Multi-Agent Reinforcement Learning tool for Job Shop Scheduling Problems

Jessica Coto Palacio[1,2] Yailen Martínez Jiménez[2] Ann Nowé[3]

[1] UEB Hotel Los Caneyes, Ave. Los Eucaliptos y Circunvalación, Santa Clara, Cuba
jcotopalacio@gmail.com
[2] Universidad Central " Marta Abreu " de Las Villas, Santa Clara, Villa Clara, Cuba
yailenm@uclv.edu.cu, jcoto@uclv.cu
[3] Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
ann.nowe@ai.vub.ac.be

Keywords: Job Shop Scheduling · Industry 4.0 · Reinforcement Learning · Multi-Agent Systems.

1 Introduction

During the last years, technological developments have increasingly benefited industry performance. The appearance of new information technologies have given rise to intelligent factories in what is termed as Industry 4.0 (i4.0) [8, 7]. The i4.0 revolution involves the combination of intelligent and adaptive systems using shared knowledge among diverse heterogeneous platforms for computational decision-making [13, 9]. In this sense, embedding Multi-Agent Systems (MAS) is a highly promising approach to handle complex and dynamic problems. A typical example of an industrial opportunity of this kind is scheduling, whose goal is to achieve resource optimization and minimization of the total execution time [11]. Given the complexity and dynamism of industrial environments, the resolution of this type of problem may involve the use of very complex solutions, as customer orders have to be executed on the resources available. In real world scheduling problems, the environment is subject to constant uncertainty, machines breakdown, orders take longer than expected, and these unexpected events can make the original schedule fail [15, 6]. Accordingly, the problem of creating a job-shop scheduling (JSSP)is considered one of the hardest manufacturing problems in literature [1]. In [3] and [4], the authors suggested and analyzed the application of reinforcement learning techniques to solve the JSSP. They demonstrated that interpreting and solving this kind of scenarios as a multi-agent learning problem is beneficial for obtaining near-optimal solutions and can very well compete with alternative approaches. Another problem that has been identified in the scheduling community is the fact that most of the research concentrates on optimization problems that are a simplified version of reality. As the author points out in [12]: this allows for the use of sophisticated approaches and guarantees in many cases that optimal solutions are obtained. The exclusion of real-world restrictions harms the applicability of those methods. What the industry needs are systems that adjust exactly to the conditions in the production plant.

2 Multi-Agent Reinforcement Learning Tool

The MARL tool proposed in this research allows the user either to keep the best result obtained by a learning algorithm or to include extra constraints of the production floor. This first version allows to fix operations to time intervals in the corresponding resources and afterwards optimize the solution based on the new constraints added by the user. This is a first approach that helps to close the gap between literature and practice. It groups several algorithms aimed at solving scheduling problems in the manufacturing industry. It focuses on the need of building a more flexible schedule, in order to adjust it to the user's requests without violating the restrictions of the JSSP scenario. The approach used to obtain the original solution that the user can afterwards modify is the one proposed in [10], it is a generic multi-agent reinforcement learning approach that can be easily adapted to different scheduling settings, such as the Flexible Job Shop (FJSSP)[5] or the Parallel Machines Job Shop Scheduling (PMJSSP) [14]. The algorithm used is the Q-Learning, which works by learning an action-value function that gives the expected utility of taking a given action in a given state. There is basically an agent per machine which takes care of allocating the operations that must be executed by the corresponding resource. Once the user chooses the scheduling scenario to solve (JSSP, FJSSP or PMJSSP), the tool proposes an initial solution based on the QL algorithm, and at the same time it enables a set of options that are the basis of this research. The user has the possibility to move the operations either using the mouse or the touch screen, and these movements must be validated once the new positions are decided.

3 Experimental Results and Conclusions

In this work we compare the performance of two alternatives for optimizing the schedule once the user has fixed some operations, the classical left shifting and a modified Q-Learning algorithm, which includes the position of the fixed operations in the learning process. In order to measure the performance of both alternatives several benchmark problems from the OR-Library [2] were used. For each instance, the same operations were fixed, and each optimization alternative had to adjust the schedule in order to minimize the makespan. The Wilcoxon test applied to the results shows that there are significant differences between the two alternatives (sig=0.08), the mean ranks confirm that the QL version with fixed operations is able to obtain better results than the classical optimization process. This is mainly because the left shifting respects the order in which the operations were initially placed along the X-axis. The QL algorithm, on the other hand, keeps the fixed positions and during the process of learning, the order in which the operations are scheduled in the resources does not have to be the same, this allows the approach to obtain better solutions in terms of makespan.

References

1. Asadzadeh, L.: A local search genetic algorithm for the job shop scheduling problem with intelligent agents. Computers & Industrial Engineering **85**, 376–383 (2015)
2. Beasley, J.E.: OR-Library: Distributing test problems by electronic mail. Journal of the Operational Research Society **41**(11), 1069–1072 (1990)
3. Gabel, T.: Multi-Agent Reinforcement Learning Approaches for Distributed Job-Shop Scheduling Problems. Phd thesis, Universität Osnabrück (2009)
4. Gabel, T., Riedmiller, M.: On a Successful Application of Multi-Agent Reinforcement Learning to Operations Research Benchmarks. In: IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning. pp. 68–75. Honolulu , USA (2007)
5. Gavin, R., Niranjan, M.: On-line Q-learning using connectionist systems. Tech. rep., Engineering Department, Cambridge University (1994)
6. Hall, N., Potts, C.: Rescheduling for new orders. Operations Research **52**, 440–453 (2004)
7. Leitao, P., Colombo, A., Karnouskos, S.: Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. Computers Industry **81**, 11–25 (2016)
8. Leitao, P., Rodrigues, N., Barbosa, J., Turrin, C., Pagani, A.: Intelligent products: The grace experience. Control Engineering Practice **42**, 95–105 (2005)
9. Leusin, M.E., Frazzon, E.M., Uriona Maldonado, M., Kück, M., Freitag, M.: Solving the Job-Shop Scheduling Problem in the Industry 4.0 Era. Technologies **6**(4) (2018)
10. Martínez Jiménez, Y.: A Generic Multi-Agent Reinforcement Learning Approach for Scheduling Problems. Ph.D. thesis, Vrije Universiteit Brussel, Brussels (2012)
11. Toader, F.A.: Production Scheduling in Flexible Manufacturing Systems: A State of the Art Survey **3**(7), 1–6 (2017)
12. Urlings, T.: Heuristics and metaheuristics for heavily constrained hybrid flowshop problems. Ph.D. thesis (2010)
13. Vogel-Heuser, B., Lee, J., Leitao, P.: Agents enabling cyber-physical production systems. AT-Autom. **63**, 777–789 (2015)
14. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning **8**, 229–256 (1992)
15. Xiang, W., Lee, H.: Ant colony intelligence in multi-agent dynamic manufacturing scheduling. Engineering Applications of Artificial Intelligence **21**, 73–85 (2008)