**D605 - Optimization**
**BRN1 Task 2: Solve an Optimization Problem**
Prepared by Shanay Murdock

---

# A. Identify Optimization Problem

The optimization problem in this scenario is a supply chain distribution problem where Amazon Air needs to minimize the total cost of transporting cargo through its hierarchical logistics network. The network has three tiers:

1. Two main hubs (Cincinnati/Northern Kentucky and Alliance Fort Worth

2. Three focus cities (Leipzig, Hyderabad, and San Bernardino)

3. Sixty-five fulfillment centers

The challenge is to determine optimal shipment quantities along all possible routes while satisfying demand at each fulfillment center and respecting capacity constraints at hubs and focus cities. This is fundamentally a **linear programming problem** since the object function (cost) and all constraints are linear functions of the decision variables (shipment quantities).

# B. Create Mathematical Representation of Optimization Problem

Below are the mathematical representations of the variables specific to this optimization problem:

1. Hub capacities
$$\sum_{j=1}^{3} x_{ij} + \sum_{k=1}^{65} y_{ik} \leq capacity,\ i = 1,\ 2$$

2. Quantity into focus cities
$$\sum_{i=1}^{2} x_{ij} \leq capacity,\ j = 1,\ 2,\ 3$$

3. Quantity out of focus cities
$$\sum_{k=1}^{65} z_{jk} = \sum_{i=1}^{2} x_{ij},\ j = 1,\ 2,\ 3$$

4. Center demand
$$\sum_{i=1}^{2} y_{ik} + \sum_{j=1}^{3} z_{jk} = requirement,\ k = 1,\ 2,\ ...65$$

## B1. Objective Function Expression

The objective function is to minimize the total transportation cost across Amazon's air logistics network. This factors in the cost of transportation from hubs to focus cities, cost of transportation from hubs to fulfillment centers, and cost of transportation from focus cities to fulfillment centers.

This can be mathematically expressed with the following equation:

$$min\left(\sum_{i=1}^{2}\sum_{j=1}^{3} c_{ij}x_{ij} + \sum_{i=1}^{2}\sum_{k=1}^{65} c_{ik}y_{ik} + \sum_{j=1}^{3}\sum_{k=1}^{65} c_{jk}z_{jk}\right)$$

Components of the equation:

- $c_{ij}$ is the cost per unit (ton) of shipping from hub *i* to focus city *j*
- $c_{ik}$ is the cost per unit (ton) of shipping from hub *i* to fulfillment center *k*
- $c_{jk}$ is the cost per unit (ton) of shipping from focus city *j* to fulfillment center *k*
- $x_{ij}$ is the quantity of cargo shipped from hub *i* to focus city *j*
- $y_{ik}$ is the quantity of cargo shipped from hub *i* to fulfillment center *k*
- $z_{jk}$ is the quantity of cargo shipped from focus city *j* to fulfillment center *k*

## B2. Optimization Constraints

1. Hub Capacity Constraints

   $$\sum_{j=1}^{3} x_{ij} + \sum_{k=1}^{65} y_{ik} \leq capacity,\ i = 1,\ 2$$

   Explanation: For each hub, the total outbound shipments cannot exceed its capacity.

2. Focus City Capacity Constraints (Inbound)

   $$\sum_{i=1}^{2} x_{ij} \leq capacity,\ j = 1,\ 2,\ 3$$

   Explanation: The total inbound shipments to each focus city cannot exceed its capacity..

3. Flow Balance at Focus Cities

   $$\sum_{k=1}^{65} z_{jk} = \sum_{i=1}^{2} x_{ij},\ j = 1,\ 2,\ 3$$

   Explanation: Each fulfillment center must receive exactly its required demand.

4. Fulfillment Center Demand Constraints
   $$\sum_{i=1}^{2} y_{ik} + \sum_{j=1}^{3} z_{jk} = requirement,\ k = 1,\ 2,\ ...65$$

   Explanation: The total amount of cargo sent to a center must meet the demand requirement by shipments from other cities.

5. Non-negativity Constraints
   $$x_{ij}, y_{ik}, z_{jk} \geq\ 0$$

   Explanation: All shipment quantities must be non-negative.

### B3. Identify Decision Variables

The decision variables represent the quantities of cargo (in tons) shipped between different locations and determine the most cost-effective shipping routes while meeting demand and staying within capacity constraints:

- $x_{ij}$: Quantity shopped from hub *i* to focus city *j*
  Where *i* ∈ {1, 2} (two hubs) and *j* ∈ {1, 2, 3} (three focus cities)
- $y_{ik}$: Quantity shipped direction from hub *i* to fulfillment center *k*
  Where Where *i* ∈ {1, 2} and *k* ∈ {1, 2, …, 65}
- $z_{jk}$: Quantity shipped from focus city *j* to fulfillment center *k*
  Where *j* ∈ {1, 2, 3} and *k* ∈ {1, 2, ..., 65}

Note: Some route combinations are disallowed due to distance constraints (marked as “NA” in the provided cost table), reducing the actual number of variables from the theoretical maximum.

## C. Approach for Solving Optimization

The approach to solving this Amazon Air optimization problem follows a systematic methodology that transforms the business problem into a mathematical model, applies computational algorithms, and produces actionable results. The overall approach consists of four integrated phases that work together to deliver an optimal solution.

First, the problem formulation phase involves translating the real-world logistics challenge into mathematical language. This means taking the network structure with its hubs, focus cities, and fulfillment centers, and representing it as a system of linear equations and inequalities. The transportation costs from the provided tables become coefficients in the objective function, while the capacity limits and demand requirements become constraints that bound the feasible solution space.

Second, the data preparation phase requires organizing all input parameters into structured formats suitable for computational processing. The demand data from Table 1 in the Amazon

Distribution Problem document needs to be indexed properly for each fulfillment center. The cost matrix from Table 2 of the same document must be structured to reflect allowed routes (excluding the “N/A” entries), and the capacity constraints need to be formatted as upper bounds for the optimization solver. This phase also involves validating data consistency and identifying any potential issues before computation begins.

Third, the solution phase applies linear programming techniques to find the optimal allocation of shipments across all routes. Since this is a continuous linear problem with approximately 189 variables and 73 constraints, the computational process will explore the solutions space systematically to identify the point that minimizes total cost while satisfying all constraints.

Finally, the solution interpretation phase translates the numerical output back into business terms. This means converting the optimal values of decision variables into specific shipping instructions, calculating the total minimized cost, and analyzing the solution for practical implementation considerations such as which routes are fully utilized and where capacity slack exists in the network.

## C1. Identification of Optimization Method or Algorithm

The optimization method I will use is **Linear Programming (LP)** solved via the **Revised Simplex Algorithm**, which is a standard implementation in modern optimization solvers. This choice is based on the mathematical structure of the problem and the proven effectiveness of this method for transportation network optimization.

The Revised Simplex Algorithm optimizes Amazon Air's logistics by finding the most efficient and cost-effective package delivery routes. Like navigating a city, it systematically searches for the best solution, minimizing delivery costs. This "revised" version is more efficient and requires less computational effort than the original.

The process starts with an initial delivery plan, then iteratively improves it while adhering to constraints like capacity. This continues until the absolute optimal solution is found. For large operations like Amazon Air, the algorithm excels due to predictable delivery patterns, managing many variables without becoming slow.Its main benefit is guaranteeing the absolute best solution, crucial for decisions involving millions in shipping costs.

## C2. Tools and Technologies

The technology stack uses the most current, stable version of Python within a Jupyter Notebook environment for interactive development.

**PuLP (Python Linear Programming)** is the main tool for setting up the optimization problem. It lets us write out the problem in a way that looks like mathematical equations, making it easy to translate our ideas into code.

**CBC (Coin-or Branch and Cut) Solver**, which comes with PuLP, is the engine that actually solves the problem. It efficiently figures out the best solution to the problem we've defined.

## D. Risks and Limitations

**Key Risks:**

- **Model assumptions and simplifications:** Real-world complexities and nuances may not be fully represented in the model, which is based on assumptions and simplifications. In any real-world setting, there are always going to be edge cases to consider and the model may fail to consider those. Examples include:
  - Static demand assumptions versus actual demand fluctuations
  - Linear cost relationship assumptions versus application of bulk discounts
  - Perfect weather and mechanical operation assumptions versus severe weather and mechanical failure delays
- **Data quality and accuracy:** We are operating under the assumption that the data is accurate and complete as best as it can be. Inaccurate or incomplete data can lead to biased results.
- **Constraint violations:** Constraints must be properly specified and properly enforced for them to remain valid and practical. Failing to enforce a constraint may invalidate the results.

## E. Sources


- WGU Course Materials