CSE537 HW2

Tim Zhang

November 4, 2016

1 Performance

Algorithm	Input	Time (ms)	Searches	Arc Prunings
DFSB++	backtrack_1	205.00199999999998	146	345
DFSB++	backtrack_2	754.674	368	546
DFSB++	backtrack_easy	0.9679999999999964	8	10
DFSB++	backtrack_hard	15331.961000000001	2679	1389
DFSB	backtrack_1	Failure	163957	0
DFSB	backtrack_2	Failure	103541	0
DFSB	backtrack_easy	0.17800000000000082	8	0
DFSB	backtrack_hard	Failure	44000	0
Min Conflicts	minconflict_1	0.4680000000000035	7	0
Min Conflicts	minconflict_2	2445.835	100036	0
Min Conflicts	minconflict_easy	1.3279999999999959	13	0
Min Conflicts	minconflict_hard	Failure	400000	0

2 Observations

2.1 DFSB

Naive DFSB could not solve any of the input sequences under the time constraints whereas under DFSB++ all instances were solvable. This is due to the massive reduction in the search space given by the heuristics used in DFSB++.

Effectively each of the heuristics in turn pruned the search tree at branches which were either destined to fail or incompatible to begin with. In this way both redundant and fruitless branches were avoided completely.

It is notable that for the easy instance of the test input DFSB had superior performance. This is due to the fact that DFSB++ uses computation time to compute the heuristics even on the simple input case. However in the more complex cases this extra computation time is more than made up for by the reduction in search space.

2.2 Min Conflicts

The observed solution time of Min Conflicts is heavily dependent on the starting configuration of the CSP, which is generated randomly in my program. For this reason there is heavy variation in observed running times on even the same input instance.

To avoid the normal pitfalls of local search I implemented my algorithm using random restart which after an arbitrary amount of searches the algorithm would terminate and be reinitialized with a new starting configuration and run again.

Due to random restart, the original hard instance "minconflict_hard" would be solvable in some cases and trivially complete depending on the starting configuration.