

Artificial Intelligence Methods for Complex Industrial Scheduling Applications

Mohammed El-Kholany, MSc.

Production Systems Research Group
Applied Informatics Department
Alpen-Adria-Universität Klagenfurt, Austria

Supervisor

Prof. Martin Gebser

Production Systems Research Group
Applied Informatics Department
Alpen-Adria-Universität Klagenfurt, Austria

December 2020

1 Introduction

The study of production systems is complex, given a large number of integrated entities and interactions. This complexity is one of the scheduling problem causes, which are generally recognized as difficult to solve [1, 2]. Scheduling of operations involves resource allocation over a period of time to perform a series of tasks. There are several types of scheduling problems, one of the most critical problems is the Job-Shop Scheduling Problem (JSP). The JSP has been known as a notoriously complex and stubborn combinatorial optimization problem since the 1950s, and it has been proven to be NP-hard in [3, 4]. Therefore, it was very difficult to reach the optimal solution, even for small-scale instances.

However, in the practical manufacturing environment, the Job-Shop Scheduling Problem scale is much larger, where the number of operations may be up to 10000 in some mechanical workshops [5]. The main purpose is to find the most appropriate sequencing of operations on the machines to optimize the performance indicator. A typical performance indicator for JSP is the makespan, which is the time needed to complete all jobs.

The real world is subject to many change sources, typically treated as random occurrences such as breakdown machines, new job arrivals, change in due dates, and canceling orders. These random occurrences might cause some disturbances and make the original schedule infeasible in [6]. Many researchers focused on Dynamic Job-shop Scheduling Problem (DJSP) to handle unforeseen events [7, 8, 6]. It is a NP-hard optimization problem and a computationally challenging issue.

In DJSP, urgent jobs can arrive at some unknown times. The generated schedules may become outdated when they are inserted to the shop floor. Since it is difficult to take into account these changes while generating a schedule. A rescheduling strategy is necessary to update the production schedule and guarantee feasibility and efficiency. The dynamic scheduling is the process of generating a schedule based on the old one to respond to disturbances [9]. DJSP applications can be found in different real-world applications such as production planning, machine scheduling, etc. This makes DJSP attracted much attention

from a significant number of researchers and many heuristics and meta-heuristics that have been proposed to find a near-optimal solution for DJSP.

2 State of the art

Since the JSP has NP-hard property, the performance of traditional methods will hardly be satisfactory. To overcome these difficulties, several decomposition-based algorithms have been proposed. The main idea of the decomposition approach is to divide the original problem into a series of smaller sub-problems and solve sub-problems respectively to obtain the final solution of the original problem. For instance, a decomposition-based hybrid approach for solving large-scale JSP is proposed in [5] while the total weighted tardiness must be minimized. In each iteration, a sub-problem is defined using Simulated Annealing and solved by a Genetic Algorithm.

In [10], a new method for detecting multi-bottleneck machines is proposed where the characteristics of bottleneck machines are used to improve the solution efficiency. As mentioned in the previous section, handling unforeseen events like (breakdown machines, the arrival of urgent jobs, and canceling orders) is one of the most critical success keys in the manufacturing systems.

For this reason, many researchers have tackled DJSP while minimizing the effect of disruptions that might occur as much as possible. Most of the schedules consider only shop efficiency. However, when a disruption occurred a new generated schedule might deviate significantly from the original one, and it can seriously affect the floor's performance. The main objective of dynamic scheduling is to generate a new schedule based on the old one considering the shop efficiency (makespan minimization) and stability (minimizing the deviation from the original schedule) [11]. Three different rescheduling strategies: Right-Shift Rescheduling, Fixed-Sequence Rescheduling, and Total Rescheduling have been studied in [12], while investigating each strategy's efficacy to minimize the total weighted tardiness. The experiments showed that breakdown machine time is sensitive to rescheduling performance.

A new rescheduling strategy based on the Genetic Algorithm and tabu search is proposed to solve the DJSP where the objective is to minimize the makespan

in [13]. This study investigated the arrival of new jobs unconditionally at the system. Another version of the Genetic Algorithm has been developed to solve JSP with machine disruptions in [14] while minimizing the makespan. We see that the previous studies aimed to optimize the shop's efficiency and considered only one disruption. Some studies tackled a different kind of disruption in addition to optimizing schedule efficiency and schedule stability simultaneously. For instance, a rescheduling approach based on the Genetic Algorithm and tabu search has been presented to solve the DJSP with machine breakdowns and random job arrivals in [15]. The proposed algorithm aimed to generate a robust and stable schedule by minimizing the makespan and the starting time deviations of the operations simultaneously.

3 Problem Statement

In the JSP, the problem consists of n jobs and a set of m machines. Each job has a series of operations. The operations must be executed sequentially and need to be processed on specific machines. Each machine can process only one operation at a time. The Dynamic JSP is an extension of the JSP, and the execution of a schedule is usually confronted with disruptions and unforeseen events. Many real-time events could occur during the scheduling process, such as random job arrivals, machine breakdowns, and cancel orders. When an unforeseen event happens, the system should react and update the schedule very fast to adapt with the new situation.

4 Contribution

In summary, the number of papers on the Dynamic Job-shop scheduling problem has been increasingly growing in recent years. However, most of them were tackling schedule efficiency that implies high machine utilization. It can improve shop efficiency. However, the schedule that only considers the schedule efficiency may deviate from the original schedule. It can have a negative impact on shop performance. Moreover, most of the work focused on small-scale benchmark instances considering only an unexpected event, which does not fit

real applications in the production systems. Therefore, it is essential to study Dynamic JSP by considering the schedule efficiency and schedule stability simultaneously. This thesis aims to develop an enhanced framework that provides a robust and stable schedule to solve large-scale instances and considers different unexpected events (breakdown machines, the arrival of new jobs, and canceling orders). More preciously, the research objectives of this Ph.D. are:

1. This package aims to develop an efficient approach that decomposes the JSP into a series of sub-problems and obtain near-optimal solutions in a reasonable time. To evaluate our proposed approach, we should perform many experiments on large-scale instances to validate its efficiency.
2. Secondly, we will consider machine breakdown and develop a technique that minimizes the classical performance measure (makespan). In this phase, we will focus only on minimizing the makespan’s deviation.
3. The third objective is to provide a robust and stable schedule that optimizes the second research objective and minimizes the deviation of the operations’ starting time and the sequence of the operations.
4. The last objective of this Ph.D. project is to consider two other types of disruptions: the arrival of new jobs and canceling orders while optimizing the same objectives.

5 Methodology

We plan to build our models using Answer Set Programming (ASP), a programming methodology rooted in artificial intelligence and computational logic research [16]. It is considered among the most popular paradigms for knowledge representation and reasoning, especially in combinatorial optimization problems [17]. ASP was successfully used in various application areas, including team building and scheduling [18], bio-informatics [19], and product configuration [20]. An ASP program contains logic rules, including variables and facts, which represent the input. Standard ASP follows a one-shot process in computing answer set(s) of a logic program, and then a new flexible paradigm of grounding and solving is proposed, which is multi-shot ASP solving [21]. The Multi-shot ASP deals with continuously changing logic programs, and it allows for ad-

vanced forms of search as in optimization problems.

6 Progress and Schedule

In October 2019, we started by doing an extensive review of the JSP papers that have been published recently. After reviewing the articles, we develop a single-shot ASP model to solve the JSP problem in January. Because of the results' limitations, we developed several techniques that decompose the problem into time windows and solve the problem using multi-shot ASP. The preliminary results were accepted as an extended abstract in workshop TAASP 2020. By submitting the extended results as a journal/conference paper, the first objective will be finalized in February 2021. The next step is to start working on the second objective, which is planned to be finished by December 2021. As the last two objectives are correlated, the fourth objective will be overlapped with the third objective, and the thesis will be completed in June 2023.

#	Phase	Time Span									
		Oct. 2019	April 2020	Oct. 2020	April. 2021	Oct. 2021	April. 2022	Oct. 2022	April. 2023		
1	Review of related models in literature	■		■	■	■	■	■	■		
2	Preliminary Experiments and develop an efficient method for large-scale JSP	■	■	■							
3	Build a model which handles breakdown machines			■	■	■					
4	Build a multi-objective model to optimize the stability and robustness					■	■	■			
5	Develop a model to handle other types of disruptions						■	■	■	■	
6	Validate and evaluate the results			■		■		■	■	■	
7	Thesis writing								■	■	■

Figure 1: Work plan.

References

- [1] M. R. Garey and D. S. Johnson, *Computers and intractability*, vol. 174. freeman San Francisco, 1979.
- [2] J. K. Lenstra, A. R. Kan, and P. Brucker, “Complexity of machine scheduling problems,” in *Annals of discrete mathematics*, vol. 1, pp. 343–362, Elsevier, 1977.
- [3] K. R. Baker, *Introduction to sequencing and scheduling*. John Wiley & Sons, 1974.
- [4] J. K. Lenstra and A. R. Kan, “Computational complexity of discrete optimization problems,” in *Annals of discrete mathematics*, vol. 4, pp. 121–140, Elsevier, 1979.
- [5] R. Zhang and C. Wu, “A hybrid approach to large-scale job shop scheduling,” *Applied intelligence*, vol. 32, no. 1, pp. 47–59, 2010.
- [6] M. A. Salido, J. Escamilla, F. Barber, and A. Giret, “Rescheduling in job-shop problems for sustainable manufacturing systems,” *Journal of cleaner production*, vol. 162, pp. S121–S132, 2017.
- [7] S. Zhang and T. N. Wong, “Flexible job-shop scheduling/rescheduling in dynamic environment: a hybrid mas/aco approach,” *International Journal of Production Research*, vol. 55, no. 11, pp. 3173–3196, 2017.
- [8] J. Xiong, L.-n. Xing, and Y.-w. Chen, “Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns,” *International Journal of Production Economics*, vol. 141, no. 1, pp. 112–126, 2013.
- [9] S. M. Sajadi, A. Alizadeh, M. Zandieh, and F. Tavan, “Robust and stable flexible job shop scheduling with random machine breakdowns: multi-objectives genetic algorithm approach,” *International Journal of Mathematics in Operational Research*, vol. 14, no. 2, pp. 268–289, 2019.
- [10] Y. Zhai, C. Liu, W. Chu, R. Guo, and C. Liu, “A decomposition heuristics based on multi-bottleneck machines for large-scale job shop scheduling problems,” *Journal of Industrial Engineering and Management (JIEM)*, vol. 7, no. 5, pp. 1397–1414, 2014.

- [11] D. Ouelhadj and S. Petrovic, “A survey of dynamic scheduling in manufacturing systems,” *Journal of scheduling*, vol. 12, no. 4, p. 417, 2009.
- [12] S. Mason, S. Jin, and C. Wessels, “Rescheduling strategies for minimizing total weighted tardiness in complex job shops,” *International Journal of Production Research*, vol. 42, no. 3, pp. 613–628, 2004.
- [13] K. B. Ali, A. J. Telmoudi, and S. Gattoufi, “Adopted rescheduling strategy for solving the dynamic job shop using ga based local search,” in *2019 International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, pp. 68–73, IEEE, 2019.
- [14] Y. M. Wang, H. L. Yin, and K. Da Qin, “A novel genetic algorithm for flexible job shop scheduling problems with machine disruptions,” *The International Journal of Advanced Manufacturing Technology*, vol. 68, no. 5-8, pp. 1317–1326, 2013.
- [15] L. Zhang, L. Gao, and X. Li, “A hybrid genetic algorithm and tabu search for a multi-objective dynamic job shop scheduling problem,” *International Journal of Production Research*, vol. 51, no. 12, pp. 3516–3531, 2013.
- [16] V. Lifschitz, “Answer set programming and plan generation,” *Artificial Intelligence*, vol. 138, no. 1-2, pp. 39–54, 2002.
- [17] M. Abseher, M. Gebser, N. Musliu, T. Schaub, and S. Woltran, “Shift design with answer set programming,” *Fundamenta Informaticae*, vol. 147, no. 1, pp. 1–25, 2016.
- [18] F. Ricca, G. Grasso, M. Alviano, M. Manna, V. Lio, S. Iiritano, and N. Leone, “Team-building with answer set programming in the gioia-tauro seaport,” *arXiv preprint arXiv:1101.4554*, 2011.
- [19] C. Guziolowski, S. Videla, F. Eduati, S. Thiele, T. Cokelaer, A. Siegel, and J. Saez-Rodriguez, “Exhaustively characterizing feasible logic models of a signaling network using answer set programming,” *Bioinformatics*, vol. 29, no. 18, pp. 2320–2326, 2013.
- [20] T. Soinenen and I. Niemelä, “Developing a declarative rule language for applications in product configuration,” in *International Symposium on Practical Aspects of Declarative Languages*, pp. 305–319, Springer, 1999.

- [21] M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub, “Multi-shot aspsolving with clingo,” *Theory and Practice of Logic Programming*, vol. 19, no. 1, pp. 27–82, 2019.