Integration of variable selection heuristics into a MaxSAT solver for solving the MRCPSP

Author: Denis Tsvetkov D.R.Tsvetkov@student.tudelft.nl; Responsible Professor: Emir Demirović; Supervisors: Konstantin Sidorov and Maarten Flippo

1. Problem

- The **M**ulti-mode **R**esource **C**onstrained **P**roject **S**cheduling **P**roblem is an extension of the RCPSP;
- The MRCPSP is characterized by:
  - 2 types of resources - renewable (e.g., machines available) and non-renewable (e.g., budget);
  - multiple execution modes for each task – 1 person needs 6 days to complete task 1, whereas 2 people – only 3 (see Fig. 1);
  - precedence relations – to work on task 3, one must first complete task 2 (see Fig. 1);
  - goal - minimize the time to complete all activities (see Fig. 2).
- MRCPSP is known to be **NP-Hard** [1];

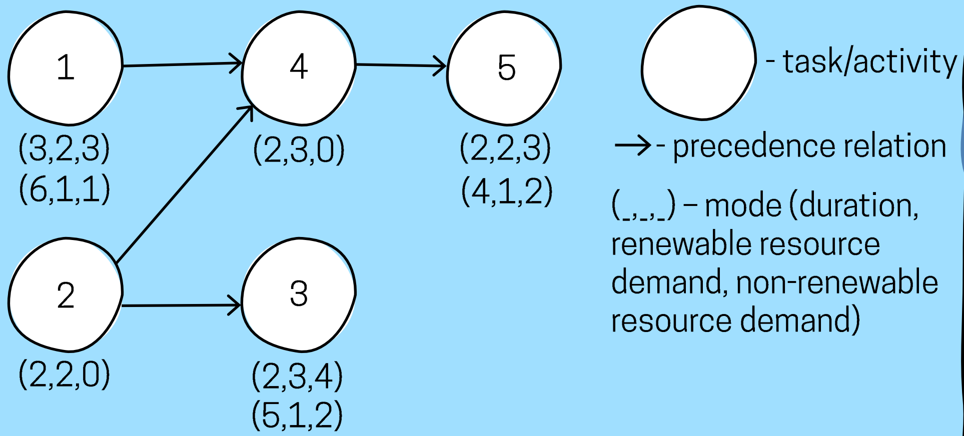

Fig. 1. Graph example of a MRCPSP instance with 5 activities.

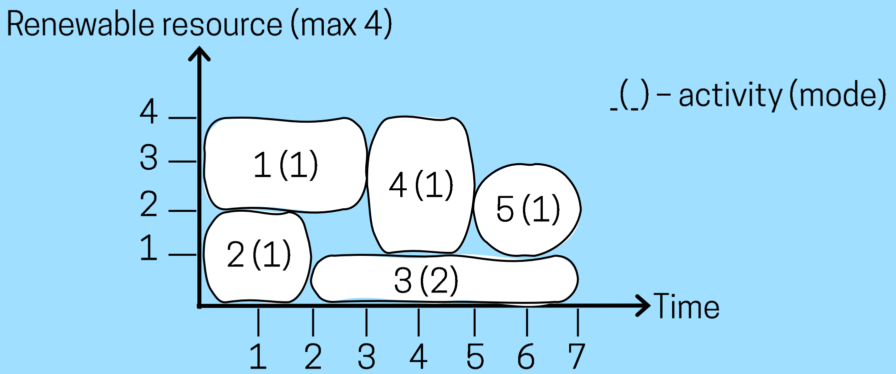

Fig. 2. Optimal schedule with makespan of 7 for the example above.

2. Question

What **heuristics**, specific to MRCPSP, can be integrated into the variable selection strategy of a **MaxSAT solver** and how do they **impact the performance** of the solver?

3. Background

What? A MaxSAT solver's goal is to find a variable assignment that maximizes the number of satisfied clauses in a given set of (weighted) clauses.

Why? MaxSAT solvers are more flexible than dedicated algorithms, as they can be applied to any problem that can be encoded as a Boolean formula in conjunctive normal form (CNF).

How? To use a MaxSAT solver, we first need to encode the MRCPSP into CNF. For example, **(**execute task 1 in mode 1 **or** mode 2**) and (**start task 1 at time 0 **or** at time 1**)**.

4. Heuristics

Variable Selection

VSIDS (baseline)

Provides dynamic ranking for the variables. Each variable is assigned a float value that is updated when a clause is learned and decayed at regular intervals.

SFM

Pick the shortest feasible mode first. Considered to be one of the "most effective scheduling rules" for the MRCPSP [2] .

EST

Schedule activities as early as possible. This is rational as we are trying to minimize the makespan. Also, it has a low overhead because it is static.

5. Experiment and Results

- 3 heuristic configurations – VSIDS, VSIDS + EST + SFM (VSIDS++), and EST + SFM (ESTSFM);
- 4 metrics – division of solutions into optimal, satisfiable, and timeouts, CPU time, number of decisions, and area under the time-objective curve;
- 6 benchmarks, ranging in terms of the number of tasks, number of modes per activity, or number of renewable and non-renewable resources;
- VSIDS++ (significantly) outperforms VSIDS on every metric, benchmark, and timeout / memory configuration (see Table 1);
- ESTSFM proves to be inefficient compared to the other 2 alternatives (see Table 1).

Table 1. Results for the 3 heuristics run on 631 instances with 30 activities, 3 modes per activity, 2 renewable and 2 non-renewable resources.

Metric	Timeout / Memory								
	5s / 1GB			20s / 1GB			60s / 2GB		
Division of solutions	38,445,148	285,216,130	60,278,293	293,188,150	360,141,130	251,123,257	371,109, 151	430,121,80	355,122,154
CPU time	4.67s	3.61s	5.19s	12.13s	7.87s	14.82s	20.7s	16.13s	32.93s
Number of decisions	7063709	536387	10056199	9053227	639097	28793461	9186846	680804	39235787
AUC	989.55	347.31	288.54	1600.39	496.91	768.41	1983.22	712.59	1626.91

1 – VSIDS 2 – VSIDS + EST + SFM (VSIDS++) 3 – EST + SFM (ESTSFM)

6. Conclusions

- VSIDS, which is widely used in MaxSAT solvers, cannot be completely replaced by domain-specific heuristics;
- Propositional-logic algorithms can be combined with domain-specific knowledge to obtain a more efficient algorithm.

7. Future Work

- Test the rest of the scheduling heuristics proposed in [2];
- Incorporate previous work – e.g., a more compact MaxSAT encoding of the problem;
- Compare the improved solver, which uses VSIDS++, to other state-of-the-art approaches for solving the MRCPSP.

References

[1] J. Blazewicz, J.K. Lenstra, and A.H.G.Rinnooy Kan."Scheduling subject to resource constraints: classification and complexity" (1983).
[2] F. F. Boctor. "Heuristics for scheduling projects with resource restrictions and several resource-duration modes" (1993).