An Industrial Application of Deep Reinforcement Learning for Chemical Production Scheduling

Christian D. Hubbs[*,†]
cdhubbs@dow.com

Adam Kelloway [†]
akelloway@dow.com

John M. Wassick[†]
jmwassick@dow.com

Nikolaos V. Sahinidis[‡]
nikos@gatech.edu

Ignacio E. Grossmann[*]
grossmann@cmu.edu

Abstract

We discuss the implementation of a deep reinforcement learning based agent to automatically make scheduling decisions for a continuous chemical reactor currently in operation. This model is tasked with scheduling the reactor on a daily basis in the face of uncertain demand and production interruptions. The reinforcement learning model has been trained on a simulator of the scheduling process that was built with historical demand and production data. The model has been successfully implemented to develop schedules on-line for an industrial reactor and has exhibited improvements over human made schedules. We discuss the process of training, implementation, and development of this system and the application of reinforcement learning for complex, stochastic decision making in the chemical industry.

1 Introduction

Scheduling in an industrial chemical process requires determining when to produce a product, in what quantity, and in what sequence in order to satisfy ever-changing market requirements. In practice, these decisions are made by human planners who are required to account for a vast amount of disparate orders from customers around the world. Additionally, the plant has production constraints such as type-dependent changeovers, daily packing and shipping capacities, and schedules are all too often thrown out of sorts due to unplanned maintenance issues or new customer requests. The role can be mentally taxing on humans who have to juggle an inordinate amount of information and uncertainty to keep customers satisfied and meet production targets. There is clear difference in planner and scheduler capabilities, and given the difficulties of the role, there is a large amount of turnover that is seen among planners and schedulers. For these reasons, we seek to develop an approach to support planners and schedulers by making production schedules automatically, allowing them to focus on customer service and business process improvement rather than difficult optimization problems.

Chemical scheduling processes have been explored by researchers since the late 1970's with pioneering works by Reklaitis [16] and Mauderli and Rippin [13] on process scheduling and industrial applications of batch scheduling (see Grossmann and Harjunkoski [2] for more on the history of the field). These early works acknowledged the existence of uncertainty and the complications that it poses, but did not tackle these issues in their formulations.

[*] Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15123, USA

[†] Dow Chemical, Digital Fulfillment Center, Midland, MI 48667, USA

[‡] H. Milton Stewart School of Industrial & Systems Engineering and School of Chemical & Biomolecular Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

Workshop on machine learning for engineering modeling, simulation and design @ NeurIPS 2020

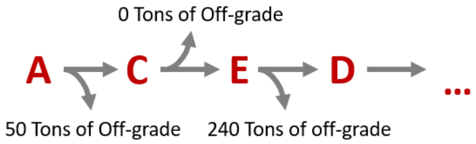


Figure 1: Losses are sequence dependent with the off-grade values denoting the average amount of off-grade material that will be manufactured before the reactor begins to produce in-spec material.

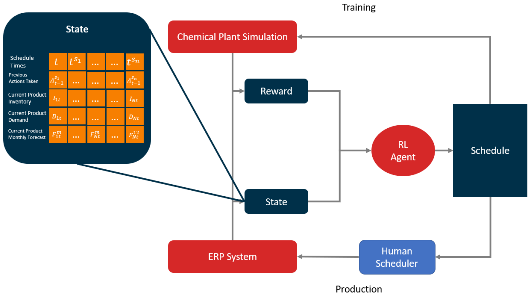
Much work has been done over the years to address scheduling under uncertainty via math programming models [17, 6, 3, 5, 4]. Despite these improvements, traditional math programming approaches are not widely used in industrial production planning roles due to the stochastic nature of the role and the size of the models, which often makes solving these models in an on-line fashion prohibitive. Instead, optimization models are used as a guide for a planner or scheduler to make changes to over the course of the planning horizon. More often than not, looking back over the previous month, the actual schedule that was implemented scarcely resembles the initial schedule that was delivered by the optimization model, and these systems tend to fall out of favor among planners who are entrusted with the outputs of the models.

To address these shortcomings and move towards a fully-automated planning and scheduling solution, we have developed and implemented a deep reinforcement learning (DRL) system. Mao et al. [12] use reinforcement learning in conjunction with a graph neural network to schedule compute processes on a cluster. Almasan et al. [1] uses a deep Q-network with a graph neural network for optimal routing. Hubbs et al. [8] provide a series of operations research problems and benchmarks showing the deep reinforcement learning performs well across a wide set of problems operating under uncertainty. Li [11] contributes an extensive review of applications of deep reinforcement learning in the literature and Pinedo [15] provides a thorough overview of the theory and design of scheduling systems. Most relevant is Hubbs et al. [7] which provides the foundations for this current work, in which the authors explore scheduling a single-stage continuous chemical reactor with change-over costs with an actor-critic model and benchmark the results against a variety of MILP models.

2 Problem Description

The chemical plant under consideration has two stages, a continuous reactor stage followed by a packaging stage. At the first stage, the planner must choose to produce one of six separate products (denoted as products $A - F$) subject to minimum campaign lengths, capacity and raw material constraints (see Figure 1 for an illustration). Additionally, the reactor can incur type-change losses whereby the thermal and chemical properties of one product differs from that of the next product causing lost production as out-of-spec product is manufactured that cannot be sold at prime prices. Some type changes are so severe that they are forbidden, while others have manageable costs associated with them. The second, packaging stage can process a single product at a time and can package each of the six products from the first stage into two separate types of packages for road or maritime shipment (products A_1, A_2 etc.). No type-change losses are incurred at the second stage. The run-rates for both stages are considered fixed under normal operating conditions, therefore the sequence for both stages is the only degree of freedom the system must cope with. The schedule must be made under demand and equipment uncertainty due to unplanned outages and delays all while maintaining product availability and inventory level targets. New orders are continuously entered into the planning system, however the DRL system only updates once per day to take into account the changes in demand from the previous 24-hours due to a limitation with the data-refresh rate. The schedule also has a fixed period in the near term whereby no changes may occur to the schedule for the next seven days in order to maintain commitments to customers, shipping schedules, and work schedules for plant employees. Given the fixed period and the shifting demand in the system, a previously optimal schedule may quickly become sub-optimal, particularly in the near term, as rush orders are entered or customers adjust their delivery dates.

The stages must work in concert in order to meet customer demands. If there is an order for product A_1, but the first stage does not replenish inventory levels for A or the packaging stage produces A_2, then a penalty is incurred.



Schedule Times	t	t^{s_1}	...	...	t^{s_n}
Previous Actions Taken	$A_{t-1}^{s_1}$	...	...	...	$A_{t-1}^{s_n}$
Current Product Inventory	I_{1t}	...	...	...	I_{Nt}
Current Product Demand	D_{1t}	...	...	...	D_{Nt}
Current Product Monthly Forecast	F_{1t}^{m}	...	F_{Nt}^{m}	...	F_{Nt}^{12}

Figure 2: Schematic of the DRL system which is trained in simulation before being integrated into the online, ERP system.

3 Methods

Given the voracious data appetite of DRL systems, we developed a simulation of the chemical plant and its planning process. This simulation is based on historical, proprietary data related to the operational history of the plant and its demand patterns. This simulation was then used to train a DRL system using the RLlib package [14]. Policy network is defined with eight fully connected layers (256 nodes per layer) with ReLU activation functions using the PPO algorithm for optimization [18]. The policy selects actions for the second, packaging stage of the production facility. We then back-calculate the first stage schedule based on the requirements of the second stage in order to develop our full facility schedules.

Inspired by the work of Jaderberg et al. [9, 10] a Population Based Training (PBT) method was used to meta-learn a set of parameter weights associated with elements of a reward function. PBT works in conjunction with reinforcement learning by allowing the reward function of the DRL agent to adapt in an evolutionary manner and by updating the hyperparameters. The system does so across a large population of competing agents (48 in this case due to resource limitations) which are subject to a higher-level objective function with the best agents according to this objective function selected for mutation and use in the next round.

This approach was chosen because no obvious, direct reward signal existed and it allows for a rich reward signal to be used for learning. The PBT objective function is determined by maximizing the profit (revenue minus costs) collected over the training episode. The revenue is the total number of orders shipped on time times the product margin. Total costs are determined by the inventory holding costs which depend on the average inventory over the episode and off-grade cost. Off-grade cost is the lost revenue due to off-grade production which is that material produced during a transition from one product to another sold at a lower margin.

4 Results and Discussion

We report two sets of results to describe the system, the historical back-test we subjected the scheduling system to and the results of the live scheduling system following 6 months of generating daily schedules under human supervision.

4.1 Historical Comparison

The trained agent's performance was tested on 100 years of simulated demand to assess its expected performance and against historical demand. A second test was then run on the actual historical data. Given that we lack decades of high-quality data for this asset, we instead took the available years of historical data and provided the initial conditions to agent as scheduling scenarios over 52-week increments to comprise one historical run. We then took the next, consecutive 52-week period, set the same initial conditions that the plant had, and created another historical testing scenario. This was done to generate all of the historical data points shown in Figure 3.[4] The results in Figure 3

[4] Note that all cost, demand, and reward function values have been re-scaled to protect any sensitive or confidential information.

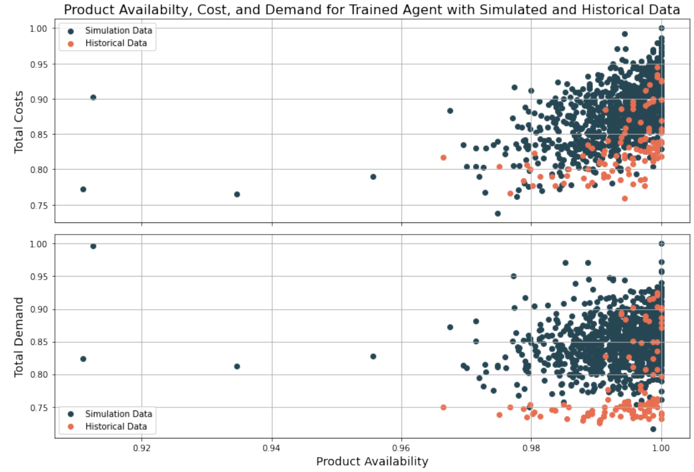


Figure 3: Performance of the trained DRL system versus 1,000 simulated years and historical data. The DRL system outperforms actual human planners over the same horizon with an average 99% Product Availability Rate compared to 85%.

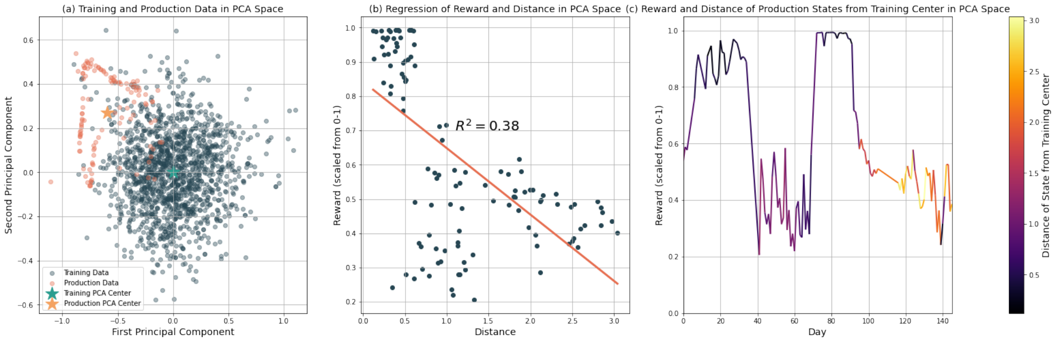


Figure 4: The DRL agent's performance decays as the distance between the state used to make scheduling decisions drifts from the center of the training data in PCA space.

show a good overlap between the simulated results and the historical back-testing, although the agent was trained in slightly higher average demand scenarios than occur in the actual, historical data. Comparing the results of the historical run to the identical time periods for our human planners, we find that the DRL system yields a large increase in our key metrics. For example, product availability - a measure that determines how often we can meet customer orders when they request it - rose from 85% to 99%.

4.2 Production Scheduling System

The trained agent is deployed as a web application which collects state information from corporate data stores to initialize the simulation with current open orders and inventory. The schedule is then produced by the DRL agent with the specific sequence it selects being passed back into commercial scheduling software. This schedule is reviewed by the planning team before being finalized and executed in the plant. With this regular feedback, the team was able to troubleshoot any issues as they arose with the system in production and make tweaks to ensure stable operation of the plant, appropriate inventory and customer service levels.

Moreover, we monitor the states the system is operating on to better understand the limitations of the system in practice but projecting the multi-dimensional state down to its three principle components based on a principle component model based on training states. We hypothesized that the DRL agent's performance would degrade as operational states becomes increasingly dissimilar from training states.

Over the months since the system has been implemented, we can see that many of the states do differ from the training data (Figure 4a). This drift has yielded poorer performance and a reduction in reward for the agent, either by higher costs, or reduced sales and customer service levels. This is shown in Figure 4b where we plot the distance of each state the agent has seen while in production from the center of the training data in PCA space. In Figure 4c, we see two areas where performance has dropped off, one coinciding with increased PCA distance. The first region of poor performance was driven by a lengthy, unplanned production outage which negatively impacted customer service. The second region, roughly from day 90 onward, was caused by pernicious data errors and double counting of orders from introduction of a new system upstream from the planning agent.

Despite these challenging areas, we are encouraged by the early results of the DRL agent in our scheduling system. To improve the quality of the schedules the system produces, we can raise alerts to call for human intervention if the state strays too far in PCA space. More long-term, we seek to develop an ensemble of models that can be intelligently swapped out to ensure that the planning and scheduling performance remains as close to optimal as possible.

5 Conclusion and Future Work

We developed a DRL agent to schedule a two-stage production process consisting of a continuous chemical reactor and a packaging line, online, under uncertainty. The agent was trained using PBT to maximize the profitability of the scheduling system. The testing results outperformed human schedulers giving confidence in deploying the live system, which has thus far met with success. While the live scheduling system currently operates with a human-in-the-loop, future developments hope to move to a fully automated system that will be capable of managing the chemical reactors and production lines autonomously.

Additional future work exists around optimizing the network of reactors in the plant and throughout the region. This is just one of many production facilities in our network that meet customer needs around the globe. Coordination between these facilities is often difficult creating higher inventory levels or lower customer satisfaction to compensate. Thus we are in the process of exploring larger, multi-agent reinforcement learning systems to address these problems.

Broader Impact

This project is explicitly aimed at automating much of the work currently done by human planners and schedulers. The job in question is typically a high-stress, high-turnover role that requires employees to optimize and re-optimize complex scheduling systems in order to meet ever-changing customer demands. While the spectre of automation is disconcerting to some, economic theory and history are in unanimous agreement that automation is a net benefit for society. Increased mechanization and digitalization of goods and services reduces their prices causing them to be more affordable, thereby raising the standard of living of individuals. Moreover, new products and services are developed based on the wider availability of these goods and services leading towards economic expansion.

While the nature of these planner and scheduler jobs will change, they will not be eliminated, certainly not in the near-term. As planning and scheduling tools have improved over the years, less of the job has been devoted towards developing new schedules and more time has been freed for higher-value work such as improvement projects and increasing customer service. This work may accelerate this trend, but employers are certainly eager to re-allocate their employee's time away from operational tasks and towards more strategic and customer-centric activities.

Acknowledgments and Disclosure of Funding

We thank Dow Chemical and our colleagues for their tremendous support and funding in this work.

References

[1] P. Almasan, J. Suárez-Varela, A. Badia-Sampera, K. Rusek, P. Barlet-Ros, and A. Cabellos-Aparicio. Deep Reinforcement Learning meets Graph Neural Networks: exploring a routing optimization use case. 10 2019. URL http://arxiv.org/abs/1910.07421.

[2] I. E. Grossmann and I. Harjunkoski. Process Systems Engineering: Academic and Industrial Perspectives. In *13th International Symposium on Process Systems Engineering*, San Diego, 7 2018. PSE.

[3] I. E. Grossmann, R. M. Apap, B. A. Calfa, P. García-Herreros, and Q. Zhang. Recent advances in mathematical programming techniques for the optimization of process systems under uncertainty. *Computers and Chemical Engineering*, 91:3–14, 2016. ISSN 00981354. doi: 10.1016/j.compchemeng.2016.03.002. URL http://dx.doi.org/10.1016/j.compchemeng.2016.03.002.

[4] D. Gupta and C. T. Maravelias. A general state-space formulation for online scheduling. *Processes*, 5(4), 12 2017. ISSN 22279717. doi: 10.3390/pr5040069.

[5] D. Gupta, C. T. Maravelias, and J. M. Wassick. From rescheduling to online scheduling. *Chemical Engineering Research and Design*, 116:83–97, 2016. ISSN 02638762. doi: 10.1016/j.cherd.2016.10.035. URL http://dx.doi.org/10.1016/j.cherd.2016.10.035.

[6] K. Huang and S. Ahmed. The Value of Multistage Stochastic Programming in Capacity Planning Under Uncertainty. *Operations Research*, 57(4):893–904, 2009. ISSN 0030-364X. doi: 10.1287/opre.1080.0623.

[7] C. Hubbs, C. Li, N. Sahinidis, I. Grossmann, and J. Wassick. A deep reinforcement learning approach for chemical production scheduling. *Computers and Chemical Engineering*, 141, 2020. ISSN 00981354. doi: 10.1016/j.compchemeng.2020.106982.

[8] C. D. Hubbs, H. D. Perez, O. Sarwar, N. V. Sahinidis, I. E. Grossmann, and J. M. Wassick. OR-Gym: A Reinforcement Learning Library for Operations Research Problems. 8 2020. URL http://arxiv.org/abs/2008.06319.

[9] M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan, C. Fernando, and K. Kavukcuoglu. Population Based Training of Neural Networks. 11 2017. URL http://arxiv.org/abs/1711.09846.

[10] M. Jaderberg, W. M. Czarnecki, I. Dunning, L. Marris, G. Lever, A. G. Castaneda, C. Beattie, N. C. Rabinowitz, A. S. Morcos, A. Ruderman, N. Sonnerat, T. Green, L. Deason, J. Z. Leibo, D. Silver, D. Hassabis, K. Kavukcuoglu, and T. Graepel. Human-level performance in first-person multiplayer games with population-based deep reinforcement learning. 7 2018. URL http://arxiv.org/abs/1807.01281.

[11] Y. Li. Deep Reinforcement Learning: An Overview. pages 1–70, 2017. ISSN 1701.07274. doi: 10.1007/978-3-319-56991-8{_}32. URL http://arxiv.org/abs/1701.07274.

[12] H. Mao, M. Schwarzkopf, S. B. Venkatakrishnan, Z. Meng, and M. Alizadeh. Learning scheduling algorithms for data processing clusters. In *SIGCOMM 2019 - Proceedings of the 2019 Conference of the ACM Special Interest Group on Data Communication*, pages 270–288. Association for Computing Machinery, Inc, 8 2019. ISBN 9781450359566. doi: 10.1145/3341302.3342080.

[13] A. Mauderli and D. W. T. Rippin. Production Planning and Scheduling for Multi-Purpose Batch Chemical Plants. Technical report, 1979.

[14] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan, and I. Stoica. Ray: A Distributed Framework for Emerging AI Applications. 2018. URL http://arxiv.org/abs/1712.05889.

[15] M. L. Pinedo. *Scheduling: Theoty, Algotithms, and Systems*, volume 4. 2012. ISBN 9780874216561. doi: 10.1007/s13398-014-0173-7.2.

[16] G. Reklaitis. Review of Scheduling of Process Operations. *AIChE Symposium*, 78:119–133, 1978.

[17] N. V. Sahinidis. Optimization under uncertainty: State-of-the-art and opportunities. In *Computers and Chemical Engineering*, volume 28, pages 971–983, 2004. ISBN 0098-1354. doi: 10.1016/j.compchemeng.2003.09.017.

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms John. *arXiv*, 2017. ISSN 09594388. doi: 10.1016/j.conb.2007.07.004. URL https://arxiv.org/pdf/1707.06347.pdf.