Naphtha Cracking Center Scheduling Optimization using Multi-Agent Reinforcement Learning

Demonstration Track

Sunghoon Hong
Deunsol Yoon
Whiyoung Jung
Jinsang Lee
Hyundam Yoo
Jiwon Ham
Suhyun Jung
Chanwoo Moon
Yeontae Jung
Kanghoon Lee
Woohyung Lim
LG AI Research
Seoul, South Korea

Somin Jeon
Myounggu Lee
Sohui Hong
Jaesang Lee
Hangyoul Jang
Changhyun Kwak
Jeonghyeon Park
Changhoon Kang
Jungki Kim
LG Chem
Seoul, South Korea

ABSTRACT

The Naphtha Cracking Center (NCC) is central to petrochemical feedstock production through the intricate process. It consists of receipt stage for unloading naphtha, blending stage for mixing naphtha, and furnace stage for producing marketable products. It is crucial to make an optimal schedule for NCC for profitability and efficiency. Traditionally managed by human experts, challenges arise in predicting complex chemical reactions and navigating real-world complexities. To address these issues, this paper aims to develop autonomous NCC operation using multi-agent reinforcement learning, where each agent is responsible for each stage and collaborates to achieve common objectives, while adhering to real-world constraints. We developed an online web service to allow the staff in LG Chem Daesan NCC facility to obtain an NCC schedule in real-time, and the staff are now operating the facility based on the schedules generated by the online web service.

KEYWORDS

Multi-Agent Reinforcement Learning, MacDec-POMDP, Petrochemical Feedstock Production, Agent-based Modeling and Simulation

ACM Reference Format:
Sunghoon Hong, Deunsol Yoon, Whiyoung Jung, Jinsang Lee, Hyundam Yoo, Jiwon Ham, Suhyun Jung, Chanwoo Moon, Yeontae Jung, Kanghoon Lee, Woohyung Lim, Somin Jeon, Myounggu Lee, Sohui Hong, Jaesang Lee, Hangyoul Jang, Changhyun Kwak, Jeonghyeon Park, Changhoon Kang, and Jungki Kim. 2024. Naphtha Cracking Center Scheduling Optimization using Multi-Agent Reinforcement Learning: Demonstration Track. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand.

1 INTRODUCTION

The Naphtha Cracking Center (NCC) is integral to petrochemical production, involving complex processes such as the receipt stage for unloading naphtha from a vessel to a tank, the blending stage for mixing naphtha to achieve the desired quality, and the furnace stage for producing marketable products from the mixed naphtha. Traditionally, these processes have been managed by human experts who face challenges due to unpredictable chemical reactions, numerous constraints, and uncertainties. Previous research efforts often focused on a subset of the overall process. Joo et al. [3], Kim et al. [4] tackled furnace control using genetic algorithm but they assume no changes in state over time, which is hardly applicable to comprehensive NCC operation. Meanwhile, Lee et al. [6] formalized the NCC problem as a mixed-integer linear programming problem. However, they require a well-defined linear mathematical model and could not account for the furnace stage due to its nonlinearity.

In response to these gaps, we introduce Reinforcement Learning (RL) agents using Multi-Agent Reinforcement Learning (MARL) for comprehensive management of overall NCC operations. Each agent is responsible for a specific stage and collaborates to achieve optimal objectives such as profit maximization, while adhering to real-world constraints. We contribute in: 1) Formulating the NCC scheduling problem as MARL with complex constraints; 2) Creating a realistic simulator for RL agents that reflects real-world NCC conditions; and 3) Developing an online web service for generating real-time, optimal NCC schedules, now in operational use at LG Chem Daesan NCC facility. To the best of our knowledge, this is the first learning-based approach to optimize the overall NCC operation, significantly advancing complex industrial manufacturing process management.

2 APPROACH

2.1 NCC Scheduling Optimization

Naphtha Cracking Center (NCC) refers to a facility equipped to crack naphtha and produce basic petrochemical feedstock. Such an

Contact: {sunghoon.hong, kanghoon.lee, w.lim}@lgresearch.ai

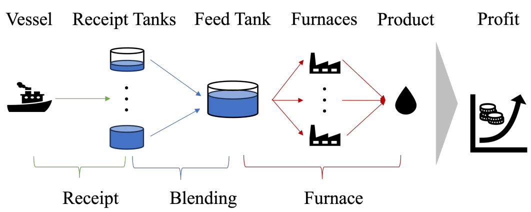

Figure 1: The flow diagram of NCC operation.

NCC consists of three consecutive stages, as illustrated in Figure 1: receipt, blending, and furnace stages. Initially, naphtha is delivered from various geographically distributed refineries via vessels. When the vessel arrives at the docking station, naphtha is unloaded into receipt tanks. The various types of naphtha from the receipt tanks are mixed at the feed tank to achieve the desired quality of naphtha for the furnace. The mixed naphtha is then cracked in the furnace and converted into marketable products.

Given a set of information of NCC, such as tank levels, vessel arrival plan, and the expected price of naphtha and marketable products, we require a schedule for controlling the overall process of the NCC in the upcoming weeks, since staff need to review it on site in advance. Once such a schedule is created, staff manage the facility in accordance with the schedule until a new schedule is generated. We refer to the creation of such a schedule as *NCC scheduling*, and optimal NCC scheduling is crucial for maximizing profits and ensuring the efficient operation of the facility.

In this paper, we consider the NCC scheduling problem as an MARL problem, with agents responsible for their corresponding stages. A *receipt agent* decides on a receipt tank to store the naphtha from the vessel and sets the receipt rate, while a *blending agent* determines which receipt tanks to draw naphtha from and at what rate. Finally, a *furnace agent* determines control variables, including the feed rate taken from the feed tank and coil outlet temperature, enabling the corresponding furnace to convert naphtha into products. These agents must collaborate to achieve common objectives such as profit and stability while adhering to the constraints.

Notably, each agent makes decisions of varying durations at different times. For example, a receipt agent takes action when the vessel arrives irregularly, and a blending action continues until the level of the chosen receipt tank reaches a threshold. In this regard, we introduce MacDec-POMDP [1, 9, 10] to account for this asynchronicity among the agents, since the MARL methods under the assumption of synchronicity, where most works focused, have limitations in realistic asynchronous settings.

2.2 NCC Simulator for MARL

We developed a simulator for MARL that reflects the real NCC environment based on OpenAI Gym [2]. This simulator takes actions from agents and provides them with next observations and rewards based on their current actions. An agent's observation consists of relevant information for solving the task the agent is responsible for. The reward is designed to encourage agents to collaborate in achieving common objectives while adhering to real-world constraints, i.e., $\texttt{Profit} - \sum_{c \in \texttt{Constraints}} w_c \cdot \texttt{Cost}_c$, shared across all agents. Here, profit is calculated by subtracting the estimated production cost of marketable products from the estimated revenue generated by selling them, taking into account the cost of energy usage and the price of naphtha. One of constraints we consider is the stability of paraffin property in naphtha which is known as a key component for efficient production of the facility [5]. For this constraint, we use change of paraffin property in naphtha stored in the feed tank as the constraint cost, $\texttt{Cost}_c$. Since other constraints are related to business and operational issues, we have concealed them. Additionally, the simulator terminates the current episode, when agents violate safety-critical or availability constraints, such as any tank level deviating beyond threshold.

2.3 Agent Training and Deployment

We modify the Proximal Policy Optimization (PPO) algorithm [8] to make it suitable for asynchronous MARL, i.e., we utilize multi-agent variant PPO [11] with special training buffer for asynchronicity [10]. And, we train RL agents on diverse scenarios, e.g., different price and initial tank levels, for robustness. We use Ray RLlib [7], an open-source library designed for highly distributed RL workloads in industry applications. After training, we generate NCC schedules by running simulations for the upcoming weeks using the NCC simulator. To further optimize the scheduling, we employ beam search, which finds and retains the top-K schedules.

We deploy the RL-based scheduling process as an online web service, as depicted in Figure 2. This service allows users to generate NCC schedules in real-time and provides a user-friendly interface.

Figure 2: The overview of the online web service.

3 DEMONSTRATION

In this section, we provide instructions on how to use the developed online web service.[1] A user can access the online web service by following these steps:

1. Select a simulator version and RL agent version to run.
2. Upload necessary input files, i.e., information of NCC.
3. Verify uploaded files and generate schedules using RL agents.
4. Check the overview for N schedules with figures and statistics.
5. Download schedules written in staff-friendly formats.

We simulated both schedules in 2023, generated by human experts and the online web service, using the NCC simulator we developed. We observed that the schedules generated by the online web service yielded higher profits and more frequently satisfied the constraints compared to the schedules created by human experts. The online web service for NCC scheduling is now in operational use at LG Chem Daesan NCC facility.

[1]Video link: https://youtu.be/TxoWG7_SLLU.

REFERENCES

[1] Christopher Amato, George Konidaris, Leslie P Kaelbling, and Jonathan P How. 2019. Modeling and planning with macro-actions in decentralized POMDPs. *Journal of Artificial Intelligence Research* 64 (2019), 817–859.

[2] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. *arXiv preprint arXiv:1606.01540* (2016).

[3] Chonghyo Joo, Hyukwon Kwon, Junghwan Kim, Hyungtae Cho, and Jaewon Lee. 2023. Machine-learning-based optimization of operating conditions of naphtha cracking furnace to maximize plant profit. *Computer Aided Chemical Engineering* 52 (2023), 1397–1402.

[4] Jeongdong Kim, Chonghyo Joo, Minsu Kim, Nahyeon An, Hyungtae Cho, Il Moon, and Junghwan Kim. 2023. Multi-objective robust optimization of profit for a naphtha cracking furnace considering uncertainties in the feed composition. *Expert Systems with Applications* 216 (2023), 119464.

[5] Ho-kyung Lee. 2012. Method for Naphtha Storage Tank Operation and System for the Same. KR Patent App. KR1020100046570A.

[6] Taeyeong Lee, Jun-Hyung Ryu, Ho-Kyung Lee, and In-Beum Lee. 2010. A study on scheduling of naphtha transportation and storage systems for naphtha cracking center. *Chemical Engineering Research and Design* 88, 2 (2010), 189–196.

[7] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. 2018. RLlib: Abstractions for distributed reinforcement learning. In *International Conference on Machine Learning*. PMLR, 3053–3062.

[8] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[9] Yuchen Xiao, Joshua Hoffman, and Christopher Amato. 2020. Macro-action-based deep multi-agent reinforcement learning. In *Conference on Robot Learning*. PMLR, 1146–1161.

[10] Yuchen Xiao, Weihao Tan, and Christopher Amato. 2022. Asynchronous Actor-Critic for Multi-Agent Reinforcement Learning. *Advances in Neural Information Processing Systems* 35 (2022), 4385–4400.

[11] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The surprising effectiveness of PPO in cooperative multi-agent games. *Advances in Neural Information Processing Systems* 35 (2022), 24611–24624.