RL for Blending

March 2024

1 Introduction

In this document we discuss conceptual and technical elements of implementing and solving, first the pooling problem, then the blending problem, using reinforcement learning.

2 Implementation details - Simple pooling problem

The following table proposes how elements of the Pooling problem should be incorporated to a Reinforcement learning problem. Please refer to annex A (page 41) of reference [1] for variable meanings.

This element...	**...is a part of...**
Flow Connection Graph (Adjacencies)	The environment (fixed at environment initialization)
Environment parameters (Source costs, Demand prices, Initial Concentrations, Concentration Requirements, Max Requirement)	The observation (randomized at each new step/episode)
Variables ($F_{i,j}, F_X, F_Y$)	The action (to be provided by the model)
Objective function/profits	The reward (calculated from Env. costs & prices)

Since the pooling problem takes place over a single time step, our environment would terminate after a single step. This means there is no need for a transition function.

3 Implementation details - Blending problem

3.1 Environment definition

The following table proposes how elements of the Blending problem should be incorporated to a Reinforcement learning problem. Please refer to Figure 1 (page 5) of reference [1] for variable meanings.

This element...	**...is a part of...**
Flow Connection Graph (Adjacencies)	The environment (fixed at environment initialization)
Environment parameters (*Ref [1], Table 2*)	Fixed at environment initialization OR randomized at each new epoch
Variables (*Ref [1], Table 1*)	The action (to be provided by the model at each time step)
Objective function/profits	The reward (calculated from β_s^T, β_d^T, $\alpha_{nn'}, \beta_{nn'}$)

This time, each time step of the blending process can correspond to a single step in our environment.

It should be noted that in the pooling problem, since everything happens at the same timestamp, tanks charge and discharge simultaneously. In the blending problem, tanks have to either charge or discharge during a given timestamp. This is another difference that will need to be reflected in the implementation, and possibly in the model as well.

Additionally, it should be determined which parameters should be made constant (fixed for each environment) and which ones should be randomized at each episode. We want enough randomized variables for the model to learn how to act on a broad range of environments, but there is a risk that the model doesn't learn anything if there are too many actions to be taken at the same time. This shall be determined by experimentations.

3.2 Transition function

...

3.3 Policy model

See reference [2]

3.4 Demand prediction

A function of the form $D_q(t) = a\sin(bt) + c + e(t)$ where a, b, c are constants chosen appropriately and $e(t)$ is small Gaussian noise, seems fit for our problem. It reflects seasonality and irregularity in the demand.

4 MDP Formulation

Note: $\mathcal{S}$ denotes the set of source tanks, $\mathcal{J}$ denotes the set of blending tanks, $\mathcal{P}$ denotes the set of demand (or "product") tanks, $\mathcal{Q}$ is the set of properties (or chemicals) being blended.

4.1 State

The inventories (before flow occurs) of source, blending and product tanks for each property at time step t

$$\begin{gathered}(I_{s,t}) \quad s \in \mathcal{S} \\ (I_{j,t}) \quad j \in \mathcal{J} \\ (I_{p,t}) \quad p \in \mathcal{P} \\ (C_{q,j,t}) \quad q \in \mathcal{Q}, j \in \mathcal{J} \\ (\tau^0_{q,s,t}) \quad q \in \mathcal{Q}, s \in \mathcal{S} \\ (\delta^0_{q,p,t}) \quad q \in \mathcal{Q}, p \in \mathcal{P}\end{gathered}$$

$\tau^0_{q,s,t}$ is the maximum amount of property q we can buy from source s at time step t
$\delta^0_{q,p,t}$ is the maximum amount of property q we can sell to demand tank p at time step t
Add costs/prices in the state

4.2 Action

The flows between each tank at time step t

$$\begin{gathered}(F_{s,j,t}) \quad s \in \mathcal{S}, j \in \mathcal{J} \\ (F_{j,j',t}) \quad j, j' \in \mathcal{J} \\ (F_{j,p,t}) \quad j \in \mathcal{J}, p \in \mathcal{P} \\ (\tau_{s,t}) \quad s \in \mathcal{S} \\ (\delta_{p,t}) \quad p \in \mathcal{P}\end{gathered}$$

We do not require the binary $X_{\cdot,\cdot,t}$ variables, as we can derive them from the action.

4.3 Transition function

We need to introduce the $\tau_{s,t}$ decision variable to represent how much product we "buy" from source s at time t. This variable is within the range $[0, \xi_{s,t}]$

$$I_{s,t+1} = I_{s,t} + \tau_{s,t} - \sum_j F_{s,j,t} \quad s \in \mathcal{S}$$

$$I_{j,t+1} = I_{j,t} + \sum_s F_{s,j,t} + \sum_{j'} F_{j',j,t} - \sum_{j'} F_{j,j',t} - \sum_p F_{j,p,t} \quad j \in \mathcal{J}$$

$$I_{p,t+1} = I_{p,t} - \delta_{p,t} + \sum_j F_{j,p,t} \quad p \in \mathcal{P}$$

$$C_{q,j,t+1} = \frac{1}{I_{j,t+1}}(I_{j,t}C_{q,j,t} + \sum_s \sigma_{s,q}F_{s,j,t} + \sum_{j'} C_{q,j',t}F_{j',j,t} - \sum_{j'} C_{q,j,t}F_{j,j',t} - \sum_p C_{q,j,t}F_{j,p,t}) \quad q \in \mathcal{Q}, j \in \mathcal{J}$$

4.4 Reward

Reminder:
- $\delta_{p,t}$ is the amount of product sold at demand tank p
- $\tau_{s,t}$ is the amount of product bought at source tank s

Simple minimizing costs formulation:

$$Q_t = \alpha \sum_{s,j,j',p} (X_{s,j,t} + X_{j,j',t} + X_{j,p,t}) + \beta \sum_{s,j,j',p} (F_{s,j,t} + F_{j,j',t} + F_{j,p,t})$$

Simple maximizing profits formulation:

$$R_t^1 = \sum_p \beta_p^T \min(\delta_{p,t}, I_{p,t}) - \sum_s \beta_s^T \tau_{s,t} - Q_t$$

Maximizing profits formulation adapted to the constraints:

$$R_t^2 = R_t^1 - \sum_j f(j,t) - \sum_{p,q,j} g(p,q,j,t)$$

where:

$$f(j,t) = \begin{cases} \text{M} & \text{if } \sum_{k \in \mathcal{J},\mathcal{P}} F_{j,k,t} > 0 \text{ and } \sum_{k \in \mathcal{S},\mathcal{J}} F_{k,j,t} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$g(p,q,j,t) = \begin{cases} \text{M} & \text{if } C_{q,j,t} < \sigma_{p,q}^L \text{ and } X_{j,p,t} = 1 \\ \text{M} & \text{if } C_{q,j,t} > \sigma_{p,q}^U \text{ and } X_{j,p,t} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$M \text{ is a large (positive) number} \tag{3}$$

Proportional to the flow ?
(1) reflects the fact that a blending tank can only be in charging or discharging mode, not both at the same time.
(2) reflects the quality constraints of each demand tank.

We could also consider adding similar penalties in the case of tank inventories getting exceeded, invalid flows being requested and so on.
If invalid flows are requested by the model's action, we can also indicate to the model it did not progress on to the next timestamp, and ask for an action until a valid flow is provided. We could also give a penalty for each invalid action and let the model learn by itself. Therefore, the time step should be part of the state as well.

5 Useful links & references

[1] An MILP-MINLP decomposition method for the global optimization of a source based model of the multiperiod blending problem
[2] Defining custom neural networks for Actor Critic policy in stable_baselines3