MIT MANAGEMENT SLOAN SCHOOL

OPERATIONS RESEARCH CENTER

Enhancing Efficiency in Microchip Distribution: Strategic Supply Chain Route Optimization

Vincent Tian (vinnyt@mit.edu)
Zeki Yan (zikaiyan@mit.edu)

November 2023

Contents

1 Introduction

A global microchip producer supplied a dataset from their outbound logistics network. This dataset includes demand information for 1000 orders which must be directed through their distribution system, consisting of 19 warehouses, 11 ports of departure, and a single port of arrival.

The project is to design an optimal distribution network that incorporates warehouses, shipping routes, and courier services to create the most economical supply chain possible. The goal is to minimize the total costs, comprising warehouse operations and transportation expenses while adhering to the constraints of demand, supply, and shipping logistics.

2 Data Source

The data for this project is from Brunel University London's public supply chain datasets.

- Link to Data Set
- Detailed descriptions of this dataset can be found in the paper: *Accelerating supply chains with Ant Colony Optimization across a range of hardware solutions*

Table 1: Tables in the dataset

Table	Explanation
OrderList	All orders that need to be assigned a route.
FreightRates	All available couriers, weight gaps for each individual lane, and rates associated.
WhCosts	Specifies the cost associated with storing products in a given warehouse, measured in dollars per unit.
WhCapacities	Lists warehouse capacities measured in the number of orders per day.
ProductsPerPlant	All supported warehouse-product combinations.
VmiCustomers	All special cases where a warehouse is only allowed to support specific customers, while any other non-listed warehouse can supply any customers.
PlantPorts	Allowed links between warehouses and shipping ports in the real world.

3 Problem Description

3.1 Index Notation

- $k \in [n_{\text{order}}]$ - order number
- $i \in [n_{\text{warehouse}}]$ - warehouse number
- $e \in [n_{\text{wport}}]$ - warehouse port number
- $j \in [n_{\text{freight}}]$ - freight number
- $u \in [n_{\text{product}}]$ - product number
- $v \in [n_{\text{customer}}]$ - customer number

3.2 Input Data

3.2.1 Order Data

The following are data of order k:

- q_k - unit quantity of order k
- s_k - service delivery:

$$s_k = \begin{cases} 1, & \text{if Door-to-Door (DTD) or Door-to-Port (DTP)} \\ 0, & \text{if Customer Referred Freight (CRF)} \end{cases}$$

- ot_k - maximum delivery time for order k
- ow_k - weight for order k
- pro_k - product type of order k
- cus_k - customer of order k

3.2.2 Freight Data

The following are data of freight j:

- o_j - original port
- d_j - destination port - in this problem, the only destination port is PORT09
- c_j - carrier number
- t_j - delivery time
- m_j - transportation mode:

$$m_j = \begin{cases} 1, & \text{if Ground} \\ 0, & \text{if Air} \end{cases}$$

- r_j - transportation rate cost per unit
- $minc_j$ - minimum transportation cost
- $maxw_j$ - maximum transportation weight

3.2.3 Warehouse Data

The following are data of warehouse i:

- p_i - storage cost per unit in warehouse i
- cap_i - order capacity per day in warehouse i
- PW_{ui}

$$PW_{ui} = \begin{cases} 1, & \text{if product } u \text{ can be stored in warehouse } i \\ 0, & \text{otherwise} \end{cases}$$

- WP_{ie}

$$WP_{ie} = \begin{cases} 1, & \text{if warehouse } i \text{ connects to warehouse port } e \\ 0, & \text{otherwise} \end{cases}$$

- WC_{iv}

$$WC_{iv} = \begin{cases} 1, & \text{if warehouse } i \text{ can serve customer } v \\ 0, & \text{if otherwise} \end{cases}$$

4 Baseline Model: Yan-Tian Greedy Algorithm

We developed a baseline model for addressing this problem, known as the Yan-Tian Greedy Algorithm. The primary concept behind this approach involves a systematic iteration through all incoming orders. For each order, we initiate a search through available warehouses and their corresponding freight options, starting from the beginning of the list. The algorithm then assigns the order to the first suitable warehouse-freight pair it encounters, following a thorough evaluation to ensure that all necessary conditions are met before making the assignment. The pseudo code is shown in the appendix as Algorithm 1. The cost of solution produced by the baseline model is **$8,878,241.89**.

5 Formulating Optimization Model

The problem can be formulated as the following MIO.

5.1 Decision Variables

- X_{ki} where $k \in [n_{\text{order}}]$, $i \in [n_{\text{warehouse}}]$ - order k should be assigned to warehouse i
- Y_{kj} where $k \in [n_{\text{order}}]$, $j \in [n_{\text{freight}}]$ - order k should be assigned to freight assignment j

5.2 Objective Function

Some key things to note for the transportation cost:

- if $s_k = 0$ then the transportation cost is 0
- if $m_j = 1$ then the transportation cost is proportional to the weight consumed by the freight assignment order kj in respect of the total weight for the freight assignment order kj
- a minimum charge $minc_j$ is applied in cases where the air transportation is less than the minimum charge

Following from above:

$$\text{Objective} = \min(\text{Warehouse Cost} + \text{Transportation Cost})$$

$$\min \sum_{k=1}^{n_{\text{order}}} \sum_{i=1}^{n_{\text{warehouse}}} X_{ki} \cdot p_i \cdot q_k + \sum_{k=1}^{n_{\text{order}}} \sum_{j=1}^{n_{\text{freight}}} Y_{kj} \cdot TC_{kj}$$

5.3 Constraints

Converting the above constraints into mathematical notation:

- each order needs to be assigned to a warehouse

$$\sum_{i=1}^{n_{\text{warehouse}}} X_{ki} = 1, \forall k$$

- each order needs to be assigned to a freight assignment

$$\sum_{j=1}^{n_{\text{freight}}} Y_{kj} = 1, \forall k$$

- each warehouse has a daily order capacity

$$\sum_{k=1}^{n_{\text{order}}} X_{ki} \leq cap_i, \forall i$$

- each product can be stored in some warehouses only

$$\text{If } PW_{pro_k i} = 0, \text{ then } X_{ki} = 0$$
$$X_{ki} \leq M(PW_{pro_k i}), \forall k, i$$

  where M is the Big-M.

- some warehouses can only service certain customers

$$\text{If } WC_{icus_k} = 0, \text{ then } X_{ki} = 0$$
$$X_{ki} \leq M(WC_{icus_k}), \forall k, i$$

  where M is the Big-M.

- each warehouse can only begin transporting things via some specific warehouse ports

$$\text{If } X_{ki} = 1 \text{ and } WP_{io_j} = 0 \text{ then } Y_{kj} = 0$$
$$Y_{kj} \leq M(1 - X_{ki} + WP_{io_j}), \forall k, i, j$$

  where M is the Big-M.

- orders need to be shipped within a certain time to the customer

$$\sum_{j=1}^{n_{\text{freight}}} Y_{kj} \cdot t_j \leq ot_k, \forall k$$

- for each route via a carrier, different parts of the carrier should not exceed a maximum weight

$$\sum_{k=1}^{n_{\text{order}}} Y_{kj} \cdot ow_k \leq maxw_j, \forall j$$

- total transportation cost

$$TC_{kj} \leq s_k[(1 - m_j) \cdot TCA_{kj} + m_j \cdot TCG_j], \forall k, j$$

- air transportation cost

$$minc_j \leq TCA_{kj}, \forall k, j$$
$$ow_k \cdot r_j \leq TCA_{kj}, \forall k, j$$

- ground transportation cost

$$z_j r_j \leq TCG_j, \forall j$$
$$z_j \leq \sum_{k=1}^{n_{\text{order}}} Y_{kj}, \forall j$$
$$Y_{kj} \leq z_j, \forall k, j$$

  where z_j is a binary variable.

6 Data Pre-Processing

Due to many variables in the dataset not being of factor type eg. PORT09, mapping tables were created to map each of these values to a number. For eg. 1 → PORT09. Mapping tables were created for the following variables:

- Order ID, Warehouse ID, Product ID, Origin Port, Carrier, Customer, Service Level, Destination Port, Transportation Mode

A variable to capture the maximum delivery time for order k, ot_k was also created

- Maximum delivery time = TPT + Ship ahead day count + Ship late day count + 2

Also, due to the large number of orders and freight paths, 1000 orders were randomly sampled and freight paths were reduced through aggregation.

7 Exploratory Data Analysis

Figure 1 shows that most warehouses are connected to only a single warehouse port and many warehouses are connected to warehouse port 4. This suggests that many warehouses may only be sent via one freight and many orders may be sent through freights going through warehouse port 4. Figure 2 shows that most products are only ordered a small number of times.

Figure 3 shows that most warehouses have negative correlation between the cost per unit cost and the daily order capacity. We should expect the warehouses with lower cost per unit to have the most number of orders allocated to it.

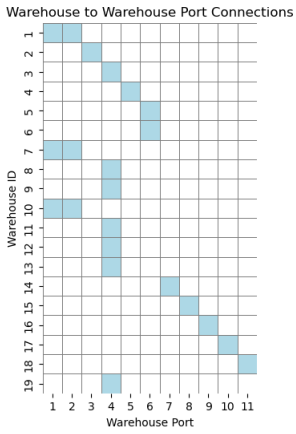

Figure 1: Warehouse to Warehouse Port Connections

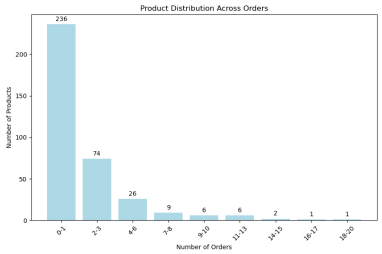

Figure 2: Distribution of Products Across Orders

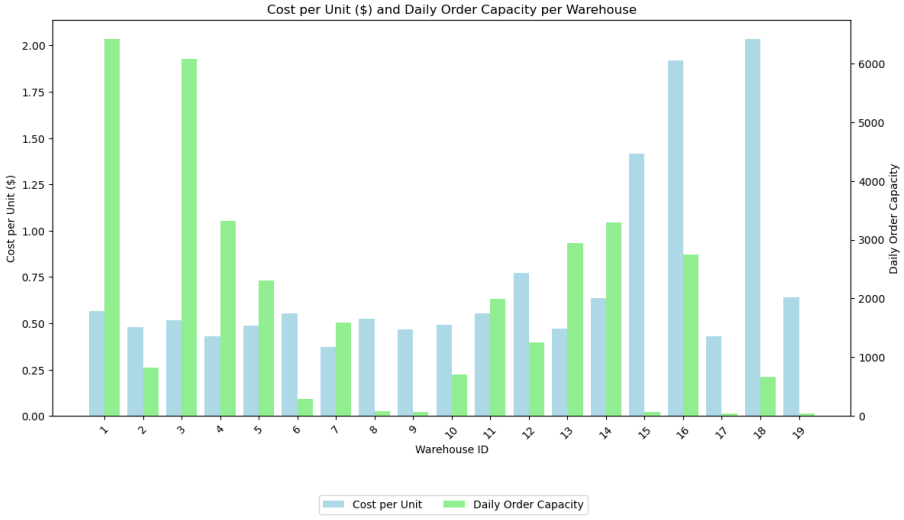

Figure 3: Cost per Unit ($) and Daily Order Capacity Per Warehouse

8 Results

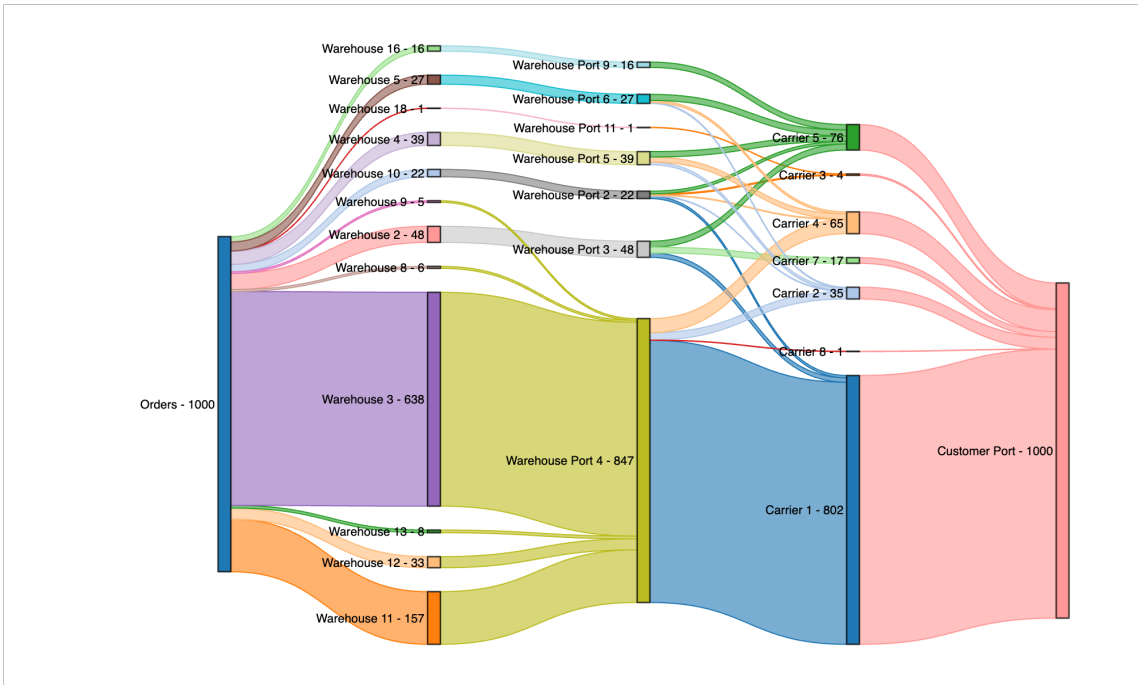

Figure 4: Optimization Result - Sankey Diagram of Order Assignment

Figure 4 shows the optimal warehouse and freight allocation for each of 1000 orders. Note that freight allocation is composed of a warehouse port and carrier.

Total cost reduction:

- The optimal cost of our solution is **$5,365,566.57**.
- The cost of solution produced by the baseline model is **$8,878,241.89**.
- Our solution reduced the total cost by **$3,512,675.32**, which is **39.5%** reduction in cost.

9 Conclusion

The solution produced by our model surpasses the previously provided solution by the freight company.

- We successfully reduced the total cost, including storage cost and transportation cost by 39.5%.

The outcomes obtained in our study align with the expectations derived from our preliminary exploratory analysis.

- many orders to pass through warehouse port 4.
- many orders allocated to warehouse 3 and warehouse 11 due to their lower daily cost per unit.
- not many orders allocated to warehouse 15, 16 or 18 due to their high daily cost.

10 Next Steps

Here are some ideas for future model improvement:

1. Split orders into multiple orders which may be fulfilled through different warehouses.
2. If a warehouse has reached full capacity, allow some orders to still be fulfilled there the next day is delivery time can still be met.

References

[1] Brunel University London. (2019). Supply Chain Logistics Problem Dataset. Brunel University London. Retrieved from https://brunel.figshare.com/articles/dataset/Supply_Chain_Logistics_Problem_Dataset/7558679

[2] Dzalbs I, Kalganova T. (2020). Accelerating supply chains with Ant Colony Optimization across a range of hardware solutions. *ScienceDirect*. Retrieved from https://www.sciencedirect.com/science/article/pii/S0360835220303442?via%3Dihub

[3] Brechter, L. (n.d.). Supply Chain Data. Kaggle. Retrieved from https://www.kaggle.com/datasets/laurinbrechter/supply-chain-data

[4] Bertsimas, D. & Tsitsiklis, J. (1997), Introduction to linear optimization , Athena Scientific .

[5] Sankey Diagrams in Python. Retrieved from https://medium.com/@cbkwgl/sankey-diagrams-in-python-fc9673465ccb

Appendix

Yan-Tian Greedy Algorithm

Algorithm 1 Yan-Tian Greedy Algorithm For Order Assignment

```
// Assign orders to warehouse and freight
for order_k from 1 to n_order do
  for warehouse_i from 1 to n_warehouse do
    for freight_j from 1 to n_freight do
      if warehouse_i can produce order_k's product then
        if warehouse_i can serve order_k's customer then
          if warehouse_i has capacity then
            if freight_j has capacity then
              if warehouse_i can transport products to freight_j's warehouse port then
                if freight_j's transportation time satisfy order_k's demanding time then
                  Assign order_k to warehouse_i and freight_j
                  BREAK
                end if
              end if
            end if
          end if
        end if
      end if
    end for
  end for
end for
// Calculate transportation cost TC and Warehouse Cost WC, Penalty Cost PC, and total cost C
Initialize TC = 0, WC = 0, PC = 0, and C = 0
for warehouse_i from 1 to n_warehouse do
  WC = WC + warehouse_i's cost
end for
for freight_j from 1 to n_freight do
  TC = TC + freight_j's cost
end for
for order_k from 1 to n_order do
  if order_k is not assigned then
    PC = PC + unit penalty cost
  end if
end for
C = PC + TC + WC
// Output results
```

Compact Optimization Model

$$
\begin{aligned}
\min \quad & \sum_{k=1}^{n_{\text{order}}} \sum_{i=1}^{n_{\text{warehouse}}} X_{ki} \cdot p_i \cdot q_k + \sum_{k=1}^{n_{\text{order}}} \sum_{j=1}^{n_{\text{freight}}} Y_{kj} \cdot TC_{kj} \\
\text{s.t.} \quad & \sum_{i=1}^{n_{\text{warehouse}}} X_{ki} = 1, \forall k; \quad \sum_{j=1}^{n_{\text{freight}}} Y_{kj} = 1, \forall k \\
& \sum_{k=1}^{n_{\text{order}}} X_{ki} \leq cap_i, \forall i; \quad \sum_{k=1}^{n_{\text{order}}} Y_{kj} \cdot ow_k \leq maxw_j, \forall j \\
& X_{ki} \leq M(PW_{pro_k i}), \forall k, i \\
& X_{ki} \leq M(WC_{icus_k}), \forall k, i \\
& Y_{kj} \leq M(1 - X_{ki} + WP_{io_j}), \forall k, i, j \\
& \sum_{j=1}^{n_{\text{freight}}} Y_{kj} \cdot t_j \leq ot_k, \forall k \\
& TC_{kj} \leq s_k[(1 - m_j) \cdot TCA_{kj} + m_j \cdot TCG_j], \forall k, j \\
& minc_j \leq TCA_{kj}, \forall k, j; \quad ow_k \cdot r_j \leq TCA_{kj}, \forall k, j \\
& z_j r_j \leq TCG_j, \forall j; \quad z_j \leq \sum_{k=1}^{n_{\text{order}}} Y_{kj}, \forall j; \quad Y_{kj} \leq z_j, \forall k, j
\end{aligned} \tag{1}
$$

Input Data Structure

Order
Order ID : integer
Order Date : date
Origin Port : string
Carrier : string
TPT : integer
Service Level : string
Ship ahead day count : integer
Ship Late Day count : integer
Customer : string
Product ID : integer
Warehouse : string
Destination Port : string
Unit quantity : integer
Weight : float

FreightRates
Carrier : string
orig_port_cd : string
dest_port_cd : string
minm_wgh_qty : float
max_wgh_qty : float
svc_cd : string
minimum cost : float
rate : float
mode_dsc : string
tpt_day_cnt : integer
Carrier type : string

WhCosts
Warehouse : string Cost/unit : float

WhCapacities
Warehouse : string Daily Capacity : integer

ProductsPerPlant
Warehouse : string Product ID : integer

VmiCustomers
Warehouse : string Customer : string

PlantPorts
Warehouse : string Origin Port : string