Minimizing Cost Flows and Deciding on a New Warehouse

By Iva Boishin, Justine Deleval, Daniela Tuiran

# Business Problem & Data

*A pied* is a company that produces and distributes shoes across France. The company is composed of two plants that manufacture the shoes, which are then sent to the four warehouses and shipped to the six stores. Instead of going through the warehouses, the shoes can also be sent to the stores directly from the plants. The main goal of our company is to maximize profit while still satisfying store demand.

The plants and warehouses have capacity constraints that they cannot exceed, and which are listed in the table below. Stores also have demand constraints that the company must meet.

To reduce delivery costs, *A pied* is also facing the choice of building a new warehouse to allow easier and faster delivery. We will use the model to decide whether it could be profitable to build this new warehouse.

*Below are the capacity and demand in terms of units for the plants, warehouses, and stores:*

|  |  |
| --- | --- |
| Plant | Capacity |
| 1 | 150,000 |
| 2 | 200,000 |

|  |  |
| --- | --- |
| Warehouse | Capacity |
| 1 | 70,000 |
| 2 | 250,000 |
| 3 | 100,000 |
| 4 | 40,000 |
| Project Warehouse 5 | 55,000 |

|  |  |
| --- | --- |
| Store | Demand |
| 1 | 50,000 |
| 2 | 10,000 |
| 3 | 40,000 |
| 4 | 35,000 |
| 5 | 60,000 |
| 6 | 20,000 |

In addition to capacity constraints, delivery costs must be considered. Below is a matrix of the cost of delivering one unit between the different entities (1 and 2 being the plants, 3 through 6 being the warehouses, 7 through 12 being the stores and 13 being the potential additional warehouse). The rows represent from where the shoes leave and the columns represent where the shoes travel to. The zeros correspond to the zeros of the adjacency matrix, showing that the units will never flow through a path connected by a zero.

# Model & Software Implementation

Using the minimum cost flow method, we built a model to minimize the cost of delivering the shoes through our network respecting the constraints that are imposed.

We implemented the model in AMPL by defining parameters, variables, an objective function and constraints.

**The parameters**

Total number of plants, warehouse and stores

 Setting up a very high C number allow the model to find the best quantity for the permitted path considering some constraints

 Costs of building the links from 1 to 13

 Cost of building the new warehouse (13)

Capacity of each warehouse. We set up a separate parameter for warehouse capacity, where very high capacities were allowed for the plants and stores to automatically enable the flow and have it later limited by the balance function.

 Adjacency matrix that list the output connections with other nodes (1: connection available, 0: no connection possible)

 Delivery cost between the nodes

 The balance variable, which inherently specifies the production capacity of the plants (positive flow) and the demand of the clients (negative flow)

**The variables**

 represents the quantity that can flow from i to j (either the plants to the warehouse, warehouse to stores and/or plants to stores).

 is a binary variable to decide whether it is profitable to build the new warehouse.

 is a binary variable that allows the flow (links) out of the new warehouse to the stores.

**The objective function**

The mathematical formula for minimizing the cost of delivering the shoes is represented below:

Using AMPL, we are summing the product of the total delivery cost from i to j (1 to 13 representing the plants, warehouse, and stores) with the product of the cost to build the new warehouse, where the binary variable y will be equal to 1 if the decision is to build or 0 if the decision is not to build, making the cost equal to 0. The last step is to add the sum of the cost of building the links between the new warehouse and the other nodes.

**The constraints**

**Subject to Balance**

We set up a constraint that calculates the difference between the sum of flow in from i to j and the sum of flow out from i to j. This difference must be less than or equal to the capacity that each plant and store can handle in order to have enough products produced for the given level of demand. If the balance function is negative instead, the problem is infeasible because the company cannot meet the store demand.

**Subject to Capacity**

We set up a constraint to limits the capacity that each warehouse cannot exceed. We are summing the quantity that can flow from i to j (with j going from 3 to 13 to reduce some computational time since we are not concerned with the plant capacity here). This sum must be less than or equal to the actual capacity of the warehouse.

**Subject to Bounds**

In terms of bounds, the flow from i to j must follow the adjacency matrix. Since this matrix is binary, once a flow is not allowed (value of 0 in the matrix), the quantity will immediately 0 therefore restricting the flow. If the flow is allowed (value of 1 in the matrix), then x will take the optimal value considering all the other constraints. We set up a value of 500000 (very high number) to allow the model to find the best quantity for the permitted path considering the capacity and demand constraints.

**Subject to New Links In and New Links Out**

We set up a constraint for the flow in from the project new warehouse to the stores. The quantity flowing from warehouse 13 to the potential stores must be less than or equal than C (which represent a very big number) times the binary variable w that allows a link or not. If w is equal to 0, then it sets up a constraint in which no quantity can flow from warehouse 13 into the stores. If w is equal to 1, then the link is built and the flow is allowed. Setting up a very high number for C allow the model to find the best quantity for the permitted path considering the capacity and demand constraints. The same applies to links out.

**Subject to New Warehouse**

Considering all the other constraints, we set up the binary variable w to be less than or equal to the binary variable y that is the decision of building a new warehouse or not. If y is equal to 0, then the decision is not to build the warehouse, restricting w to be 0 therefore not building a link neither the warehouse. On other hand, if y is equal to 1 it then allows the link and the flow out of this new warehouse.

**Subject to Self-Link**

We set up a constraint that restrict the link of warehouse 13 to 0 since there is no flow that can go from warehouse 13 to itself.

# Numeric Results

Using the cplex solver method and AMPL software, after 31 simplex iterations we found the optimal solution for the problem under consideration. The objective function takes a value of € 204,004 for the cost of transportation through the network. The diagram represents the optimal paths that will be taken by the model; the paths in red go from plants to warehouses, the paths in green go from plants to stores and blue from warehouses to stores.



The matrix below shows the resulting values for the quantities that will be transported between each node. The rows in the matrix represents the quantities going out the corresponding node to their optimal destination.

# Further Avenues of Improvement

In this example, we only considered the flow of one type of product through various levels and made a decision whether or not to build a new warehouse. For a future product, it could be interesting to create a similar model with multiple products, where each warehouse not only has a capacity constraint but a rent cost as well. The consideration of different types of products would make this optimization example a little more realistic as stores don’t only sell one product. Additionally, the extra cost constraint might change the current cost flow as the marginal cost decrease of delivery costs of one channel might not be worth the added costs of renting the extra warehouse. This is most likely to be the case of Warehouse 1 (node 3) as it only stores with a small amount of products, which might not be worth the larger added cost of renting the additional warehouse.