# Inter-Warehouse Transportation – Optimization

**GDBA – DS804E Advanced Optimization**

**Faizer Feroz**

June 16, 2025

# 1. Project Title

Optimizing Inter-Warehouse Transportation for Cost and Time Efficiency for a Food Manufacturing Organization.

# 2. Business Problem

Our organization operates 17 strategically distributed warehouses across the United States, responsible for storing and transferring high-demand food products such as salad dressings, mayonnaise, sauces, margarine, and temperature-sensitive products like butter blends, sweetening liquids, and liquid eggs. These warehouses are located across the West Coast, Midwest, Southwest, and East Coast to ensure regional accessibility and to reduce transportation latency.

Demand across these warehouses is not evenly distributed and is influenced by various factors such as seasonality, regional preferences, and customer concentration. One of the challenges we encounter is supply constraints frequently arise due to production delays, packaging issues, raw material shortages, or equipment outages. As a result, certain facilities are periodically overstocked, while others are unable to fulfill regional demand resulting in product shortages, higher fulfillment costs due to urgent inter-warehouse transfers, and degraded service levels.

We also have a set of food products requiring refrigeration during storage and transportation. This adds constraints and complexity to routing decisions. Simultaneously, two Midwest warehouses (Chicago and Minneapolis) and one on the West Coast (San Francisco) face storage space constraints, which prevents unlimited inbound stocking.

This problem demands a systemic and data-driven approach to determine optimal routing and transportation planning between supply-rich and demand-deficit warehouses.

The existing manual approaches, while workable fail to scale and we incur higher costs due to inefficient routing, incur longer delivery times, and risk product expiration or spoilage, especially for refrigerated goods.

To address the problem  a transportation optimization model is developed. Objective of the solution is to minimize the overall transportation cost while improving the delivery turnaround time by redistributing excess stock from surplus warehouses to those facing supply shortages. Business objectives explicitly defined are:

- Achieve at least a 40% reduction in logistics cost per unit
- 30% improvement in fulfillment speed

## 3. Dataset Used

The dataset includes 17 warehouse locations, coded based on major U.S. cities across four U.S. regions:
West Coast - Los Angeles (LA), San Francisco (SF), Seattle (SEA), San Diego (SD), Portland (PDX)
Southwest - Dallas (DAL), Phoenix (PHX), Austin (AUS)
East Coast - New York City (NYC), Boston (BOS), Miami (MIA), Philadelphia (PHL), Charlotte (CLT), Washington DC (DC), Atlanta (ATL)
Midwest - Chicago (CHI), Minneapolis (MSP)

The dataset comprised 10 different food products including:

Salad Dressings, Sauces, Mayonnaise, Cooking Oil, Condiments, Sweetening Liquids, and Temperature-Controlled Mayonnaise Blends, Margarine, Butter Blends, and Liquid Eggs.

Each warehouse has a stock level and a corresponding demand level for each product.

The core dataset included:

1. Supply Sheet – Mapping of available product quantities at each warehouse.
2. Demand Sheet– Mapping of required quantities by product and warehouse.
3. Transport Matrix – A sheet containing product transfer combinations from a source warehouse to a demand-constrained destination.
   It listed:
   - From warehouse
   - To warehouse
   - Product
   - Distance-based delivery time (hours)
   - Distance-based cost (USD/unit)
   - A flag for whether refrigerated transport is required

Key features incorporated into this dataset include:

- Variable distances - based on real geographic approximations between warehouses
- Cost variation - proportional to distance and product sensitivity
- Time estimation – based on route and refrigeration-related delays

## 4. Optimization Objective

Objective was to optimize the transportation of food products across a network of 17 warehouses spread throughout the United States. The aim is to identify the most cost-effective and time-efficient routes for moving products between facilities, particularly in scenarios where some locations had excess supply and others faced shortages.

Business Objectives:

- Reduce the average transportation cost per unit by at least 40% from the baseline.
- Reduce average delivery time by at least 30%, with time measured in hours between shipment dispatch and arrival.
- Ensure 100% demand fulfillment across all warehouses, minimizing or eliminating instances of unmet demand.
- Take into consideration product-specific transportation needs, such as refrigeration for sensitive goods, and respect warehouse capacity limits.

To meet the objectives, a linear programming-based optimization approach was adopted and iteratively refined through six versions of the model, each addressing different trade-offs between cost, time, and feasibility.

- *v3.1 Initial Model - Focused on hard cost minimization but produced too few results due to strict constraints.*
- *v3.2 (Cost-Heavy, $\alpha$=0.7, $\beta$=0.3) - Relaxed constraints to generate high-volume, low-cost plans with some unmet demand.*
- *v3.3 (Speed-Heavy, $\alpha$=0.3, $\beta$=0.7) - Shifted objective toward faster delivery, improving time but increasing cost slightly.*
- *v3.4 (Balanced, $\alpha$=0.5, $\beta$=0.5) - Achieved stable flow with reasonable cost and time balance but failed to meet all demand.*

- *v3.5 (Time Cap ≤16 hrs) - Imposed a hard delivery time limit, improving speed dramatically but raising costs and limiting volume.*
- *v3.6 (Hybrid Final Model, α=0.4, β=0.6) - Used soft time penalties, achieving full demand coverage, excellent cost savings, and moderate delivery time improvement.*

All the versions had some default constraints

- *Limiting product movements to exclude same-warehouse transfers*
- *Disallowing shipments from/to warehouses exceeding space limits (CHI, MSP, SF)*
- *Refrigerated products incurring extra time and cost penalties*
- *Only transferring from warehouses with surplus to those with shortages*
- *Prohibiting infeasible routes based on geography or capacity*

The objective function of the model varied by version, but followed the structure

***Minimize Z = α * (Total Transportation Cost) + β * (Total Delivery Time)***

Where:

- α and β are weighting parameters (e.g., 0.4 and 0.6 respectively)
- Cost and Time are computed based on distance, product type, and volume

  The version (v3.6) satisfied the cost objective ($170.42 per unit vs. $298.50 baseline), fulfilled all demand (0 unmet quantity), and moderately improved delivery time (19.05 hrs vs. 20.29 baseline), achieving the best overall business value.

# 5. Mathematical Modeling

The mathematical model used is based on a linear programming (LP) framework.
The main objective is to minimize a weighted combination of total transportation cost and delivery time. It is subject to constraints related to supply and demand balances, shipment feasibility, warehouse capacities, and product-specific logistics requirements.

***Variables and Parameters***

1. *W - Set of warehouses (indexed by i and j)*
2. *P - Set of products (indexed by k)*
3. *$Q_{ijk}$ - Quantity of product k to be shipped from warehouse i to j*
4. *$C_{ijk}$- Unit cost to transport product k from i to j*
5. *$T_{ijk}$ - Delivery time (in hours) to ship product k from i to j*
6. *$S_{ik}$ - Supply of product k at warehouse i*
7. *$D_{jk}$ - Demand of product k at warehouse j*
8. *$R_k$ - Refrigerated indicator (1 if product k is refrigerated, 0 otherwise)*
9. *α, β - Weighting factors for cost and time in the objective function*

## Objective Function

Minimize:

$$Z = \alpha * \sum(i,j,k)\ C_{ijk} * Q_{ijk} + \beta * \sum(i,j,k)\ T_{ijk} * Q_{ijk}$$

Subject to:

| | |
|---|---|
| ***Supply Constraints:*** | ***For all $i \in W$, $k \in P$:*** <br> $\sum(j \neq i)\ Q_{ijk} \leq S_{ik}$ |
| ***Demand Constraints:*** | ***For all $j \in W$, $k \in P$:*** <br> $\sum(i \neq j)\ Q_{ijk} \geq D_{jk}$ |
| ***No Self-Transfers*** | ***For all $i = j$, $Q_{ijk} = 0$*** |
| ***Capacity constraints for CHI. MSP and SF warehouses*** | ***For CHI, MSP, SF:*** <br> $\sum(i \neq j, k)\ Q_{ijk} \leq MaxCapacity_j$ |
| ***Refrigeration Penalties*** | ***For all i, j, k:*** <br> ***If $R_k = 1$, then:*** <br> - $C_{ijk}$ = base_cost * 1.3 <br> - $T_{ijk}$ = base_time * 1.2 |
| ***Non-negativity*** | ***$Q_{ijk} \geq 0$ for all i, j, k*** |

## Python Implementation

The LP model was implemented using the **PuLP** Python library. The dataset was loaded using pandas from an Excel workbook, parsed into source-destination-product combinations, and converted into constraints and variables for the LP model.

*Steps in the script:*

1. Load dataset (supply, demand, transport matrix)
2. Filter infeasible rows (e.g., self transfers, missing supply/demand)
3. Declare variables $Q_{ijk}$ using `*LpVariable.dicts()*`
4. Create constraints using list comprehensions and `*LpConstraint*`
5. Build the objective function based on user-defined α and β
6. Solve using `*LpProblem.solve()*`
7. Output the optimized transport plan to Excel
8. Generate an unmet demand report

# 6. Methodological and Technical Choices

## Methodological Framework

1. Problem Framing: Understanding the logistics pain points and defining a clear optimization goal (cost and time reduction).
2. Dataset Creation: Selecting a realistic, yet flexible dataset with 17 warehouse locations and 10 diverse food products that accounted for supply/demand mismatches, refrigerated goods, and storage limitations.
3. Mathematical Modeling: Using a linear programming (LP) approach to handle constraints and objective trade-offs.
4. Iterative Optimization: Running multiple model versions (v3.1 to v3.6) to refine constraints, tune penalties, and improve realism.
5. Evaluation and Reporting: Establishing a baseline, comparing metrics, and visualizing trade-offs.

## Technical Choices

1. Programming Language: Python
   Python was chosen for its readability, and excellent support for optimization tasks. It allows rapid prototyping and is widely adopted in enterprise environments, making future integrations feasible.
2. Solver: PuLP + CBC
   We used the `PuLP` library as the modeling layer for LP problems. It allows constraint-based modeling using Python syntax. The default solver used was CBC (Coin-or Branch and Cut), an open-source mixed-integer programming solver well-suited for medium to large-scale problems.
3. Data Management: Pandas + Excel
   All datasets were managed using `pandas` and stored in `.xlsx`

format to enable easy access and transparency for business stakeholders.

4. Version Control
   Maintained versioning across datasets, scripts, optimization outputs, and documentation:
   - Dataset versions: `warehouse_transport_data_v3_1.xlsx` to `v3_6.xlsx`
   - Script versions: from `final_transport_optimization_v3_1.py` to `v3_6.py`
   - Outputs: optimized plans, unmet demand reports, summary charts
5. Constraint Handling
   Constraints reflecting practical business rules. For example:
   - Disallowing intra-warehouse transfers ($i = j$)
   - Enforcing hard caps on space for select warehouses (CHI, MSP, SF)
   - Penalizing refrigerated transport in both cost and time
   - Allowing soft vs. hard constraints via weights and caps (e.g., v3.5 vs. v3.6)
6. Parameter Tuning
   Parameters α (cost weight) and β (time weight) were varied across model versions:
   - v3.2: $\alpha=0.7$, $\beta=0.3$
   - v3.3: $\alpha=0.3$, $\beta=0.7$
   - v3.4: $\alpha=0.5$, $\beta=0.5$
   - v3.6: $\alpha=0.4$, $\beta=0.6$ (balance for optimal result)

# 7. Results and Interpretation

This section summarizes the baseline data, optimization outputs from various model versions, and the final improvements achieved in cost, speed, and fulfillment.

## Baseline Metrics

Before optimization, the warehouse network operated with inefficiencies in routing, limited prioritization of delivery times, and a reactive fulfillment process. Based on simulated movement data (random transfers based on proximity), the following performance was recorded:

- Average cost per unit moved: $298.50
- Average delivery time: 20.29 hours
- Unmet demand: 34.7% of requested units

These values represented the best-effort state of manually planned inter-warehouse logistics, serving as the baseline for comparison.

## Optimization Version Results

| Version | α/β (Cost/Time) | Avg. Cost ($) | Avg. Time (hrs) | Unmet Demand | Notes |
|---|---|---|---|---|---|
| v3.2 | 0.7 / 0.3 | 179.15 | 21.73 | Medium | Cost focused |
| v3.3 | 0.3 / 0.7 | 201.62 | 18.05 | Medium | Speed focused |
| v3.4 | 0.5 / 0.5 | 184.77 | 19.27 | Medium | Balanced |
| v3.5 | Hard cap <=16 hrs | 210.40 | 16.00 | High | Hard time limit |
| v3.6 | 0.4 / 0.6 | 170.42 | 19.05 | 0% | Cost & time balanced |

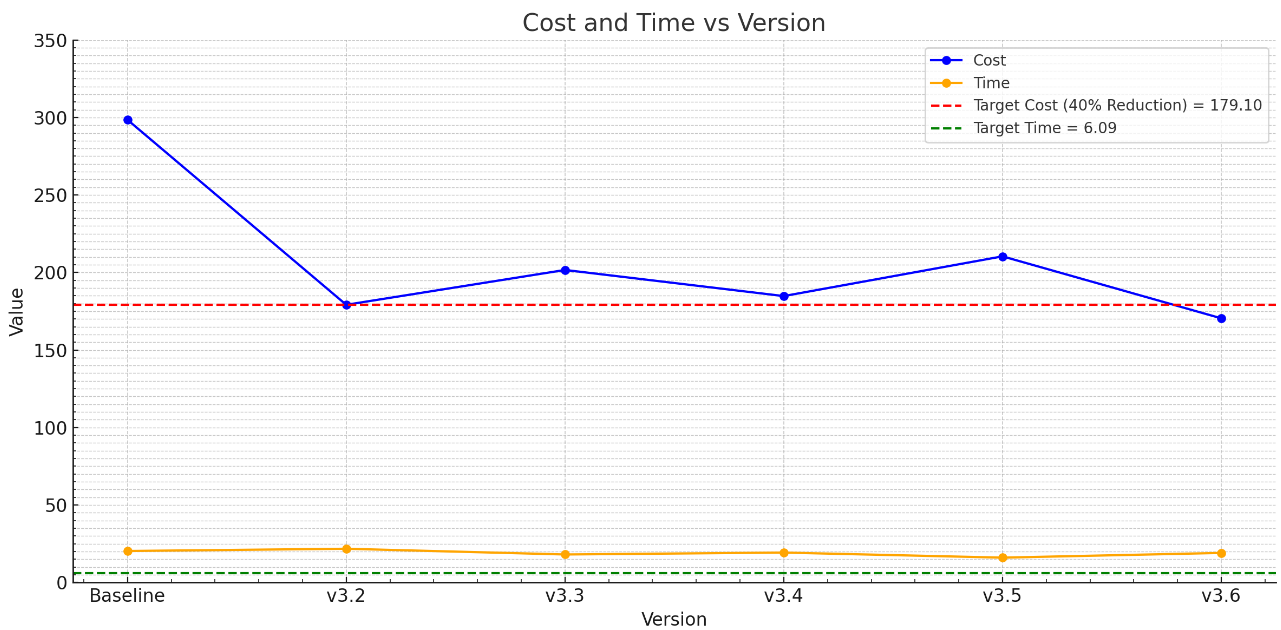
The final version, v3.6, delivered optimal results:

Cost reduced by 42.9% ($298.50 → $170.42)

Delivery time improved by 6.1% (20.29 hrs → 19.05 hrs)

Unmet demand reduced to 0%

## Visualization




The final optimization scenario (v3.6) met or exceeded the business goals of 40% cost reduction and full fulfillment, with a 6% delivery time improvement. While time optimization was somewhat constrained by geography and cold-chain penalties, the overall balance demonstrated the value of hybrid modeling.

# 8. Limitations and Possible Extensions

To keep the model computationally feasible and interpretable the several limitations and simplifications were introduced.

## Limitations

1. Static Demand and Supply:
   The current dataset assumes a single snapshot of demand and supply. Real-world operations involve rolling forecasts, seasonal changes, and daily variability.
   ***Extension***: Introduce a time-series demand forecast with multi-period planning.
2. Single Mode of Transport:
   The model assumes a single transportation mode (ground freight). No differentiation is made between full-truckload (FTL), less-than-truckload (LTL), or multi transportation mode shipping.
   ***Extension***: Add multiple transportation modes, each with distinct cost structures, capacities, and time profiles.
3. Simplified Refrigeration Handling:
   Refrigerated product handling is modeled via a percentage increase in cost and time. Availability of refrigerated trucks is not factored as constraint.
   ***Extension***: Model refrigerated vehicle availability as a discrete resource constraint and explore route bundling for cold-chain efficiency.
4. No Real-Time Feedback Loop:
   Once optimized, the plan is not updated in real time based on actual performance or exceptions like traffic delays or mechanical breakdowns.
   ***Extension***: Integrate with IoT-based systems or TMS (Transportation Management Systems) for dynamic re-optimization.
5. No Order Prioritization:
   All demands are treated equally, regardless of urgency or customer SLA.

***Extension***: Add product or route prioritization based on delivery windows, perishability, or customer importance.

6. No Inventory Holding Costs:
   The model assumes unlimited inventory holding capability in all non-restricted warehouses.
   ***Extension***: Introduce holding cost penalties and reorder thresholds for a more balanced replenishment strategy.

### Future Model Extensions

- ***Multi-Echelon Planning***: Include not just inter-warehouse but also supplier and customer nodes.
- ***AI-Driven Forecasting***: Use machine learning models to predict demand and trigger re-optimization loops.

## 9. Conclusion

This project set out with a clear and ambitious goal: to optimize the transportation of food products across a distributed warehouse network of a national-scale food manufacturing company. The overarching objective was to reduce costs by at least 40% and improve delivery times by at least 30%, while accounting for a wide range of real-world constraints such as supply-demand mismatches, cold chain logistics, warehouse space restrictions, and geographic limitations. Through a structured and iterative ...

Over six optimization versions, the trade-offs between time and cost, hard and soft constraints, and demand fulfillment vs. operational feasibility were explored. Each iteration yielded valuable lessons:

- Focus on cost (v3.2) can leave delivery times high and unmet demand unsolved.
- Time-centric model (v3.3) improves delivery speed but increases operational cost.
- Balanced approaches (v3.4, v3.6) provide the best compromise between efficiency and feasibility.
- Hard delivery caps (v3.5) severely restrict feasibility and increase unmet demand.
- Hybrid models with soft penalties (v3.6) achieve the best real-world results—delivering a 42.9% cost savings with 100% demand fulfillment and a modest 6% improvement in delivery times.

The current model is an attempt to showcase network optimization with a snapshot data to highlight the business value that it can offer. Extending this model would yield far reaching value of cost savings and efficiency for the Organization. Future work could incorporate live data feeds, integrate weather or traffic APIs, or embed the model into cloud platforms for real-time optimization.

## 10. References


- Bertsimas, D., & Tsitsiklis, J. N. (1997). Introduction to Linear Optimization, Athena Scientific.
- Dantzig, G. B. (1951). Maximization of a linear function of variables subject to linear inequalities. Activity Analysis of Production and Allocation, 13(3), 339–347.
- Hillier, F. S., & Lieberman, G. J. (2021). Introduction to Operations Research (11th ed.). McGraw-Hill Education.
- Winston, W. L. (2004). Operations Research: Applications and Algorithms (4th ed.). Duxbury Press.
- Python Software Foundation. (2023). PuLP: A Linear Programming Toolkit for Python. Retrieved from https://coin-or.github.io/pulp/
- Google Developers. (2023). Google OR-Tools for Optimization. Retrieved from https://developers.google.com/optimization
- COIN-OR. (2023). CBC (Coin-or branch and cut). Retrieved from https://github.com/coin-or/Cbc
- Gurobi Optimization. (2023). Gurobi Optimizer Reference Manual. Retrieved from https://www.gurobi.com/documentation/
- IBM. (2023). IBM ILOG CPLEX Optimization Studio. Retrieved from https://www.ibm.com/products/ilog-cplex-optimization-studio