BUDT732 Course Project - Part 2 Report

Solution Status: Feasible | **Profit Achieved:** $58,920.91

Google Colab Link:

https://colab.research.google.com/drive/16f9br6FbP24mLI8jvu1Pet9T5clr2d1B?usp=sharing

Methodology

Our team focused on maximizing cash inflows, then minimizing costs starting at the supply level, then at the transportation level, while ensuring feasibility through a systematic three-step process.

1. Store Pricing Optimization

The objective in the first step of the process, the store pricing optimization, was to determine the profit-maximizing prices and resulting demands for each store in the network, accounting for store-specific demand curves, store rental costs, and customer travel time substitution effects. Before we started this process, we had to prepare the data, which was stored in the storedata.csv (containing demand curve parameters like slope and intercept and store rental costs) and storett.csv (containing a matrix describing travel time between each store) files. The storedata.csv file is used to extract a store list from its index, as well as the AMPL parameters for demand curve (slope and intercept) and store rental costs. The storett.csv is “melted” into long-form to fulfill the parameter for travel time between any two given stores.

The next step in the process was forming the mathematical optimization model formulation in AMPL. We created one set of stores and four parameters for store demand curve intercept, store demand curve slope, store rental cost, and travel time between stores. We defined two decision variables: the price to sell the product at each store, and the resulting demand for the product at each store. The objective function in this formulation is to maximize store profit, which is the summation of the following: the product of price and demand, subtracting rent, and the variable cost of stocking each unit of the product (10 * demand), for each store in the network. Finally, we add two constraints: the demand curve constraint, and the time-pricing variance constraint. The demand curve constraint ensures the model abides by each store’s demand curve - it sets the demand equal to the intercept plus the product of the price and slope for each store. The time-pricing variance constraint makes sure the price difference between two stores does not differ by more than $0.05 per minute of travel time between the stores - it sets the product of the time difference and 0.05 as greater than or equal to the difference between the two stores’ prices.

The final step in the process was to pass the necessary values into the set and parameters. The set of stores is built by passing the list created by the index of the storedata DataFrame. The parameters for demand curve intercepts, slopes, store rental costs, and time between stores are all populated by passing values from the storedata and storett DataFrames. Finally, as was standard in this course, we used the “gurobi” solver in AMPL and displayed the optimum profit, prices, and demands.

2. Warehouse Optimization

The objective in this step of the process, the warehouse supply optimization, was to determine the profit-maximizing allocation of goods from each warehouse to the stores in the network. This required accounting for warehouse supply limits, store-level demand requirements, transportation costs, fixed warehouse operating costs, and the time needed to ship goods between facilities. Before beginning the optimization, we prepared the necessary data inputs. These included the lists of warehouses and stores, the supply available at each warehouse, the demand required at each store, unit prices, fixed operating costs, warehouse rental costs, and both warehouse-to-store and store-to-store travel times. These values formed the basis for the AMPL parameters used in the model. We also computed a BigM value based on the maximum supply in the dataset, which is used in several constraints of the formulation.

The next step in the process was to load the mathematical optimization model written in AMPL. The model contains two sets, one for warehouses and one for stores, and several parameters that represent supply, demand, pricing, transportation time, transportation costs, fixed costs, and rental costs. The model includes decision variables for the quantity shipped from each warehouse to each store and a binary variable indicating whether a warehouse–store connection is used. The objective in this formulation is to maximize total profit across the network. Profit is calculated as revenue from supplying stores, minus shipping costs, minus fixed operating costs for any warehouse that is opened, and minus rental costs. The constraints in the model ensure that shipments do not exceed warehouse supply, that store demand is satisfied, and that the binary trip-existence variable correctly reflects whether a warehouse supplies a given store. After loading the model and assigning all values to its sets and parameters, we selected the “gurobi” solver and executed the optimization. AMPL then reported the resulting profit and the chosen shipment quantities.

The final step in this process was to extract, process, and organize the optimization results for further analysis. After solving, we pulled the values of the binary variables that indicate whether a shipment route is used and the corresponding shipment quantities. Because the next stage of the analysis focuses on a specific warehouse, we filtered the results to include only the stores served by a selected warehouse. We combined these filtered shipment results with the relevant travel-time data and converted them into structured DataFrames for easier interpretation. We also created dictionaries capturing store-level shipment quantities, warehouse-store travel times, and store-store travel times. These were supplemented with artificial source and sink nodes, which are useful in later routing analysis. Finally, we identified which stores were served, computed the warehouse’s total shipped volume, and constructed node-level supply and demand values. Together, these processed outputs provide the necessary inputs for subsequent modeling steps, such as routing, sequencing, or evaluating transportation efficiency across the network.

3. Transportation Optimization

For this part of the project we focused on the transportation model. The goal was to create delivery routes that follow the limits given in the project. These limits include vehicle capacity, total travel time and unloading time. Once the warehouse assignment model was complete we already knew which stores were linked to each warehouse and how much product each store needed. This allowed us to build the routing step in a clear and organized way.

We started by collecting the store lists for warehouses 4, 6, 7 and 9 along with the quantities they needed. We also used the travel time tables for warehouse to store and store to store movement. Every van can carry up to 500 units and a route cannot exceed 480 minutes. These 480 minutes include all driving time and 30 minutes of unloading at each store. Because of this, we expected most routes to be short and limited to a few stores.

The routing model used a few main rules. A van that arrives at a store must also leave it. A van that is not used cannot appear in any route. The total load on a van cannot go over capacity. The time used by a van must be within the eight hour limit. The model also included the daily cost of a van and the cost based on time spent driving and unloading.

We solved this model for each warehouse separately. The results gave us small sets of routes where each route visited two or three stores. This pattern worked well because too many stops would exceed the time limit once unloading time was added. The final set of routes stayed within the required limits and matched the quantities from the earlier supply model. After we confirmed feasibility we wrote the output in the format required for the route file.

This step completed the link between pricing, supply allocation and transportation. It showed that it was possible to meet demand while respecting all operating limits.

4. Final Evaluation

We used the evaluation script provided for the project to check our full solution. The script read the store file, warehouse file and route file and then calculated revenue, costs and total profit. It also checked route feasibility. Our final profit was 58920.91 dollars per day. This is higher than the target of 48000 dollars and also above the ten percent improvement threshold. The solution passed all feasibility checks which confirmed that the final routes, stocking levels and prices followed the project rules.

5. Reflections

Working on the transportation part helped us see how important the earlier decisions were. Once prices and warehouse assignments were fixed the routing model had to work with the quantities already assigned. The capacity rule and the unloading time rule made the routing step more restrictive than it appeared at first. Routes that looked fine in distance sometimes failed once unloading time was added. This made it important to limit each route to only a few stores.

Splitting the entire project into stages made the work manageable. Solving everything together would have been too difficult. By keeping routing as its own part we were able to focus only on time limits, capacity limits and van usage without having to worry about pricing or supply decisions. The final profit showed that this staged approach was effective.

6. List of Documents

Below is the set of documents used or produced in our project.

- storedata file with intercepts, slopes and rent values
- whdata file with capacity and rent values for warehouses
- whtt file with warehouse to store travel times
- storett file with store to store travel times
- store output file with final store prices
- warehouse output file with final stocking levels
- route output file with final routes
- Jupyter notebooks used to run pricing, supply and transportation models
- AMPL model files for pricing, supply and transportation
- evaluation notebook used to compute the final profit