Course: MSc DS

Optimisation

Module: 5

Learning Objectives:

1. Grasp foundational principles of the Simplex Method.
2. Differentiate among Dual, Revised, and Primal-Dual Simplex variants.
3. Implement linear programming solutions in Excel and Python.
4. Understand core concepts of network analysis in optimization.
5. Interpret and visualise results from linear and network analyses.
6. Apply techniques to real-world data science challenges.

Structure:

5.1 Introduction to the Simplex Method

5.2 Variants of the Simplex Method: Dual Simplex, Revised Simplex, and Primal-Dual Simplex

5.3 Introduction to Network Analysis in Optimization

5.4 Interpreting Results from Network Analyses

5.5 Summary

5.6 Keywords

5.7 Self-Assessment Questions

5.8 Case Study

5.9 Reference

5.1 Introduction to the Simplex Method

The Simplex Method, a cornerstone in the study of linear programming, was developed by George Dantzig in 1947. It was born out of the necessity of planning and decision-making processes during World War II, specifically for military logistics and planning. Over the decades, the method has been refined and enhanced, but its core principles remain intact. It stands out not just for its historical significance but also its resilience to changes in computational methods over time.

5.1.1 Key Principles and Terminologies:

- **Linear Programming (LP):** Linear programming (LP) is a mathematical method for maximisation or minimization of a linear objective function under the restrictions of linear equality and linear inequality.
- **Feasible Region:** A set of points that satisfy all the constraints of the linear programming problem.
- **Basis:** In the context of the Simplex Method, a basis refers to a subset of columns from the A matrix (constraint

coefficients) that form a linearly independent set. The corresponding solution is called a basic feasible solution.

- **Objective Function:** A function, linear in nature, that needs to be optimised (either maximised or minimised).
- **Vertices (or extreme points):** Corner points of the feasible region. The Simplex Method is based on the principle that the optimal solution, if it exists, will be at one of these vertices.

5.1.2 Basic Steps in the Simplex Algorithm:

1. **Initialization:** Begin at a feasible corner point (vertex) of the solution space.
2. **Iteration:** Move from one vertex to an adjacent one in such a way that the objective function's value improves (increases for a maximisation problem and decreases for a minimization problem).
3. **Termination:** Stop when no adjacent vertex can improve the objective function. At this point, the current solution is optimal.

5.1.3 Geometric Interpretation and Visualization:

The Simplex Method can be visualised geometrically, especially in two-dimensional and three-dimensional problems.

- In a **two-dimensional** linear programming problem, the feasible region will be a polygon, and its vertices will be the potential solutions that the Simplex Method evaluates.
- In **three dimensions**, this region transforms into a polyhedron, and its corner points (vertices) again serve as potential solutions.

This geometric interpretation is beneficial for grasping the iterative nature of the Simplex Method. However, it's worth noting that in real-world scenarios, especially in the domain of data science, LP problems can have thousands or even millions of variables, making direct visualisation impossible. In such cases, the mathematical representation and computational approach become crucial.

5.1.4 Benefits and Limitations:

Benefits:

- **Efficiency:** For many problems, especially those with a large number of constraints and variables, the Simplex Method is very efficient compared to alternative optimization techniques.
- **Versatility:** The method can handle both bounded and unbounded feasible regions, making it applicable to a wide array of problems in diverse fields.
- **Robustness:** Even though it's an iterative method, the Simplex has demonstrated a high degree of stability and robustness in real-world applications.

Limitations:

- **Cycling:** In rare cases, the Simplex Method can get caught in a cycle, where it keeps revisiting the same vertices without finding an optimal solution. Various pivot rules have been proposed to mitigate this issue.
- **Linear Restrictions:** The Simplex Method is specifically designed for linear programming problems. It cannot handle

nonlinear objective functions or constraints directly.

- **Large-scale Problems:** While the Simplex is efficient, other interior point methods might outperform it for some large-scale LP problems.

5.2 Variants of the Simplex Method: Dual Simplex, Revised Simplex, and Primal-Dual Simplex

The Dual Simplex method is a specialised algorithm used to solve linear programming problems when the primal solution is infeasible. Unlike the primal simplex that seeks feasible solutions from an optimal point, the dual simplex seeks optimal solutions from an infeasible point. It can be visualised as operating on the dual problem while manoeuvring within the primal infeasibility.

- Key Characteristics:
    - Useful for problems where initial solutions violate some constraints.
    - It iteratively modifies the basic feasible solution until an optimal solution is obtained.
- Applications:

    - Situations where starting from an infeasible solution is more advantageous.
    - Real-world problems like resource allocation where constraint adjustments are necessary.

5.2.1 The Revised Simplex: Improving Computational Efficiency

The Revised Simplex method is an optimised version of the original simplex method. Its primary focus is to improve computational efficiency by updating only relevant parts of the inverse matrix, rather than computing it entirely in every iteration.

- Key Characteristics:
    - Reduces computational effort by taking advantage of the sparsity in constraint matrices.
    - Avoids the direct inversion of matrices by using systematic updates.
- Applications:
    - Problems with a large number of constraints and variables, where standard simplex becomes computationally expensive.

    - Dynamic environments where the linear program might change frequently and can be updated incrementally.

5.2.2 Primal-Dual Simplex: Bridging the Gap between Primal and Dual Problems

The Primal-Dual Simplex method is an amalgamation of the primal and dual simplex algorithms. It is a methodology that's instrumental in resolving problems where neither a feasible primal nor dual solution is readily available. It switches between primal and dual spaces to find an optimal solution.

- Key Characteristics:
    - Utilises information from both the primal and the dual problem, providing a comprehensive approach.
    - When primal infeasibility is detected, it pivots in the dual space and vice versa.
- Applications:
    - Problems where both primal and dual information is beneficial for deriving optimal solutions.

- Environments where the initial setup doesn't provide an obvious starting point for either the primal or dual simplex.

5.2.3 Comparing the Variants: Use Cases and Scenarios

- Dual Simplex:
  - **Use Cases**: Problems with infeasible starting solutions, particularly when some constraints are temporarily violated.
  - **Scenarios**: Resource allocation tasks, especially where adjustments are needed due to sudden changes in constraints.
- Revised Simplex:
  - **Use Cases**: Situations demanding high computational efficiency or problems with large-scale dimensions.
  - **Scenarios**: Dynamic industrial processes, transportation, and logistics optimization where frequent updates to the linear program are commonplace.

- Primal-Dual Simplex:
    - **Use Cases**: Complex problems where it's challenging to find a starting point in either the primal or the dual space.
    - **Scenarios**: Financial optimization problems, multi-objective optimization scenarios, or multi-criteria decision-making problems.

5.3 Introduction to Network Analysis in Optimization

Network analysis in optimization provides a structured approach to tackling problems where entities and their interactions can be represented by graphs. It is a cornerstone of many real-world applications ranging from supply chain management to telecommunications and beyond. By understanding and utilising the foundations of network analysis, data scientists can address complex problems in a structured and efficient manner.

5.3.1 Foundations of Network Analysis

Network analysis is rooted in graph theory, a branch of discrete mathematics that studies networks of interconnected nodes and edges. The fundamental principles include:

- **Graph**: A collection of nodes (or vertices) and edges (or links) that connect pairs of nodes.
- **Directed Graph (Digraph)**: Where each edge has a direction, indicating a one-way relationship between two nodes.
- **Undirected Graph**: Edges don’t have a direction, indicating a mutual relationship.
- **Weighted Graph**: Edges have associated weights which can represent distances, costs, or any quantifiable relationship between nodes.
- **Connected Graph**: It is the path between every pair of nodes.

5.3.2 Graph Theory Basics: Nodes, Edges, and Paths

The rudimentary components of graph theory crucial for optimization include:

- **Nodes (or Vertices)**: Fundamental entities in a graph. They can represent cities, computers, individuals, etc. depending on the application.

- **Edges (or Links)**: Represent relationships or connections between nodes. In transportation, for example, an edge might represent a road connecting two cities.
- **Paths**: A sequence of nodes in which each adjacent pair is connected by an edge. A shortest path between two nodes is of special interest in optimization as it identifies the least-costly way to move from one node to another.
- **Cycles**: It is the path that begins and ends at the same node without retracing any edge.

5.3.3 Applications of Network Analysis in Real-world Scenarios

Network analysis permeates numerous real-world applications:

- **Supply Chain Optimization**: Determining the most efficient ways to manufacture and distribute products, considering various constraints and objectives.
- **Telecommunication Networks**: Designing efficient routing of data packets to ensure swift and reliable communication.
- **Social Network Analysis**: Understanding the structure, dynamics, and influential entities within social networks.

- **Transportation**: Finding optimal routes for delivery, public transit system design, and traffic flow optimization.

5.3.4 Network Flow Problems: Max Flow, Min Cut

Network flow problems are classic optimization challenges in which the goal is to maximise or minimise the flow through a network:

- **Max Flow Problem**: Given a source node and a sink (target) node, the objective is to push as much flow as possible through the network from the source to the sink without violating any capacity constraints on the edges.
  - Application: Telecommunication bandwidth, traffic management, and more.
- **Min Cut Problem**: Identifying the smallest set of edges which, when removed, will disconnect the source from the sink. The 'cut' capacity represents the maximum flow that can pass from source to sink.
  - Application: Vulnerability analysis in networks, system reliability, and more.

5.4 Interpreting Results from Network Analyses

Network analysis is a rapidly expanding field in data science, offering tools and techniques to understand and interpret complex relationships and structures in various types of data. Given the vastness of network data and the myriad techniques available for its analysis, it becomes pivotal to understand how to interpret the results.

5.4.1 Reading and Understanding Network Diagrams

- **Nodes and Edges:** At its core, a network consists of nodes (entities) and edges (relationships). A proper understanding of what these nodes and edges represent in your specific problem domain is crucial.
  - Example: In a social network, nodes might represent individuals, while edges could represent friendships or interactions.
- **Layouts:** The visual presentation of a network can heavily influence its interpretation. Different layouts can emphasise

different aspects of the data, such as clusters, central nodes, or peripheral nodes.

  - o Force-directed layouts often produce aesthetically pleasing diagrams where the placement of nodes reflects their relationships in the network.

- **Node Size and Color:** Often, nodes might be sized or coloured based on specific attributes or metrics. For example, in a citation network, a node might be sized by the number of citations a paper has received, indicating its influence in the field.
- **Edge Thickness and Color:** Similar to nodes, edges can be adjusted in terms of thickness or colour to represent various attributes. An edge might be thicker if there are multiple connections or relationships between two nodes.

5.4.2 Performance Metrics in Network Analysis

- **Degree Centrality:** This metric calculates the number of direct connections a node has. Nodes with high degree centrality are often pivotal in the network.

- **Betweenness Centrality:** This measures how closely a node is situated on pathways leading to other nodes. High betweenness centrality nodes serve as bridges and significantly affect the flow of information or resources.
- **Closeness Centrality:** This metric represents how close a node is to all other nodes in the network. Nodes with high closeness centrality can quickly interact with all other nodes.
- **Eigenvector Centrality:** This is a measure of a node's power within a network. It considers a node's indirect connections as well as those with its neighbours and other nodes.
- **Clustering Coefficient:** This metric captures the degree to which nodes in a network tend to cluster together. It's beneficial in identifying tightly-knit groups within a network.

5.4.3 Making Data-driven Decisions from Network Insights

- **Identify Key Players:** By analysing centrality measures, one can identify influential nodes in a network, which can be pivotal for strategies such as marketing campaigns or information dissemination.

- **Detecting Communities:** Network analyses can help identify clusters or communities within networks. These clusters can represent, for example, groups with similar tastes or interests.
- **Optimising Routes:** In logistics, transportation, or even telecommunication, identifying the shortest or most efficient path between nodes is critical. Network analyses can provide such insights.
- **Predictive Modelling:** Using network structures and attributes, predictive models can be developed to forecast trends, behaviours, or even node importance evolution over time.

5.5 Summary

- ❖ A widely-used algorithmic approach for solving linear programming problems, the Simplex Method optimises a linear objective function subjected to linear equality constraints. It iteratively moves along the edges of the feasible region defined by the constraints to find the optimal solution.

- Handles infeasible solutions in the primal problem by optimising the dual.Streamlines calculations using only necessary variables.Combines features of both primal and dual Simplex for greater efficiency.
- Practical solutions can be developed using tools such as Excel's Solver and Python libraries like SciPy. Both platforms allow model formulation, solution, and interpretation of linear programming problems using the Simplex method.
- An approach that uses graph theory principles to study and optimise interconnected systems. It analyses structures composed of nodes (entities) and edges (connections) to optimise specific objectives, such as shortest paths or maximum flows.
- Beyond deriving solutions, interpreting network analysis results involves understanding network diagrams, gauging performance metrics, and using the derived insights to make informed decisions.

- ❖ Using Excel and Python, one can visualise, analyse, and solve network-based problems. Python's NetworkX and PyGraphviz libraries, for instance, are powerful tools for analysing and visualising complex networks.

5.6 Keywords

- **Simplex Method**: The Simplex Method is an iterative mathematical algorithm used to solve linear programming problems. It operates by moving along the edges of the feasible region (defined by the constraints) to find the optimal solution, often visualised as navigating on a polyhedron in multi-dimensional space to find the highest (or lowest) point.
- **Dual Simplex**: The Dual Simplex method is a variation of the Simplex algorithm. It is particularly useful for problems that are infeasible at the start. While the Simplex method moves towards optimality from a feasible solution, the Dual Simplex moves towards feasibility from an optimal solution.
- **Network Analysis**: Network Analysis in optimization refers to

the study and analysis of networks to determine the most efficient way of achieving a particular objective, like the shortest path, maximum flow, etc. It employs concepts from graph theory and is used in various fields like transportation, telecommunications, and logistics.

- **Graph Theory**: Graph theory is a branch of mathematics that studies networks of interconnected nodes and edges. It provides the foundational structures (like nodes, edges, paths, cycles) and concepts (like connectivity, flow, cuts) for network analysis in optimization.
- **Linear Programming**: Linear programming (LP) is a method to achieve the best outcome, such as maximum profit or lowest cost, in a mathematical model whose requirements are represented by linear relationships. It involves three main components: objective function (what you want to maximise/minimise), decision variables (the variables that decide the outcome), and constraints (the limitations).
- **NetworkX**: NetworkX is a Python library used for the

creation, manipulation, and study of complex networks of nodes and edges. It provides tools to work with graph structures, algorithms like shortest path computations, and visualisation capabilities, making it a popular choice for network analysis in Python.

5.7 Self-Assessment Questions

1. How does the Revised Simplex method improve computational efficiency compared to the traditional Simplex method?
2. What are the fundamental differences between the Primal and Dual problems in the context of the Simplex method?
3. Which Python library would be most suitable for implementing the Simplex method to solve linear programming problems, and why?
4. What are the core components of Graph Theory that are essential for understanding network analysis in optimization?
5. How can one interpret results from a network analysis to

make data-driven decisions in real-world scenarios?

5.8 Case Study

Title: Optimization in E-Commerce Logistics in China

Introduction:

Alibaba, a leading e-commerce giant in China, has always been at the forefront of leveraging technology for business optimization. With the growing number of online shoppers and the vast geographical expanse of China, Alibaba faced challenges ensuring timely delivery, reducing transportation costs, and maintaining customer satisfaction. The company's extensive logistics network required an overhaul to stay ahead in the e-commerce race.

Challenge: One major issue Alibaba confronted was optimising its warehouse locations and determining the best route for the delivery of products. The vast inventory spread across multiple suppliers, and the diverse locations of customers made this a complex problem. Each wrong location or inefficient route translated into increased costs and decreased customer satisfaction.

Solution:

Using advanced linear programming and network analysis, Alibaba devised a solution. The company collated data from its numerous orders, identifying high-density order regions and frequently ordered product categories. By modelling this data, Alibaba identified strategic warehouse locations to ensure the shortest average distance to its customers. Moreover, by applying network analysis, they outlined the most efficient routes for their delivery vehicles, taking into consideration road conditions, traffic patterns, and peak shopping times.

Outcome:

As a result, Alibaba experienced a 15% reduction in transportation costs within a year. The average delivery time was reduced by 10%, leading to a surge in customer satisfaction. The optimization not only resulted in cost savings but also enabled Alibaba to promise and deliver faster shipments, a competitive advantage in the aggressive e-commerce market of China.

Questions:

1. What challenges did Alibaba face concerning its logistics and customer satisfaction?
2. How did linear programming and network analysis assist Alibaba in optimising its logistics operations?
3. Considering the outcome, how did the optimization influence Alibaba's competitive standing in the e-commerce market of China?

5.9 References

1. "Linear Programming and Network Flows" by Mokhtar S. Bazaraa, John J. Jarvis, and Hanif D. Sherali.
2. "Introduction to Linear Optimization" by Dimitris Bertsimas and John N. Tsitsiklis.
3. "Network Flows: Theory, Algorithms, and Applications" by Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin.
4. "Convex Optimization" by Stephen Boyd and Lieven Vandenberghe.
5. "Combinatorial Optimization: Algorithms and Complexity" by Christos H. Papadimitriou and Kenneth Steiglitz.