Data Table

Problem	Solver	VAH	#Node	#BT	Time (ms)
d-10-01	BT	VAH1	128	70	17
	BT	VAH2	*	*	*
	BT	**VAH3**	**58**	**0**	**12**
	BT	VAH4	332	274	23
	BT	VAH5	5546625	5546567	4787
	FC	**VAH1**	**67**	**9**	**7**
	FC	VAH2	657944	657886	3669
	FC	VAH3	255	197	11
	FC	VAH4	301	243	17
	FC	VAH5	10448	10390	91
d-10-06	BT	VAH1	67	9	14
	BT	VAH2	*	*	*
	BT	**VAH3**	**58**	**0**	**6**
	BT	VAH4	58	0	29
	BT	VAH5	125712002	125711944	108198
	FC	VAH1	60	2	12
	FC	VAH2	355513	355455	2158
	FC	**VAH3**	**58**	**0**	**8**
	FC	VAH4	58	0	10
	FC	VAH5	376395	376337	669
d-10-07	**BT**	**VAH1**	**95**	**37**	**5**
	BT	VAH2	*	*	*
	BT	VAH3	80	22	6
	BT	VAH4	58	0	8
	BT	VAH5	3562257	3562199	3518
	FC	**VAH1**	**91**	**33**	**5**
	FC	VAH2	269387	269329	1685
	FC	**VAH3**	**58**	**0**	**4**
	FC	VAH4	61	3	12
	FC	VAH5	24095	24037	130

d-10-08	BT	VAH1	58	0	48
	BT	VAH2	*	*	*
	BT	VAH3	701	643	32
	BT	**VAH4**	**188**	**130**	**23**
	BT	VAH5	3871379	3871321	4423
	FC	**VAH1**	**58**	**0**	**6**
	FC	VAH2	882225	882167	6157
	FC	VAH3	205	147	17
	FC	VAH4	90	32	17
	FC	VAH5	6453	6395	85
d-10-09	**BT**	**VAH1**	**61**	**3**	**8**
	BT	VAH2	*	*	*
	BT	VAH3	58	0	13
	BT	VAH4	63	5	12
	BT	VAH5	*	*	*
	FC	**VAH1**	**60**	**2**	**9**
	FC	VAH2	*	*	*
	FC	VAH3	58	0	12
	FC	VAH4	130	72	26
	FC	VAH5	132376	132318	549
d-15-01	**BT**	**VAH1**	**299884**	**299777**	**1542**
	BT	VAH2	*	*	*
	BT	VAH3	378538	378431	2705
	BT	VAH4	549177	549070	11152
	BT	VAH5	*	*	*
	FC	**VAH1**	**77945**	**77838**	**588**
	FC	VAH2	*	*	*
	FC	VAH3	245066	244959	2096
	FC	VAH4	732620	732513	16892
	FC	VAH5	*	*	*

Value Ordering Heuristic

In this assignment, after a variable has been selected using a variable-order-heuristic, to decide on the order in which to examine its values, **least-constraining-value** heuristic had been applied.

It prefers the value that rules out the fewest choices for the neighboring variables (in this problem, the neighboring variables are the initial empty cells that are either in same row or same column) in the board. As it happens to leave the maximum flexibility for assignments of subsequent unassigned neighbor variables, it is more likely to go through the path which will provide a solution faster.

Analysis

To apply the heuristics given, a **modified backtracking** method had been used where the modification is: When a value of a variable is successfully assigned, then the domain of its neighboring variables has been pruned accordingly. The difference between the **forward checking** and the **modified backtracking** is: in forward checking, after updating the domain of neighboring variables, if any of those domains becomes empty, then our solver immediately backtracked from that node, whereas no such decision is made based on the size of the domain in the modified backtrack.

Reason to use a Modified Backtracking: If a pure backtrack solver is used, then the heuristics will not get the updated domain. Therefore, it will work on the backdated domain every time and this will not be able to provide the power of the heuristics.

Observations

From the data table, it is clear that the **Forward Checking** scheme performs better than **Backtracking** for all the test cases **as it detects failure earlier and reduces the number of nodes in the search tree by pruning larger parts of the tree earlier.** For example, for test case **d-10-08**, the **BT with heuristic VAH-2**

searches total **3871379 nodes** where the FC searches only **6453 nodes** (a factor of 600 improvement!).

Performance Comparison among the five Variable-Order-Heuristics

The "Minimum Remaining Values (MRV)", that is, **VAH-1** heuristic seems to be the best from the data table as it gives the best performance (considering number of nodes, backtracks and time to find the solution) in most of the test cases. The reason is - **it picks a variable that is most likely to cause a failure soon, thereby pruning the search tree; hence avoiding pointless searches through other variables.**

After VAH-1, the VAH-3 gives the 2nd best performance, and then VAH-4. The VAH-3 is a extended version of VAH-1, where the tie is broken using the **forward degree heuristic** (selecting the variable that is involved with the largest number of constraints on other unassigned variables). Therefore, some extra work has to be done to get the degree of the variables that make a tie. On the other hand, VAH-4 tries to combine heuristic VAH-1 and VAH-2 by minimizing the ratio. It gives similar performance to VAH-3 in some test cases.

The VAH-2 heuristic can be used to break tie of VAH-1 heuristic. It alone does not provide much improvement in performance which is evident from the data table.

Finally, the VAH-5 heuristic picks variables randomly and its performance changes in different runs but on average, it provides poor performance than VAH-1, VAH-3 and VAH-4 as it is not really a heuristic.

Conclusion

Forward checking combining with VAH-1 seems the best scheme. The reason is, it picks a variable that is most likely to cause a failure soon and checking the size of the domain of neighboring variables, it detects failure early, thereby pruning the search tree - avoiding pointless searches through other variables.