Decision Making - ex 4

Filippo Brajucha, Youssef Hanna

November 2023

1 RCPSP - Table

	default		**smallest**	
	Makespan	**Time**	**Makespan**	**Time**
rcpspData1.dzn	90	265ms	90	259ms
rcpspData2.dzn	53	857ms	- 54	-
rcpspData3.dzn	- 82	-	- 75	-

2 JSP - Table

	default		**smallest**	
	Makespan	**Time**	**Makespan**	**Time**
jobshop1.dzn	664	823ms	- 670	-
jobshop2.dzn	- 882	-	- 922	-

3 Observations

What do you observe? Is searching on the smallest (earliest) start times is always a good idea?

The results for the two exercises are similar, we can observe that the default method is better for the resolution of the problem.
For the **RCPSP** model the execution timed out for `rcpspData3.dzn` with both methods. For `rcpspData2.dzn` only with *smaller* heuristic the execution fails, we can also observe that the solver is near the best solution for makespan but the time taken is very different, the default solver is very fast. Observing the `rcpspData3.dzn` it's clear that the *default* solver is slower than the *smallest* one.
Looking at **JSP** model we can observe that there's a failure on `jobshop2.dzn` for both solver heuristics and the *default* method is always faster than the *smaller* one.
It's clear that the use of the *smallest* solving heuristics isn't a good idea for this exercise, in fact, searching for the earliest start time (EST) is the *greedy solution*, this is just linked to the problem's nature, using the earliest start time isn't good idea because of the propagation. The EST strategy has a propagation problem, when the solver finds a solution it's very difficult to change the order of the works because the solver can't postpone easily the executions, when it finds a low data for the EST it tries to execute the job.