Decision Making - ex 1

Filippo Brajucha, Youssef Hanna

October 2023

1 nQueens

1.1 Tables

1.1.1 R, RC1, RC2, RC3, AllDiff

n	#sols	R	RC1	RC2	RC3	AllDiff
8	92	891	500	593	864	254
9	352	4262	2656	2772	4603	849
10	724	23291	13996	13593	23195	3722
12	14200	773550	355041	380595	820127	75823

1.1.2 AllDiffSym

n	#sols	AllDiffSym
8	12	78
9	46	296
10	92	965
12	1787	16343

1.2 Q & A

1.2.1 Questions

1. What is happening when going r $\rightarrow$ rc1 $\rightarrow$ alldiff ? Why?
2. What is happening when going rc1 $\rightarrow$ rc2 $\rightarrow$ rc3 ? Why?
3. What is happening when going alldiff $\rightarrow$ alldiffsym? Why?

1.2.2 Answers

1. The main differences between "r", "rc1" and "alldiff" are the number of constraints used. More global constraints are gradually introduced (going from 1 to 2 and then 3). As a result, the search space decreases because the problem is better specified to the solver. You can observe a reduction in failures, which decreases from 891 to 500 and then to 254. The constraints used can also be described as "global" because the number of solutions remains the same, but the new logical connections in the problem domain help the solver to reach solutions more easily.
   The introduction of the intensional constraints (those preventing diagonal attacks) reduces the search space, making it easier to reach solutions. The third model ("alldiff") remains the best because it only uses global constraints, which, as per their definition, are much more efficient than the simply ones. They are semantically redundant, but they also help the solver in problem formulation, making the structure of the problem more evident and complete. Adding global constraints is so important for the resolution of the problem; even though they are not as specific as ad-hoc simply constraints, they resolve the main problems of the simpler constraints that are lack of expressiveness and domain reduction weakness. Applying global constraints is a bit like adding multiple constraints with a single command.

2. The use of models "rc1," "rc2," and "rc3" demonstrates how, by removing global constraints, the search space of the problem increases, and therefore, failures also increase. The solver is no longer aware of the perfect structure of the problem, and as a result, failures increase gradually, by removing only the global constraints, there is an increase in failures ranging from +19% for n=8 to +4% for n=9 (except for n=10, where there is a decrease in failures by 3%). However, when removing the intensional constraint (from rc2 to rc3), there is a significantly higher increase, with a minimum of +46% for n=8 and a maximum of +115% for n=12.
   Moving between the 3 models (rc1 $\rightarrow$ rc2 $\rightarrow$ rc3) we can see that initially the global constraints are removed, followed by the removal of an intensional constraint. Despite these various removals, the number of solutions remains the same because the constraints were semantically redundant. However, the situation for the solver deteriorates as it now has to solve the same problem with less information. In fact, it reaches conclusions more slowly and commits more failures. By removing the constraints, the efficiency of the filtering algorithm decrease, so it has to process more data to find solutions. Essentially, the search space is expanded.

3. Adding symmetry to an already highly precise and structured resolution as the "alldiff" model is, the result are better in speed with fewer failures, but also significantly fewer solutions. Redundant permutations are excluded a priori because the search space has been ordered lexicographically. Considering the resolution method without symmetry, the number

of solutions decreases by approximately 87% for each value of n. Meanwhile, the number of failures gradually decreases from around -70% for n=8 to -80% for n=12.

2 Sequence

2.1 Table

n	Base		Base + Implied	
	Fails	**Time**	**Fails**	**Time**
500	617	18s 711ms	495	12s 321ms
1000	1247	1m 57s	995	54s 793ms

2.1.1 Question

What is happening when going base → base+implied ? Why?

2.1.2 Answer

The solver has a greater understanding of the problem's structure, allowing it to narrow down the search space and solve it with fewer failures. The implied constraints added later enable the solver to logically connect some data in the problem that it wouldn't otherwise be able to connect. Thanks to these new insights, the execution time drops significantly, and the number of failures also decrease.
Thanks to the implied constraints, the search space is reduced, and some variables are excluded from the set of solutions.