Using Heuristics To Guide Constraint Solvers

Can using problem-specific branching heuristics improve the performance of constraint solvers for PRCPSP-ST?

Author: Codrin Ogreanu
[C.Ogreanu@student.tudelft.nl]

Responsible professor: Dr. Emir Demirović
Supervisors: Maarten Flippo, Imko Marijnissen

1. Background

The **Resource Constrained Project Scheduling Problem** (RCPSP) is an NP-Hard problem[1]. The goal is to generate the shortest schedules, under resource and precedence constraints.

Preemptive RCPSP with **Setup Times** is an extension, where activities can be **pre-empted** and resumed after a **setup time.**

Constraint Programming (CP) is a powerful method of solving general combinatorial problems. However, the **search space** of PRCPSP-ST is much larger compared to base problem, making it less practical.

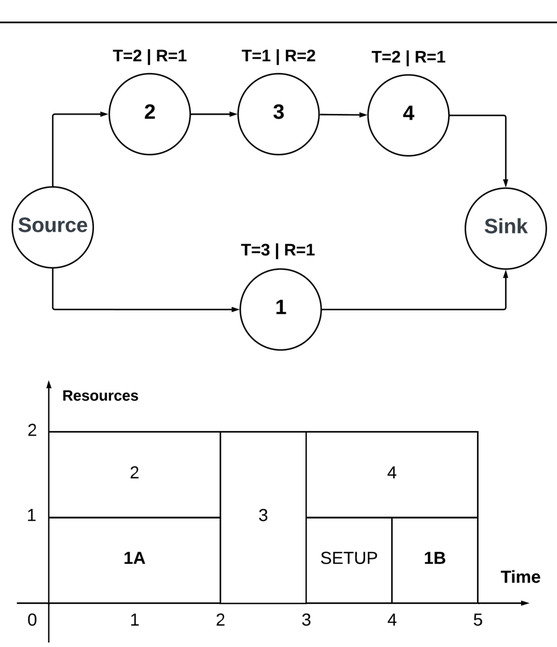

Figure 1: A PRCPSP-ST instance with 4 activities.

2. Research Questions

- Can including domain-specific knowledge improve the search procedures of constraint solvers?
- Can these new heuristics be used to improve existing branching heuristics?

3. Methodology

Two heuristics inspired by existing methods[2] were created:

- **Greatest Resource Demand** (GRD): prioritize scheduling most demanding activities;
- **Dynamic Earliest Starting Time** (dEST): try to schedule activities as early as possible.

By prioritizing demanding activities, only small gaps in the schedule are left, which can be filled by less demanding tasks.

4. Experimental Results

Three algorithms were tested:

- A baseline configuration, **VS/SG**, which used VSIDS[3] and solution-guided Phase Saving[4];
- Our new heuristics, **GRD/dEST**;
- A combination of VSIDS and dEST, **VS/dEST**.

Results can be seen in Table 1 and Figure 2.

VS/dEST had a **10-fold speedup** over the baseline, and a **47-fold decrease** in decisions made by the solver.

While **VS/SG** found fewer solutions, they were **better schedules**, especially for large datasets.

5. Discussion and Conclusion

- The speedup can partly be attributed to the baseline initially considering all time slots equally.
- Using **VSIDS with dEST** resulted in the **best performance for most metrics**.
- Heuristic strategies may be more suitable for applications with short solver timeouts.

Table 1: The measured metrics.

Dataset	Method	Metrics						
		%DEV	%RU	#Imp	#DC	T	#PRE	#Ins
J30	VS/SG	0.65	48.81	107	161050	23.87	12	404
	GRD/dEST	0.54	44.52	79	30676	14.10	2	366
	VS/dEST	0.64	46.53	106	11685	6.42	2	407
J60	VS/SG	0.31	48.40	38	618776	156.44	28	342
	GRD/dEST	0.28	44.16	31	26202	13.09	5	333
	VS/dEST	0.30	46.24	39	6279	6.61	7	362
J90	VS/SG	0.11	46.47	10	890968	286.16	42	207
	GRD/dEST	0.09	43.28	13	17385	16.54	8	313
	VS/dEST	0.12	46.52	21	8079	13.06	13	361
J120	VS/SG	**0.40**	56.66	4	975005	366.55	65	40
	GRD/dEST	**0.06**	51.43	2	65982	50.99	22	113
	VS/dEST	**0.18**	57.42	5	22998	45.07	35	176
All	VS/SG	0.42	48.49	159	**503642**	**138.01**	26	993
	GRD/dEST	0.31	44.76	125	**29200**	**18.19**	6	1125
	VS/dEST	0.36	47.91	171	**10714**	**13.52**	11	1306

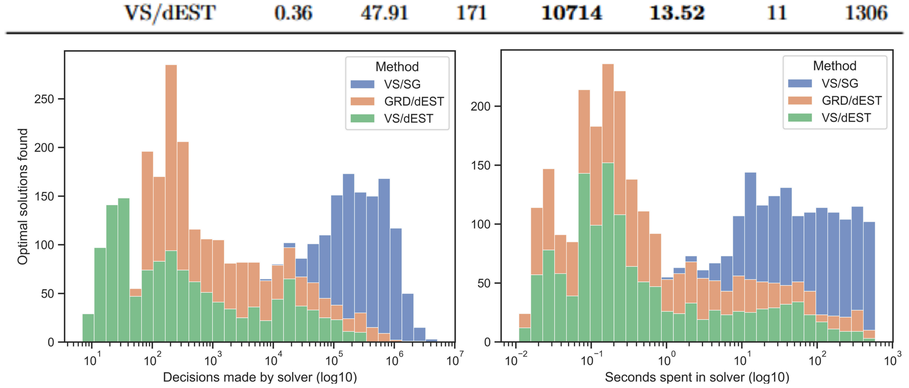

Figure 2: The progress of the different methods, by Decision Count and Time.

6. Future Work

- Longer timeouts may return different results, especially for **VS/SG**;
- Different types of setup times can be tried by scaling activity durations.

References

[1]: Blazewicz, J., J. K. Lenstra, and A. H. G. Rinnooy Kan. 'Scheduling Subject to Resource Constraints: Classification and Complexity'. Discrete Applied Mathematics 5, no. 1 (1 January 1983): 11–24. https://doi.org/10.1016/0166-218X(83)90012-4.

[2]: Klein, Robert. 'Bidirectional Planning: Improving Priority Rule-Based Heuristics for Scheduling Resource-Constrained Projects'. European Journal of Operational Research 127, no. 3 (16 December 2000): 619–38. https://doi.org/10.1016/S0377-2217(99)00347-1.

[3]: Moskewicz, Matthew W., Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. 'Chaff: Engineering an Efficient SAT Solver'. In Proceedings of the 38th Annual Design Automation Conference, 530–35. DAC '01. New York, NY, USA: Association for Computing Machinery, 2001. https://doi.org/10.1145/378239.379017.

[4]: Demirovic, Emir, Geoffrey Chu, and Peter J. Stuckey. 'Solution-Based Phase Saving for CP: A Value-Selection Heuristic to Simulate Local Search Behavior in Complete Solvers'. In Principles and Practice of Constraint Programming, edited by John Hooker, 11008:99–108. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018. https://doi.org/10.1007/978-3-319-98334-9_7.