Compsci 367 A3 Report

Table of Results

# queens or Map for 4-colour	method	Average # of assignments ± std dev	Average time to solve ± std dev (seconds)	Average # of backtracks ± std dev	Average # of repair assignments ± std dev	Probability of Success (%)
10 queens	1	91 ± 0	0.0013 ± 0.000464	81 ± 0	NA ± NA	100
30 queens	1	1000 ± 0	0.0273 ± 0.0022	985 ± 0	NA ± NA	0
50 queens	1	1000 ± 0	0.0505 ± 0.0116	966 ± 0	NA ± NA	0
70 queens	1	1000 ± 0	0.0585 ± 0.00618	954 ± 0	NA ± NA	0
90 queens	1	1000 ± 0	0.0714 ± 0.00238	941 ± 0	NA ± NA	0
australia_map	1	6 ± 0	9.98e-05 ± 0.000299	0 ± 0	NA ± NA	100
usa_csp	1	49 ± 0	0.000798 ± 0.000399	0 ± 0	NA ± NA	100
zebra	1	1000 ± 0	0.0288 ± 0.00468	987 ± 0	NA ± NA	0
10 queens	2	73 ± 0	0.0011 ± 0.000544	35 ± 0	NA ± NA	100
30 queens	2	1000 ± 0	0.0236 ± 0.00128	570 ± 0	NA ± NA	0
50 queens	2	1000 ± 0	0.0455 ± 0.00136	568 ± 0	NA ± NA	0
70 queens	2	1000 ± 0	0.0775 ± 0.00244	542 ± 0	NA ± NA	0
90 queens	2	1000 ± 0	0.152 ± 0.038	538 ± 0	NA ± NA	0
australia_map	2	6 ± 0	0 ± 0	0 ± 0	NA ± NA	100
usa_csp	2	49 ± 0	0.000697 ± 0.000456	0 ± 0	NA ± NA	100
zebra	2	1000 ± 0	0.0215 ± 0.00434	650 ± 0	NA ± NA	0
10 queens	3	50 ± 0	0.00808 ± 0.000706	40 ± 0	NA ± NA	100
30 queens	3	1000 ± 0	0.803 ± 0.133	979 ± 0	NA ± NA	0
50 queens	3	1000 ± 0	2.52 ± 0.269	966 ± 0	NA ± NA	0
70 queens	3	1000 ± 0	5.77 ± 0.68	953 ± 0	NA ± NA	0
90 queens	3	1000 ± 0	12.9 ± 0.825	942 ± 0	NA ± NA	0
australia_map	3	6 ± 0	0.000299 ± 0.000457	0 ± 0	NA ± NA	100
usa_csp	3	49 ± 0	0.00359 ± 0.00049	0 ± 0	NA ± NA	100
zebra	3	1000 ± 0	0.0986 ± 0.0163	989 ± 0	NA ± NA	0
10 queens	4	225 ± 279	0.00508 ± 0.0065	215 ± 279	NA ± NA	100
30 queens	4	818 ± 249	0.0433 ± 0.0141	790 ± 250	NA ± NA	40
50 queens	4	772 ± 351	0.0616 ± 0.0282	725 ± 353	NA ± NA	40
70 queens	4	908 ± 168	0.14 ± 0.0359	842 ± 170	NA ± NA	40
90 queens	4	1000 ± 0	0.136 ± 0.00446	919 ± 2.65	NA ± NA	0
australia_map	4	9.2 ± 5.23	0.000199 ± 0.000399	3.2 ± 5.23	NA ± NA	100
usa_csp	4	905 ± 285	0.0356 ± 0.0123	863 ± 288	NA ± NA	10
zebra	4	1000 ± 0	0.0801 ± 0.00509	987 ± 2.74	NA ± NA	0
10 queens	5	18.1 ± 7.29	0.00479 ± 0.00208	8.1 ± 7.29	NA ± NA	100
30 queens	5	49.8 ± 28.3	0.132 ± 0.0283	19.8 ± 28.3	NA ± NA	100
50 queens	5	82 ± 41.6	1.16 ± 0.352	32 ± 41.6	NA ± NA	100
70 queens	5	446 ± 391	4.05 ± 1.11	381 ± 397	NA ± NA	70
90 queens	5	356 ± 320	10.1 ± 1.77	268 ± 323	NA ± NA	90
australia_map	5	6 ± 0	0.000399 ± 0.000489	0 ± 0	NA ± NA	100
usa_csp	5	49 ± 0	0.00529 ± 0.00046	0 ± 0	NA ± NA	100
zebra	5	45.5 ± 2.97	0.00768 ± 0.000456	20.5 ± 2.97	NA ± NA	100
10 queens	6	93.7 ± 73.9	0.00289 ± 0.0027	NA ± NA	83.7 ± 73.9	100
30 queens	6	65.7 ± 13.4	0.00319 ± 0.000873	NA ± NA	35.7 ± 13.4	100
50 queens	6	102 ± 29.6	0.00718 ± 0.00308	NA ± NA	52.1 ± 29.6	100
70 queens	6	132 ± 19.7	0.0124 ± 0.00225	NA ± NA	62.4 ± 19.7	100
90 queens	6	137 ± 50.3	0.0157 ± 0.00715	NA ± NA	47.4 ± 50.3	100
australia_map	6	6 ± 0	9.97e-05 ± 0.000299	NA ± NA	0 ± 0	100
usa_csp	6	65.4 ± 33.8	0.00409 ± 0.00787	NA ± NA	16.4 ± 33.8	100
zebra	6	997 ± 84.9	0.255 ± 0.0231	NA ± NA	972 ± 84.9	10

Discussion of Results

a) Which settings do you think work best for the constraint satisfaction algorithm and whether they are specific to certain problems or are always better or worse? Why do you think they work best?

The best settings for the constraint satisfaction algorithm are settings 5 and 6. This is because these settings solve all the assignment problems at least once throughout the ten iterations performed. In comparison, the other settings are not able to solve some of the assignment problems.

Setting 5 uses backtracking search with minimum remaining value, least constrained values and maintaining arc consistency. These conditions are core reasons for the improvement in the performance of backtracking search when compared to the other settings. By using minimum remaining value, we can optimise variable selection, using least constraining value we can optimise value selection and by maintaining arc consistency we can detect failure earlier. Therefore, using these settings overall we expand less nodes and backtrack less meaning faster runtimes and higher probability of solving. This setting is good across all problems.

Setting 6 uses the minimum conflict solver. This setting iteratively repairs conflicting constraints by reassignment, eventually resulting in a valid assignment for all variables. This setting works well on problems such as n-queens. However, it does not perform as well as setting 5 on map-colouring problems.

b) Talk about the growth of the number of assignments and time as the N-Queens problem and the Map-Colour Problem grow. Do the two domains act the same? If not, why not?

As the number of queens in the n-queens problem and number of regions in the map-colouring problem grow, on average, the number of assignments and time to solve increase. Generally, this is because there are more variables to assign values.
However, the two domains act quite differently. This is because as the size of the n-queens problem increases, the number of locations each queen can be placed in increases, hence the size of the domain also increases. In comparison in map-colouring problems, as the size of the map increases, the number of possible colourings for each region remains the same, hence the size of the domain remains the same.

c) When would you use Minimum Conflict instead of a Constraint Satisfaction Solver? Why?

The minimum conflict solver works better on problems such as n-queens rather than map-colouring problems. This is because problems such as n-queens fall outside the critical ratio region (where $Ratio = \frac{number\ of\ constraints}{number\ of\ variables}$). Therefore, for CSP problems that fall outside the critical ratio region the minimum conflicts solver should be used.

However, problems such as map-colouring lie within this critical ratio region. In this region, the average time to solve spikes and thus constraint satisfaction solvers such as variants of backtracking search should be used.

d) How does Minimum Conflict (which is essentially a local search algorithm) perform compared to the Local Search Algorithms you used in Assignment 2?

Minimum Conflict performed much better than the Assignment 2 algorithms run on the same number of queens. Generally, minimum conflict had a quicker time to solve and greater probability of success than algorithms used in Assignment 2.

e) In some problems there might be a large number of assignments, but no backtracking (or very little), why would this occur?

Problems where there are many assignments with little/no backtracking occur for two key reasons. Firstly, because the problem itself has a large number of variables, but each variable only has a few values that it can be assigned. Therefore, the probability of picking a legal assignment is much higher and resulting in no/few backtracks. Secondly, the constraints defined are easy to satisfy, therefore legal assignments are much easier to find, again resulting in little to no backtracking.