Report

Offline on Constraint Satisfaction Problem (Latin Square)

Date of Submission: 9th January 2023

Submitted by:

Kazi Ababil Azam
1805077
L-3/T-2
CSE-B1

Value Order Heuristic:

The value order heuristic in use chooses from the values in the domain in the ascending order. The given problems are solvable with this heuristic, barring some.

The heuristic will simply pick the numerically lowest value from the domain and try to assign it to the variable. This heuristic serves better in two ways, one is that it does not require any sorting or any manipulation of the domain list, the other being the subsequent assignments of the same variable are much faster as we are trying to assign values in the predefined ascending order (the array indicating the domain).

Table:

Value Order Heuristic	Problem	Solver	VAH	# Nodes	# BT	Runtime (ms)
Ascending Ordered Value	d-10-01	BT	VAH1	208	151	17
			VAH2	1060855716	1060855659	465713
			VAH3	57	0	12
			VAH4	59	2	3
			VAH5	2510558158	2510558101	748198
		FC	VAH1	208	151	5
			VAH2	28677475	28677418	14322
			VAH3	57	0	6
			VAH4	59	2	2
			VAH5	2196147	2196090	1488
	d-10-06	BT	VAH1	57	0	6
			VAH2	200860911	200860854	84340
			VAH3	57	0	5
			VAH4	58	1	6
			VAH5	8195533	8195476	3067
		FC	VAH1	57	0	2
			VAH2	6936956	6936899	4124
			VAH3	57	0	4
			VAH4	58	1	3
			VAH5	34365	34308	69
	d-10-07	BT	VAH1	1555	1498	20
			VAH2	299113274	299113217	146407
			VAH3	465	408	14
			VAH4	576	519	15
			VAH5	*	*	*
		FC	VAH1	1555	1498	12
			VAH2	4571968	4571911	2750
			VAH3	465	408	15
			VAH4	548	491	12
			VAH5	3548641	3548584	2224

	d-10-08	BT	VAH1	1040	983	11
			VAH2	409798240	409798183	189264
			VAH3	134	77	8
			VAH4	121	64	4
			VAH5	*	*	*
		FC	VAH1	1040	983	12
			VAH2	12162652	12162595	6076
			VAH3	134	77	10
			VAH4	120	63	4
			VAH5	352349	352292	366
	d-10-09	BT	VAH1	57	0	1
			VAH2	192995049	192994992	84043
			VAH3	24504	24447	267
			VAH4	22691	22634	87
			VAH5	*	*	*
		FC	VAH1	57	0	9
			VAH2	7684384	7684327	4011
			VAH3	24504	24447	211
			VAH4	44512	44455	151
			VAH5	133465533	133465476	58843
	d-15-01	BT	VAH1	932672	932566	3846
			VAH2	*	*	*
			VAH3	482936	482830	3348
			VAH4	1064131	1064025	3949
			VAH5	*	*	*
		FC	VAH1	932672	932566	3305
			VAH2	*	*	*
			VAH3	482936	482830	3026
			VAH4	1172354	1172248	3846
			VAH5	*	*	*

The star (*) marked rows had been run for about 30-40 minutes without results.

Conclusion:

We can conclude easily from the above table that forward checking drastically improves the performance of the solver. The number of nodes and backtracks are lesser, and the times taken are more or less lower, depending on the problems.

Among the variable order heuristics used in the solver, the random variable picker and the heuristic using maximum forward degree to pick the next variable have very poor performance. The random heuristic has issues as it backtracks very frequently due to the next variable picked being arbitrary. The second variable order heuristic on the other hand, is picking the variable with the most unassigned variables in the row and column of the said variable, creating more conflicts and thus backtracking more often than not.

The best performances thus directly or indirectly involve the heuristic picking the variable on the basis of the smallest domain size (the first heuristic). The cause is clear, the choice on the basis of domain size takes a lot less time than the calculation of forward degrees of each variable, and also the chosen variable has the smallest domain, ensuring that the generated nodes are lesser causing a lower number of nodes created and backtracks to be followed.

The third and fourth heuristics are apparent to be better than the first, which implies that the combination of the min-domain and max-forward-degree is a feasible choice for the heuristic. The fourth heuristic involving the minimization of the ratio of the two is performing the best, but not as much when larger problems (N = 15) are tried to be solved. Therefore, it is very apparent that the solution runtimes and overall performance depend on the problem itself, i.e. the given latin square. The third heuristic performs better, so does the first, as the third heuristic only appends the max-forward-degree metric as a tiebreaker for variables with equally small domains.

To conclude, the variables with the smallest domains are to be picked first for a better performance in such constraint satisfaction problems, with some modifications if further improvement is necessary.
