Resource-constrained project scheduling with task preemption and setup times by heuristically augmented SAT solver

Jasper Vermeulen[1]
Supervisor: Emir Demirović[1]
[1]EEMCS, Delft University of Technology, The Netherlands
J.Vermeulen-1@student.tudelft.nl, E.Demirovic@tudelft.nl

Abstract

TBD in final paper.

1 Introduction

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). The standard RCPSP has a set of tasks that each require a specified resource for a given amount of time and precedence restrictions are given for the tasks. On its own this problem definition for standard RCPSP is limited and of little use to realistic applications where additional constraints must be followed or more options are available. 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 [8] [2]. More recently the variations and extensions have also been surveyed and put into a structured overview [7].

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 [5] and including setup times [9] 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 [10]. 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 [4].

In the research done on solving RCPSP variants the focus has been on heuristic and meta-heuristic algorithms. These algorithms are usually variants of branch-and-bound algorithm [5] or a form of genetic algorithms [6] that were established for the standard RCPSP.

The use of SAT solvers as a part of the algorithm was mentioned earlier but using a SAT solver as a complete solution to try and solve PRCPSP-ST instances has not been researched before. This leaves room to try and find out if a SAT solver that can outperform the heuristic and meta-heuristic algorithms. Because the RCPSP is known to be strongly NP-hard [1] the SAT solver might not be more efficient than the the heuristic and meta-heuristic at first. The heuristic part means the algorithms are specialized by knowledge or an insight into the problem variation translated into a heuristic rule for the algorithm to use. The intention is that this heuristic rule will increase the performance of the algorithm. A SAT solver can also be modified to include a heuristic for the PRCPSP-ST variant. The augmentation of the SAT solver making it more specialized for the PRCPSP-ST variant could result in a better performance than other algorithms. So, the expectation for this research is to show that a heuristically augmented version of the SAT solver algorithm will result in a lower makespan than a heuristic algorithm when running an equal amount of time.

References

[1] J. Blazewicz, J. K. Lenstra, and A. H. G. Rinnooy Kan. Scheduling subject to resource constraints: classification and complexity. *Discret. Appl. Math.*, 5:11–24, 1983.

[2] P. Brucker, A. Drexl, R. Möhring, K. Neumann, and E. Pesch. Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112(1):3–41, 1999.

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

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

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

[6] S. Hartmann. A competitive genetic algorithm for resource-constrained project scheduling. *Naval Research Logistics (NRL)*, 45(7):733–750, 1998.

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

[8] W. Herroelen, E. Demeulemeester, and B. De Reyck. *A Classification Scheme for Project Scheduling*, pages 1–26. Springer US, Boston, MA, 1999.

[9] L. Kaplan. Resource-constrained project scheduling with setup times. 1991.

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