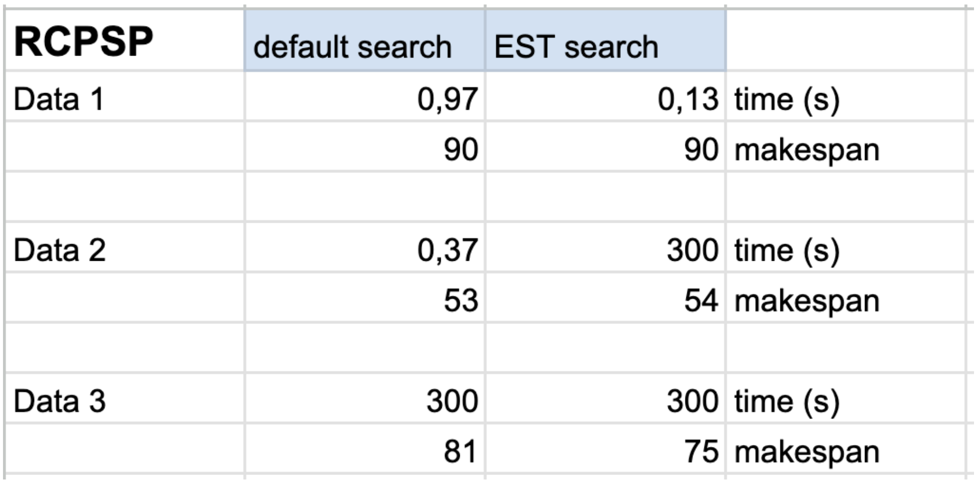
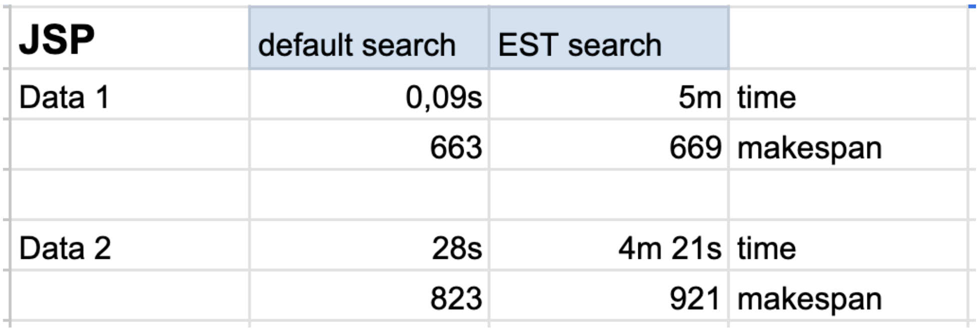
Team:

- Andrea Corradetti - andrea.corradetti2@studio.unibo.it
- Francesco Corigliano - francesco.coriglian2@studio.unibo.it

Report



RCPSP	default search	EST search	
Data 1	0,97	0,13	time (s)
	90	90	makespan
Data 2	0,37	300	time (s)
	53	54	makespan
Data 3	300	300	time (s)
	81	75	makespan



JSP	default search	EST search	
Data 1	0,09s	5m	time
	663	669	makespan
Data 2	28s	4m 21s	time
	823	921	makespan

Note: when time is 300s / 5m, the solver timed out without proving optimality.

- What do you observe? Is searching on the smallest (earliest) start times always a good idea? Justify your answer.

EST search does not seem to guarantee improved performance.
A greedy search like choosing according to EST can give us good solutions in many cases, but does not guarantee optimality.

After finding one solution, the solver poses a new constraint on the maximum value for the objective function and backtracks as needed. In the case of chronological backtracking, this will be to the first node.
Unfortunately on choosing new values, the solver will only be able to pose weak constraints on the variable and exclude previously tried values (eg. $X_0 \neq 0$). This does not reduce the search space significantly.
The time to prove optimality probably depends on the shape of the search tree which is determined by the interaction between the starting parameters and the search strategy.
For some parameters, the default search seems to result in a more convenient search tree.

Still you will see that searching on the earliest start times will be bad most of the times. Please justify the reason by studying the part where we motivated the schedule or postpone search strategy (yes $X0 \neq 0$ will not be strong but your description after here is vague. Why this way of searching delays finding the optimal solution?)

Because the cost function of this problem is regular, given a schedule (task ordering), assigning starting values other than EST to the tasks cannot improve the objective value. To see improvements we must try different task orderings, but an EST strategy will require many failures before the tasks are sufficiently pushed ahead to reshuffle the order. Indeed, with inDomainMin as value heuristic, values from a domain will be eliminated one at a time starting from the smallest and the tasks will slowly be pushed forward. Many of these assignments will be fruitless because of the nature of the cost function.