**A:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

As outlined in Task 2, the optimization problem is, “How can we minimize the total shipment costs while ensuring capacity needs are met?”. As seen above, we have optimal shipment amounts, with an optimal minimum cost of $199,390.25, and optimal storage capacity for the hub and focus cities’ capacity. We can now plan accordingly to use the best shipment amounts and capacity for hubs and focus cities while maintaining our optimal minimum cost of $199,390.25 (WGU Learning, n.d.). I would also like to add that when creating this optimization problem, I utilized the proper name, “Amazon\_Distribution,” and specified the objective function as minimize. This would instruct Python to minimize the cost of the function I created (GeeksforGeeks, 2019).

**B:**

**A math equations on a white background

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.**

The constraints I initially established and replicated in my code are pictured above from Task 2. For example, as you can see above, the cost for hub AFW to San Bernardino is 0.5, and the hub capacity AFW is 44,350 (Sane’s Academy of Artificial Intelligence, 2023).

A screenshot of a computer program

AI-generated content may be incorrect.

A black text on a white background

AI-generated content may be incorrect.

The solution includes the decision variables, constraints, and the objective function. As mentioned before, the constraints are listed above. The solution includes the decision variables x, y, and z. This covered the shipment from hub to focus city, hub to center, and focus city to center, as previously established in Task 2. I also set up the objective function accordingly (Sane’s Academy of Artificial Intelligence, 2023). First, I set my problem to the “Amazon\_Distribution” and utilized LpMinimize, to define the task (WGU Learning, n.d.). I put the objective first to equal 0 to establish the empty variable; then, I added each part of the costs from hub to focus city, hub to center, and the focus city to center. Lastly, I set the problem of minimizing cost to my objective (*Transportation Cost Flow Optimization Using Python Scipy Minimize*, n.d.).

This solution matches my expectations of the process. I have followed my guidelines for task 2 by setting my constraints, decision variables, and the objective function while also finding an accurate optimization problem. As pictured below, my minimal distribution cost would be only $199,390.25. This satisfies the constraints, as the 2 hubs used for storage were still within their capacity limits. CVG used all its storage without going over, but AFW still has 6,425 left after using 37,925. These two hubs could cover all our storage needs and allow our focus city to maintain its free storage space (Stojiljković, 2023).

A screenshot of a computer program

AI-generated content may be incorrect.

**C:**

The development of my approach matched my expectations. The problem was simple and linear while still having many data points. We could further see that in the simple solution it used. Relying on the hubs for storage needs and utilizing the right methods according to costs, the solution was straightforward and simple. We broke down the steps in Task 2, which helped as an outline. In task 2, we had our parts of the problem broken down into optimization problems, decision variables, constraints, objective functions, and approaches. Task 2 served as a great outline of expectations of the process and allowed me to reference it when establishing the code for this linear programming (WGU Learning, n.d.).

Sources:

Cloud academy. (2024, Sept). *Linear optimization problems*. <https://platform.qa.com/lab/linear-programming-s1-2190cd8c-4676-44aa-8070-142d7655a0d8/>

Felkl, L. (2023, September 30). Multi-objective LP with PuLP in Python. SCDA. <https://www.supplychaindataanalytics.com/multi-objective-linear-optimization-with-pulp-in-python/>

GeeksforGeeks. (2019, August 26). *Python | Linear Programming in Pulp*. GeeksforGeeks. <https://www.geeksforgeeks.org/python-linear-programming-in-pulp/>

Scribbr. (n.d.). Scribbr. <https://www.scribbr.com>

Sane’s Academy of Artificial Intelligence. (2023, May 5). *Linear Optimization with Python (PuLP) | Linear Programming Problem(LPP)* [Video]. YouTube. <https://www.youtube.com/watch?v=ybA5zfvnfFI>

Stojiljković, M. (2023, June 16). *Hands-On Linear Programming: Optimization with Python*. <https://realpython.com/linear-programming-python/>

*Transportation cost flow optimization using Python Scipy Minimize*. (n.d.). Stack Overflow. <https://stackoverflow.com/questions/48895750/transportation-cost-flow-optimization-using-python-scipy-minimize>

WGU Learning. (n.d.). *Lesson 3.1:Optimization Approaches and Models*. WGU. <https://apps.cgp-oex.wgu.edu/wgulearning/course/course-v1:WGUx+OEX0417+v01/block-v1:WGUx+OEX0417+v01+type@sequential+block@a25e2216b64d45ab95a3f492f6d72379/block-v1:WGUx+OEX0417+v01+type@vertical+block@b6a7fb6685d449bd8494488431588f27>