CP Second Assignment

Students: Massimo Rondelli e Gianmarco Gabrielli

Poster Placement problem
How the solver performance changed from the decomposed model to the global model?

The main difference between the naive and the global model in the **poster placement problem** is how we treat the no overlap constraint. As global constraint to achieve this goal we decided to use *diffn.* The geometrical packing constrians diffn arranges a given set of multidimensional boxes in n-space such that they do not overlap. This approach, as we can see from the table, speed up the computation.

So, we can say that global constraints are more powerful and efficient than simple constraints (used in the naive model) because they provide a higher level of abstraction and can capture complex relationships among variables more effectively. Global constraints are normally implemented with optimized algorithms, making them more efficient in terms of constraint propagation and search space reduction. They can reduce the search space faster than decomposed simple constraints, so that's why they take less time to find a solution.

The constraint can also provide pruning techniques that eliminate invalid assigment early in the search, reducing accordingly the size of the search space and can incorporate simmetry-breaking mechanisms which help to avoid symmetrical solutions that would inflate the search space. In our cases in particular, the diffn use implicitly the constructive disjunction. The idea is to try to each alternative of a disjunction and remove values that were pruned in all alternatives. That lead us to reduce the search space of our problem.

Talking about the **nQueens problem** we have seen the same behavior. Looking the results we obtained, it's clear to see that the model with global constraints is better than the decomposition. Using the global constraints, the search space is reduced and the number of failures drastically decrease.

The Sequence Puzzle problem
Going from Base $\rightarrow$ Global, and from Base + Implied $\rightarrow$ Global + Implied: what is the main advantage of using a global constraint? Why?

Looking at the model's statistics, we can see that the variables used in the base and global constraints models are quite different. Base models use 500500 variables and global models only 500. (those numbers are true if when n = 500). The number of propagation is drastically different as well, but it has a meaning since it is proportional to the variable's number. The cost of propagation should be approximately $O(n^2)$ due the meta constraint

constraint forall(i in 0..n-1)(x[i] = sum(j in 0..n-1)(x[j] = i));

since for each i in the range $0..n-1$, $x[i]$ is the sum of all the j if $x_j = i$. In the end, we can say simple constraints are unefficient compared to the global constraints. Using global constraints, in this case, help us to avoid a computation cost of $O(n^2)$.

Is there an implied constraint that now becomes redundant in the Global + Implied model? Why?

We can see that the first implied constraint is redundant for the model since it is already insiede the gcc global constraint. We define $gcc(X, v, O)$, the constraint, which searches for an assignment of each variable of X such that each value $y_j \in D_X$ must be taken at least v_j times and at most O_j times. In this way we can observe that the sum of occurences will be exactly equal to n, which is the size of X.

Using this approach help us to avoid the use of the first implied constriant since it's goal is literally check if the sum of occurences is equal to n. In this case the use of the implied constraint is useless. It does not give us computationl advantage (which it should be it's purpose).