**Report**

**CSE 318**

**Artificial Intelligence Sessional**

Name: A. H. M. Osama Haque

ID: 1805002

**Value Order Heuristics:**

The heuristics I used for value ordering is **Least Constraining Domain Value.**

It searches for the value which is least conflicting with the unassigned variables residing in the same row or column of the chosen variable. This ensures us a more direct approach to solve a latin board, that is to try for a value that has the most potential.

Downside of this heuristic is that we have to always sort the values by conflicting count every time we are working with a specific variable. Therefore when a problem has many unassigned variables, the run time may increase for some cases.

However this heuristic works pretty well with the given test cases, thus proven to be a decent approach.

**Table:**

The table formed by applying the variable heuristics and value ordering heuristics and using both backtracking and forward checking is given below:

| **Problem** | **Solver** | **VAH** | **#Node** | **#BT** | **Time (ms)** |
| --- | --- | --- | --- | --- | --- |
| d-10-01 | BT | VAH1 | 181 | 123 | 36 |
| BT | VAH2 | 183506796 | 183506738 | 503222 ms = 8.39 min |
| BT | VAH3 | 58 | 0 | 16 |
| BT | VAH4 | 275239 | 275181 | 1797 |
| BT | VAH5 | 383121727 | 383121669 | 1166709 ms = 19.45 min |
| FC | VAH1 | 169 | 111 | 42 |
| FC | VAH2 | 19834 | 19776 | 548 |
| FC | VAH3 | 58 | 0 | 16 |
| FC | VAH4 | 96847 | 96789 | 1409 |
| FC | VAH5 | 756298 | 756240 | 5725 |

| **Problem** | **Solver** | **VAH** | **#Node** | **#BT** | **Time (ms)** |
| --- | --- | --- | --- | --- | --- |
| d-10-06 | BT | VAH1 | 58 | 0 | 18 |
| BT | VAH2 | \* | \* | \* |
| BT | VAH3 | 147 | 89 | 31 |
| BT | VAH4 | 23099 | 23041 | 383 |
| BT | VAH5 | 10944141 | 10944083 | 35392 |
| FC | VAH1 | 58 | 0 | 17 |
| FC | VAH2 | 2838983 | 2838925 | 20492 |
| FC | VAH3 | 138 | 80 | 28 |
| FC | VAH4 | 10636 | 10578 | 390 |
| FC | VAH5 | 713250 | 713192 | 5170 |

| **Problem** | **Solver** | **VAH** | **#Node** | **#BT** | **Time (ms)** |
| --- | --- | --- | --- | --- | --- |
| d-10-07 | BT | VAH1 | 261 | 203 | 41 |
| BT | VAH2 | 1391417056 | 1391417116 | 4291929 ms = 71 min |
| BT | VAH3 | 63 | 5 | 19 |
| BT | VAH4 | 56559 | 56501 | 755 |
| BT | VAH5 | 173723784 | 173723726 | 551476 ms = 9.2 min |
| FC | VAH1 | 245 | 187 | 47 |
| FC | VAH2 | 1184177 | 1184119 | 8989 |
| FC | VAH3 | 62 | 4 | 18 |
| FC | VAH4 | 13519 | 13461 | 422 |
| FC | VAH5 | 2365 | 2307 | 141 |

| **Problem** | **Solver** | **VAH** | **#Node** | **#BT** | **Time (ms)** |
| --- | --- | --- | --- | --- | --- |
| d-10-08 | BT | VAH1 | 80 | 22 | 25 |
| BT | VAH2 | \* | \* | \* |
| BT | VAH3 | 139 | 81 | 31 |
| BT | VAH4 | 25285 | 25227 | 428 |
| BT | VAH5 | 3465929 | 3465871 | 16961 |
| FC | VAH1 | 78 | 20 | 22 |
| FC | VAH2 | 164099686 | 164099627 | 1268257 ms = 21.14 min |
| FC | VAH3 | 134 | 76 | 28 |
| FC | VAH4 | 11578 | 11520 | 400 |
| FC | VAH5 | 10346 | 10288 | 348 |

| **Problem** | **Solver** | **VAH** | **#Node** | **#BT** | **Time (ms)** |
| --- | --- | --- | --- | --- | --- |
| d-10-09 | BT | VAH1 | 67 | 9 | 18 |
| BT | VAH2 | \* | \* | \* |
| BT | VAH3 | 63 | 5 | 18 |
| BT | VAH4 | 2719 | 2661 | 112 |
| BT | VAH5 | 432368 | 432310 | 2074 |
| FC | VAH1 | 66 | 8 | 23 |
| FC | VAH2 | 10373714 | 10376656 | 82973 |
| FC | VAH3 | 62 | 4 | 20 |
| FC | VAH4 | 149 | 91 | 45 |
| FC | VAH5 | 1211 | 1153 | 107 |

| **Problem** | **Solver** | **VAH** | **#Node** | **#BT** | **Time (ms)** |
| --- | --- | --- | --- | --- | --- |
| d-15-01 | BT | VAH1 | 92633 | 92526 | 1630 |
| BT | VAH2 | \* | \* | \* |
| BT | VAH3 | 137649 | 137542 | 1880 |
| BT | VAH4 | \* | \* | \* |
| BT | VAH5 | \* | \* | \* |
| FC | VAH1 | 57242 | 57135 | 1144 |
| FC | VAH2 | \* | \* | \* |
| FC | VAH3 | 142520 | 142413 | 2259 |
| FC | VAH4 | \* | \* | \* |
| FC | VAH5 | \* | \* | \* |

**Conclusion:**

For 2nd and 4th test cases having 10x10 cells, VAH1 takes the least time to run. Also for the case having 15x15 cells, VAH1 takes less time than others.

This prompts me to believe that VAH1 is a very safe heuristic to try for any given dimension.

So if the dimension increases, chances are that it can be solved with VAH1 without any issues.

VAH3 gives good responses with the test cases 1 and 3. It basically mimics VAH1 except for the tie breaker. But the tie breaking heuristic VAH2 did not turn out to be a good heuristic as for most of the cases, either it took a really long time or it could not finish at all. Therefore even as a tiebreaker it can interfere with the overall performance of VAH1.

VAH4 on the other hand, worked well but still took a significant amount of time because of the VAH2 degree count in the denominator.

VAH5 is fairly unpredictable with the results. But it is evident that, higher the dimension of latin square, the more insignificant it becomes as a heuristics.

To sum up, as dimension increases, the type of variable heuristic matters a lot. In our experiment, VAH1 and VAH3 were the most useful variable heuristics.