Decision Making - ex 3

Filippo Brajucha, Youssef Hanna

October 2023

1 nQueens

1.1 Table

	30	**35**	**45**	**50**
input order				
min value	**1.588.827**	-	-	-
random value	9	10	**6**	42
min domain size				
min value	15	21	**6**	123
random value	1	**0**	1	10
domWdeg				
min value	15	21	**6**	123
random value	1	**0**	1	10

1.2 Comment

https://www.minizinc.org/doc-2.7.6/en/mzn_search.html
We can observe that the number of failures with the "input_order - min_value" is extremely high compared to other ones. It is in the order of the 10^6, other datas are in the order of 10^2; this is the only case where the system cannot perform the research for `n = 35, 45, 50`.
Generally we can observe that the random model is always better in terms of failures for this problem.

2 Poster Placement

2.1 Table

	19x19		**20x20**	
	Fails	Time	Fails	Time
input_order				
min value	**1.315.598**	**11s 35ms**	26.063.823	3m 12s
random value	-	-	-	-
min domain size				
min value	239.954	1s 796ms	**1.873**	**244ms**
random value	2.929.153	19s 172ms	5.797.312	35s 987ms
domWdeg				
min value	236.024	1s 820ms	**1.873**	**244ms**
random value	2.929.030	19s 30ms	5.797.456	35s 957ms

2.2 Table with sorted rectangles

	19x19		**20x20**	
	Fails	Time	Fails	Time
min value	-	-	7.151.303	56s 120ms
random value	-	-	-	-

2.3 Comment

3 Quasigroup

3.1 Table

		default	**domWdeg** - random	**domWdeg** + Luby
qc30-03	Fails	234.522	234.522	234.522
	Time	27s 502ms	**15s 569ms**	19s 109ms
qc30-05	Fails	36.866	36.866	36.866
	Time	3s 121ms	2s 909ms	**2s 820ms**
qc30-08	Fails	324	324	324
	Time	**373ms**	394ms	583ms
qc30-12	Fails	470	470	470
	Time	409ms	399ms	**396ms**
qc30-19	Fails	2.192	2.192	2.192
	Time	513ms	**500ms**	574ms

3.2 Comment

4 Questions

1. When are random decisions (not) useful? Why?
2. Are dynamic heuristics always better than static heuristics? Why?
3. Is programming search and/or restarting always a good idea? Why?

4.1 Answers

1. The outputted results are very different, concerning nQueens problem, the random solution are always the best, in the Poster Placemente problem the random solutions are always the worse and in the Quasigroup problem there's alternation of different results.
   We can be sure that the worst situation is the Poster Placement. This problem has different roots, we think that the main reason is mathemathical, in fact if we sort the rectangles by perimeter in crescent mode the program can't compile.
2.
3.