Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)
Doctoral Consortium Track

Adaptive Artificial Intelligence Scheduling Methods for Large-Scale, Stochastic, Industrial Applications

Pierre Tassel
University of Klagenfurt, Austria
pierre.tassel@aau.at

Abstract

Traditional scheduling techniques suffer from a lack of flexibility. The problem's instances need to be deterministic, and results on datasets with small benchmark instances do usually not transfer to large-scale instances. We propose to develop adaptive algorithms that can leverage the similarities between instances of industrial scheduling problems. In particular, we focus on applications of modern machine learning techniques to combinatorial optimization problems, an emerging and promising research area. Traditional scheduling techniques such as constraint, mixed-integer, or answer set programming are highly generic, domain-independent, and, therefore, do not explicitly exploit the specificities of a problem domain. However, in a production facility, the settings between two consecutive schedules are often very similar. The machines, workers, production capacity, etc., usually stay the same or do not change significantly. Traditional scheduling techniques do not take advantage of such similarities, while machine learning, especially deep learning, can discover and exploit relationships in the data. Therefore, our research aims to incorporate machine learning into combinatorial optimization.

1 Problem Statement

Many real-world industrial problems are NP-hard, in particular, in the area of planning and scheduling. This is bad news, as it is considered unlikely that an algorithm whose running time is polynomial in the size of the input exists [Gasarch, 2019]. As we don't have such an algorithm to compute good-quality solutions in an acceptable time, we can apply combinatorial optimization techniques such as constraint, mixed-integer, or answer set programming. While such methods can successfully perform small problem instances, they often fail to scale up to large industrial instances.

A weakness of general problem-solving techniques is their generic approach, which does not exploit the specificities of a problem domain and instead applies rigid search methods independently of instance characteristics to derive information about the instance during solving without reusing it. Hence, traditional combinatorial optimization techniques cannot take advantage of the similarities between instances of industrial scheduling problems. In particular, factories are rather static entities in which machines, workers, production capacity, and produced goods are often very similar from one schedule to another.

Moreover, recurring planning and scheduling tasks are often linked, e.g., when an existing schedule needs to be changed or extended in view of specific events like operational disruptions, new production orders, etc. Traditional techniques do not consider scheduling as a continuous task. They assume instances to be deterministic and independent of each other, limiting their modeling and solving capacities to coarse abstractions of the real world. In practice, however, factories handle many stochastic events such as the processing times of jobs, varying relative to worker skills, operational and transport delays, etc., or machine breakdowns [Hassoun *et al.*, 2019]. Finally, traditional combinatorial optimization techniques only benefit little from the high parallelism capacities of modern CPUs. The improvements due to allocating more CPUs are often marginal and absorbed by some inherently sequential bottleneck tasks like, e.g., pre-solving in constraint programming [1]. As with mixed-integer programming solvers, using a single CPU can sometimes result in better solutions than parallel solving on multiple CPUs [2]. Such behavior is problematic as modern CPUs boost computing performance through high concurrency, exploiting an increasing number of cores [Sutter, 2005]. Reinforcement Learning (RL) is an area of Machine Learning (ML) where an autonomous agent operates in an environment and learns, based on experimentation, how to optimize a quantitative reward over time. At each iteration, the agent observes the environment, selects an action, performs it, and receives a reward that corresponds to the impact of the actions taken on the desired outcome. For any state, the agent then aims to identify the action that maximizes the expected cumulative rewards for the given task. Such a mapping from states to actions, called policy, can be viewed as a heuristic for achieving a desirably high reward at the end of an episode. Recently, the combination of RL with deep learning (DRL) has been the source of breakthroughs in Artificial Intelligence. These success sto-

[1] https://github.com/google/or-tools/issues/1588

[2] https://gurobi.com/documentation/9.1/refman/threads.html

ries include winning against the world's best GO player [Silver and et al., 2016], super-human performance on Atari's games [Mnih and et al., 2013], and beating human professional players on the complex, information-imperfect, real-time game Dota2 [Berner and et al., 2019]. In view of such showcases, researchers soon recognized the potential of DRL for practically important Combinatorial Optimization Problems (COPs) [Bengio *et al.*, 2021].

2 Contributions

In [Tassel and Rbaia, 2021], we initially leverage problem properties and propose a decomposition of the Aircraft Routing and Maintenance Planning problem. We propose a multi-shot Answer Set Programming (ASP) model which incrementally increases the allowed on-ground time between two flights at each iteration. As a result, compared to the original, single-shot ASP model, we found solutions of 40% lower cost while dividing the runtime by 3.

In [Tassel *et al.*, 2021], we designed an optimized reinforcement learning environment for the job-shop scheduling problem [Taillard, 1993]. We could outperform state-of-the-art reinforcement learning approaches when training a simple PPO agent with a multi-layer perceptron network. However, compared to traditional approaches such as Constraint Programming (CP), we could not improve their performance. The state-space is instance dependent (i.e., depends on the number of jobs to schedule) which can reduce the applicability of our approach.

In [Kovács *et al.*, 2021], we developed a heuristic and an Integer Linear Programming (ILP) and CP model to solve a large-scale, industrial Resource-Constrained Project Scheduling Problem (RCPS). Compared to previous RCPS approaches, focusing on small academic benchmarks of less than 100 jobs to allocate, our meta-heuristic combining a heuristic and a CP model could solve industrial instances with more than 6000 jobs. This work has been done in collaboration with an industrial partner.

In [Tassel *et al.*, 2022], we proposed a new and efficient Markov Decision Process (MDP) formulation for scheduling problems, where the state-transition function considers sequences rather than a fixed number of actions. Closely related to dispatching heuristics. We also proposed a training approach using Evolution Strategies to train a neural network to dispatch jobs. Combined with a Beam-Search solution generator, it provides better results than the previously proposed meta-heuristic while reducing the time needed by a factor of 60.

In [Kovács *et al.*, 2022], we developed a realistic simulation of a semiconductor factory. This work provides a scalable, open-source tool for simulating semiconductor factories and encouraging the development of new scheduling methods.

3 Conclusion

During the first two years of my Ph.D., I focused on developing adaptive scheduling methods to leverage challenging industrial scheduling problems. My recent work focused on reinforcement learning and simulation produced encouraging results. In future works, I want to further improve the methods I have already developed and combine them with traditional scheduling techniques such as CP.

References

[Bengio *et al.*, 2021] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: A methodological tour d'horizon. *Eur. J. Oper. Res.*, 290(2):405–421, 2021.

[Berner and et al., 2019] Christopher Berner and et al. Dota 2 with large scale deep reinforcement learning. *CoRR*, abs/1912.06680, 2019.

[Gasarch, 2019] William I. Gasarch. Guest column: The third p=?np poll. *SIGACT News*, 50(1):38–59, 2019.

[Hassoun *et al.*, 2019] Michael Hassoun, Denny Kopp, Lars Mönch, and Adar Kalir. A new high-volume/low-mix simulation testbed for semiconductor manufacturing. In *WSC*, pages 2419–2428. IEEE, 2019.

[Kovács *et al.*, 2021] Benjamin Kovács, Pierre Tassel, Wolfgang Kohlenbrein, Philipp Schrott-Kostwein, and Martin Gebser. Utilizing constraint optimization for industrial machine workload balancing. In *CP*, volume 210 of *LIPIcs*, pages 36:1–36:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[Kovács *et al.*, 2022] Benjamin Kovács, Pierre Tassel, Martin Gebser, and Georg Sidel. A Customizable Simulator for Artificial Intelligence Research to Schedule Semiconductor Fabrication Plants. In *ASMC*, SC Manufacturing, pages 122–130. IEEE Computer Society, 2022.

[Mnih and et al., 2013] Volodymyr Mnih and et al. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

[Silver and et al., 2016] David Silver and et al. Mastering the game of go with deep neural networks and tree search. *Nat.*, 529(7587):484–489, 2016.

[Sutter, 2005] Herb Sutter. The free lunch is over: A fundamental turn toward concurrency in software. *Dr. Dobb Journal*, 30(3):202–210, 2005.

[Taillard, 1993] E. Taillard. Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64(2):278–285, 1993.

[Tassel and Rbaia, 2021] Pierre Tassel and Mohamed Rbaia. A multi-shot ASP encoding for the aircraft routing and maintenance planning problem. In *JELIA*, volume 12678 of *Lecture Notes in Computer Science*, pages 442–457. Springer, 2021.

[Tassel *et al.*, 2021] Pierre Tassel, Martin Gebser, and Konstantin Schekotihin. A reinforcement learning environment for job-shop scheduling. *CoRR*, abs/2104.03760, 2021.

[Tassel *et al.*, 2022] Pierre Tassel, Benjamin Kovács, Martin Gebser, Konstantin Schekotihin, Wolfgang Kohlenbrein, and Philipp Schrott-Kostwein. Reinforcement Learning of Dispatching Strategies for Large-Scale Industrial Scheduling. *ICAPS*, 2022.