Dynamic Scheduling of Ethylene Cracking Furnaces System Leveraging Deep Reinforcement Learning

Haoran Li,a,b Yixin Wei, a,b Tong Qiua,b\*

a Department of Chemical Engineering, Tsinghua University, Beijing 100084, China

b Beijing Key Laboratory of Industrial Big Data System and Application, Tsinghua University, Beijing 100084, China

qiutong@tsinghua.edu.cn

Abstract

The profitability and resilience of traditional scheduling algorithms for ethylene steam cracking systems in the face of supply chain fluctuations are comparatively weak. To address this issue, a dynamic ethylene scheduling framework based on deep reinforcement learning is proposed in this study. Through a comparative analysis with literature cases, this framework demonstrates a notable enhancement of 5.7 % in daily revenue, showcasing strong resilience to supply chain fluctuations.

**Keywords**: deep reinforcement learning, ethylene, dynamic scheduling

* 1. Introduction

Ethylene stands as one of the most prolifically produced chemicals globally, with the prevailing method of production being steam cracking(Bi et al., 2021). Owing to the inherent volatility in the supply and demand dynamics along the value chain, the profitability of ethylene cracking furnace systems (ECFS) is susceptible to perturbations. While traditional static mixed-integer nonlinear optimization (MINLP) models effectively optimize the scheduling of various feedstocks across multiple cracking furnaces to maximize economic benefits(Li et al., 2022), their resilience to fluctuations is limited. Consequently, dynamic scheduling tailored to conditions of supply chain volatility has emerged as a pivotal strategy for enhancing the profitability of ethylene cracking furnace systems(Li et al., 2023).

In recent years, deep reinforcement learning has garnered attention from researchers in the field of process systems engineering, owing to its robust handling of stochastic scenarios. In this study, we have developed a deep reinforcement learning framework specifically tailored for the ethylene scheduling system, comprising an ethylene scheduling Markov Decision Process (MDP) environment and a Deep Q-Network (DQN) decision network to address the dynamic scheduling challenges under supply chain fluctuations. The subsequent sections are organized as follows: Part 2 outlines the definition of the ethylene scheduling problem, Part 3 provides the methodology, Part 4 presents the results of a case study, and the final conclusions are provided in Part 5.

* 1. Problem statement

The steam cracking production of ethylene is a semi-continuous process with slow dynamic shifts during the batch because of coking inside the furnace tube. The cracking furnace will be switched out and decoked once the coke accumulates to a certain extent to ensure safety. An ethylene cracking furnaces system usually consists of *NC* types of feed, *NF* cracking furnaces and *NP* products. The scheduling of ECFS on each day is to decide:

1) whether to end the current batch and perform decoking

2) the next feed type after decoking

The objective of the scheduling problem is to maximize the net profits while satisfying the constraints including non-simultaneous decoking, feed inventory, and batch length lower and upper bounds, etc.

* 1. Method



Figure 1. Framework of Dynamic Ethylene Scheduling Environment and DRL-based Decision Model.

* + 1. Ethylene Scheduling System Environment Formulation

We considered the scheduling problem of an ECFS with 3 types of feeds, 5 cracking furnaces, and 3 types of products. Then we formulated it as an MDP with state (), action (), reward (), and probability transition function ().

* + - 1. State space

The state is a 27-dimensional vector consisting of the furnace states, historical supply chain parameters, and penalty terms.

|  |  |
| --- | --- |
|  | (1) |

Where are the indexes for feeds, furnaces, and products. and are the current inlet feed type and batch length of furnace , is the current feed inventory level of feed , and is the flowrate of product. are historical value of the supply amount of feed , price of feed , and price of product .

* + - 1. Action space

The action is designed as an integer variable from [0,15] indicating the decision on whether to perform decoking and the next inlet feed type after decoking. If *a=0*, no furnaces will perform decoking. Otherwise, the decoking furnace and next feed type will be determined as Eqs. (2) and (3).

|  |  |
| --- | --- |
|  | (2) |
|  | (3) |

For example, if , then decoking will be performed on Furnace 3 on that day () and Feed B will be processed afterwards ().

* + - 1. Reward function design

The reward function is designed as net profit minus the penalty terms which are used to account for the violation of constraints, as shown in Eq. (4). The net profit is calculated by the product revenue minus the feed cost, operation cost, and the decoking cost.

|  |  |
| --- | --- |
|  | (4) |

Where and are auxiliary binary variables. If furnace is processing feed and is not under decoking then . If furnace j was processing feed and is under decoking then . is the feed processing flowrate, is the operation unit cost, and is the decoking unit cost.

 and correspond to penalties for violating hard constraints and soft constraints, respectively. The distinction lies in the fact that hard constraints are inviolable—if violated, action will be rewritten, whereas soft constraints are not. In this case, the hard constraint is the batch length lower and upper bounds. If violated, the furnaces will be forced to be decoked or not to be decoked, as shown in Eqs.(5) and (6).

|  |  |
| --- | --- |
|  | (5) |
|  | (6) |

The soft constraint is the feed inventory level. If the inventory is negative, the usage of the feed will pay a price surplus .

|  |  |
| --- | --- |
|  | (7) |

* + - 1. Probability transition function

The uncertainties originate from the fluctuating daily feed supply amount , feed prices , and product prices . On each day, once the state and action are given, the next state will be updated by the probability transition function.

If furnace is not decoked, then the will the same as and will be incremented by 1. Otherwise, will be set as and will be set as 0.

The inventory level of feed *i* will be updated by the supply amount and consumption on that day, as in Eq. (8). is the total processing amount of feed *i* on day *t*, and is determined by the .

|  |  |
| --- | --- |
|  | (8) |

The product yield model is in the exponential form, as shown in Eq. (9). And the flowrate of products will be calculated by the yields and feed flowrate processed by each furnace.

|  |  |
| --- | --- |
|  | (9) |

Where are the reaction dynamics parameters.

The supply chain parameters will be updated by the newly observed values.

|  |  |
| --- | --- |
|  | (10) |

* + 1. Deep Reinforcement Learning-based Model

A version of DQN algorithm(Mnih et al., 2013) is applied to train the deep reinforcement learning model. The Q-network () is the used the predict the future rewards of the given combination of and . The action is chosen using the ε-greedy strategy. With a probability of ε, a random action is selected; otherwise, the action with the highest prediction is chosen.

|  |  |
| --- | --- |
|  | (11) |

The target network () is used to predict the future rewards of . The loss function is the mean squared error between the prediction value and the target value, and the network is updated by the calculation of the loss on a minibatch of N samples from the experience replay buffer.

|  |  |
| --- | --- |
|  | (12) |

* 1. Case study

We utilized a case study in the literature (Liu et al., 2010). Firstly, the MINLP model in the original literature was formulated into an MDP environment following the method in Section 3.1. Then, the DQN model was trained by interacting with this environment. Finally, the scheduling decisions generated by the original MINLP and DQN models are both validated to compare the performances.

* + 1. Model training

Figure 2 shows the loss function curve and the episode reward curve during model training. The loss function value decreased drastically in the beginning and converged after around 1 million environment steps. At the same time, the episode reward increased from negative values to positive values and also stabilized after 1 million steps. The negative value was caused by the penalty terms.



Figure 2. (a) Loss Function Curve and (b) Episode Reward During Model Training.

* + 1. Scheduling performance validation

After training, the original MINLP model and the trained DQN model are tested on a 1000-day case in which the daily feed supply, feed price and product price parameters fluctuate from day to day, as shown in Figure 3. The fluctuations in these parameters bring the necessity of dynamic scheduling of ECFS because once the parameter changes, the original optimal solution will become sub-optimal or even infeasible.



Figure 3. Fluctuating supply chain parameters of (a) Daily feed supply and (b) Feed and Product Price.

Figure 4 is the Gantt chart which demonstrates the feed allocation among the five cracking furnaces using the decision results generated by the two models. Figure 4(a) is the result from the original MINLP which did not consider the parameter fluctuations in the optimization model. Thus, the Gantt chart shows a repeated pattern. On the contrast, the DQN model was trained to tackle the fluctuations and did not show the repeated pattern.



Figure 4. Scheduling Gantt Chart of (a) MINLP Model and (b) DQN Model.

Figure 5 shows the daily net profit under these two solutions. The solid line represents DQN solution and dash line represents MINLP solution. On most of the days during the testing length, the daily net profit from DQN solution is higher than MINLP solution. The profit gaps between the two solutions are especially significant at around day 400 and 550, where large net profit losses appear extensively because of the penalty from the inventory constraint violations. Overall, the average daily net profits from the MINLP and DQN solutions are 103,227 $/day and 161,703$/day. An improvement of 5.7 % is achieved by the DQN model in this case, showing the strong capability of the DQN model to handle the supply chain parameter fluctuations during ethylene cracking scheduling.



Figure 5. Daily Net Profit During the Testing Length.

* 1. Conclusions

In this work, a deep reinforcement learning framework of dynamic ethylene cracking scheduling systems including an MDP environment and a DQN model is proposed. The performance on a literature case is demonstrated. Compared to the original MINLP model, the proposed model improves the average net profit by 5.7 %, showing its enhanced capability of tackling supply chain fluctuations and increasing profitability.

References

K. Bi, S. Zhang, C. Zhang, H. Li, X. Huang, H. Liu, T. Qiu, 2021. Knowledge expression, numerical modeling and optimization application of ethylene thermal cracking: From the perspective of intelligent manufacturing. Chinese Journal of Chemical Engineering 38, 1–17.

H. Li, S. Zhang, T. Qiu, 2023. Receding Horizon Optimization of Ethylene Cracking Operation and Scheduling under Supply Chain Fluctuations. Ind. Eng. Chem. Res. 62, 5125–5140.

H. Li, S. Zhang, T. Qiu, 2022. Two-Level Decoupled Ethylene Cracking Optimization of Batch Operation and Cyclic Scheduling. Ind. Eng. Chem. Res. 61, 16539–16551.

C. Liu, J. Zhang, Q. Xu, K. Li, 2010. Cyclic scheduling for best profitability of industrial cracking furnace system. Computers & Chemical Engineering 34, 544–554.

V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, 2013. Playing Atari with Deep Reinforcement Learning.