Research Plan for Combinatorial Optimisation for Scheduling

Jasper Vermeulen

April 22, 2022

Background of the research

The problem of scheduling tasks arises in industries all the time. It is not hard to imagine that generating an optimized schedule can be of great profit for production or logistic operations. Optimization can for example be minimizing the overall required time or minimizing the delay before starting a task. Because this type of problem is so prevalent it has already been subject to much research.
Formally this specific type of problem is known as the resource-constrained project scheduling problem (RCPSP). On its own the problem definition for RCPSP is to limited to be of use for realistic application. To make sure the researched algorithms solving the scheduling problem would have a wider use case many variations and extensions to the problem definition have been classified over time [7], [2]. More recently the variations and extensions have also been surveyed and put into a structured overview [6].
For this research the preemptive resource-constrained project scheduling problem with setup times (PRCPSP-ST) variant is under study. Preemption allows an activity to be interrupted during its scheduled time by another activity. Each interruption can be seen as a split of the activity into multiple smaller activities. The setup times are introduced for each interruption in an activity to discourage endless splits resulting in a chaotic schedule. This model had already been established [8] and a proposed algorithm was found to result in a reduction of the makespan compared to the optimal schedule without activity preemption [9]. Within this algorithm the activities are split into all possible integer time segments and a SAT solver made a selection from these segments [3]. The resulting list was used to construct a schedule with a genetic algorithm established in earlier research [5].

Research Question

This research will try to answer the following question:

> Can the addition of a simple heuristic to a SAT solver algorithm used to solve to PRCPSP-ST models reduce the average makespan of the resulting schedule in an equal amount time?

Because the RCPSP is known to be strongly NP-hard [1] any general algorithm that can be applied to the problem might be outperformed by an algorithm specialized for the specific variation. To specialize an algorithm this way knowledge or an insight into the problem variation can be translated into a heuristic rule for the algorithm to use. The intention is that this heuristic rule will increase the performance of the algorithm. Because scheduling is so prevalent in industries performance better performing algorithms are desirable ways to increase profits.
It has already been shown that allowing for preemption in schedules can lead to a reduction in the resulting makespan even if penalties are given for each activity that is not finished from start to finish [9]. For this research a similar algorithm will be setup that uses a SAT solver algorithm to make a selection of activity segments and constructs a schedule from the selected segments. The expectation

for this research is to show that a heuristic version of the SAT solver algorithm will result in a lower makespan when running an equal amount of time.

- What is a model for the activities in a RCPSP problem?
- How can the RCPSP model be transformed to include activity preemption?
- How can setup times be included in the RCPSP model for splits in an activity schedule?
- What is the role of a SAT solver in an RCPSP optimization algorithm?
- What is the required input of a SAT solver?
- How can PRCPSP-ST problems be modelled as an input to a SAT solver for activity segment selection?
- How can the output of a SAT solver be used to construct an activity schedule?
- What could be a possible heuristic for the SAT solver algorithm?
- Where can a heuristic be used by a SAT solver algorithm?
- Why could a heuristic be beneficial for solving PRCPSP-ST problems?
- How does the heuristic change the resulting activity schedule?
- How does the heuristic change the time required by the SAT solver algorithm?

Method

The schedule construction might be done using a simpler method than the genetic algorithm that has been used before [5] because this study will try the use of a heuristic in the SAT solver part of the algorithm. When this algorithm is implemented a benchmark can be made using a known and efficient SAT solver algorithm like DPLL [4]. With this baseline benchmark a heuristic version of the DPLL algorithm can be tested and results will show if a there is a reduction in the makespan after running the algorithm for the same amount of time.

Planning of the research project

Meetings with the project supervisor and peergroup will be held weekly on Wednesdays at 13:00. These meetings are used for feedback on project process and problem resolution.

0.1 Project timeline

Research phase	Objectives	Deadline
1. Background research	• Learn about existing research • Gather information on modelling RCPSP problems • Make a model for PRCPSP-ST model variant • Design a complete algorithm from dataset to schedule • Theorize multiple possible heuristics	May 6, 2022
2. Implementation	• Implement the PRCPSP-ST problem model • Implement a complete algorithm • Add multiple heuristic SAT algorithm alternatives	May 20, 2022
3. Performance tests	• Generate a baseline benchmark • Test multiple heuristic algorithms vs baseline • Optimize algorithms if necessary • Gather analytical result data	June 3, 2022
4. Data analysis and report	• Aggregate result data • Perform statistical analysis • Generate (graphical) representations of the data • Report findings on the research question according to data	June 17, 2022
5. Presentation	• Inform examiner, supervisor and peers on results	June 24, 2022

0.2 Deliverables

Deliverable	Deadline
Research proposal: first week plan	April 19, 2022
Information Literacy	April 20, 2022
Research proposal: a document describing what will be done and when	April 24, 2022
Research proposal presentation	April 24, 2022
Academic Communication Skills: First 300 words	May 7, 2022
Academic Communication Skills: Midterm poster (for feedback)	May 12, 2022
Midterm presentation (+ poster)	May 16, 2022
Academic Communication Skills: Improve first 300 words, and add section (300 words)	May 19, 2022
Scientific paper: v1 for peer feedback on writing and content feedback by supervisor	May 30, 2022
Peer review on v1 paper from another student	June 2, 2022
Scientific paper: v2 for feedback on both content and writing by supervisor	June 8, 2022
Poster summarizing research	June 17, 2022
Scientific paper: final version	June 19, 2022
Software programmed to obtain results	June 19, 2022

References

[1] J. Blazewicz, J. K. Lenstra, and A. H. G. Rinnooy Kan. Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, 5(1):11–24, 1983.

[2] Peter Brucker, Andreas Drexl, Rolf Möhring, Klaus Neumann, and Erwin Pesch. Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112(1):3–41, 1999.

[3] José Coelho and Mario Vanhoucke. Multi-mode resource-constrained project scheduling using rcpsp and sat solvers. *European Journal of Operational Research*, 213(1):73–82, 2011.

[4] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, 1962.

[5] Dieter Debels and Mario Vanhoucke. A decomposition-based genetic algorithm for the resource-constrained project-scheduling problem. *Operations Research*, 55(3):457–469, 2007.

[6] Sönke Hartmann and Dirk Briskorn. A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1):1–14, 2010.

[7] Willy Herroelen, Erik Demeulemeester, and Bert De Reyck. *A Classification Scheme for Project Scheduling*, pages 1–26. Springer US, Boston, MA, 1999.

[8] Lori Kaplan. Resource-constrained project scheduling with setup times. 1991.

[9] Mario Vanhoucke and José Coelho. Resource-constrained project scheduling with activity splitting and setup times. *Computers Operations Research*, 109:230–249, 2019.