IMPROVED FILTERING OF SCHEDULING PROBLEMS USING REDUNDANT TABLE CONSTRAINTS

ModRef21

Hélène Verhaeghe[1], Roger Kameugne[2], Christophe Lecoutre[3], Pierre Schaus[1]

25 October 2021

[1] ICTEAM, UCLouvain, Place Sainte Barbe 2, 1348 Louvain-la-Neuve, Belgium, $\{firstname.lastname\}@uclouvain.be$
Department of Mathematics and Computer Science, Faculty of Science, University of Maroua, P.O. Box 814, Maroua, Cameroon, $rkameugne@gmail.com$ [3] CRIL-CNRS UMR 8188, Université d'Artois, F-62307 Lens, France, $lecoutre@cril.fr$

UCLouvain

Tasks:

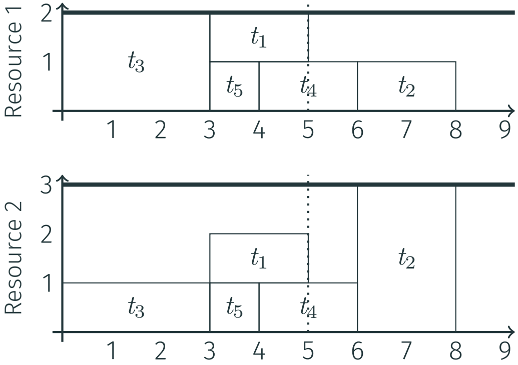
	t_1	t_2	t_3	t_4	t_5
durations	2	2	3	2	1
resource 1 uses	1	1	2	1	1
resource 2 uses	1	3	1	1	1

Precedences:

0	1,2,3,4,5
1	2,6
2	6

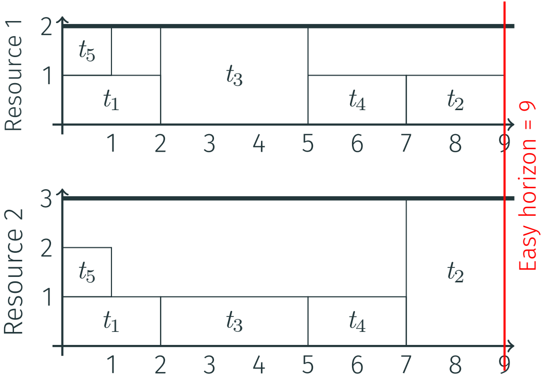
3	4,6
4	6
5	6

Optimal solution:

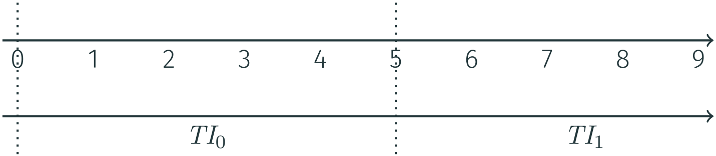


How to improve the solving time of such scheduling problem?

- **Step 1**: Reduce horizon (first easy solution)
- **Step 2**: Generate table of relaxed solutions (branching on abstract variables)
- **Step 3**: Find solution using redundant table (branching on start variables)



Step 1: Reduce horizon (first easy solution)

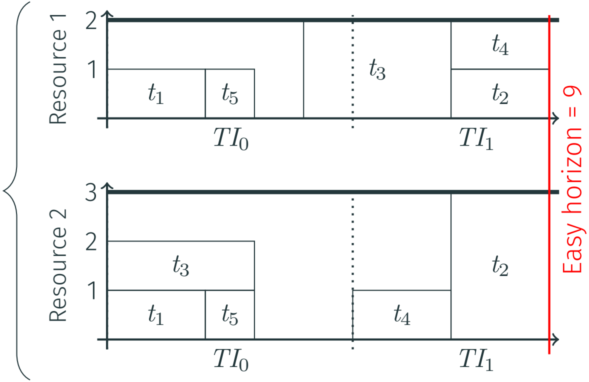


Step 2: Generate table of relaxed solutions (branching on abstract variables)

Step 2a: Create the abstract variable

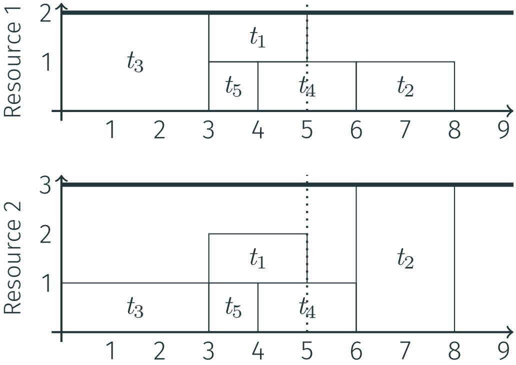
METHODOLOGY

	s^a_0	s^a_1	s^a_2	s^a_3	s^a_4	s^a_5	s^a_6
τ_1	0	0	1	0	0	0	1
τ_2	0	1	1	0	0	0	1
τ_3	0	0	1	0	0	1	1
τ_4	0	1	1	0	0	1	1
τ_5	0	0	0	0	1	0	1
τ_6	0	0	0	0	1	1	1
τ_7	0	0	1	0	1	0	1
τ_8	0	0	1	0	1	1	1
τ_9	0	1	1	0	1	0	1



Step 2: Generate table of relaxed solutions (branching on abstract variables)

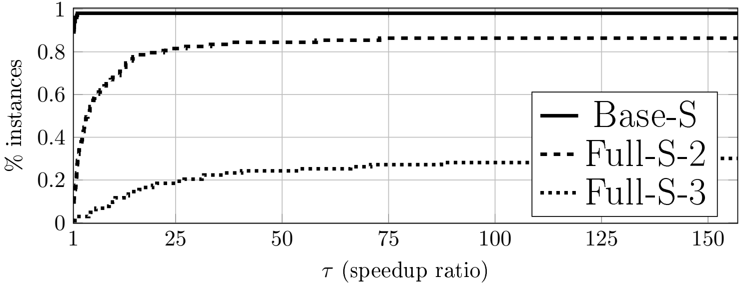
Step 2b: Generate table



Step 3: Find solution using redundant table (branching on start variables)

RESULTS: CASE STUDY 1: FULL TABLE

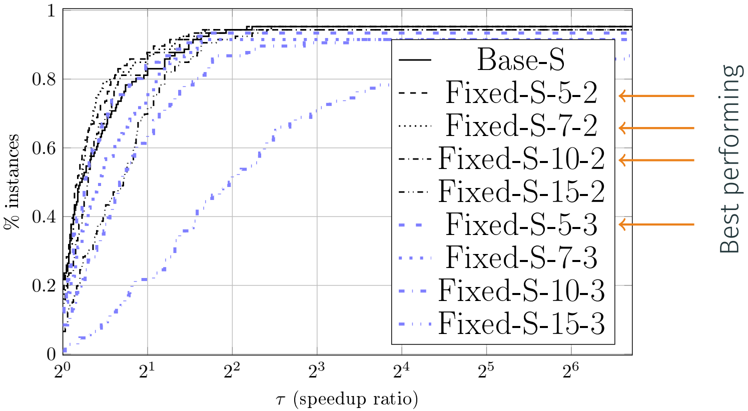
Settings: J30 instances, all the variables have corresponding auxiliary variables, strongest consistency for cumulative, 2 or 3 time intervals



Conclusion: Complete table too big to compute

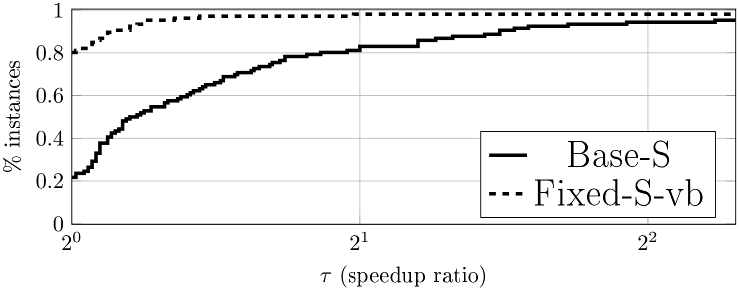
RESULTS: CASE STUDY 2: FIXED TABLE

Settings: J30, 5, 7, 10 or 15 variables have corresponding auxiliary variables (chosen by decreasing energy), strongest consistency for cumulative, 2 or 3 time intervals



Best performing

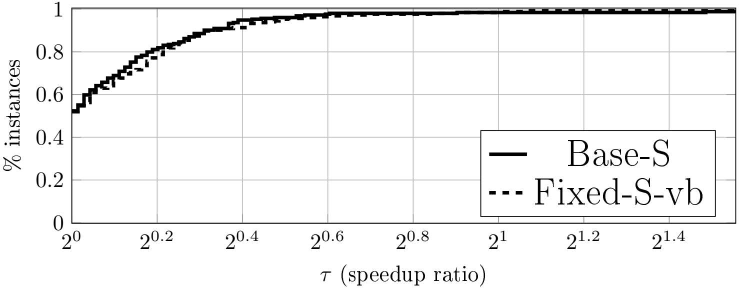
Settings: J30, 5, 7, 10 or 15 variables have corresponding auxiliary variables (chosen by decreasing energy), strongest consistency for cumulative, 2 or 3 time intervals



Conclusion: Successfull, given an oracle to choose the best suited parameters

RESULTS: CASE STUDY 2: FIXED TABLE

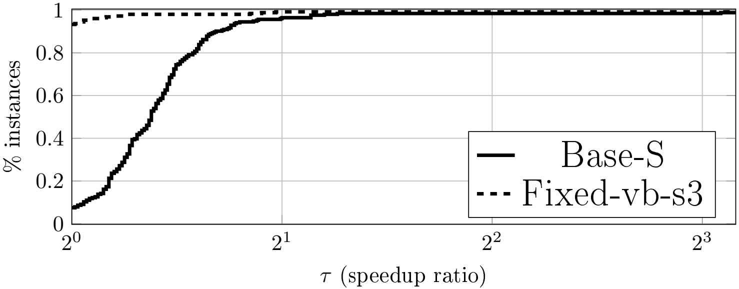
Settings: J60, 5, 10, 15 or 20 variables have corresponding auxiliary variables (chosen by decreasing energy), strongest consistency for cumulative, 2 or 3 time intervals



Conclusion: Complexity of strong propagators may lead to a bigger overhead of the step 2

RESULTS: CASE STUDY 2: FIXED TABLE

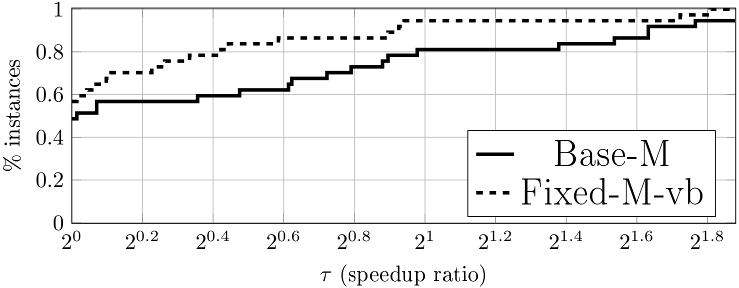
Settings: J60, 5, 10, 15 or 20 variables have corresponding auxiliary variables (chosen by decreasing energy), strongest consistency for cumulative, 2 or 3 time intervals



Conclusion: Confirmation that step 2 has too big an overhead.

RESULTS: CASE STUDY 2: FIXED TABLE

Settings: J60, 5, 10, 15 or 20 variables have corresponding auxiliary variables (chosen by decreasing energy), medium consistency for cumulative, 2 or 3 time intervals



Conclusion: Successfull, given an oracle to choose the best suited parameters

CONCLUSION

Current state:

- Applicable to any scheduling problem
- Solver-dependent
- Good preliminary results

Perspectives:

- Automatic selection of the parameters
- Test the approach with other problems
- Study the pruning effect of the redundant constraint

Thank you for listening!

Any questions?