Tackling Scheduling Problems with Graph Structured Reinforcement Learning

Seppe Renty, Raphaël Avalos, Andries Rosseau, and Ann Nowé

Vrije Universiteit Brussel, Belgium, https://www.vub.be/

Abstract. Job scheduling is a central element of our society. While this problem has been actively researched by the field Operations Research (OR), yielding good algorithms, these solutions are usually not generalizable across different problem instances. They also tend to behave poorly when there is uncertainty during execution. In this thesis, we research whether dispatching rules that generalize to groups of similar problem instances can be learned through Reinforcement learning. To leverage the complex structure of a Job scheduling problem we designed a graph representation of the problem. We then used a combination of Graph Neural Networks (GCN) and Multi-Agent Reinforcement Learning (MARL) to improve on existing dispatching rules. We found that the technique generates shorter schedules on average than popular dispatching rules for multiple families of problems.

Keywords: Job Shop Scheduling Problem · Multi-agent Reinforcement Learning · Graph Neural Networks · Graph Convolutional RL

1 Introduction

Scheduling problems are a fundamental part of our economy. Take, for example, a car manufacturer that builds several car models. Numerous machines and processes are involved and scheduling will be very important to make full use of the infrastructure and to improve efficiency. This research focuses on the popular Job Shop Scheduling Problem (JSP) [1]. The goal is to find generalizable rules that generate efficient schedules for a family of similar instances. This means that the technique should create schedules that are as short as possible, yet able to adapt to uncertainties during execution. The JSP has been studied for several decades and is thus a problem for which many solution methods already exist. In this thesis, we focus on combining both the scheduling and execution phases. Other areas of research focus on methods that distinguish between the two phases. Although techniques that combine both phases generally produce less efficient schedules, these techniques are much faster and more resilient to uncertainties during execution.

2 Methods and Experiments

The JSP can be represented by an inter-agent graph, which is defined by the machines and the jobs of the scheduling problem. The machines are the nodes,

and the paths of the jobs form the edges. In this thesis, we explore the idea of representing dispatching rules with a Graph Convolutional Neural Network (GCN) [3]. A GCN is a type of neural network that can leverage the graph structure of its input. Additionally they have the capacity to generalize well and inference of a neural network is very fast. This means that these rules can contain very complex logic, while still maintaining the robustness of a simple dispatching rule. Figure 1 shows a visual representation of the procedure. The GCN extracts information from the inter-agent graph at each time step. This information is then used by the GCN to determine the next action for each machine. This results in a reactive schedule, which is a schedule that can react to uncertainties during execution. An adaptation of Graph Convolutional Reinforcement Learning [4] is used to train the GCN, which uses Deep Q-Learning [8] as general training procedure.

The experiments are divided into three parts. A case study on the ft06 benchmark [5], followed by experiments on the Lawrenche benchmarks [6], which are considered harder than the ft06 benchmark. We end with some preliminary experiments on curriculum learning [7]. The most commonly used priority dispatching rules [2] are used as a comparison for all experiments. For example, MWKR (Most Work Remaining) and FIFO (Fist In First Out).

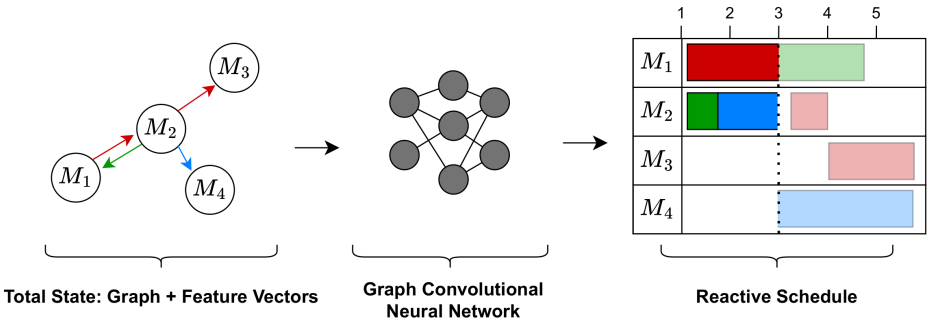

Fig. 1. The inter-agent graph is the input to the GCN at each time step, and creates a reactive schedule.

3 Results and Conclusion

The ft06 case study produced some promising results. On average, the trained model on the single benchmark generated shorter schedules than the dispatching rules for all derivatives of this benchmark, ranging between 10 and 80 jobs per instance. The same set of experiments was conducted with the Lawrence benchmark. The results are less significant, yet the generalizable qualities were still evident. The preliminary experiments on curriculum learning showed no significant improvement. However, extending the curriculum could be very interesting for future research. We conclude that there is potential in using Graph Convolutional Reinforcement Learning to learn complex dispatching rules for scheduling problems, especially when looking for a rule that generalizes well on multiple similar problems.

References

1. Baker, K.R.: Introduction to sequencing and scheduling. First ed. John Wiley & Sons, New York (1974)
2. Panwalker, S.S., Iskander, W.: A survey of scheduling rules. Operations research **25**(1), 45–61 (1977)
3. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations (ICLR 2017), (2016)
4. Jiang, J., Dun, C., Lu, Z.: Graph Convolutional Reinforcement Learning. arXiv preprint arXiv:1810.09202 (2018) https://doi.org/10.48550/ARXIV.1810.09202
5. Fisher, H., Thompson, G. L.: Probabilistic learning combinations of local job-shop scheduling rules. In: Industrial Scheduling, pp. 225–251. Prentice-Hall, Englewood Cliffs, New Jersey (1963)
6. Larence, S.: Resouce constrained project scheduling: An experimental investigation of heuristic scheduling techniques (Supplement). Graduate School of Industrial Administration, Carnegie-Mellon University (1984)
7. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Bottou L., Littman M. (eds.) Proceedings of the 26th International Conference on Machine Learning (ICML 2009), pp 41–48. Omnipress, Montreal (2009)
8. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G.,... others.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)