Alfonso Esposito 1057171
Isabella Marasco 1040993

Decision Making with Constraint Programming: Exercises 2

To Do - 1.

Objective: Starting from N-Queens exercises observe how the variation of the constraints changes the fails and time for solutions.

Comments: We wrote 2 different models with several constraints for the problem, the global constrain and the decomposition. The first type of constraint is GAC (Generalized Arc Consistency) while the second is AC (Arc Consistency).

In this case, GAC on the original constraint is stronger than AC on the constraints in the decomposition, this why AC must achieve the max before backtracking and then it must consider more attempt than the GAC, and from that derives more failures and consequently more time.

N	**Alldifferent GC**		**Decomposition**	
	Fails	**Time**	**Fails**	**Time**
28	78.847	2s29msec	417.027	7s 589msec
29	31.294	1s 88msec	212.257	4s 319msec
30	1.588.827	42s 351msec	7.472.978	2m 14s

.

To Do - 2.

Objective: Starting from puzzle exercises globalize the problem constraints and observe how the variation of constraints changes the fails and time for solutions.

Comment: We wrote 4 different models with several constraints for the problem: base, base + implied constraints, global and global + implied constraint.

We observed that for both, the basic and global models, upon adding the implicit constraints there is an improvement on both the number of failures and the time taken. This happen because implied constraints introduce constraints that cannot be deduced by the solver, and this reduce the search space making the solver faster.

Furthermore, we observed that the models with global constraints and global + implied get results in less time than the other two models. This happens because the global constraints have higher propagation, namely, it achieves a certain level of consistency on a constraint by removing the inconsistent values from the domains of the variables.

Lastly, we noted that the number of failures for the base model is lower than using global. This could be because the first makes fewer attempts but takes longer to figure out what the solution is while, the second makes more attempts taking much less time because of propagation.

N	**Base**		**Base + implied**		**Global**		**Global + implied**	
	Fails	**Time**	**Fails**	**Time**	**Fails**	**Time**	**Fails**	**Time**
500	617	39s 611msec	495	26s 218msec	989	837msec	493	530msec
1000	1.247	4m 6s	995	2m 3s	1.989	1s 818msec	993	1s 27msec

To Do - 3.

Objective: Resolve a poster problem and observe how the variation of the model changes the search for solutions.

Comment: Observing the results, we can see that the naïve model has obtained a greater number of failures than the model with global. In fact, the use of global constraint has allowed the solver, through propagation algorithms, to eliminate from the search space all those variables that did not lead to a correct solution, consequently the number of attempts and the failures decreases with the total time.

In the naive solution, since forall is present, the solver is forced to cycle through all the possible results, making a considerably greater number of attempts, which therefore lead to a higher number of failures and therefore a longer time spent.

The substantial difference is the presence of propagation, which allows to obtain better results by decreasing the search space.

Instances	**Naïve Model**		**Global Model**	
	Fails	**Time**	**Fails**	**Time**
19x19	1.678.013	18s 866msec	300.649	2s 900msec
20x20	2.504.120	29s 282msec	2.030	270msec