N-QUEENS PROBLEM

n	Alldifferent GC		Decomposition	
	Fails	Time	Fails	Time
28	78,847	1.64s	417,027	3.78s
29	31,294	509ms	212,257	1.727s
30	1,588,827	19.173s	7,472,978	55.6s

The table above shows the number of failures and the time spent by each model in the search for a solution. We can clearly see that the model that uses global constraints is more efficient both in terms of failures and time required. This should be because global constraints have optimized search algorithms: with global constraints it's known immediately if a constraint will fail on an assignment because of the algortihm used that guarantees GAC or BC properties.

POSTER PLACEMENT PROBLEM

Istance	Naive Model		Global Model	
	Fails	Time	Fails	Time
19x19	1,678,013	11.781s	300,649	1.789s
20x20	2,504,520	17.978s	2030	100ms

Q: What are the variables and the domains?

A: The variables are represented as arrays that identify the horizontal and vertical positions of our 'n' posters, while the domain is represented by the area available for placing the posters in a non-overlapping manner.

Q: What are the constraints?

A: The constraints we have defined fall into two categories:

- *Constraints that prevent posters from exceeding the domain:* To achieve this, we have two similar constraints (one for height and one for width). Let's focus on the width constraint without loss of generality (the height constraint works similarly). This constraint ensures that the sum of the widths of the already placed posters, along with the width of the poster under consideration, does not exceed the maximum width.

- *Constraints that prevent poster overlap during placement:* To achieve this, we used two different constraints for the naive model and the global model.

In the naïve model, given a poster *A* for each other poster *B* (where *B* is a poster that have been already placed on the wall), we check that the starting coordinates (x_A, y_A) summed with the width (w_A) and height (h_A) of the poster *A* ($x_A + w_A$, $y_A + h_A$) are not greater than the initial coordinates of the poster *B* (x_B, y_B).

In the global model, we relied on the global constraint 'diffn(x, y, dx, dy)' which works as follows *"It constrains rectangles i, given by their origins (x[i], y[i]) and sizes (dx[i], dy[i]), to be non-overlapping."*

Q: How does solver performance change from the naive model to the global model?

A: Solver performance varies because, in the global model, we have a specific global constraint for our problem that optimizes operations compared to the naive model with unoptimized sub-constraints. This naturally leads to an increase in computational complexity in the naive model, resulting in longer execution times and potential failures, as seen in the table. This is because global constraints reduce the number of **partial solutions**.

THE SEQUENCE PUZZLE PROBLEM

n	Base		Base+Implied		Global		Global+Implied	
	Fails	Time	Fails	Time	Fails	Time	Fails	Time
500	618	23.18s	496	16.35s	989	270ms	493	198ms
1000	Too much time		996	1.8m	1989	829ms	993	349ms

Q: Going from Base → Global and going from Base + Implied → Global + Implied: what is the main advantage of using a global constraint? Why?

A: As we can easily deduce from the table, what significantly changes when transitioning from a basic implementation of the model to the global one is the execution time. In both cases (Base, Base+Implied), the shift to the global implementation notably reduces execution times for the models, often to a significant degree. This substantial variation in execution times is due to the fact that global constraints are optimized for specific operations. This specialization reduces the problem's search space and maximizes the utilization of the system's computational resources.

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

A: We have observed that, when one removes the first implied constraint from the global+implied model, the number of failures of the search doesn't change at all as well as the execution time. We think this must be due to the fact that every solution found by the global_cardinality constraint, are built respecting the first implied constraint. We observed that this does not works for the base+implied model which requires the first implied constraint to notice invalid assignments immediately. For this reason, in the basic model (the one without implied constraints) we see some improvement.