

# Algorithms for SAT

Ranadeep Biswas

## 1 Introduction

Donald E. Knuth discussed Satisfiability (7.2.2.2.) of CNF in Volume 4B of *The Art of Computer Programming*. We tried to analyze these algorithms using the notion of pruning of search tree.

## 2 SAT solving algorithms

Typically, these algorithms are basically a family of algorithms to search a set of literals in a search space that satisfies the CNF. We can consider it as a tree such that it satisfies these following conditions.

- $E(u, v)$  if  $u \subset v$  and  $|u \setminus v| = 1$
- if  $|u| = n$ , then  $\{|l| \mid l \in u\} = [1 \dots n]$

Then root node is the empty assignment and leaves are full assignments. Rest of the nodes are partial assignments.

### 2.1 Backtrack

This is the naive approach to solve CNF SAT problems. It starts from the root node i.e. empty assignment and try to decide next undecided literal with positive value. If some assignment makes the formula unsatisfied, it will try the negative value. If it is still unsatisfied, it will backtrack to previous literal try to decide its next choice.

It continues to do so until it reaches the root node i.e. the formula is UNSAT else a partial assignment that satisfies the formula.

TL;DR, it does DFS on the search tree.

### 2.2 Backtrack with watchee

This underlying idea of this algorithm is same as Backtrack. But it uses a lazy data structure for faster computation.

### 2.3 Cyclic DPLL

This one uses the *forced literals* to decide first. If assignment of some literal gets decided for current partial assignment, choose it first. Rest of the algorithm is almost same as backtrack using watchee.

But the advantage of this algorithm is, it prunes more search space than first two algorithms. Because earlier pruning was depend on current partial assignment. DPLL is using future decisions along with current partial assignment to prune search space.

### 2.4 Clause learning

If  $C_1$  and  $C_2$  are two clauses in CNF and  $C$  is their resolvent, clause learning algorithm uses this resolvent to prune the search tree. DPLL used only forced decisions on one single literal. But clause learning actually uses the forced relations (newly formed resolvents) on multiple literals to prune the search space.

## 2.5 Random walk

This tries to guess a leaf of the search tree and using a heuristic it tries to search a satisfying assignment.

Note, this algorithm can not report UNSAT, because it never knows if it visited all the leaves of the search it. It only reports SAT when it finds a satisfying assignment.

Knuth discussed, doing random walk once may not give the assignment (because of local minima). But running the algorithm for several times may find the assignment.

## 3 Extended $N$ -queens problem

The conditions are same as the original  $N$ -queen problem such that place  $N$  queens in  $N \times N$  chess board such that no two queens are in attacking position with an extra condition such that there are no *skewed diagonals* such that no three queens are in a straight line.

We will use this problem for different  $N$  as our CNF benchmark.

### 3.1 Running times

**Backtrack** This is the native algorithm. Running time is small for smaller  $N$  but large for larger  $N$ .

**BT with watchee** It is faster than naive backtrack since it uses lazy data structure using watchee.

**Cyclic DPLL** It is supposed to be faster than previous two, but finding forced literals is costly and it searches for forced literals in each deciding step. So it is slower than previous two.

**Clause learning** This is very slow since finding the right literal should use some right heuristic. Otherwise it will go on creating new unnecessary resolvents which does not help pruning the tree, yet we have to account all those new clauses.

**Random walk** This algorithm runs really fast with respect other algorithms for SAT CNFs. But it of course, it can not report UNSAT, so for UNSAT CNF it will go through all the loops.

Following table compares times taken by implement codes for each algorithm for different board sizes.

Board size	5	8	10	12	14	16	18	20
backtrack	0.0	0.01	0.01	0.02	0.10	1.49	9.75	184.00
BT with watchee	0.0	0.01	0.02	0.04	0.09	1.00	6.63	96.23
Cyclic DPLL	0.0	0.11	0.25	7.46	54.81	25.05	-	-
Clause learning	0.9	12.91	9.86	-	-	-	-	-
Random walk	0.59	0.01	0.02	0.12	0.46	0.39	0.93	4.55

- '-' means it took really long time.
- Every board has a SAT assignment except board 5.