AUGMENTING VSIDS HEURISTIC FOR THE RCPSP/T BY INITIALIZING ACTIVITY VALUES USING DOMAIN-SPECIFIC INFORMATION

Tijs Lenssen

Responsible Professor: Emir Demirović

Supervisors: Maarten Flippo, Konstantin Sidorov

The VSIDS heuristic is one of the most effective variable selection heuristics for CDCL SAT solvers. It works by keeping track of activity values for each variable, which usually start out arbitrarily at zero.

This research presents several adaptations of the VSIDS heuristic specialized for solving the Resource-Constraint Project Scheduling Problem with time varying resource availabilities and demands (RCPSP/t). The research aims to answer the Research Question:

> **TO WHICH EXTEND CAN INITIALISING VSIDS ACTIVITY VALUES USING PROBLEM-SPECIFIC INFORMATION HELP A SAT SOLVER FIND SOLUTIONS FOR THE RCPSP/T?**

MOTIVATION

The RCPSP and its variants find practical usage in the scheduling of a wide range of projects from a lot of different fields:

- Construction Projects [5]
- Medical Research Projects [2]
- Industrial Processes
- And many more!

There have been promising results based on solving the RCPSP/t with a SAT solver, but that solver did not make use of problem-specific knowledge. Therefore it would be interesting to know if augmenting VSIDS with problem-specific information could result in faster solve times.

RCPSP/T

- A number of tasks need to be scheduled with the goal of minimising the total length of the schedule, the *makespan*, under certain constraints:
  - Precedence Constraints: Some tasks need to be scheduled before other tasks (see Figure 1)

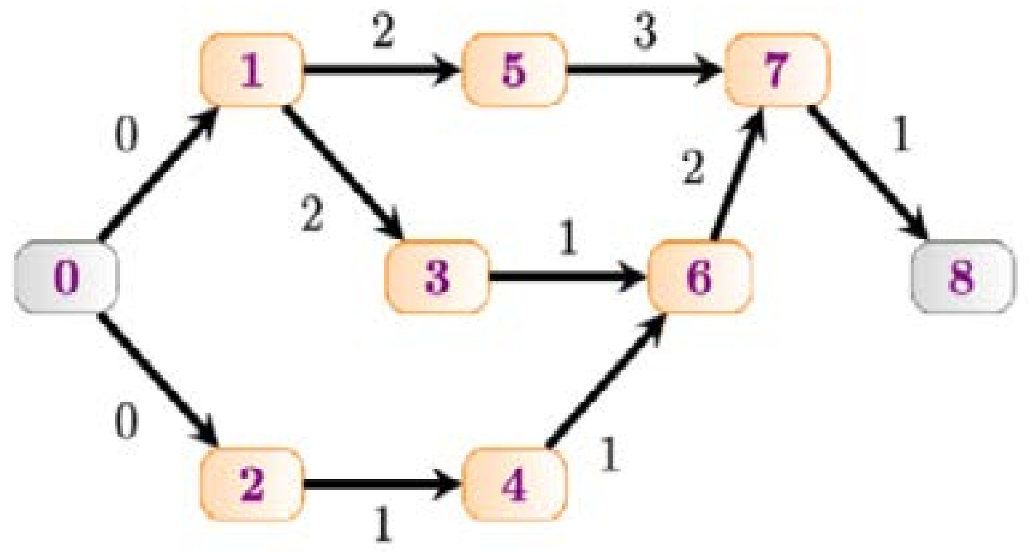

Figure 1: Precedence Graph, Adapted from [1]

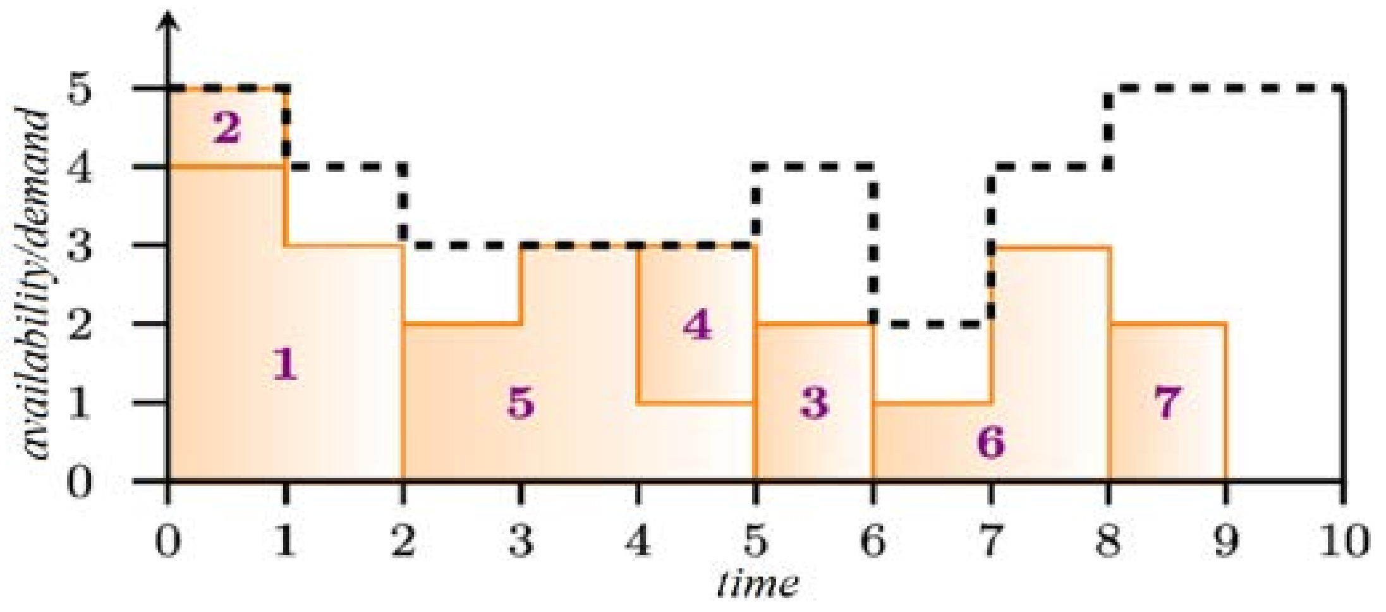

Figure 2: Optimal Schedule for the example, Adapted from [1]

- Resource Constraints: There are some resources available, which tasks can require a part of. In the RCPSP/t these resource availabilities and demands don't have to be constant and can vary over time (see Figure 2)

CDCL & VSIDS

Conflict-Driven Clause-Learning:

- Selects Variable and assigns value based on heuristic
- Propogates clauses using *unit propagation*
- Analyses conflicts to learn clauses and backtracks

Variable State Independent Decaying Sum heuristic:

- *bumps* activity values of variables that were recently involved with a conflict
- Periodically also *decays* all activity values
- Results in a heuristic that focusses on variables that have recently appeared in conflicts

VSIDS AUGMENTATION

- Three types of variables in the SAT encoding (*start*, *process* & *resource* variables)
- Simple method: initialise all *start* variable activity values to a constant *c*
- Other methods: initialise activity based on earliest/latest time, or most or least restricted tasks

EXPERIMENTAL SETUP

- MAX-SAT solver from our supervisor
- Instances from the J120t dataset from PSPLIB [3]
- RCPSP/t encoding from research by Pleunes [4]
- Experiments ran on *DelftBlue*[1] with a memory limit of 32GB and a time limit of 60s

[1] https://www.tudelft.nl/dhpc

RESULTS & DISCUSSION

Table 1: Results for dataset J120t (3600 instances, $c = 10^4$)

	VSIDS	Aug. VSIDS	Improvement
#solutions	2402	2806	17%
#optimal	1660	1743	5%

- Augmented heuristic found solutions for 17% more instances
- Augmented heuristic certified optimality for 5% more instances
- Biggest effect is at the beginning of the search (see Figure 3)
- Only the very first decisions seem to have a big effect on the performance, which could also explain:
  - Little difference for any $c > 0$
  - Little difference between different methods

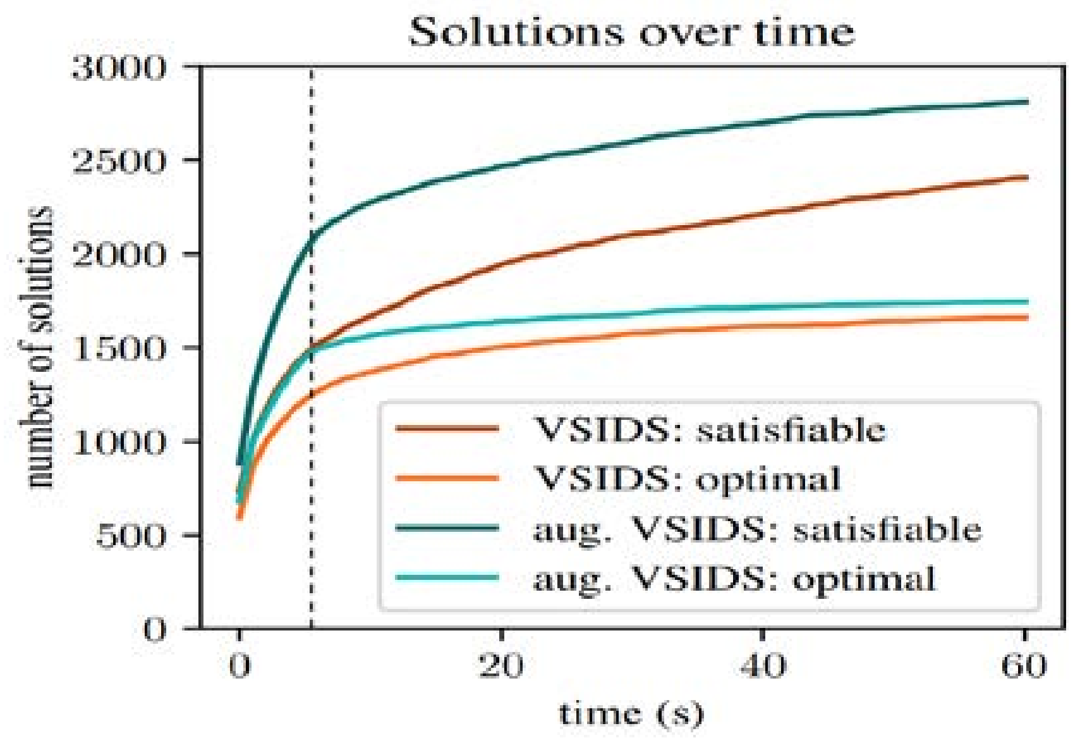

Figure 3: The number of first solutions and optimal solutions found over time for the original VSIDS and the augmented VSIDS ($c = 10^4$)

CONCLUSION & FUTURE WORK

- It is possible to get better results by initialising VSIDS activity values!
- It would still be interesting to find out:
  - Are there other problem-specific improvements?
    - Even better initialisation of activity values
    - Changing the *bump* amount per variable
  - Is this method applicable for other problems?
    - Other RCPSP variants
    - Totally different problems that have SAT encodings

REFERENCES

[1] M. Bofill, J. Coll, J. Suy, and M. Villaret, "SMT encodings for Resource-Constrained Project Scheduling Problems," *Computers & Industrial Engineering*, vol. 149, 2020, doi: 10.1016/j.cie.2020.106777.

[2] S. Hartmann, "Project scheduling with resource capacities and requests varying with time: a case study," *Flexible Services and Manufacturing Journal*, vol. 25, no. 1-2, pp. 74-93, 2013, doi: 10.1007/s10696-012-9141-8.

[3] R. Kolisch and A. Sprecher, "PSPLIB - A project scheduling problem library: OR Software - ORSEP Operations Research Software Exchange Program," European Journal of Operational Research, vol. 96, no. 1, pp. 205-216, 1997/01/10/ 1997, doi: https://doi.org/10.1016/S0377-2217(96)00170-1.

[4] J. Pleunes, "SAT-based optimisation for the resource-constrained project scheduling problem with time-dependent resource capacities and requests," BSc Computer Science and Engineering Bachelor thesis, EEMCS faculty Delft University of Technology, Delft University of Technology, Delft University of Technology Education Repository, 2022. [Online]. Available: http://resolver.tudelft.nl/uuid:698692ab-3a12-4a66-ae6a-214d77d4a013

[5] M. Vanhoucke, "The Westerschelde Tunnel Project," in *Project Management with Dynamic Scheduling: Baseline Scheduling, Risk Analysis and Project Control*, M. Vanhoucke Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 175-185.