

Henro926

Task 1

a) Depth first graph search - does not work for any puzzle. There are ALOT of nodes to consider, which the algorithm can not handle. Time complexity $O(d^n)$, where $d = 9$

Backtracking search - Works within 45 seconds for the easy puzzle, does not work for medium and hard. This is because of the BT is a DFS but with constrains, which results in it working for some easy sudokus, but not for harder ones.

AC3 (Arc consistency) - Works super fast for the easy puzzle, however, since as more moves are required for medium and hard, the time complexity grows to quickly and the algorithm can not solve it. Time complexity: $O(n^2 * d^3)$

Min conflicts - Does not work for any of the puzzles. This algorithm looks for the minimum amount of conflicts, which will iteratively become higher and higher, which is why it does not work.

b) BT+FC+MRV works super fast for even the hardest puzzle, since it does a lot of reducing of search space with forward checking, combined with least constraining value, returns a very prominent result each iteration.

-> Best alternative is regular BT+FC+MRV

Task 2 - NQueens

a) BT

Since backtracking depends on which settings are applied, if default value (no forward checking, no least constraining values)

$n = 28$ is top limit in a reasonable time for my computer. (ish a minute)

AC3

Does not succeed at all. This algorithm does not contain any backtracking, which results in no solution.

Min_conflicts

$n = 6000$ takes around a minute to run, so it should work for very large n .

Depth first tree search

Is very fast for $n < 20$,

$n = 22$ is the limit to run in one minute.

b)

BT -> n=28 runs within a minute

BT+FC -> n<28, (not working)

BT+MRV -> n>28, Works better than forward checking, constrains seem to contribute more.

BT+MRV+FC -> n = 500 runs in a minute

BT+MRV+FC is once again the better choice of settings.

c) It is always random, depending on where the algorithm starts to solve, here are some outputs i got:

for n = 20, steps = 49

for n = 30, steps = 119

for n = 100, steps = 155

d) In a traditional state-based search a heuristic is created as a particular solution to the given problem.

In CSP-based problem solving, we have more general heuristics, such as LCV, which checks the least constraining values. These heuristics can be used to solve more general problems.