## 4.4 Hands-on: Formulating Linear Programming Problems in Excel and Python

### Setting Up an LP Problem in Excel

Linear Programming (LP) problems involve allocating limited resources among several competing activities in an optimal way. Excel's Solver Add-in provides a user-friendly interface to define and solve LP problems. To set up an LP problem:

1. **Define Decision Variables**: Decide what your decision variables are. These are typically what you are trying to determine. For example, in a production problem, it might

be the number of units of each product to produce.

2. **Set Up Objective Function**: This is typically a formula in Excel that calculates the value you are trying to maximise or minimise (e.g., profit or cost). Use Excel formulas to represent this based on your decision variables.
3. **Establish Constraints**: These are the limitations or restrictions on the decision variables. For instance, constraints can represent limits on available resources or minimum/maximum production levels. Again, use Excel formulas to represent these constraints.

### 4.4.1 Using the Solver Add-in: A Step-by-Step Guide

Solver is an Excel Add-in that allows users to find optimal solutions to decision problems. To utilise it:

1. **Enable Solver**:
    - Go to the File menu, choose Options, and then Add-Ins.
    - In the Add-Ins available list, select Solver Add-in and click on Go.

- Check Solver Add-in and click OK.

2. **Open Solver**:
   - Go to the Data tab and select Solver from the Analysis group.
3. **Configure Solver**:
   - Set Objective: Select the cell which contains the formula of the objective function.
   - Equal to: Choose either Max or Min based on whether you want to maximise or minimise the objective function.
   - By Changing Variable Cells: Select the cells which represent the decision variables.
   - Add Constraints: Click on 'Add' to define the constraints.
4. **Run Solver**:
   - Click on "Solve" to let Solver find the optimal solution.
5. **Review & Save Solution**:
   - Once Solver finds a solution, you can either keep the

Solver solution or restore the original values. You can also generate reports to analyse the solution.

### 4.4.2 Understanding the Solver Reports

Solver provides three primary reports:

1. **Answer Report**:
   - Displays the original and final values of the objective function and decision variables.
   - Provides status of the solution, indicating if it’s optimal, feasible, or infeasible.
2. **Sensitivity Report**:
   - Reveals information on how the optimal solution is affected by changes in the objective function's coefficients and the right-hand sides of the constraints.
   - Displays shadow prices, which show how much the objective function value will change with a one-unit change in the right-hand side of a constraint.
3. **Limiting Report**:
   - Highlights how much you can change a coefficient

before the current solution point is no longer optimal.

### 4.4.3 Common Issues and Troubleshooting

1. **No Feasible Solution**: This implies the constraints are too restrictive. You should:
    - Review the constraints for possible errors.
    - Consider if all constraints are necessary or if some can be relaxed.
2. **Model Not Converging**: This could be due to non-linearity or other complexities. Try:
    - Simplifying the model.
    - Using different algorithms available in Solver.
3. **Incorrect Solution**: This might be due to errors in the Excel formula or model setup.
    - Double-check all formulas and constraints.
    - Ensure that all variables are correctly defined.
4. **Solver Not Available**: If Solver doesn't appear in the list of Excel Add-ins, you might need to:
    - Install the Solver Add-in through Excel options.

- Check if the add-in is activated.

### 4.4.4 Introduction to Python for Linear Programming

A mathematical technique known as linear programming (LP) is used to get the optimal result in a mathematical model whose needs are represented by linear connections.

**Essential Libraries for Linear Programming in Python**

1. **PuLP**:
   - **Overview**: PuLP is an open-source linear programming package that provides tools for describing and solving LP and Mixed Integer Programming (MIP) problems.
   - **Features**:
     - Intuitive Python-based modelling syntax.
     - Provides a range of solution techniques, both open-source and commercial.
     - Supports various problem types: Linear, Integer, and Mixed Integer problems.
2. **SciPy**:
   - **Overview**: While SciPy is a library for mathematics,

science, and engineering, it has a specific submodule **scipy. optimise** for optimisation, including LP.

- **Features**:
  - o **linprog** function for linear programming.
  - o Incorporates the Simplex method and other optimisation algorithms.
  - o Handles bounded and unbounded LP problems.

3. **Others**:

- **CVXPY**: A Python-embedded modelling language for convex optimisation problems. It allows you to express your problem in a natural way and solves with several backend solvers.
- **Gurobi and CPLEX**: These are commercial optimisation solvers. Both offer Python APIs to define and solve LP problems, and often they're known for their speed and robustness.

### 4.4.5 Defining and Solving an LP Problem in Python using PuLP

Example: Let's consider a simple production optimisation problem. Suppose a company produces two products (A & B). Producing each unit of A earns \$3 profit and B earns \$5 profit. Each unit of A requires 1 unit of raw material and 2 hours of work, while B requires 2 units of raw material and 1 hour of work. The company has 10 units of raw material and 12 hours of work available.

Objective: Maximise the profit.

Here's how you'd solve this using PuLP:

```
from pulp import LpMaximize, LpProblem, LpVariable

# Define the model

Model =

LpProblem(name="production-problem",sense=LpMaximize)

# Define the decision variables

x = LpVariable(name="A_units", lowBound=0)  # Number of units

of product A

y = LpVariable(name="B_units", lowBound=0)  # Number of units

of product B
```

```python
# Add the constraints

model += (x + 2*y <= 10, "raw_material_constraint")

model += (2*x + y <= 12, "work_hours_constraint")


# Define the objective function

model += 3*x + 5*y


# Solve the problem

model.solve()


# Print the results

print(f"Produce {x.varValue} units of product A.")

print(f"Produce {y.varValue} units of product B.")
```

In this manner, Python, paired with libraries like PuLP and SciPy, becomes a formidable tool in the domain of linear programming, enabling data scientists, operations researchers, and other professionals to model and solve real-world optimisation problems.

**4.5 Summary**

- ❖ Linear programming is a mathematical technique used to find the optimal way to allocate limited resources among competing activities to achieve a specific objective.
- ❖ An LP problem comprises decision variables (representing decisions to be made), an objective function (representing the goal, either to maximise or minimise), and constraints (conditions or limitations).
- ❖ Widely applied in various industries including manufacturing, transportation, finance, and logistics, LP helps in optimising operations and making effective decisions.
- ❖ Not all LP problems have solutions. A solution is feasible if it satisfies all constraints, with the feasible region being the set of all viable solutions.
- ❖ Among the feasible solutions, the one that best meets the objective function (maximise or minimise) is termed optimal.

This solution can often be found at the vertices or extreme points of the feasible region.

❖ LP problems can be formulated and solved using various software and programming languages, with Excel's Solver Add-in and Python's PuLP library being popular choices for many practitioners.

**4.6 Keywords**

- **Linear Programming (LP)**: The goal of the optimisation technique known as linear programming is to maximise or minimise a linear function under linear constraints. In a mathematical model whose requirements are represented by linear relationships, it is a method of obtaining the optimal result (such as highest profit or lowest cost).

---

- **Decision Variables**: Decision variables represent the unknowns in the problem which need to be determined. For instance, in a business model, they could represent quantities of products to be produced. In LP, these variables are what we adjust to optimise the objective function.

- **Objective Function**: The objective function is the primary criterion that needs to be optimised in a linear programming problem. It's a linear function of the decision variables. For instance, it might represent total profit or total cost.
- **Constraints**: Constraints are the restrictions or limitations on the decision variables. They could represent things like resource limitations (e.g., budget constraints, manpower limitations) or requirements that need to be satisfied (e.g., demand constraints). They define the feasible region within which the solution must lie.
- **Feasible Region**: In the geometric representation of LP, the feasible region is the set of all possible solutions that satisfy all the constraints of the problem. It's often visualised as a polygon (in 2D) or a polyhedron (in 3D) where each point within this shape is a potential solution that meets the set constraints.

- **Sensitivity Analysis**: This refers to the study of how changes in the coefficients of an LP problem (both in the objective function and constraints) affect the optimal solution. For example, how would a change in resource availability or cost coefficients alter the optimal solution or objective function value? Sensitivity analysis provides insight into the robustness of the solution and its susceptibility to changes in the problem parameters.

### 4.7 Self-Assessment Questions

1. How does linear programming differ from other optimisation methods? Provide a brief explanation of its unique characteristics.
2. What are the primary components required to formulate a linear programming problem? List and briefly describe each component.
3. Which of the following scenarios is most suitable for solving with linear programming?
    - Determining the best combination of ingredients for a

new snack mix given costs, availability, and nutritional requirements.

- Predicting the stock market prices for the next month.
- Analysing sentiment in customer reviews.
- Developing a new image recognition algorithm for identifying animals.

4. What role do extreme points or vertices play in determining the optimal solution in a linear programming problem? Why are they crucial in the context of feasibility and optimality?
5. Which tool or platform would you prefer for solving a moderately complex linear programming problem: Excel's Solver Add-in or Python with the PuLP library? Justify your answer by considering ease of use, scalability, and flexibility.

## 4.8 Case Study

**Title: Optimising Supply Chain for a Leading Japanese Electronics Company**

**Introduction:**

In 2018, a leading electronics manufacturer in Japan, "NihonTech," faced a significant challenge. With an increasing demand for its products globally, they struggled to optimise their supply chain, resulting in increased costs and delivery delays. A careful analysis revealed that multiple suppliers, inefficient warehouse operations, and lack of demand forecasting were the core issues.

**Background:**

Seeking a solution, NihonTech decided to harness the power of data science to improve its supply chain optimisation. They collaborated with a team of data scientists to gather data on every element of their supply chain, from raw materials to the final product delivery.

Using linear programming techniques, the team formulated an optimisation model that considered various constraints such as production capacity, warehouse storage limits, transportation logistics, and lead time for suppliers. This model aimed to minimise the total cost, including production, storage, and transportation, while ensuring timely delivery.

Additionally, the team developed a demand forecasting model using historical sales data and market trends. This model enabled NihonTech to predict the demand for its products in different regions, thus aiding in production planning.

The results were astounding. Within a year, NihonTech reduced its supply chain costs by 15%, improved its on-time delivery rate by 10%, and significantly reduced excess inventory in their warehouses. The optimisation not only led to financial gains but also increased customer satisfaction levels, giving NihonTech a competitive edge in the global electronics market.

**Questions:**

1. What were the primary challenges faced by NihonTech in their supply chain before the optimisation?
2. How did the application of linear programming help NihonTech in supply chain optimisation?
3. Discuss the significance of the demand forecasting model for NihonTech's production planning. How do you think it influenced the supply chain costs and delivery efficiency?

## 4.9 References

1. "Introduction to Linear Optimization" by Dimitris Bertsimas and John N. Tsitsiklis.
2. "Linear Programming: Foundations and Extensions" by Robert J. Vanderbei.
3. "Convex Optimization" by Stephen Boyd and Lieven Vandenberghe.
4. "Optimisation by Vector Space Methods" by David G. Luenberger.
5. "Linear and Nonlinear Programming" by David G. Luenberger and Yinyu Ye.