Algorithmics	Student information	Date	Number of session
	UO: 294665	18/04/2024	7
	Surname: Garcia Castro		
	Name: Gonzalo		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. [The Numerical Square]

What I have done in the heuristic is to go further to the board that have more nodes initialized (the basic example that I have learn from the e-campus). I attempted to make an heuristic that when, for example, the value is 0 all the nodes must be below 5 or make an heuristic that if the number to reach is low (I have left the previous approximation commented at the bottom of the NumericSquareBaB class), all the number in the column must be low too but I have had a problem with the multiplication and division and I have been trying to modified the code so that it works but I have not reached an appropriate solution. For this second approach I have left the code of the two functions (one for the row and another for the column) in the HeuristicState class.

Test case	Time for first solution (backtracking)	Number of developed nodes (backtracking)	Time for first solution (branch and bound)	Number of developed nodes (branch and bound)
Test00	LoR	4	LoR	9
Test01	LoR	287	LoR	129
Test02	LoR	166	LoR	36
Test03	LoR(27 ms)	69240	0,215 ms	295657
Test04	LoR(38 ms)	79341	7,5 s	14523155
Test05	0,05 s	224207	LoR	112
Test06	LoR	3204	LoR	298
Test07	5,5 s	35152380	LoR	1485

The code of my branch and bound is similar to the one of backtracking, but instead of doing recursive calls, we save in a queue order by the heuristic value, all the possible nodes containing a solution, so we go with all the nodes in an order that might get a solution before than using backtracking, as they are ordered by the probability of obtaining a solution.

As we can see, branch and bound works better than backtracking but for reaching all the possibilities obviously is much better backtracking as branch and bound is impossible (or very unworthy) to implement for all the solutions.

I think this happens because with branch and bound we came closer and closer to one solution without taking in account other possibilities just focusing on the solution, meaning prune the leaves that are far away for the solution.