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

Decomposition Methods for Solving Scheduling Problem Using Answer Set Programming*

Mohammed M. S. El-Kholany
University of Klagenfurt
Cairo University
mohammed.el-kholany@aau.at

Abstract

This study proposes solving scheduling problems in industrial applications using the decomposition approach. The proposed model has been built using Multi-shot Answer Set Programming with Difference Logic. We tested our model with some benchmark instances and the results showed that our model is comparable to Constraint Programming and other heuristics in the literature.

1 Introduction

Scheduling is a crucial problem appearing in various domains, such as manufacturing, transportation, or healthcare, where the goal is to schedule given operations on available resources such that the operations are completed as soon as possible. Unfortunately, most scheduling problems cannot be solved efficiently, so research on suitable approximation methods is primary importance. Job-shop Scheduling Problem (JSP) is one of the most critical scheduling problems. The JSP has been proven to be NP-hard in [Baker, 1974]. Therefore, it was difficult to reach optimal solutions, even for small-scale problems. However, in the practical manufacturing environment, the problem scale is much larger, where the number of operations may be up to 10000 in some mechanical workshops [Zhang and Wu, 2010]. The main purpose of scheduling is to find the most appropriate sequencing of operations assigned to each machine while optimizing the performance indicator. A typical performance indicator for JSP is the makespan, which is the time needed to complete all jobs.

2 Problem Statement

The types of scheduling problems in industrial applications are quite big. We started with building a model that solves the classical JSP that consists of n jobs and a set of m machines. Each job has a series of operations executed sequentially and processed on a specific machine. The job must visit each machine once and it is not possible to interrupt the machine while it is processing. However, in reality, the number of operations is huge, making the problem more complex and the traditional techniques cannot find a good solution in a reasonable time. Such a problem requires applying an approach that can provide an efficient schedule in a reasonable time. One of the most effective approaches for solving JSP is decomposition which will be described in the next section [Ovacik and Uzsoy, 2012].

3 Contributions Accomplished

We have decided to build our model using Answer Set Programming (ASP), a declarative programming language rooted in artificial intelligence and computational logic research [Lifschitz, 2008]. ASP proved its effectiveness for solving combinatorial optimization problems in different fields [Banbara *et al.*, 2013; Abseher *et al.*, 2016]. Furthermore, an extension of ASP with Difference Logic (DL) has been proposed and applied to solve different scheduling problems [Gebser *et al.*, 2016; Kaminski *et al.*, 2017]. In addition, as mentioned before, we aim to apply the decomposition approach to solving the JSP and the multi-shot ASP allows us to apply the iterative optimization process [Gebser *et al.*, 2019].

In the first phase, we built an ASP model with DL that solves the whole JSP problem in a single shot. This model could reach optimal solutions for small instances. However, when the number of operations increased, the solver could not obtain near-optimal solutions in a reasonable time. This led us to apply a decomposition approach that splits the problem into sub-problems called Time Windows (TW). The main idea of the decomposition is to assign the operations in the most appropriate TWs where the precedence constraint must be respected. After the decomposition process, the optimization phase starts where the multi-shot ASP scheduler starts to optimize the first TW with a time limit and after finishing it, it optimizes the next TW(s), where the solution of each TW is an input for the next window. In the last step, the TW(s) solutions are merged to obtain the solution to the master problem.

Developing an efficient decomposition method is challenging because no optimal decomposition strategy is known for solving the JSP. We have proposed different decomposition strategies based on different techniques. Firstly, we have used a constrained k-Means clustering algorithm to gather the similar operations into one cluster based on some features gen-

*This Ph.D. study is under the supervision of Prof. Martin Gebser and Assoc. Prof. Konstantin Schekotihin

erated from the problem itself and/or some heuristics such as (FIFO and MTWR) where each cluster is a TW. This work has been published in PADL 2022 [El-Kholany *et al.*, 2022]. Secondly, we have applied four other decomposition strategies that rank the operations based on the Earliest possible Starting Time and Total Remaining Work with/without considering bottleneck machines. This work has been accepted in ICLP 2022. In order to improve our results, we have applied two different techniques. The first is the overlapping between TWs to overcome the decomposition mistakes. The second is the compression (post-processing phase) applied after optimizing each window. This process aims to check if there are operations that can start processing earlier if they do not violate the problem constraints. In order to assess our proposed model, we performed computational experiments with benchmark instances generated by [Taillard, 1993; Demirkol *et al.*, 1998]. We compared our obtained results with (FIFO/MTWR dispatching rules and Constraint Programming). We found that our model can provide better results than the heuristics in most cases and is comparable to Constraint Programming with 2000 operations.

4 Remaining Work

The classical JSP is limited and has many assumptions that make the problem unrealistic. Currently, we aim to build an ASP model that deals with a scheduling problem in semiconductor manufacturing systems. In order to accomplish this, we are working on a dataset based on a simulator called SMT2020 [Kopp *et al.*, 2020]. This dataset considers many factors such as batching, rework and cascading. We aim to build a model based on ASP while considering the mentioned factors and testing that model on a small-scale dataset. Afterward, our goal is to link the developed model with SMT2020 and investigate how our model behaves with such a problem compared to some dispatching rules in the literature.

5 Main Contribution

Our main contribution is to investigate the success/failure of ASP in scheduling problems. Furthermore, the scheduling problem is a hot topic in that period and there is no particular method/algorithm that could solve the problem efficiently. So that investigating the impact of applying different decomposition strategies followed by the optimization process using multi-shot ASP would be beneficial and provide insights that could help other researchers solve this problem in the future. For example, what we have found out from our studies since I started my Ph.D. study is that:

- Considering the bottleneck machines during the decomposition process is beneficial.
- The overlapping between TWs can partially overcome the decomposition mistakes in most cases.
- Compressing the TWs after the optimization process improves the solution quality.

We compared our obtained results to the optimal solutions of the benchmark instances that have 750 to 2000 operations to be scheduled. The evaluation showed that our model could reach solutions close to the optima by $1 - 10$ % in most cases in 1000 seconds. At the end of this study, we aim to have an effective model that could generate a short period (week) schedule in a semiconductor manufacturer on a particular fab.

References

[Abseher *et al.*, 2016] Michael Abseher, Martin Gebser, Nysret Musliu, Torsten Schaub, and Stefan Woltran. Shift design with answer set programming. *Fundamenta Informaticae*, 147(1):1–25, 2016.

[Baker, 1974] Kenneth R Baker. *Introduction to sequencing and scheduling*. John Wiley & Sons, 1974.

[Banbara *et al.*, 2013] Mutsunori Banbara, Takehide Soh, Naoyuki Tamura, Katsumi Inoue, and Torsten Schaub. Answer set programming as a modeling language for course timetabling. *Theory and Practice of Logic Programming*, 13(4-5):783–798, 2013.

[Demirkol *et al.*, 1998] Ebru Demirkol, Sanjay Mehta, and Reha Uzsoy. Benchmarks for shop scheduling problems. *European Journal of Operational Research*, 109(1):137–141, 1998.

[El-Kholany *et al.*, 2022] Mohammed El-Kholany, Konstantin Schekotihin, and Martin Gebser. Decomposition-based job-shop scheduling with constrained clustering. In *International Symposium on Practical Aspects of Declarative Languages*, pages 165–180. Springer, 2022.

[Gebser *et al.*, 2016] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, Max Ostrowski, Torsten Schaub, and Philipp Wanko. Theory solving made easy with clingo 5. In *Technical Communications of the 32nd International Conference on Logic Programming (ICLP 2016)*, 2016.

[Gebser *et al.*, 2019] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Multi-shot asp solving with clingo. *Theory and Practice of Logic Programming*, 19(1):27–82, 2019.

[Kaminski *et al.*, 2017] Roland Kaminski, Torsten Schaub, and Philipp Wanko. A tutorial on hybrid answer set solving with clingo. *Reasoning Web International Summer School*, pages 167–203, 2017.

[Kopp *et al.*, 2020] Denny Kopp, Michael Hassoun, Adar Kalir, and Lars Mönch. Smt2020—a semiconductor manufacturing testbed. *IEEE Transactions on Semiconductor Manufacturing*, 33(4):522–531, 2020.

[Lifschitz, 2008] Vladimir Lifschitz. What is answer set programming? *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 1594–1597, 2008.

[Ovacik and Uzsoy, 2012] Irfan M Ovacik and Reha Uzsoy. *Decomposition methods for complex factory scheduling problems*. Springer Science & Business Media, 2012.

[Taillard, 1993] Eric Taillard. Benchmarks for basic scheduling problems. *european journal of operational research*, 64(2):278–285, 1993.

[Zhang and Wu, 2010] Rui Zhang and Cheng Wu. A hybrid approach to large-scale job shop scheduling. *Applied intelligence*, 32(1):47–59, 2010.