Research Plan for Combinatorial Optimisation for Scheduling

Jasper Vermeulen

April 23, 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 [10], [2]. More recently the variations and extensions have also been surveyed and put into a structured overview [9].
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. Both a model for allowing preemption [8] and including setup times [11] have already been established and a proposed algorithm for a combined model was found to result in a reduction of the makespan compared to the optimal schedule without activity preemption [13]. 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 [6].

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 [13]. 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.

The answers to the following supporting sub-questions will help towards answering the main research question.

- 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 required input of a SAT solver?
- How can PRCPSP-ST problems be modelled as an input to a SAT solver for activity segment selection?
- What could be a possible heuristic for the SAT solver algorithm?
- Where can a heuristic be used by a SAT solver algorithm?
- Does the heuristic reduce the resulting activity schedule makespan compared to the default algorithm?
- Does the heuristic reduce the time required by the SAT solver algorithm compared to the default algorithm?

Method

This research will first need a model definition for PRCPSP-ST problems. The model takes a standard RCPSP dataset and includes the possibility for splitting the activities at integer parts for a time penalty trade-off. There are existing activity network for both setup times [11] and preemption [8] that can be combined. This combined model results in a network that is the same as a standard RCPSP activity network and can therefore be solved by any RCPSP solving algorithm. The extended activity network includes all possible activity segments and needs to be transformed into an input for the SAT solver. For the algorithm to work the input of the SAT solver must force the result to only contain each activity segment once. The SAT solver that is going to be used will be one of the implementations found on `https://maxsat-evaluations.github.io/2021/descriptions.html` using the DPLL algorithm [5][4]. When the SAT solver returns a selection of segments it can be used to construct a schedule with a minimum makespan. The process going from the combined model to the segment selection has been done in a research before [13].
For the construction of the schedule from the selected activity segments the established branch-and-bound algorithm [7] can be updated to specific PRCPSP-ST requirements. Previous attempts at altering the branch-and-bound algorithm to include activity preemption has already proven to be successful [14]. When the schedule is constructed the objective value must be calculated and for this research it will be the schedule makespan. The algorithm will make multiple iterations within a given time ensuring different results from the SAT solver and when a schedule with a lower makespan is found it is saved. When the time runs out the final makespan is returned.

With the first algorithm implementation a baseline benchmark can be generated. Because the RCPSP class of problems is widely studied has a well known and researched dataset [12] that can be downloaded at `https://www.om-db.wi.tum.de/psplib/data.html`. This will also provide optimal schedule solutions without preemption and setup times to check the quality of the results if necessary.

A last step is including a heuristic in the SAT solver algorithm to make use of problem specific knowledge. If this research is to be successful the resulting set of selected activity segments will result in a lower makespan after the branch-and-bound algorithm is run. A single iteration might take longer but as long as the final makespan after the same given time is lower the heuristic algorithm will be considered more optimal.

Planning of the research project

Meetings with the project supervisor and peer group 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] Jacek Blazewicz, Jan Karel Lenstra, and Alexander H. G. Rinnooy Kan. Scheduling subject to resource constraints: classification and complexity. *Discret. Appl. Math.*, 5:11–24, 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] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, 1960.

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

[7] Erik Demeulemeester and Willy Herroelen. A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Management Science*, 38(12):1803–1818, 1992. doi: 10.1287/mnsc.38.12.1803.

[8] Erik L. Demeulemeester and Willy S. Herroelen. An efficient optimal solution procedure for the preemptive resource-constrained project scheduling problem. *European Journal of Operational Research*, 90(2):334–348, 1996.

[9] 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.

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

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

[12] Rainer Kolisch, Arno Sprecher, and Andreas Drexl. Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science*, 41(10):1693, 1995. Copyright - Copyright Institute for Operations Research and the Management Sciences Oct 1995 CODEN - MNSCDI.

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

[14] Mario Vanhoucke and Dieter Debels. The impact of various activity assumptions on the lead time and resource utilization of resource-constrained projects. *Computers Industrial Engineering*, 54(1):140–154, 2008.