**Revisiting the Hunt Company Example from Module 14. Using Gurobi.**

***The best example to review from the Gurobi documentation called “facility” and “multiscenario”***

***PDF page 465-468; 485-490 &*** [***https://www.gurobi.com/resource/modeling-examples-using-the-gurobi-python-api-in-jupyter-notebook/***](https://www.gurobi.com/resource/modeling-examples-using-the-gurobi-python-api-in-jupyter-notebook/)

**Example:** Hunt Company produces tomato sauce at five different plants. The capacity (in tons) of each plant is given in Table 1. The tomato sauce is stored at one of three warehouses. The per-ton cost (in hundreds of dollars) of producing tomato sauce at each plant and shipping it to each warehouse is given in Table 2. Hunt has four customers. The cost of shipping a ton of sauce from each warehouse to each customer is as given in Table 3. Each customer must be delivered the amount (in tons) of sauce given in Table 4. Fixed costs for opening plants and warehouses are provided in Table 5. Prepare a formulation and solve to minimize total cost.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table 1 |  |  |  |  |  |
|  | Plant | | | | |
|  | 1 | 2 | 3 | 4 | 5 |
| Tons | 300 | 200 | 300 | 200 | 400 |

|  |  |  |  |
| --- | --- | --- | --- |
| Table 2 |  |  |  |
|  | To | | |
|  | Warehouse 1 | Warehouse 2 | Warehouse 3 |
| Plant 1 | 8 | 10 | 12 |
| Plant 2 | 7 | 5 | 7 |
| Plant 3 | 8 | 6 | 5 |
| Plant 4 | 5 | 6 | 7 |
| Plant 5 | 7 | 6 | 5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table 3 |  |  |  |  |
|  | To | | | |
|  | Customer 1 | Customer 2 | Customer 3 | Customer 4 |
| Warehouse 1 | 40 | 80 | 90 | 50 |
| Warehouse 2 | 70 | 70 | 60 | 80 |
| Warehouse 3 | 80 | 30 | 50 | 60 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table 4 |  |  |  |  |
|  | Customer | | | |
|  | 1 | 2 | 3 | 4 |
| Demand | 200 | 300 | 150 | 250 |

|  |  |
| --- | --- |
| Table 5 |  |
|  | Fixed Annual Cost (in thousands) |
| Plant 1 | 35 |
| Plant 2 | 45 |
| Plant 3 | 40 |
| Plant 4 | 42 |
| Plant 5 | 40 |
| Warehouse 1 | 30 |
| Warehouse 2 | 40 |
| Warehouse 3 | 30 |

**Formulation:**

Indexed Sets:

i: plant #

j: warehouse #

k: customer #

(Note, we will use two derived indexed sets: ij and jk)

Data: (Hint – look at the tables!)

Ui = the capacity of plant i in tons

Cij = production and shipping cost from plant i to warehouse j

Sjk = shipping cost from warehouse j to customer k

Dk = demand for customer k in tons

Fi = fixed cost for plant i

Gj = fixed cost for warehouse j

T = total demand

Variables:

Xij = tons produced at plant i and shipping to warehouse j

Yjk = tons shipped from warehouse j to customer k

Mi = 1 if plant i is open, 0 otherwise

Wj = 1 if warehouse j is open, 0 otherwise

Minimize Total Cost (Similar to fixed charge problem – but two things – plants and warehouses.):

Minimize

Constraints:

for all k (Meet demand)

for all i (Don’t exceed plant capacity)

for all j (Node balance at warehouses)

for all j (Don’t exceed warehouse capacity)

Xij >= 0, Yjk >= 0, Mi ~ binary, Wj ~ binary for all combinations

\*\*\*Note, there are stronger formulas (i.e., we could add some additional constraints to help), but the above formulation will work.\*\*\*

*Original Formulation (above): mod18-original*

Benefit to solving the problem as a Linear Program (i.e., eliminating the binary constraints).

{Changing variable restrictions.}

*Continuous Formulation: mod18-cont*

Need to add more constraints (that are still valid) to make the formulation stronger.

*Thought process … if warehouse j ships anything to customer k, then warehouse j must be open and the most we can ship is Dk.*

What about adding the following constraints, in place of the last constraint?

Yjk <= Dk \* Wj for all j and k pairs

*Thought process: If plant i ships anything to warehouse j, then warehouse j must be open and plant i has to be open, most we can ship is Ui.*

What about adding the following constraints? The goal is to decouple (disaggregate) the X’s and Y’s with respect to the warehouse.

Xij <= Ui\*Wj for all i and j pairs

(number of inventory goes in needs to be <= plant capacity x warehouse)

*Final Formulation: mod18-final*

Recap:

Adding more constraints to reduce the number of integer (binary) variables. Problem is easier to solve.

Original IP Formulation: 671,500

Relaxed LP Formulation: 641,500 (with Warehouses not binary due to T = 900)

Relaxed LP Formulation with both sets of new constraints: 671,500 (with Warehouses binary)

*Now we can use the above formulation to solve for reduced costs and shadow prices if we want.*