

The 2nd best Sudoku solver

Kirin, Gary, Ramil, Felix



Tools

- Python
 - Tkinter - python gui package to display our board
- PyCharm
- Git & Github

Board Generation

- Computer generated
 - Based on the all-difference constraint for each row, column and subgrid
- Pseudorandom
- 25% filled
- Not a true sudoku
- Can't guarantee it is solvable

Brute Force

Recursive DFS searching Algorithm

Class: BruteForce (one 2D list storing the state of the board)

Heuristics:

- Backtracking: rule out invalid numbers for each square
Maintains a set for each row, column and subgrid of used numbers
- MRV:
Empty square to assign next: with the least number of valid options

Limits:

- Max number of fail:
 - Initial set as grid size * 1000
 - Increase by 1.05 each fail
- Max time lapse: 5 mins

CSP

Classes: Square, Arc, CSP

Process:

- Initialize constraints (generate Arcs, populate neighbours fields)
- Preprocess search space using ac3
- Multiprocessing
- Start backtrack
 - MRV - with degree heuristic for tiebreaker
 - Least constraining value
 - MAC
 - Uniqueness
 - Naked Pair

Multiprocessing

- Sudoku is CPU dependent
 - Multiprocessing is more appropriate vs multithreading
- Ineffectively used
 - Single usage
 - Unable to have recursive calls also employ multiprocessing
- Puzzles still solved quickly, up to a point
 - Ideal cell determined by MRV usually had <16 values

Naked Pairs

- Cell domain of just a pair of values
- Neighbouring cell with the same domain
- All other neighbours domain can have that pair dropped from them

Uniqueness

- If a cell's domain only contains one value, it's effectively assigned
- Greatly reduced our processing time

AWS

- Ineffective for our purposes
- The free tier only provides a t2 or t3 micro
 - max frequency of 3.3 or 3.1 GHz respectively
 - 1 or 2 cores max
- Our devices have 10+ cores with a higher frequency
- Usage of AWS is simply not feasible unless we pay

Brute Force Stats

Unit: seconds

Puzzle Size	Mean	St. Dev	Completed
9x9	2.2744	8.7766	15/15
12x12	0.8426	2.7584	15/15
16x16	2.0199	4.7494	12/15
25x25	N/A	N/A	0/15
100x100	N/A	N/A	0/5

Data from 9 x 9 Sudoku Board

Unit: seconds

S\G	1	2	3	4	5
1	0.1553	0.1281	0.1368	0.3827	0.3993
2	0.1229	0.1093	0.1453	0.1598	0.2683
3	0.1384	0.1291	0.1470	0.1436	0.2522

Mean	St. Dev
0.1879	0.0937

Non-naked

S\G	1	2	3	4	5
1	0.1450	0.1394	0.1604	0.3189	0.3056
2	0.1486	0.1247	0.1411	0.1624	0.2985
3	0.2054	0.1424	0.1470	0.1395	0.2341

Mean	St. Dev
0.1875	0.0682

Naked

Data from 12 x 12 Sudoku Board

Unit: seconds

S\G	1	2	3	4	5
1	0.2089	0.1960	0.2141	0.1560	1.2143
2	0.1668	0.1986	0.2477	0.1790	0.2151
3	0.2076	0.1877	0.1920	0.2105	0.1861

Mean	St. Dev
0.2654	0.2634

Non-naked

S\G	1	2	3	4	5
1	0.2297	0.2275	0.2161	0.1694	1.0178
2	0.2061	0.2094	0.2899	0.1826	0.2013
3	0.2254	0.1836	0.2207	0.2219	0.2041

Mean	St. Dev
0.2670	0.2095

Naked

Data from 16 x 16 Sudoku Board

Unit: seconds

S\G	1	2	3	4	5
1	0.6248	0.5477	0.2060	0.8974	41.5477
2	0.4456	0.6748	1.4303	1.9287	0.8942
3	0.3631	0.4809	0.4102	2.0752	0.7595

Mean	St. Dev
3.5524	10.5258

Non-naked

S\G	1	2	3	4	5
1	0.5868	0.4259	0.2029	0.6220	5.4800
2	0.6991	0.4351	1.2181	0.7470	0.7306
3	0.3626	0.5137	0.4317	2.0292	0.7401

Mean	St. Dev
1.0150	1.3102

Naked

Data from 25 x 25 Sudoku Board

Unit: seconds

S\G	1	2	3	4	5
1	12.5972	DNF	230.2718	DNF	DNF
2	75.6233	116.6268	86.9199	DNF	DNF
3	393.1911	DNF	102.7015	DNF	DNF

Mean	St. Dev
145.4188	127.2661

Naked

CSP Overall Stats with Naked Pairs

Puzzle Size	Mean	St. Dev	Completed
9x9	0.1875	0.0682	15/15
12x12	0.2670	0.2095	15/15
16x16	1.0150	1.3102	15/15
25x25	145.4188	127.2661	7/15
100x100	N/A	N/A	0/5

Resources

<https://medium.com/@davidcarmel/solving-sudoku-by-heuristic-search-b0c2b2c5346e>