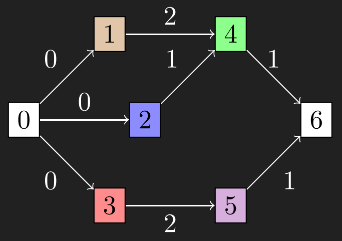
Comparing schedule generation of VSIDS against CPRU for RCPSP-t solvers

Wouter Breedveld
(W.J.Breeveld-1@student.tudelft.nl)

1 Background

RCPSP is a class of problems where some tasks, which require resources, must be scheduled. **RCPSP-t** is a subproblem of RCPSP, where the resources produced and taken vary over time. The goal is to minimize the time from start to end **[1]**. **CPRU** is a heuristic designed for RCPSP-t and uses the critical path and resource utilization.

Precedence graph with durations:



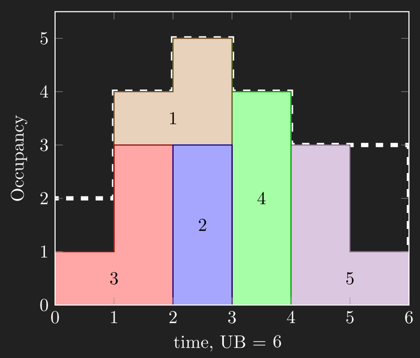
Resource capacities:

$B_{1,0}$	$B_{1,1}$	$B_{1,2}$	$B_{1,3}$	$B_{1,4}$	$B_{1,5}$
2	4	5	4	3	3

Activity durations and demands:

	0	1	2	3	4	5	6
p_i	0	2	1	2	1	2	0
$b_{i,1,0}$	-	1	3	1	4	3	-
$b_{i,1,1}$	-	2	-	3	-	1	-
$b_{i,1,2}$	-	-	-	-	-	-	-

An optimal solution:



2 Research Question

- **How does selection randomization for CPRU compare in runtime and average deviation from optimal to a version without?**
- **Can CPRU be improved by using an adaptive heuristic calculation?**
- **How does CPRU compare against meta-heuristics like VSIDS[2]?**

3 Methodology

RCPSP-t can be solved using **constraint programming (CP)**. In CP, variables which may take a range of values and constraints which enforce relations are used. A solver assigns each variable a value such that the constraints hold.

CPRU calculates the critical path by finding the longest path between the task and the end on the precedence graph. The resource utilization is the amount of resource a task and its successors take, relative to the time they could be scheduled in. **CPRU** only looks at a subset of the variables when selecting a variable.

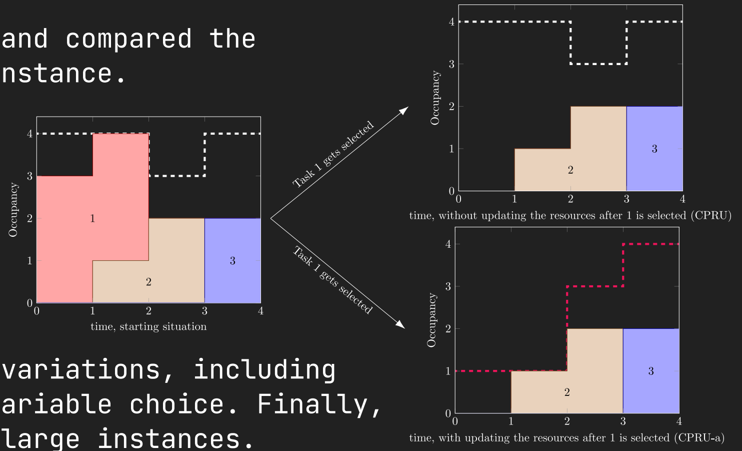
4 Experiments & Conclusion

We evaluated 2 different variations of CPRU: CPRU without randomization and CPRU with the resource utilization updating as shown in the figure below (CPRU-a). We tested the variations and CPRU against VSIDS running them for 60 seconds. We also tested them running for a maximum of 10 schedules within 9 hours.

We used the testing set by Hartmann **[1]** and compared the result against the lower bound on each instance.

Test results:

algorithm\test set	60-sec J30	60-sec J120	10-schedules J30
CPRU	44%	52%	43%
CPRU-nt	44%	52%	43%
CPRU-a	44%	52%	43%
VSIDS	44%	117%	92%



We concluded there was no significant performance difference between the CPRU variations, including showing CPRU-a had little influence on variable choice. Finally, we concluded CPRU outperforms VSIDS for large instances.

References

[1]: S. Hartmann, Project scheduling with resource capacities and requests varying with time: a case study. Flexible Services and Manufacturing Journal, 25(1-2):74-93, 06 2013

[2]: M.W. Moskewicz et al. Chaff: engineering an efficient sat solver. In Proceedings of the 38th Annual Design Automation Conference, DAC ’01, pages 530-535, New York, NY, USA, 2001. Association for Computing Machinery.

Supervisors: Imko Marijnissen
Emir Demirović
Maarten Flippo

TUDelft