CP First Assignment

Students: Massimo Rondelli e Gianmarco Gabrielli

1. What is happening when going r → rc1 → alldiff ? Why?
We find out that going from *r* to *rc1* and to *alldiff* the number of failuers decrease since the constraints are more tight. Using a combined model like rc1 is better than the row model because it has benefits like the facilitation of the expression of constraints, the enhanced constraint propagation and more option for search variables. Combined model needs a channeling constraints to maintain consistency between the variabiles of the two models. The potential advantage of combining viewpoints in this way comes from propagating the constraints of the two models during the search for a solution. It means that if we have two viewpoints, V_1 and V_2, a complete model can be created from each viewpoint, M_1 and M_2. It is possible to combine the two models with variables $X_1 \cup X_2$ and constraints $C_1 \cup C_2 \cup C_c$, where C_c is a set of channeling constriants. As we said before, while search proceeds, propagating the constraints C_1 removes values from the domains of the variables in X_1. The channeling constriants now may allow values to be removed from the domains of the variables in X_2. The result that more values are removed within viewpoint V_1 than by the constraints C_1 alone, and viceversa for the viewpoint V_2.

Regarding *allDiff*, it is better than the other models because, during the process, a single solver is attached to each constraints of the model itself, in order to identify and remove domain values that never appear in any solution to the constraint in the remaining search space. Doing it, the number of combinations is reduced and the search space became smaller, which means the number of failures will be minor.

2. What is happening when going rc1 → rc2 → rc3 ? Why?
As we said in the answer above, the number of constraints is important to reduce the search space, which means it can reduce the number of failures. Going from *rc1* to *rc2* and to *rc3*, we can see that again. From *rc1* to *rc2* the global constraints alldifferent are removed since they are redundant with the other constraints (forall and the channeling constraint). Redundant constraints can be removed from the model since they only add an unnecessary overhead. When we pass from *rc1* to *rc3* we reduce the number of constriants but the search space size become bigger. Doing that, the number of failures increse since we're allowing more flexibility.

3. What is happening when going alldiff → alldiffsym? Why?
When we pass from alldiff to alldiffsym we use symmetry breaking combined to global constraint because symmetry breaking constraints are usually considered separately to other (global) constraints in a problem. However, the interaction between problem and symmetry breaking constraints can often have a significant impact on search. For instance, the interaction between problem and symmetry breaking constraints gives an exponential reduction in the search required to solve certain problems: The use of lexicographic simmetry constraints can also refine the solution space, order the solution space in a specific way. The add of alldiffsym improve the efficency of the solver reducing the dimension of the search space and delete the solution that are simmetrically equivalent.

4. What is happening when going base → base + implied ? Why?
Going from *base* to *base + implied* the execution time and the number of failuers decresead since we have more constraints. We are looking for just one solution in this problem. In *base + implied* we have more constraints than the *base* one. Having more constraints take the solver to discard some possible solution that were achiavable with the *base* problem.

In the *base* model we only have the meta constraint which is going to solve the problem task. The implied constriants added do not change the set of solutions. They are logically redundant. The aim in adding implied constraints is to reduce the search effort to solve the problem.