

Exploring Heuristic Methods in SAT Solvers for the RCPSP-Log*

Melle Schoenmaker,[©] Supervised by: Emir Demirović

[©]M.Schoenmaker@student.tudelft.nl

1: Background Information

- Optimal project scheduling is important for the efficiency of factory processes and logistics
- There exist meta-heuristic and exact approaches to solve instances of the **R**esource **C**onstrained **P**roject **S**cheduling **P**roblem (**RCPSP**)
- The RCPSP can be encoded into **C**onjunctive **N**ormal **F**orm (**CNF**) and solved with a satisfiability (SAT) solver¹.

2: Problem Statement

- The RCPSP is an NP-Hard problem, with the goal of minimising the project's makespan
- Set of activities, resources and AND precedence relations. AND precedence requires all predecessors to be finished.
- RCPSP-Log extends precedence with OR and BI constraints
- OR requires at least one predecessor to be finished
- The BI relation prevents parallel execution

Figure 1 shows an example project. Yellow, green, blue and red are used for dummy activities, activities with AND, OR and BI precedence constraints, respectively

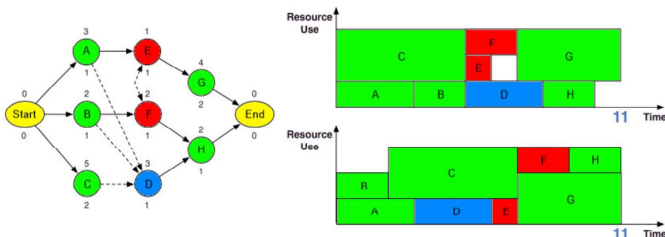


Figure 1: Activity network and schedules corresponding to an RCPSP instance and its RCPSP-Log variant. Adapted from: [1]

3: Problem Analysis

- Research in algorithms for NP-Hard problems has developed various meta-heuristic approaches and applied them to the RCPSP and its variants
- Little research has gone into simple heuristic approaches

4: Research Question

May SAT solving algorithms be augmented with domain-specific information to improve the solving of the resource-constrained project scheduling problem with logical constraints?

*Resource Constrained Project Scheduling Problem with Logical Constraints

¹<https://maxsat-evaluations.github.io/2021/>

²<https://www.om-db.wi.tum.de/psplib/main.html>

5: Methodology

The steps taken in this research are as follows:

1. Create a heuristic algorithm for the RCPSP-Log
2. Encode the RCPSP-Log into CNF without and with a heuristically calculated upper bound to the makespan
3. Compare the encodings and their solutions found by the pumpkin SAT solver

6: Results

Results have been found by comparing the execution times of encoding and SAT solver for the encoding with and without reduction by a heuristic upper bound. The problem instances published in PSPLIB² were used to make fair comparisons.

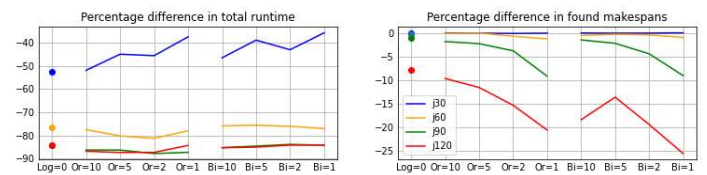


Figure 2: Total runtime of creating and solving reduced encoding, and the found makespans, compared to the standard encoding.

Conclusion

- Significantly less time is needed to compute optimal solutions when the solver is given a reduced encoding.
- Reduction is greater for instances with more jobs, since the horizon of these instances grows faster than the heuristically found makespan.
- Differences in the makespans found after timeout also show that the reduced encoding provides better results when given the same amount of time as the standard encoding.

References

- [1] M. Vanhoucke and J. Coelho. An approach using sat solvers for the rcpsp with logical constraints. European Journal of Operational Research, 249(2):579, 2016.