

Thesis experiments

yifan he

June 2021

1 Backjumping and QDLL

Instance	Baseline	Backjumping
gttt.3x3.b	1213	1213
gttt.3x3.torus.b	357	357
chain12v.13	8215	8215
chain13v.14	16409	16409
chain14v.15	32795	32795
chain15v.16	65565	65565
chain16v.17	131103	131103
chain17v.18	262177	262177
chain18v.19	524323	524323
TOILET6.1.iv.11	1441537	1441537
TOILET6.1.iv.12	218581	218581
TOILET7.1.iv.14	3036638	3036638
adder-2-sat	4463	3543
adder-2-unsat	54539	46335
k.d4_n-1	710	582
k.lin_p-2	3678092	1188617
k.path_n-2	39841	10927
k.path_n-3	981769	505581
k.path_p-3	31128	5546
k.path_p-4	632520	151620
k.t4p_n-1	5173935	1816773
k.lin_n-2	6230305	139775
impl20	2097151	41
k.dum_n-2	11326381	4590898
k.dum_p-3	891072	200362
k.dum_p-4	—	15705268

Table 1: Compare backjumping solver and baseline solver

This experiment compares the effect of the back jumping solver and the

baseline solver. The time limit is 15 minutes per instance and both solvers use the most frequent branching heuristic. The overall result shows back jumping can reduce the searching space, and 1 instance that was originally unsolvable within the time limit was solved by the back jumping trick.

The paper Backjumping for Quantified Boolean Logic satisfiability by Giunchiglia mentioned 3 instances of the BLOCK group that are impossible to be solved by the baseline solver can be solved by the back jumping solver [2]. However, even if the pruning condition was triggered many times, none of the BLOCK instances can be solved by my back jumping solver. This might because of the branching heuristic and the tie-breaking rule.

Another reason for the failure of replicating the QUBE-BJ experimental result might be the QUBE-BJ solver had bugs in its implementation. The output of the 3qbf-5cnf-100var-800cl.1 instance evaluated by the state of art QBF solver depQBF is SAT but QUBE-BJ printed UNSAT during the QBFeval 2004[1]. From my implementation experience, if there are wrong answers in the code, invalid pruning might occur and the searching space can be reduced. Thus, because QUBE-BJ can potentially print out wrong answers, the result shown in the 2004 paper cannot be completely trusted.

Another interesting thing about the back jumping trick is, the pruning often happens in a very deep level of the search tree, the number of nodes pruned is limited. This might be a reason that almost all state of art QBF solvers use clause learning instead of back-jumping.

References

- [1] Results for instance 3qbf-5cnf-100var-800cl.1 QBFEVAL'04. http://www.qbflib.org/benchmark_solvers.php?idBench=9308&year=2004&track=1, 2004. Accessed: 2021-06-09.
- [2] Enrico Giunchiglia, Massimo Narizzano, and Armando Tacchella. Back-jumping for quantified boolean logic satisfiability. *Artificial Intelligence*, 145(1-2):99–120, 2003.