SHOP ORDERING PROBLEM

PARAMETERS AND DECISION VARIABLES:

for simulation in python ↓

Θ : set of products → num_products

Λ_i : set of suppliers with product i → num_suppliers

T : time period → time_period

C_{Fj} : fixed cost for supplier j → min_fixed_cost ≤ C_{Fj} ≤ max_fixed_cost

e_i^{τ} : extracost due to unsatisfied demand → min_extracost ≤ e_i^{τ} ≤ max_extracost

C_{ij} : cost of one item i from supplier j → min_cost ≤ C_{ij} ≤ max_cost

$u(\,)$: discount function

h_i : unitary holding cost for product i → min_holding_cost ≤ h_i ≤ max_holding_cost

I_i^{τ} : inventory of product i @ time τ

D_i^{τ} : demand of product i @ time τ

M : arbitrary big number → M

p_i : selling price of product i → min_price ≤ p_i ≤ max_price

T_i : time for the product i to arrive → min_time_steps ≤ T_i ≤ max_time_steps

O_{ij}^{τ} : amount of product i ordered from supplier j @ time τ

$$y_j^{\tau} = \begin{cases} 1 & \text{if supplier } j \text{ is chosen for an order @ time } \tau \\ \emptyset & \text{otherwise} \end{cases}$$

$$b_i^{\tau} = \max\left[D_i^{\tau} - I_i^{\tau}, 0\right]$$

MODEL:

OBJECTIVE FUNCTION:

$$\max \left[\min\left[D_i^{\tau}, I_i^{\tau}\right] \cdot p_i - \left[\sum_{j \in \Lambda_i} \sum_{\tau \in T} C_{Fj}\, y_j^{\tau} + \sum_{i \in \theta} \sum_{j \in \Lambda_i} \sum_{\tau \in T} C_{ij} \cdot O_{ij}^{\tau} + \right.\right.$$

$$\left.\left. - \sum_{j \in \Lambda_i} u \left(\sum_{i \in \theta} \sum_{\tau \in T} O_{ij}^{\tau} \right) + \sum_{i \in \theta} \sum_{\tau \in T} h_i\, I_i^{\tau} + \sum_{i \in \theta} \sum_{\tau \in T} e_i^{\tau}\, b_i^{\tau} \right] \right]$$

SUBJECT TO:

$$I_i^{\tau} = I_i^{\tau-1} + O_i^{\tau-\tau_i} - D_{\tau}$$

$$b_i^{\tau} \geq D_i^{\tau} - I_i^{\tau}$$

$$\sum_{i \in \theta} O_{ij}^{\tau} \leq M \cdot y_j^{\tau} \qquad \forall j \in \Lambda_\theta, \forall \tau \in T$$

$$O_{ij}^{\tau} \geq 0 \qquad \forall i \in \theta, \forall j \in \Lambda_\theta, \forall \tau \in T$$

$$y_j^{\tau} \in \{0,1\} \qquad \forall j \in \Lambda_\theta, \forall \tau \in T$$

instance.py

- initial condition on Inventory and demand