Decision procedures and verification - CDCL

Jaroslav Šafář

September 19, 2020

1 Experimental evaluation

Some smaller Uniform Random-3-SAT examples from SATLIB Benchmark Problems were used for experimental evaluation of CDCL algorithm.

The CDCL solver uses restarts. Restarts occur when the number of conflicts reaches some limit. We tested two possible initial limits: 100 (cdcl1) and 200 (cdcl2). This limit increases geometrically by a factor of 1.1 every time the restart occurs.

During restarts, the CDCL solver also deletes some learned clauses with LBD greater than a specified limit. This limit increases geometrically by a factor of 1.1 every time the restart occurs.

1.1 SAT examples

In the case of SAT problems with sizes 20, 50, 75, 100, 125 and 150, the first few examples (usually between 5 and 10) were selected for the experimental evaluation. Their resulting CPU times, number of decided variables, propagated variables and restarts were averaged, and the results can be found in the following figures [1], [2], [3] and [4].

size	dpll3-wl	cdcl1	cdcl2
20	0.004784	0.004381	0.004781
50	0.056514	0.056075	0.059285
75	0.364641	0.289177	0.208851
100	2.509378	1.668616	1.788764
125	3.991277	3.810911	2.732437
150	183.760093	12.613605	15.036094

Figure 1: Average time results of SAT examples

size	dpll3-wl	cdcl1	cdcl2
20	10.8	8.0	8.0
50	78.5	44.5	44.5
75	355.7	138.9	104.3
100	2074.3	594.2	600.5
125	2804.1	1097.6	776.7
150	109526.2	2651.2	3285.6

Figure 2: Average number of decisions of SAT examples

size	dpll3-wl	cdcl1	cdcl2
20	24.5	21.0	21.0
50	437.3	349.1	349.1
75	2979.1	1659.8	1231.3
100	20882.4	9191.5	9764.5
125	31352.54	18561.0	13535.9
150	1438649.0	50661.4	66305.0

Figure 3: Average number of unit propagations of SAT examples

size	cdcl1	cdcl2
20	0.0	0.0
50	0.0	0.0
75	0.5	0.1
100	2.5	1.3
125	4.2	1.6
150	8.2	6.4

Figure 4: Average number of restarts of SAT examples

1.2 UNSAT examples

In the case of UNSAT problems with sizes 20, 50, 75, 100, 125 and 150, the first few examples (usually between 5 and 10) were again selected for the experimental evaluation. Their resulting CPU times, number of decided variables, propagated variables and restarts were averaged, and the results can be found in the following figures [5], [6], [7] and [8].

size	dpll3-wl	cdcl1	cdcl2
50	0.122273	0.098447	0.097491
75	1.328260	0.711433	0.800416
100	7.895830	5.064517	4.897087
125	40.661927	16.499824	17.654869
150	250.665202	80.995486	82.773766

Figure 5: Average time results of UNSAT examples

size	dpll3-wl	cdcl1	cdcl2
50	155.8	58.8	58.8
75	1295.2	320.2	311.6
100	6077.2	1544.2	1472.0
125	26638.0	3443.2	3890.4
150	139255.6	9172.0	10104.8

Figure 6: Average number of decisions of UNSAT examples

size	dpll3-wl	cdcl1	cdcl2
50	1080.3	692.8	692.8
75	10699.2	4352.8	4477.2
100	62715.6	25546.2	24602.0
125	313781.8	65679.4	74513.2
150	1819359.0	199157.0	221677.2

Figure 7: Average number of unit propagations of UNSAT examples

size	cdcl1	cdcl2
50	0.0	0.0
75	1.6	0.6
100	6.8	3.8
125	12.0	8.2
150	20.2	15.0

Figure 8: Average number of restarts of UNSAT examples

1.3 Conclusion

As we can see from the results, the CDCL solver is better than the DPLL solver with watched literals in all measurements. Especially, we can see that it is significantly faster in both SAT and UNSAT examples.

If we compare the CDCL solver with different initial limits for restarts, we can see that the CDCL solver with the initial limit of 100 performed slightly better than the CDCL solver with the initial limit of 200. In the case of larger examples of the size 150, it was a little faster, and it also performed fewer decisions and unit propagations. It seems that the initial limit of 100 is a better choice because the CDCL solver performs restarts more often. On the other hand, if the initial limit was too small, the restarts would occur too often, which would slow down the algorithm's performance.