CSE 318

Assignment on CSP

Submitted By:

Name: Ahmed Mahir Sultan Rumi

ID: 1805015

Department: CSE

Section: A1

Date of Submission: **08.01.2023**

Value Order Heuristic:

I used Least-constraining value heuristics in this offline. In this heuristic, we choose a value that eliminates smallest number of values from the domain of the variables related to the current variable by constraints.

Therefore, in this problem, for every value in the domain for our current variable, we calculate the number of cells in that row and column which contains that value in their domain. If we use that value, domain size will be reduced for all those variables by 1. Therefore, we choose the value for which this count is the smallest.

The reason we use the heuristic is that we want to reach our solution as fast as possible and want to keep the probability of the future cells lower in the tree finding legal values higher. So, we want to keep their domain as large as possible.

Table:

Problem	Solver	VAH	#Node	#BT	#Runtime (ms)
d-10-01	BT	VAH1	62	1	3
	BT	VAH2	199514851	61578657	7980598
	BT	VAH3	58	0	1
	BT	VAH4	62	1	2
	BT	VAH5	24261626	7470902	963476
	FC	VAH1	61	1	1
	FC	VAH2	2122340	461014	98759
	FC	VAH3	58	0	2
	FC	VAH4	61	1	3
	FC	VAH5	310	40	12

Problem	Solver	VAH	#Node	#BT	#Runtime (ms)
d-10-06	BT	VAH1	145	9	4
	BT	VAH2	*	*	inf
	BT	VAH3	253	17	17
	BT	VAH4	294	30	10
	BT	VAH5	36667436	10859397	1264936
	FC	VAH1	136	9	9
	FC	VAH2	1161123	293168	50597
	FC	VAH3	236	17	11
	FC	VAH4	264	27	17
	FC	VAH5	247194	67350	9082

Problem	Solver	VAH	#Node	#BT	#Runtime (ms)
d-10-07	BT	VAH1	90	4	4
	BT	VAH2	*	*	inf
	BT	VAH3	58	0	2
	BT	VAH4	62	2	1
	BT	VAH5	71123278	20649723	2695156
	FC	VAH1	86	4	2
	FC	VAH2	15678782	4036905	833402
	FC	VAH3	58	0	3
	FC	VAH4	60	1	2
	FC	VAH5	112806	26839	4498

Problem	Solver	VAH	#Node	#BT	#Runtime (ms)
d-10-08	BT	VAH1	73	1	2
	BT	VAH2	254181450	78451064	10126751
	BT	VAH3	70	1	2
	BT	VAH4	90	4	4
	BT	VAH5	1262019	360084	51939
	FC	VAH1	72	1	1
	FC	VAH2	1602959	448876	79051
	FC	VAH3	69	1	2
	FC	VAH4	86	4	8
	FC	VAH5	12052	3715	560

Problem	Solver	VAH	#Node	#BT	#Runtime (ms)
d-10-09	BT	VAH1	58	0	1
	BT	VAH2	*	*	inf
	BT	VAH3	5591	613	223
	BT	VAH4	6440	786	254
	BT	VAH5	23110126	4174983	529645
	FC	VAH1	58	0	1
	FC	VAH2	6397614	1773282	280992
	FC	VAH3	4978	613	202
	FC	VAH4	5654	770	229
	FC	VAH5	11048121	3033654	411355

Problem	Solver	VAH	#Node	#BT	#Runtime (ms)
d-15-01	BT	VAH1	1011058	79912	58903
	BT	VAH2	*	*	inf
	BT	VAH3	587590	58173	43055
	BT	VAH4	910576	99401	64172
	BT	VAH5	*	*	inf
	FC	VAH1	931146	79912	52039
	FC	VAH2	486723267	92532940	18723985
	FC	VAH3	529417	58173	38918
	FC	VAH4	811175	98868	57772
	FC	VAH5	*	*	inf

Conclusion:

VAH1, VAH3 and VAH4 performs pretty decent for all of the cases.

VAH2 is a less powerful heuristics that tries to reduce the branching factor for the future choices and its performance turns out to be poor.

VAH5 is just random choice so it does not work that well as expected.

We can clearly observe that VAH1 and VAH3 both perform better than others. VAH1 chooses the variable with smallest domain size. Thus we can backtrack from the wrong paths early and reach our solutions more easily. VAH3 uses VAH2 as tie breaker and tries to make it more efficient.

In case of using VAH2 alone, choosing variable with max forward degree is not that good for our problem because wrong paths are detected way later.

VAH4 also has decent performance. But VAH5 is just random choice and does not work well.

Of course, Forward checking works better than backtracking because we abandon the paths leading to failure way earlier by checking domain of the future variables.

SO FC+VAH1 or FC+VAH3 performs the best. However, FC + VAH3 is a good choice because it uses VAH2 as a tie breaker though it may take a bit more time for calculation.