

# Why Midas is a terrible secretary:

How can heuristics make SAT faster for the Preemptive Resource-Constrained project scheduling problem with set up time?

George Hellouin de Menibus<sup>1</sup>

<sup>1</sup>Tu Delft

## Aim and Background

Preemptive Resource-Constrained project scheduling problem with set up time describes a class of problems derived from more generic scheduling. A set of Projects needs to be scheduled. Our objective is to minimize the makespan (The time from the start of the first project till the end of the last project) We have the following challenges:

- **Project Precedence** Some projects can only be scheduled after others are complete
- **Renewable Resource Constraint** All projects draw from a set of renewable resources with limited availability. Projects might be scheduled one after the other even if they have no precedence relation
- **Preemption** Projects can be "preworked", which means a wide amount of combinations needs to be considered.

## Visualising Pre-emption

Here is how one task can be pre-empted:

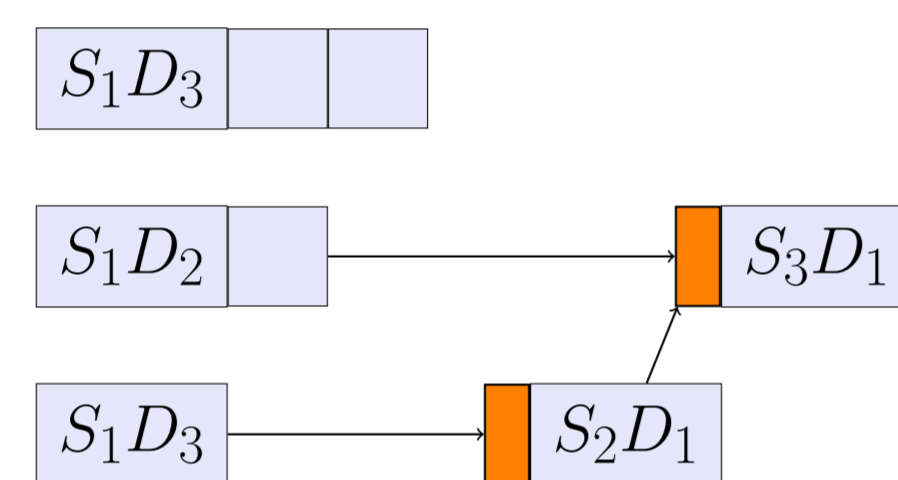


Figure 1. Possible Ways to preempt a Project of duration 3. Orange represents the set up time cost

## Heuristics and Modeling In SAT

### Heuristic

Heuristics are "rules of thumb" that help the solver make decisions. They represented educated guesses based on where we are in the solver. Heuristics are found everywhere from pathfinding, to virus detection. They are domain specific.

### SAT Solvers

Sat Solvers are used to solve any problem that can be expressed in Conjunctive Normal Form. This is a series of AND and OR statements, where Boolean variables represent a part of the problem. SAT solvers do not rely on domain-specific heuristics and usually favour VSIDS for everything.

### Modeling PRCPSP-ST

To model PRCPSP-ST, we use [2], which developed a SAT encoding for the problem. Four hard rules for all variables:

- Completion
- Consistency
- Precedence
- Resource

## Bounding our domain

We used a the horizon (sum of all task durations) to be the latest starting time for all projects. Other works, [2] use critical method path. This change represents a change to our domain.

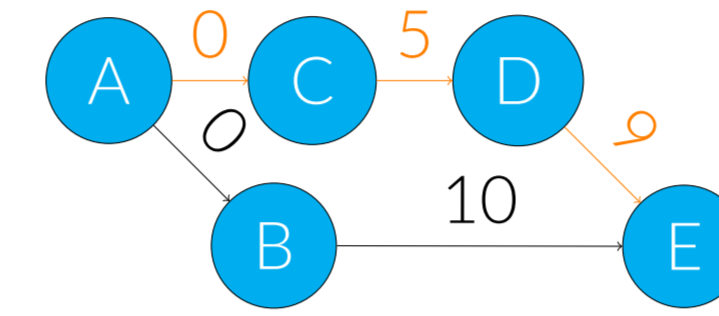


Figure 2. Example of Dependency graph with critical path analyse. The orange line shows the critical path( Length of 14)

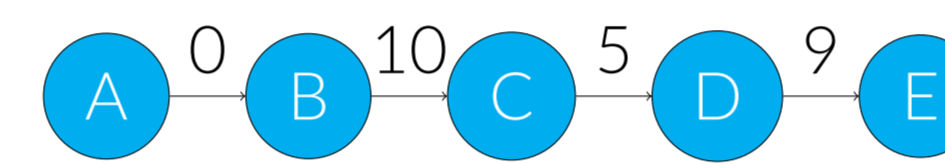


Figure 3. Order of projects if executed sequentially because resource cap is 10

## Heuristics explored

Our Heuristic needs to interface with SAT Solver. We picked a heuristic that would influence variable ordering. The main intuition is:

- Avoid paying set up time
- Consider "best" first
- Rule out uvars as quickly as possible

### Algorithm 1 Heuristic order

```
1: procedure Heuristic ordering( $S, U, Aux$ )           ▷ Lists for each variable type
2:    $S \leftarrow$  Sort by key  $S$                        ▷ Sort by set up time incr, start time inc, duration dec
3:    $vars \leftarrow$  Append  $S U Aux$ 
4:   return  $vars$ 
```

## Results

Method	Avg solve time(s)	# of Timeouts	# of Errors
Greedy	1.07	5	54
VSIDS	0.162	0	54

Table 1. Processed Results of the Runs

Setup #	Greedy		VSIDS	
	Optimal	#UnSat	Optimal	#UnSat
1	40	26	41	26
2	61	23	61	23
3	64	18	65	18
4	67	16	68	16
5	70	13	71	13
6	70	13	71	13

Table 2. Breakdown of Optimal and unsatisfied results

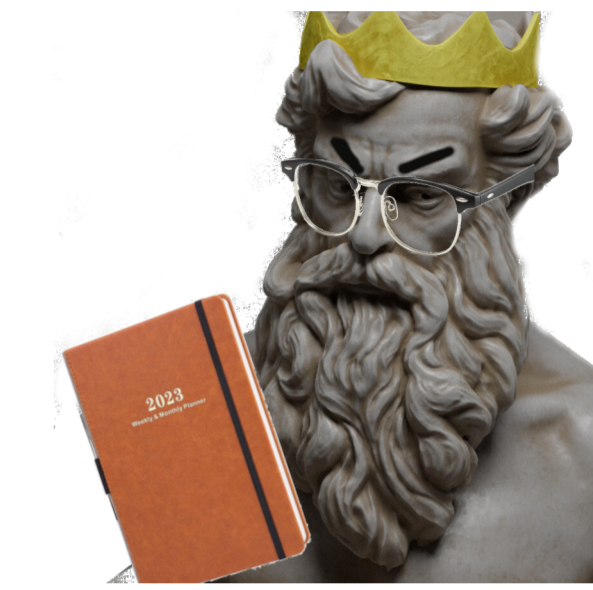


Figure 4. One angry secretary, based on [1]

## Technical Challenges

- Had to mix complex data structures (arena) with foreign function interface
- Attempted multiple approaches to multi threading
- Ran out of working memory consistently
- One problem instance generates a file per set up time value.
- Problem has higher load due to horizon method for upper bound

## Limitations

- **Small Dataset** Difficult to draw definitive take aways because the J12 set only has 12 projects per schedule, very small amount.
- **Limited insight** Not enough collection related to reasons for time difference
- **Still have unsat instances** This points to something being wrong with either dataset or encoder

## Conclusion

Key take aways:

- **Fixed variable encoding is not good** Performs 6x worse
- **Modeling work is incomplete** Both this paper and last year's have unexplained unsatisfied instances
- **PRCPSP-ST grows particularly fast.** The addition of segments, although useful to represent the problem, become incredibly large.

Potential Future work:

- Develop different Heuristics for same problem
- Add Heuristic to different section of sat
- Initialize variables with different starting values
- More insight into potential causes

## References

- [1] Lazar Isailović. Minos, the infernal judge - divine comedy. <https://www.artstation.com/artwork/aYmJdX/>, 2021.
- [2] Jasper Vermeulen. Resource-constraint project scheduling with task preemption and setup times by boolean satisfiability encoding and satisfiability (sat) solver. 2022.