**Task 3: Solve an Optimization Problem**

Joseph Emmanuel Cayetano

Western Governors University

D605: Optimization Task 3

April 16, 2025

**GitLab Repository URL:** <https://gitlab.com/wgu-gitlab-environment/student-repos/jcayet5/d605-optimization/-/tree/master?ref_type=heads>

**GitLab Repository Branch History Screenshot**

A screenshot of a computer

AI-generated content may be incorrect.

**A. Using the mathematical representations that you developed in Task 2, write a program in Python to solve the optimization problem computationally**

A screenshot of a computer

AI-generated content may be incorrect.

In this code chunk, I created three dictionaries. First is "hubs" and it defines the monthly capacity for each hub. Second is "focus\_cities" and it defines the monthly capacity for each focus city. Third is "fulfillment centers" and it shows the monthly demand for each fulfillment center.

A screenshot of a computer screen

AI-generated content may be incorrect.

In this code chunk, I created a dictionary called "cost\_data" that stores shipping costs for each route. Each key is a route pair (origin, destination), and each value is the cost per ton. I grouped the data into three types of routes: from hubs to focus cities, from hubs to fulfillment centers, and from focus cities to fulfillment centers.

A screenshot of a computer program

AI-generated content may be incorrect.

In this code chunk, I started by creating a new linear programming problem called "Amazon\_Air\_Optimization". I used the argument "LpMinimize" to specify that the goal is to minimize the objective function. Next, I split the "cost\_data" dictionary based on route types to make the model easier to work with. The variable "cost\_hub\_to\_focus" contains costs for routes from hubs to focus cities. The variable "cost\_hub\_to\_center" contains costs for routes directly from hubs to fulfillment centers. The variable "cost\_focus\_to\_center" contains costs for routes from focus cities to fulfillment centers.

Then, I created three decision variables that represent shipment quantities for each route type. The variable "x[i, j]" is the quantity shipped from hub i to focus city j. The variable "y[i, k]" is the quantity shipped directly from hub i to fulfillment center k. The variable "z[j, k]" is the quantity shipped from focus city j to fulfillment center k. I made sure that the variables are continuous and that there are no negative shipment quantities. Finally, I created the objective function to minimize the total transportation cost. This was done by multiplying each shipment quantity by its cost and summing all route costs.

A screenshot of a computer program

AI-generated content may be incorrect.

In this code chunk, I defined the constraints for the optimization problem. These constraints make sure the solution follows real-world limits like capacity and demand. The first constraint is the hub capacity constraint, which ensures that the total cargo shipped out from each hub does not exceed its monthly capacity. For example, the Cincinnati hub cannot ship more than 95,650 tons per month to focus cities and fulfillment centers combined. The second constraint is the focus city capacity constraint, which ensures that the total cargo entering each focus city stays within its handling capacity. For example, the focus city Leipzig can receive up to 85,000 tons from all hubs each month. The third constraint is the flow balance at focus cities, which ensures that all cargo received by a focus city is sent to fulfillment centers without any surplus or shortage. For example, if the focus city Leipzig receives 50,000 tons, it must send exactly 50,000 tons to the fulfillment centers. The fourth constraint is the fulfillment center demand constraint, which ensures each fulfillment center receives exactly the amount of cargo it needs. For example, the Paris fulfillment center must receive 6,500 tons per month, either directly from hubs or through focus cities.

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

In this code chunk, I solved the optimization problem and displayed the results. First, I used PuLP's default CBC solver to find the solution that minimizes total transportation cost while meeting all constraints. Then, I printed the solution status to check if the solver found an optimal result. After that, I printed the total minimized cost. Lastly, I printed only the routes that were used, which are the ones with shipment quantities greater than zero.

Looking at the output, the status is “optimal”, which means the solver found a solution that meets all constraints and gives the lowest possible total transportation cost. The total cost is $199,476.25. All the active routes and their shipment amounts are listed. For example, the route "x\_('Cincinnati',\_'Leipzig') = 43470.0" shows that the Cincinnati hub shipped 43,470 tons to the Leipzig focus city, which likely forwarded those shipments to fulfillment centers. Another example is "y\_('Cincinnati',\_'Atlanta') = 3000.0", which means 3,000 tons were shipped directly from the Cincinnati hub to the Atlanta fulfillment center. Lastly, "z\_('Leipzig',\_'Delhi') = 19000.0" shows that the Leipzig focus city sent 19,000 tons to the Delhi fulfillment center.

There are no routes from the San Bernardino and Hyderabad focus cities, which means the solver chose not to use them in the optimal solution. This likely happened for a few reasons. First, shipping directly from the Cincinnati and Fort Worth hubs or through Leipzig was cheaper than using San Bernardino or Hyderabad. Second, all fulfillment center demands and capacity limits were met without needing those two cities. Finally, because the goal was to minimize total transportation cost, the solver left out any routes that were not needed or cost-effective.

**A1. Demonstrate that the solver provided a solution to the optimization problem**

A screenshot of a computer code

AI-generated content may be incorrect.

The output shows that the solver status is “optimal”, which means the optimization problem was solved successfully and all constraints were met. The total transportation cost is $199,476.25, which is the lowest cost needed to meet all demand while staying within hub and focus city capacities.

A close-up of a list of numbers

AI-generated content may be incorrect.

The routes listed above show the shipment amounts for each route used in the optimal solution. For example, the route "x\_('Cincinnati',\_'Leipzig') = 43470.0" shows that the Cincinnati hub shipped 43,470 tons to the Leipzig focus city, which likely forwarded those shipments to fulfillment centers. Another example is "y\_('Cincinnati',\_'Atlanta') = 3000.0", which means 3,000 tons were shipped directly from the Cincinnati hub to the Atlanta fulfillment center. Lastly, "z\_('Leipzig',\_'Delhi') = 19000.0" shows that the Leipzig focus city sent 19,000 tons to the Delhi fulfillment center. This output confirms that the solver assigned values to the decision variables while meeting all constraints.

**B. Analyze the output of your model to demonstrate that the solution satisfies the requirements of the problem**

The goal of this optimization problem was to minimize the total transportation cost while making sure all fulfillment centers receive the cargo they need and that hubs and focus cities stay within their capacity limits.

A screenshot of a computer code

AI-generated content may be incorrect.

The solver returned a status of ”optimal”, which means the optimization problem was solved successfully and all constraints were met. The total transportation cost is $199,476.25, which is the lowest cost needed to meet all demand while staying within hub and focus city capacities.

A close-up of a screen

AI-generated content may be incorrect.

The solution satisfies all four constraints. For the hub capacity constraint, both Cincinnati and Fort Worth hubs shipped less than their maximum limits. Looking at the output, the Cincinnati hub shipped a total of 95,518 tons, which is 132 tons under its 95,560-ton capacity. The Fort Worth hub shipped 38,229 tons, staying 6,121 tons below its 44,350-ton capacity.

For the focus city capacity constraint, each focus city received only as much as it could handle. For example, the route "x\_('Cincinnati',\_'Leipzig') = 43470.0" shows that the Leipzig focus city only received 43,470 tons, which is well within its 85,000-ton capacity.

For the flow balance constraint at focus cities, the total amount received by a focus city equals the amount it sent out. In the output, the Leipzig focus city received 43,470 tons and sent out the same total amount to the fulfillment centers. It shipped 9,100 tons to Bangalore, 570 tons to Coimbatore, 19,000 tons to Delhi, and 14,800 tons to Mumbai. This shows that there was no leftover cargo or shortage at the Leipzig focus city.

For the fulfillment center demand constraint, every fulfillment center received exactly what it needed. For example, the route “z\_('Leipzig',\_'Delhi') = 19000.0” shows that the Delhi fulfillment center received its full demand from the Leipzig focus city, and the route “y\_('Cincinnati',\_'Atlanta') = 3000.0” shows that the Atlanta fulfillment center received 3,000 tons directly from the Cincinnati hub.

There are no routes from the San Bernardino and Hyderabad focus cities, which means the solver chose not to use them in the optimal solution. This likely happened for a few reasons. First, shipping directly from the Cincinnati and Fort Worth hubs or through Leipzig was cheaper than using San Bernardino or Hyderabad. Second, all fulfillment center demands and capacity limits were met without needing those two cities. Finally, because the goal was to minimize total transportation cost, the solver left out any routes that were not needed or cost-effective.

**B1. Demonstrate that the constraints of the optimization problem are satisfied**

A screen shot of a computer code

AI-generated content may be incorrect.

The solver returned an “optimal” status, which means it found a feasible solution that satisfies all constraints, including hub capacity, focus city capacity, flow balance at focus cities, and fulfillment center demand. The total transportation cost is $199,476.25, which is the lowest cost needed to meet all demand while staying within hub and focus city capacities.

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

The solution satisfies all four constraints. For the hub capacity constraint, both Cincinnati and Fort Worth hubs shipped less than their maximum limits. Looking at the output, the Cincinnati hub shipped a total of 95,518 tons, which is 132 tons under its 95,560-ton capacity. The Fort Worth hub shipped 38,229 tons, staying 6,121 tons below its 44,350-ton capacity.

For the focus city capacity constraint, each focus city received only as much as it could handle. For example, the route "x\_('Cincinnati',\_'Leipzig') = 43470.0" shows that the Leipzig focus city only received 43,470 tons, which is well within its 85,000-ton capacity.

For the flow balance constraint at focus cities, the total amount received by a focus city equals the amount it sent out. In the output, the Leipzig focus city received 43,470 tons and sent out the same total amount to the fulfillment centers. It shipped 9,100 tons to Bangalore, 570 tons to Coimbatore, 19,000 tons to Delhi, and 14,800 tons to Mumbai. This shows that there was no leftover cargo or shortage at the Leipzig focus city.

For the fulfillment center demand constraint, every fulfillment center received exactly what it needed. For example, the route “z\_('Leipzig',\_'Delhi') = 19000.0” shows that the Delhi fulfillment center received its full demand from the Leipzig focus city, and the route “y\_('Cincinnati',\_'Atlanta') = 3000.0” shows that the Atlanta fulfillment center received 3,000 tons directly from the Cincinnati hub.

**B2. Demonstrate that the solution includes decision variables, constraints, and the objective function**

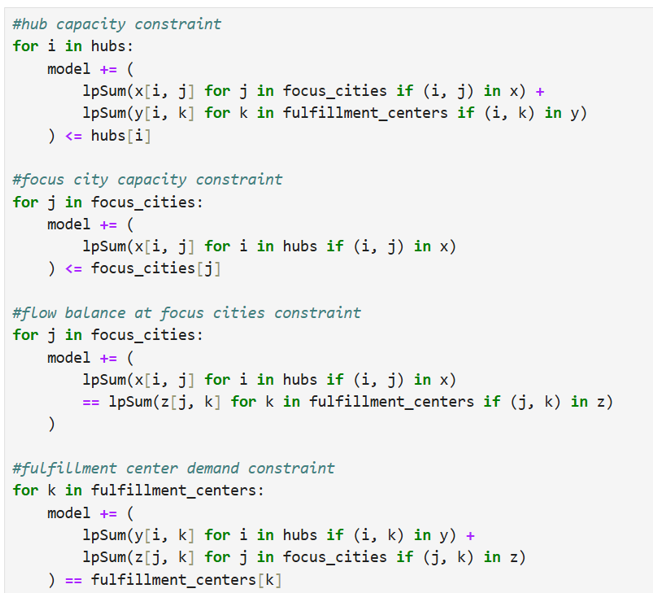
The optimization model that I created includes decision variables, constraints, and the objective function.

A screenshot of a computer code

AI-generated content may be incorrect.

Here, I started by creating a new linear programming problem called "Amazon\_Air\_Optimization". I used the argument "LpMinimize" to specify that the goal is to minimize the objective function. Next, I split the "cost\_data" dictionary based on route types to make the model easier to work with. The variable "cost\_hub\_to\_focus" contains costs for routes from hubs to focus cities. The variable "cost\_hub\_to\_center" contains costs for routes directly from hubs to fulfillment centers. The variable "cost\_focus\_to\_center" contains costs for routes from focus cities to fulfillment centers.

Then, I created three decision variables that represent shipment quantities for each route type. The variable "x[i, j]" is the quantity shipped from hub i to focus city j. The variable "y[i, k]" is the quantity shipped directly from hub i to fulfillment center k. The variable "z[j, k]" is the quantity shipped from focus city j to fulfillment center k. I made sure that the variables are continuous and that there are no negative shipment quantities.



Here, I defined the constraints for the optimization problem. These constraints make sure the solution follows real-world limits like capacity and demand. The first constraint is the hub capacity constraint, which ensures that the total cargo shipped out from each hub does not exceed its monthly capacity. For example, the Cincinnati hub cannot ship more than 95,650 tons per month to focus cities and fulfillment centers combined. The second constraint is the focus city capacity constraint, which ensures that the total cargo entering each focus city stays within its handling capacity. For example, the focus city Leipzig can receive up to 85,000 tons from all hubs each month. The third constraint is the flow balance at focus cities, which ensures that all cargo received by a focus city is sent to fulfillment centers without any surplus or shortage. For example, if the focus city Leipzig receives 50,000 tons, it must send exactly 50,000 tons to the fulfillment centers. The fourth constraint is the fulfillment center demand constraint, which ensures each fulfillment center receives exactly the amount of cargo it needs. For example, the Paris fulfillment center must receive 6,500 tons per month, either directly from hubs or through focus cities.

A close-up of a computer code

AI-generated content may be incorrect.

Here, I defined the objective function to minimize the total transportation cost. This is done by multiplying the cost per ton for each route by the amount shipped on that route, and then summing these values across all routes.

**B3. Explain why the solution that your code outputs matches your expected output**

The solution produced by my code matches my expected output because the model was designed to minimize the total transportation cost while satisfying all constraints. The solver returned an “optimal” status, which confirms that it found a valid and cost-effective solution. This indicates that the model successfully minimized the total transportation cost while meeting every constraint. The route output also shows that all four constraints were satisfied. For the hub capacity constraint, both the Cincinnati and Fort Worth hubs shipped less than their maximum limits. For the focus city capacity constraint, each focus city received only as much as it could handle. For the flow balance constraint at focus cities, the total amount received by each focus city equaled the amount it sent out. For the fulfillment center demand constraint, each fulfillment center received exactly what it needed. The solver also avoided using more expensive focus cities like San Bernardino and Hyderabad, which shows that it chose the most cost-effective routes. This all aligns with the goals and expectations of the model.

**C. Provide a reflection on how the development of your approach matched your expectations of the process**

When I first started this task, it took me a while to enter all the hubs, focus cities, and fulfillment centers into the program because there were so many. I also made a few typos in the route data, which I didn’t notice until I ran the program and the solver returned an “infeasible” status, which means the model “has no solution that satisfies all the constraints” (Hofmann, 2021, par. 1). Aside from that, everything else went smoothly. Building the optimization model was fairly straightforward, which I expected since I had already defined the objective function, constraints, and decision variables in Task 2. I did face some challenges in organizing the data, especially when I accidentally placed decision variables in the wrong parts of the formulas.

I was surprised by how efficiently the solver found the most cost-effective routes. I also didn’t expect that some focus cities, like San Bernardino and Hyderabad, would be completely unused, but it made sense once I saw the results. The solver naturally avoided more expensive or unnecessary routes.

Overall, the development process matched my expectations in both structure and outcome. It also reminded me how important it is to carefully check the code and formulas.

**References**

Hofmann, F. (2021, March). *Trouble shooting - Infeasible Model.* Linopy. https://linopy.readthedocs.io/en/latest/infeasible-model.html