Scheduling Solutions For Flexible Manufacturing Systems - Thesis Abstract

Eghonghon-aye Eigbe
Delft University of Technology, The Netherlands
e.eigbe@tudelft.nl

Abstract

In my research, I aim to develop scheduling algorithms that are both widely applicable and able to provide high quality solutions. My work has taken two main directions namely (i) expanding the capabilities of heuristic algorithms, and (ii) developing a general decision diagram solver for manufacturing systems that can be modelled as shops. These form two main parts of the thesis and its contributions are expanded below.

1 Introduction

Scheduling – the process of efficiently allocating limited resources – is often a critical determinant of the performance of manufacturing systems. Many relevant scheduling problems posed by manufacturing systems are also known to be NP-Hard. The difficulty is compounded by the increasing complexity of modern day manufacturing systems with tightly coupled cyber-physical systems playing a significant role.

Many models of manufacturing systems exist ranging from single and multi-machine models. One of the most common models is the shop model where manufacturing systems are viewed as collections of jobs and machines with constraints between them. Shop models are quite universal and are flexible enough to accommodate many problem-specific constraints and are thus able to cover a significant amount of real world manufacturing systems. As such, we primarily deal with this model.

Our research is part of the Scheduling Adaptive Modular Flexible Manufacturing Systems (SAM-FMS) project which aims to develop widely applicable scheduling algorithms fit for the challenges faced by smart manufacturing systems.

2 Contributions

My thesis is split into two main bodies of work. In the first part, I have looked into extending a class of scheduling algorithms to handle maintenance actions. In the second part of the thesis, I have developed a decision diagram solver that generalises easily to many problem setups and can absorb many ideas from other scheduling methods. In this section, we highlight the contributions of both bodies of work.

2.1 Maintenance Aware Heuristic Scheduling

Maintenance is an inevitable reality in manufacturing systems. Machines are not perpetually available and wear from use. Many scheduling algorithms ignore this reality when making production schedules and those that consider maintenance usually propose dedicated algorithms for the maintenance scenario they consider.

In [Eigbe *et al.*, 2023], we introduce a framework for allowing priority dispatching rules – which are quite common in industrial settings – to account for maintenance scheduling. The general idea is to keep the core algorithm unchanged but add a deterioration model and maintenance policy such that the effect of maintenance on the schedule can be evaluated at every decision point.

This work was done in collaboration with an industrial partner and some of the maintenance scheduling ideas are currently being valorised in a product development cycle.

2.2 Decision Diagrams for Shop Scheduling

The significant portion of my thesis research has also been on creating a decision diagram based solver for shop scheduling. Decision diagrams are graphical data structures used in discrete optimisation to represent possible assignments to variables [Bergman *et al.*, 2016] and have shown promise for solving scheduling problems. However, they still face scalability issues when used as solvers and are often used instead to find bounds on the objective [van den Bogaerdt and de Weerdt, 2018].

We chose to study this particularly for the potential of decision diagrams to generalise many other methods. The construction of a decision diagram closely mirrors dynamic programming formulations and can also be treated as a sequential decision making process. Thus, many constructive heuristics can be directly applied over a decision diagram structure. On the other hand, paths in a decision diagram are solutions and they can also be manipulated as a whole. This gives us a chance to also apply methods that manipulate whole solutions at once over a decision diagram. The diagram also encodes the entire solution space and so finding a provable optimal solution is still a possibility.

Part of this work has resulted in a workshop paper [Eigbe *et al.*, 2024] and the rest of the contributions are planned as 2 submissions (1 journal and 1 conference) before graduation.

A Model for Multi-Machine Scheduling

We introduced a model for multi machine scheduling to fit into the problems solvable by *decision diagram* based solvers. The model is in essence a dynamic programming model with markovian states and transitions.

Empirical results on real-world benchmarks with an industrial partner show that a generic decision diagram solver with this model is competitive with state-of-the-art mixed integer programming solvers and constraint programming solvers – while being demonstratively more scaleable to industrial-sized instances.

Incremental Refinement

The solver in the previous section was based on top-down construction paradigm for decision diagrams but it is also possible to treat the diagram as a constraint store and incrementally refine it to remove infeasible arcs until only feasible paths remain. We present the first model for multi-machine sequencing where machines are not the same and jobs have specific assignments to (a set of) machines. This generalises the multi machine scheduling work of [van den Bogaerdt and de Weerdt, 2018].

Reinforcement Learning Based Restricted Decision Diagrams

The step most likely to produce solutions in a branch and bound based decision diagram is the restriction step. The heuristic used to compute the restriction is thus very important for the quality of solutions provided. Typically, the DD literature uses restrictions that greedily discard nodes at every layer but this could be a good place to insert domain knowledge by using user defined heuristics to compute the diagram. We can go a step further and use a learned heuristic. We do just that in [Eigbe *et al.*, 2024] by combining a reinforcement learning agent with the decision diagram so that anytime a restriction is built, it is built by a trained RL agent. This incorporation helps improve the solving capability of the agent providing promising results both for the improvement of the RL agent and the DD solving process.

The Role of Dominance

Dominance allows us to safely prune unpromising portions of decision diagram leading to reduction in diagram size. We also contribute to the literature by developing a suite of dominance conditions for different constraints.

Clustering and Conflict Based Refinement Strategies

When refining decision diagram, a key step is node splitting. Inspired by the work of [Nafar and Römer, 2024], we introduce a clustering based node-splitting algorithm where edges coming into a node are clustered and each cluster is redirected to the same node. In addition to splitting the nodes themselves, we also have to make a decision of which nodes to split. We propose to look at conflict based selection of which nodes to split where nodes with the highest amount of conflicts between incoming and outgoing edges are split first.

Warm Start Strategies

Motivated by the fact that heuristics still provide state of the art performance for many scheduling problems, we also design our approach to start out with an initial solution provided by a heuristic and strategically improve it while retaining the properties of an exact solver. In top down construction, the diagram can simply be seeded with a path as in [Gillard and Schaus, 2022] but for incremental refinement this is not so clear. We design a method that peels off the warm start solution(s) from the initial diagram by employing the peel operation introduced in [Rudich *et al.*, 2023] to only a single path instead of a single node. Thus, ensuring the warm start exists as a unique path within the diagram with the possibility to prune other dominated paths or influence the refinement operations.

3 Summary

In summary, the thesis deals with scheduling manufacturing system and handles two main topics. The first topic is on the integration of context information with existing scheduling algorithms and the second topic is on the development of a decision diagram based solver for the class of problems discussed. Contributions have led to 3 papers (1 journal and 2 workshops) with 2 more papers (1 journal and 1 conference) in the works.

References

[Bergman *et al.*, 2016] David Bergman, Andre A Cire, Willem-Jan Van Hoeve, and John Hooker. *Decision Diagrams for Optimization*. Springer, 2016.

[Eigbe *et al.*, 2023] Eghonghon-Aye Eigbe, Bart De Schutter, Mitra Nasri, and Neil Yorke-Smith. Sequence- and time-dependent maintenance scheduling in twice re-entrant flow shops. *IEEE Access*, 2023.

[Eigbe *et al.*, 2024] Eghonghon-aye Eigbe, Christoph Schmidl, Bart De Schutter, Nils Jansen, Mitra Nasri, and Neil Yorke-Smith. Neural decision diagrams. In *6th Data Science Meets Optimisation Workshop (DSO) at the 33rd International Joint Conference on Artificial Intelligence(IJCAI)*, 2024.

[Gillard and Schaus, 2022] Xavier Gillard and Pierre Schaus. Large neighborhood search with decision diagrams. In *IJCAI*, pages 4754–4760, 2022.

[Nafar and Römer, 2024] Mohsen Nafar and Michael Römer. Using clustering to strengthen decision diagram bounds for discrete optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8082–8089, 2024.

[Rudich *et al.*, 2023] Isaac Rudich, Quentin Cappart, and Louis-Martin Rousseau. Improved peel-and-bound: Methods for generating dual bounds with multivalued decision diagrams. *Journal of Artificial Intelligence Research*, 77:1489–1538, 2023.

[van den Bogaerdt and de Weerdt, 2018] Pim van den Bogaerdt and Mathijs de Weerdt. Multi-machine scheduling lower bounds using decision diagrams. *Operations Research Letters*, 46(6):616–621, 2018.