**Task 2: Solve an Optimization Problem**

Joseph Emmanuel Cayetano

Western Governors University

D605: Optimization Task 2

April 11, 2025

**A. Identify the optimization problem in the given scenario**

The optimization problem in the scenario is a multi-echelon supply chain network optimization problem. The goal is to minimize total cargo shipping costs while making sure each fulfillment center gets the amount it needs and no hub or focus city exceeds its capacity. The supply chain has multiple layers, including two main hubs, three focus cities, and 68 fulfillment centers. Because the shipment amounts are continuous and the constraints are linear, I will be solving this problem using linear programming.

**B. Create mathematical representations of the components of the optimization problem you identified**

Here, I provided the mathematical representations of the key components in the multi-echelon supply chain network optimization problem for Amazon Air:

The **decision variables** represent how much cargo is shipped between different locations in the network:

xij = Quantity of cargo shipped from hub *i* to focus city *j*

yik = Quantity of cargo shipped from hub *i* to fulfillment center *k*

zjk = Quantity of cargo shipped from focus city *j* to fulfillment center *k*

Where:

*i* ∈ {1, 2} = two hubs: Cincinnati/Northern Kentucky (CVG) and Fort Worth Alliance (AFW)

*j* ∈ {1, 2, 3} = three focus cities: Leipzig, Hyderabad, and San Bernardino

*k* ∈ {1, 2, …, 65} = 65 fulfillment centers (not including the 3 focus cities)

The objective of the optimization problem is to minimize the total transportation cost across Amazon’s air logistics network. The **objective function** is represented as:

A black and white symbol

AI-generated content may be incorrect.

Where:

cij = Cost per ton for shipping from hub *i* to focus city *j*

cik = Cost per ton for shipping from hub *i* to fulfillment center *k*

cjk = Cost per ton for shipping from focus city *j* to fulfillment center *k*

xij, yik, zjk = Decision variables representing the quantity of cargo shipped along each route.

The optimization problem includes constraints to ensure operational feasibility. These are the **constraints**:

**Hub Capacity Constraints:**

The total cargo leaving each hub cannot exceed its capacity. This makes sure that hubs are not overloaded.

A black text on a white background

AI-generated content may be incorrect.

Where:

xij = Quantity of cargo shipped from hub *i* to focus city *j*

yik = Quantity of cargo shipped from hub *i* to fulfillment center *k*

HubCapacity, *i* = Maximum monthly capacity of hub *i*

**Focus City Inflow Capacity Constraints:**

The total cargo entering each focus city cannot exceed its capacity. This makes sure that focus cities stay within their handling limits.

A black text on a white background

AI-generated content may be incorrect.

Where:

xij = Quantity of cargo shipped from hub *i* to focus city *j*

FocusCityCapacity, *j* = Maximum monthly capacity of focus city *j*

**Flow Balance at Focus Cities**

The total cargo entering a focus city must equal the total cargo leaving it. This makes sure that there is no extra or missing cargo at focus cities.

A black and white math equation

AI-generated content may be incorrect.

Where:

zjk = Quantity of cargo shipped from focus city *j* to fulfillment center *k*

xij = Quantity of cargo shipped from hub *i* to focus city *j*

**Fulfillment Center Demand Constraints**

Each fulfillment center must receive the right amount of cargo, either from hubs or focus cities.

A black and white math equation

AI-generated content may be incorrect.

Where:

yik = Quantity of cargo shipped from hub *i* to fulfillment center *k*

zjk = Quantity of cargo shipped from focus city *j* to fulfillment center *k*

Demand, *k* = Monthly demand for fulfillment center *k*

**Non-negativity Constraints**

All cargo flow quantities must be zero or positive because negative shipments are impossible.

A close up of a sign

AI-generated content may be incorrect.

These mathematical expressions show how cargo is shipped from hubs to focus cities and fulfillment centers while minimizing cost, meeting demand, and staying within capacity limits.

**B1. Write an expression that represents the objective function**

The objective of the optimization problem is to minimize the total transportation cost across Amazon’s air logistics network. It takes into account the cost of sending cargo from hubs to focus cities, from hubs to fulfillment centers, and from focus cities to fulfillment centers.

The objective function is represented as:

A black and white math equation

AI-generated content may be incorrect.

Where:

cij, cik, cjk = Cost per ton of shipping between different locations in the network.

cij = Cost per ton for shipping from hub *i* to focus city *j*

cik = Cost per ton for shipping from hub *i* to fulfillment center *k*

cjk = Cost per ton for shipping from focus city *j* to fulfillment center *k*

xij, yik, zjk = Decision variables representing the quantity of cargo shipped along each route.

xij = Quantity of cargo shipped from hub *i* to focus city *j*

yik = Quantity of cargo shipped from hub *i* to fulfillment center *k*

zjk = Quantity of cargo shipped from focus city *j* to fulfillment center *k*

**B2. Write expressions that represent the constraints of the optimization problem**

The following constraints help ensure the optimization model is realistic and follow the limits of Amazon Air’s logistics network:

**Hub Capacity Constraints:**

The total cargo leaving each hub cannot exceed its capacity. This makes sure that hubs are not overloaded.

A black text with black text

AI-generated content may be incorrect.

Where:

xij = Quantity of cargo shipped from hub *i* to focus city *j*

yik = Quantity of cargo shipped from hub *i* to fulfillment center *k*

HubCapacity, *i* = Maximum monthly capacity of hub *i*

**Focus City Inflow Capacity Constraints:**

The total cargo entering each focus city cannot exceed its capacity. This makes sure that focus cities stay within their handling limits.

A black text on a white background

AI-generated content may be incorrect.

Where:

xij = Quantity of cargo shipped from hub *i* to focus city *j*

FocusCityCapacity, *j* = Maximum monthly capacity of focus city *j*

**Flow Balance at Focus Cities:**

The total cargo entering a focus city must equal the total cargo leaving it. This makes sure that there is no extra or missing cargo at focus cities.

A black and white math equation

AI-generated content may be incorrect.

Where:

zjk = Quantity of cargo shipped from focus city *j* to fulfillment center *k*

xij = Quantity of cargo shipped from hub *i* to focus city *j*

**Fulfillment Center Demand Constraints:**

Each fulfillment center must receive the right amount of cargo, either from hubs or focus cities.

A black and white math equation

AI-generated content may be incorrect.

Where:

yik = Quantity of cargo shipped from hub *i* to fulfillment center *k*

zjk = Quantity of cargo shipped from focus city *j* to fulfillment center *k*

Demand, *k* = Monthly demand for fulfillment center *k*

**Non-negativity Constraints**

All cargo flow quantities must be zero or positive because negative shipments are impossible.



**B3. Identify the decision variables of the optimization problem**

The decision variables represent how much cargo is shipped between different locations in the Amazon Air’s logistics network. They help determine the most cost-effective shipping routes while meeting demand and staying within capacity limits:

xij = Quantity of cargo shipped from hub *i* to focus city *j*

yik = Quantity of cargo shipped from hub *i* to fulfillment center *k*

zjk = Quantity of cargo shipped from focus city *j* to fulfillment center *k*

Where:

*i* ∈ {1, 2} = two hubs: Cincinnati/Northern Kentucky (CVG) and Fort Worth Alliance (AFW)

*j* ∈ {1, 2, 3} = three focus cities: Leipzig, Hyderabad, and San Bernardino

*k* ∈ {1, 2, …, 65} = 65 fulfillment centers (not including the 3 focus cities)

**C. Describe the approach you will use to solve the optimization**

To solve the optimization problem, I will use a linear programming (LP) approach. This method works well for this task because the decision variables are continuous, and both the objective function and constraints are linear. I will build the model in Python using PuLP, which is a library that works well with Jupyter Notebook and makes it easy to define variables, the objective function, and constraints. The model will be solved using CBC (Coin-or Branch and Cut), which is the default solver that comes with PuLP. CBC is an “open-source solver for linear programming and mixed-integer programming problems” (Ali, 2024, par. 33).

Before solving the model, I will prepare the necessary input data such as demand at fulfillment centers, shipping costs between locations, and capacity limits for hubs and focus cities. After solving, I will review the results to see how much cargo should be shipped on each route and what the total minimized transportation cost is. This will help find the most cost-effective shipping plan that satisfies all constraints.

**C1. Identify the optimization method or algorithm you will use to solve the problem**

The optimization method I will use to solve the problem is Linear Programming (LP). This method works well for this task because the decision variables are continuous, and both the objective function and constraints are linear. To solve the model, I will use the CBC (Coin-or Branch and Cut) solver, which is the default solver in the PuLP library for Python. CBC is an open-source solver that efficiently handles both linear and mixed-integer programming problems. In this case, since all variables are continuous and the model is linear, CBC will use the simplex algorithm to find the best solution. The simplex algorithm works well for LP problems because it efficiently searches the feasible region defined by linear constraints to minimize the objective function, which in this case is the total transportation cost.

**C2. Describe the tools and technologies you will use to solve the problem**

To solve the optimization problem, I will use Jupyter Notebook as my development environment and Python as the programming language. I will build the optimization model using PuLP, which is a Python library made for linear programming. PuLP makes it easy to define decision variables, the objective function, and constraints directly in code. To solve the model, I will use the CBC (Coin-or Branch and Cut) solver, which is the default solver that comes with PuLP. CBC is an open-source solver that efficiently handles both linear and mixed-integer programming problems. Since all variables in the model are continuous and the structure is linear, CBC will use the simplex algorithm to find the best solution. I will use all of these tools together to model and solve the optimization problem.

**D. Assess the risks and limitations involved in your recommended optimization approach**

There are some risks and limitations with using linear programming to solve the optimization problem. First, linear programming assumes that all relationships in the model are linear, but real-world logistics are often more complex. For example, shipping costs may decrease when sending larger volumes, and delivery times might not always be consistent. Because of this, the model might oversimplify the problem and miss some real-world details. Second, the model uses continuous variables, so it can suggest shipping amounts like 2.5 tons. However, in real life, cargo usually needs to be packed in full units, like whole containers or pallets. This means the solution might not be practical unless the numbers are rounded. Third, the model assumes that all input data, such as demand at fulfillment centers and hub capacities, is accurate and doesn’t change. If the data is incorrect or changes later, the solution may no longer work well. Even with these limitations, linear programming is still an effective method for this type of optimization problem.

**References**

Ali, M. (2024, October 28). *Optimizing with Pyomo: A Complete Step-by-Step Guide.* DataCamp. https://www.datacamp.com/tutorial/pyomo