

**ARTIFICIAL INTELLIGENCE**  
**CSE-402**

Report on

*Solving Puzzle Game “Sudoku” by CSP*

&

*Performance Analysis of Heuristics*

*Submitted By:*

*Mahmudur Rahman Hera.*

*Student No: 1105001.*

# Implemented Heuristics:

We were asked to implement the following heuristics:

a. For Variable Ordering:

1. MRV – Minimum Remaining Value variable
2. Degree – The variable with most unassigned variables as its degree
3. MRV with Degree – MRV breaks tie with Degree heuristic
4. First available – The variable that comes first
5. Random – Picking a random unassigned variable

b. For value ordering:

1. LRV – The value that leaves the others most choice
2. First available – Picking the first value from domain
3. Random – Picking a random value

c. Constraint Propagation:

1. FC – Forward checking
2. MAC – Maintaining Arc Consistency
3. Powerful propagator – A propagator that removes singletons and twins was implemented

**Table 1:** Comparison of inference rules: 1. Forward checking, 2. maintaining Arc Consistency, 3. propagator with detection and removal of singletons and twins.

Problem in Setting	Difficulty	Inference Rule	Variable Ordering	Value Ordering	Total Consistency Checks	Total times variable selected	Total times value selected	Total runtime
1	EASY	FC	MRV	LRV	168	167	174	< 1 sec
		MAC	MRV	LRV	168	167	174	< 1 sec
		POWERFUL	MRV	LRV	168	167	174	< 1 sec
2	EASY	FC	MRV	LRV	94	93	95	< 1 sec
		MAC	MRV	LRV	94	93	95	< 1 sec
		POWERFUL	MRV	LRV	94	93	95	< 1 sec
3	MEDIUM	FC	MRV	LRV	107	106	107	< 1 sec
		MAC	MRV	LRV	105	104	105	< 1 sec
		POWERFUL	MRV	LRV	99	98	99	< 1 sec
4	MEDIUM	FC	MRV	LRV	1463	1462	1476	< 1 sec
		MAC	MRV	LRV	1463	1462	1476	< 1 sec
		POWERFUL	MRV	LRV	1463	1462	1476	< 1 sec
5	HARD	FC	MRV	LRV	766	765	805	< 1 sec
		MAC	MRV	LRV	765	764	804	< 1 sec
		POWERFUL	MRV	LRV	765	764	804	< 1 sec
6	HARD	FC	MRV	LRV	299	298	298	< 1 sec
		MAC	MRV	LRV	385	384	384	< 1 sec
		POWERFUL	MRV	LRV	385	384	384	< 1 sec
7	HARD	FC	MRV	LRV	6456	6455	6458	< 1 sec
		MAC	MRV	LRV	6328	6329	6328	< 1 sec
		POWERFUL	MRV	LRV	6296	6295	6298	< 1 sec

**Observation:**

Among “Forward Checking” and “Maintaining Arc Consistency”, the latter performs better. Among MAC and the Powerful propagator that implements removing the singletons and the twins from a Sudoku grid, the performance is almost the same, but for some tricky boards, the powerful propagator performs better, although this is not true always.

**Table 2:** Comparison between the value ordering heuristics: 1. least restraining value, 2. First available value and 3. random value selection heuristics.

Problem in Setting	Difficulty	Inference Rule	Variable Ordering	Value Ordering	Total Consistency Checks	Total times variable selected	Total times value selected
1	EASY	MAC	MRV	LRV	168	167	174
		MAC	MRV	First available	170	169	690
		MAC	MRV	Random	63	62	196
2	EASY	MAC	MRV	LRV	94	93	95
		MAC	MRV	First available	92	91	302
		MAC	MRV	Random	59	58	193
3	MEDIUM	MAC	MRV	LRV	106	105	106
		MAC	MRV	First available	85	84	256
		MAC	MRV	Random	753	752	2622
4	MEDIUM	MAC	MRV	LRV	1463	1462	1476
		MAC	MRV	First available	1373	1372	4735
		MAC	MRV	Random	1484	1483	5331
5	HARD	MAC	MRV	LRV	765	764	804
		MAC	MRV	First available	727	726	2426
		MAC	MRV	Random	1193	1192	4022
6	HARD	MAC	MRV	LRV	385	384	384
		MAC	MRV	First available	834	833	2966
		MAC	MRV	Random	1767	1766	6635
7	HARD	MAC	MRV	LRV	6296	6295	6298
		MAC	MRV	First available	3396	3395	12095
		MAC	MRV	Random	5976	5975	20562

**Observation:**

Random and First available – these two heuristics usually pick a lot of values that are cut off during the inferences. There may be cases when these two show better performance in the count of consistency checks compared to LRV, but the better performance of LRV in picking the values that ultimately leads to the solution overflows the effect. Therefore, among these three, LRV is the best one to be used.

**Table 3:** Comparison between variable selection heuristics: 1. minimum remaining value with degree of unassigned variables as tiebreaker, 2. minimum remaining value, 3. Variable with maximum degree, 4. First available variable and 5. Random variable selection.

Problem in Setting	Difficulty	Inference Rule	Variable Ordering	Value Ordering	Total Consistency Checks	Total times variable selected	Total times value selected
1	EASY	MAC	MRV-deg	LRV	84	83	92
		MAC	MRV	LRV	94	93	98
		MAC	Degree	LRV	99	98	146
		MAC	First	LRV	89	88	246
		MAC	Random	LRV	1784	1783	10765
2	EASY	MAC	MRV-deg	LRV	51	50	143
		MAC	MRV	LRV	62	61	230
		MAC	Degree	LRV	71	70	71
		MAC	First	LRV	76	75	77
		MAC	Random	LRV	5299	5298	5931
3	MEDIUM	MAC	MRV-deg	LRV	106	105	106
		MAC	MRV	LRV	434	433	435
		MAC	Degree	LRV	1630	1629	1663
		MAC	First	LRV	435	434	434
		MAC	Random	LRV	504184	504184	587079
4	MEDIUM	MAC	MRV-deg	LRV	1463	1462	1476
		MAC	MRV	LRV	909	908	933
		MAC	Degree	LRV	62	61	63
		MAC	First	LRV	898	897	952
		MAC	Random	LRV	507879	507878	623246
5	HARD	MAC	MRV-deg	LRV	765	764	804
		MAC	MRV	LRV	486	484	504
		MAC	Degree	LRV	14504	14503	16053
		MAC	First	LRV	130	129	130
		MAC	Random	LRV	3127652	3127651	3678870
6	HARD	MAC	MRV-deg	LRV	385	384	384
		MAC	MRV	LRV	933	932	938
		MAC	Degree	LRV	193	192	202
		MAC	First	LRV	1650	1649	1675
		MAC	Random	LRV	1749005	1749004	2113675

**Observation:**

The random ordering of variable selection is too bad to be used in solving boards. For even easiest boards it consumes much time. For the easy ranked problems, the others are close to MRV with degree heuristic as a tiebreaker. For harder problems, performance of the others except for MRV with degree heuristic as tiebreaker starts to deteriorate. The performance of those are sometimes quite satisfactory, but worse in others and not consistent. However, MRV with degree keeps showing

consistent performances. Random variable ordering often fails to solve the Sudoku grid and needs to be restarted.

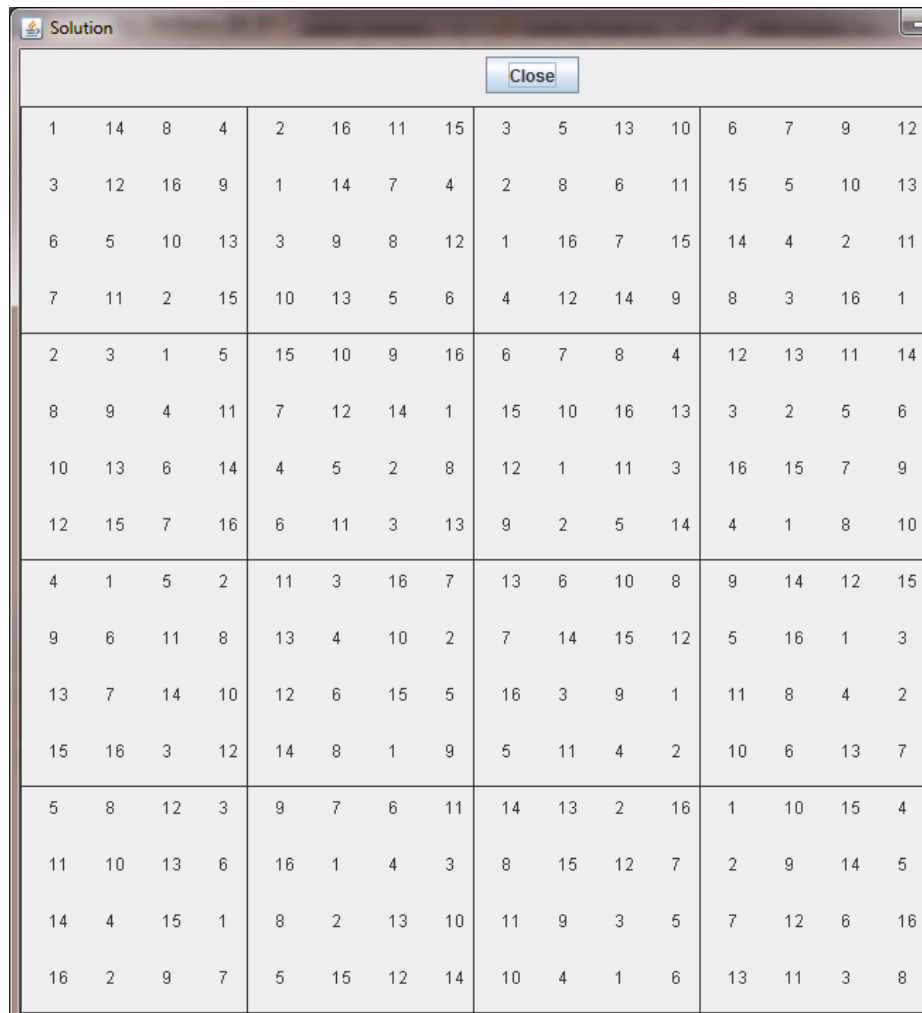
## Final Solver:

Combing all these findings, the solver is initialized to run with

- MRV heuristic for variable ordering
- LRV heuristic for value ordering
- Maintaining Arc Consistency and removing singletons and twins along the path as the inference rule.

This can solve a 16x16 Sudoku board in around 2 seconds and 25x25 in 4~5 seconds. Image of a solved 16x16 board and a partial image of a 25x25 boards are given below:

- A solved 16x16 Sudoku board:



1	14	8	4	2	16	11	15	3	5	13	10	6	7	9	12
3	12	16	9	1	14	7	4	2	8	6	11	15	5	10	13
6	5	10	13	3	9	8	12	1	16	7	15	14	4	2	11
7	11	2	15	10	13	5	6	4	12	14	9	8	3	16	1
2	3	1	5	15	10	9	16	6	7	8	4	12	13	11	14
8	9	4	11	7	12	14	1	15	10	16	13	3	2	5	6
10	13	6	14	4	5	2	8	12	1	11	3	16	15	7	9
12	15	7	16	6	11	3	13	9	2	5	14	4	1	8	10
4	1	5	2	11	3	16	7	13	6	10	8	9	14	12	15
9	6	11	8	13	4	10	2	7	14	15	12	5	16	1	3
13	7	14	10	12	6	15	5	16	3	9	1	11	8	4	2
15	16	3	12	14	8	1	9	5	11	4	2	10	6	13	7
5	8	12	3	9	7	6	11	14	13	2	16	1	10	15	4
11	10	13	6	16	1	4	3	8	15	12	7	2	9	14	5
14	4	15	1	8	2	13	10	11	9	3	5	7	12	6	16
16	2	9	7	5	15	12	14	10	4	1	6	13	11	3	8

2. A Solved 25x25 Sudoku Board: (Image is too large to fit)

Solution																								
Close																								
1	23	11	5	12	2	24	19	8	10	3	25	16	14	17	4	21	22	20	13	9	6	7	18	15
3	24	17	6	16	1	20	4	25	11	21	22	13	18	10	23	7	9	15	19	14	5	12	2	8
7	20	19	13	15	3	17	12	9	16	24	8	2	5	4	1	11	6	14	18	10	22	21	23	25
8	18	21	2	4	14	15	13	7	22	23	20	6	9	19	10	25	5	12	24	1	3	11	17	16
9	22	25	14	10	5	18	23	21	6	1	12	11	7	15	2	8	3	16	17	13	19	20	4	24
2	19	1	18	13	24	22	11	20	21	12	14	23	3	5	7	16	4	25	15	6	8	17	10	9
10	14	7	20	6	9	23	3	15	25	4	13	24	8	1	21	12	17	11	5	18	16	2	22	19
12	5	9	22	8	17	4	6	10	19	16	2	21	20	25	18	14	23	24	3	15	13	1	11	7
15	25	3	23	11	13	2	1	16	14	22	17	7	19	18	6	9	20	8	10	12	24	5	21	4
16	21	4	24	17	8	7	5	12	18	10	6	15	11	9	13	22	19	2	1	3	20	25	14	23
4	16	6	1	22	18	10	25	5	9	13	3	8	12	11	20	19	24	21	2	23	14	15	7	17
11	13	14	3	24	20	12	7	1	17	2	19	22	25	23	15	10	16	9	8	21	4	6	5	18
17	10	20	7	2	4	19	8	6	23	14	15	18	24	21	5	1	13	3	11	22	25	9	16	12
19	12	23	9	5	16	3	21	14	15	20	1	4	10	6	22	18	25	17	7	2	11	24	8	13
21	15	8	25	18	11	13	22	24	2	7	5	9	17	16	14	6	12	23	4	19	1	10	3	20
5	3	2	8	9	23	1	15	22	24	25	18	14	16	13	12	20	7	19	21	17	10	4	6	11
13	4	10	11	14	12	5	9	2	3	8	21	19	6	7	24	17	18	1	25	16	15	23	20	22