Algorithmic Behavior of DPLL on Random XOR-SAT and a NP-Complete Generalization of XOR-SAT

Presentation at the
Ontario Combinatorics Workshop
16 April 2005

Harold Connamacher

Department of Computer Science
University of Toronto

Overview

The Goal: Prove there is an exact threshold in the clause density of random XOR-SAT formulae (and a NP-complete generalization of XOR-SAT) that distinguishes instances on which DPLL using the unit clause heuristic (DPLL+UC) will require exponential time to find a satisfying assignment from instances on which DPLL+UC will take linear time, w.u.p.p.

k-SAT

- n variables, each may be assigned 0 or 1
- given variable x, a *literal* is either x or $\overline{x}$
- a *clause* is a set of k literals
  ex: $(x, \overline{y}, z)$

Question: Is there an assignment of the variables such that each clause has exactly one true literal?

If "yes", the formula is satisfiable (SAT).
If "no", the formula is unsatisfiable (UNSAT).

Complexity results:

$$k\text{-SAT} \in \begin{cases} \text{P} & \text{if } k = 2 \\ \text{NP-complete} & \text{if } k \geq 3 \end{cases}$$

Some Definitions

- All formulae considered will be *uniformly random* (u.r.)
- n: # variables
- m: # clauses
- $m = cn$: assume m is *linear* in n
- c is the *clause density*

DPLL+UC

At each step, DPLL:

- Assigns a variable v a value
- Removes satisfied clauses
- Removes v from unsatisfied clauses
- Recurses on the subformula
- Backtracks on a contradiction

Heuristic for choosing the next variable:

Unit Clause (UC):

- If there is a clause of size 1, choose it.
- Otherwise choose a variable at random

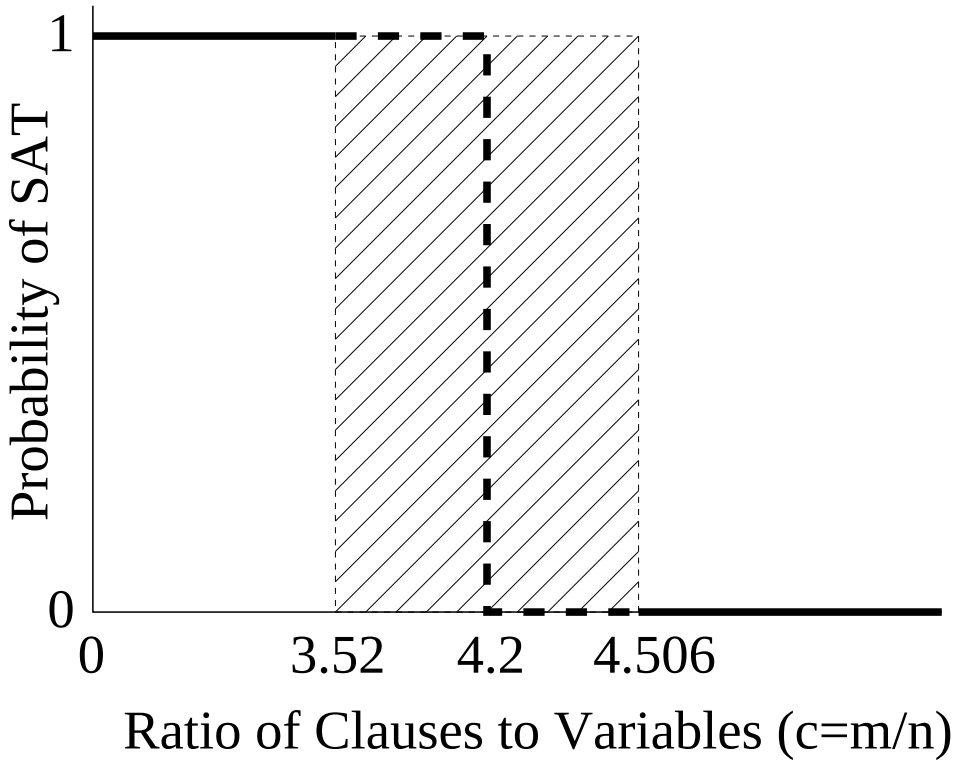
The Satisfiability Threshold Conjecture

Does there exist c_3^* s.t. a random 3-SAT formula on n variables and cn clauses is:

- a.s. SAT if $c < c_3^*$
- a.s. UNSAT if $c > c_3^*$?

2-SAT: $c_2^* = 1$ (Chvátal, Reed '92; Goerdt '96; Fernandez de la Vega '92)

k-SAT: Not known if c_k^* *exists*, $k > 2$



The $(2+p)$-SAT Model

A random SAT formula on a mixture of 2- and 3-clauses where p is the *proportion* of 3-clauses.

- n variables
- m clauses
- pm 3-clauses
- $(1-p)m$ 2-clauses

Def: Call a clause of size i an i-clause.

Conjecture: $(2+p)$-SAT has an exact satisfiability threshold for each value of p.

Known Results

The running time of DPLL+UC on 3-SAT is (w.u.p.p.):

- linear for $\leq \frac{8}{3}n$ clauses (Chao, Franco '86)
- exponential for $\geq 3.81n$ clauses (Achlioptas, Beame, Molloy '01)

Satisfiability threshold for $(2+p)$-SAT:
(Achlioptas, Kirousis, Kranakis, Krizanc '01)

- Exact threshold for $p \leq \frac{2}{5}$.
- $(1-\epsilon)n$ 2-clauses + λn 3-clauses is a.s.
  - SAT if $\lambda \leq \frac{2}{3}$ *for any* $\epsilon > 0$
  - UNSAT if $\lambda \geq 2.28$ *for some* $\epsilon > 0$

Conjecture: $(1-\epsilon)n$ 2-clauses + $\left(\frac{2}{3}+\delta\right)n$ 3-clauses is a.s. UNSAT (for any δ there is $\epsilon > 0$) .

XOR-SAT

- A variation of SAT using "exclusive-or".
- A clause is satisfied if exactly 1 or exactly 3 literals are true.
- For 3-XOR-SAT, there are 8 possible constraints, corresponding the two quasigroups (Latin squares) of size 2.

$(x, y, \overline{z})$
$(x, \overline{y}, z)$
$(\overline{x}, y, z)$
$(\overline{x}, \overline{y}, \overline{z})$

	0	1
0	0	1
1	1	0

(x, y, z)
$(x, \overline{y}, \overline{z})$
$(\overline{x}, y, \overline{z})$
$(\overline{x}, \overline{y}, z)$

	0	1
0	1	0
1	0	1

- XOR-SAT is in P because it can be solved by Gaussian elimination (modulo 2).

(k, d)-UE-CSP

- Constraints of size k
- Domain $\{0, \ldots, d-1\}$, $d \geq 2$
- Each constraint is *uniquely extendible*
  - For any setting of $k-1$ variables in a constraint, there is a unique value for the kth variable
- For $k = 3$, each constraint is a *quasigroup* of size d.

Complexity results:

$$(3, d)\text{-UE-CSP} \in \begin{cases} \text{P} & \text{if } d \leq 3 \\ \text{NP-complete} & \text{if } d \geq 4 \end{cases}$$

Threshold results: The exact satisfiability threshold of:

- $(3, d)$-UE-CSP is .917935 ...
- $(2, d)$-UE-CSP is $\frac{1}{2}$.

Relation to Graphs

We can model a formula as a hypergraph.

- Each variable is a vertex.
- Each clause is a hyperedge on its corresponding variables.

The Random Model

- Choose a hypergraph on n variables and m hyperedges, u.r.
- On each hyperedge, choose a quasigroup of size d u.r. for its constraint.

Main Theorems

Theorem: On a u.r. instance of $(3, d)$-UE-CSP with n variables and cn clauses, DPLL+UC will take (w.u.p.p.)

- linear time if $c \leq \frac{2}{3}$
- exponential time if $c > \frac{2}{3}$.

Theorem: A u.r. instance of UE-CSP with $(\frac{1}{2} - \epsilon)n$ 2-clauses and λn 3-clauses is

- w.u.p.p. SAT if $\lambda \leq \frac{1}{6}$ *for any* $\epsilon > 0$
- a.s. UNSAT if $\lambda > \frac{1}{6}$ *for some* $\epsilon > 0$

Proof Steps

Start with a u.r. random $(3, d)$-UE-CSP formula with n variables and cn clauses.

1. If $c \leq \frac{2}{3}$, DPLL+UC will find a satisfying assignment *without backtracking* (w.u.p.p.)
2. If $c > \frac{2}{3}$, DPLL+UC will produce a u.r. subformula with $n' = \alpha n$ variables, $(\frac{1}{2} - \epsilon)n'$ 2-clauses and $(\frac{1}{6} + \delta)n'$ 3-clauses (w.u.p.p.)
3. Such a formula is a.s. UNSAT.
4. DPLL will require $2^{\Omega(n')}$ steps to backtrack out of this UNSAT subformula (w.u.p.p.)

Step 1: Prove DPLL+UC will find a satisfying assignment without backtracking if $c \leq \frac{2}{3}$, w.u.p.p.

Technique: Trace UC (i.e. DPLL+UC without backtracking) with differential equations. (Achlioptas, et al. '01)

$$\begin{aligned}
\mathbf{E}[C_3(t+1) - C_3(t)] &= -\frac{3C_3(t)}{n-t} \\
\mathbf{E}[C_2(t+1) - C_2(t)] &= \frac{3C_3(x)}{n-t} - \frac{2C_2(x)}{n-t},
\end{aligned}$$

$C_i(t)$ is the number of i-clauses after t variables have been set.

Lemma: (Wormald '95) Solving the differential equations gives a.s.

$$\begin{aligned}
C_3(t) &= c_3(0)(1-t/n)^3 \cdot n + o(n) \\
C_2(t) &= (c_2(0) + 3c_3(0)(t/n))(1-t/n)^2 \cdot n + o(n)
\end{aligned}$$

where $c_i(0)$ is the initial density of i-clauses

Step 1: Prove DPLL+UC will find a satisfying assignment without backtracking if $c \leq \frac{2}{3}$, w.u.p.p.

Lemma: Until DPLL+UC backtracks, the subformula produced at each step of the algorithm is uniformly random.

Fact: DPLL only backtracks on a contradiction.

Lemma: If for all steps $0 \leq t \leq t_0$, a.s. $C_2(t) < \left(\frac{1}{2} - \epsilon\right)(n - t)$ then w.u.p.p. DPLL+UC will reach step t_0 *without producing a contradiction* and w.u.p.p. there will be no unit clause at step t_0, $t_0 = n - \gamma n$.

Pick γ small enough that the formula induced by the variables unassigned at step t_0 is "easy".

Step 1: Prove DPLL+UC will find a satisfying assignment without backtracking if $c \leq \frac{2}{3}$, w.u.p.p.

$$
\begin{aligned}
C_3(t) &= c_3(0)(1-t/n)^3 \cdot n + o(n) \\
C_2(t) &= (c_2(0) + 3c_3(0)(t/n))(1-t/n)^2 \cdot n + o(n) \\
C_2(t) &< \left(\frac{1}{2} - \epsilon\right)(n-t)
\end{aligned}
$$

Result 1: Set $c_3(0) = c$, $c_2(0) = 0$.

DPLL+UC does not produce a contradiction (w.u.p.p.) if $c \leq \frac{2}{3}$.

Result 2: Set $c_3(0) = cp$, $c_2(0) = c(1-p)$.

DPLL+UC does not produce a contradiction (w.u.p.p.) on a u.r. instance with $(\frac{1}{2} - \epsilon)n$ 2-clauses and βn 3-clauses if $\beta < \frac{1}{6}$.

Step 2: Prove $c > \frac{2}{3}$ implies DPLL+UC will produce a u.r. subformula with $n' = \alpha n$ variables, $(\frac{1}{2} - \epsilon)n'$ 2-clauses and $(\frac{1}{6} + \delta)n'$ 3-clauses, w.u.p.p.

$$
\begin{aligned}
C_3(t) &= c_3(0)(1 - t/n)^3 \cdot n + o(n) \\
C_2(t) &= (c_2(0) + 3c_3(0)(t/n))(1 - t/n)^2 \cdot n + o(n) \\
C_2(t) &< \left(\frac{1}{2} - \epsilon\right)(n - t) \\
C_3(t) &> \left(\frac{1}{6} + \delta\right)(n - t)
\end{aligned}
$$

Set $c_3(0) = c$, $c_2(0) = 0$.

If $c > \frac{2}{3}$, DPLL+UC will produce a formula with the desired clause densities without backtracking. Thus, the formula is u.r. random.

Step 3: Prove a formula on $(\frac{1}{2} - \epsilon)n$ 2-clauses and $(\frac{1}{6} + \delta)n$ 3-clauses is a.s. UNSAT.

First Moment Bound: Count the expected number of solutions of a u.r. formula with αn variables and βn clauses:

$$\mathbf{E}[\#\text{ solutions}] = d^{\alpha n} \left(\frac{1}{d}\right)^{\beta n}$$

If $\beta > \alpha$, $\mathbf{E}[\#\text{ solutions}] = \text{o}(1)$.

By Markov's Inequality, a formula is a.s. UNSAT if $\beta > \alpha$.

Goal: Find a u.r. subformula with more clauses than variables.

Step 3: Prove a formula on $(\frac{1}{2} - \epsilon)n$ 2-clauses and $(\frac{1}{6} + \delta)n$ 3-clauses is a.s. UNSAT.

- A random formula with a linear number of clauses has many variables of degree < 2.
- A clause with a variable of degree 1 can always be satisfied.
- Variables of degree 0 are trivially satisfiable.

Trim the variables of degree < 2 from the formula to get the *2-core*.

2-Core: The unique, maximal subformula with minimal degree 2.

Step 3: Prove a formula on $(\frac{1}{2}-\epsilon)n$ 2-clauses and $(\frac{1}{6}+\delta)n$ 3-clauses is a.s. UNSAT.

Use a *Branching Process* to compute the size of a 2-core. (Łuczak '91, Molloy '04)

Idea: The probability a vertex is trimmed when reducing to the 2-core is the probability all but one child is trimmed.

Theorem: A u.r. formula with n variables, $c_2 n$ 2-clauses, $c_3 n$ 3-clauses a.s. has a 2-core with $\alpha(c_2, c_3)$ variables, $\beta_2(c_2, c_3)$ 2-clauses and $\beta_3(c_2, c_3)$ 3-clauses where:

$$\begin{aligned} \alpha(c_2, c_3) &= 1 - e^{-x} - xe^{-x} \\ \beta_2(c_2, c_3) &= c_2(1 - e^{-x})^2 \\ \beta_3(c_2, c_3) &= c_3(1 - e^{-x})^3 \end{aligned}$$

where x is the largest solution to

$$x = (1 - e^{-x})^2 3c_3 + (1 - e^{-x})2c_2.$$

Lemma: $\alpha(c_2, c_3) < \beta_2(c_2, c_3) + \beta_3(c_2, c_3)$ if $c_2 = \frac{1}{2} - \epsilon$ and $c_3 = \frac{1}{6} + \delta$.

Step 4: Prove DPLL will require exponential time to backtrack out of an unsatisfiable u.r. formula F with $\left(\frac{1}{2} - \epsilon\right) n$ 2-clauses and Δn 3-clauses, w.u.p.p.

The running time of DPLL on an unsatisfied formula F can be bounded by the *resolution complexity* of F, the length of the shortest resolution refutation of F.

- Resolution initially defined for CNF boolean formulae
- Can adapt resolution to work on CSPs (Mitchell '02)

Step 4: Prove an unsatisfiable u.r. formula F with $\left(\frac{1}{2}-\epsilon\right)n$ 2-clauses and Δn 3-clauses has exponential resolution complexity, w.u.p.p.

Exponential resolution complexity is a consequence of the following three properties holding a.s. for some $\alpha, \zeta > 0$. (Ben-Sasson, Wigderson '01; Mitchell '02; Molloy, Salavatipour '03)

(a) Every subproblem on at most αn variables is satisfiable.

(b) Every subproblem on v variables, $\frac{1}{2}\alpha n \leq v \leq \alpha n$, has at least ζn variables of degree ≤ 1.

(c) The problem is *extendible*

If (a)-(c) hold, DPLL will require $2^{\Omega(n)}$ steps to show the subformula is UNSAT.

Prove:

(a) *Every subproblem on at most αn variables is satisfiable.*

(b) *Every subproblem on v variables, $\frac{1}{2}\alpha n \leq v \leq \alpha n$, has at least ζn variables of degree ≤ 1.*

- Find a configuration that exists in every formula that has few variables of degree ≤ 1.
  - *Note:* A minimal unsatisfiable formula must contain this configuration.
- Prove there a.s. can not be such a configuration on $\leq \alpha n$ variables. (Markov's Inequality)
- Prove that if there is no such configuration on $\frac{1}{2}\alpha n \leq v \leq \alpha n$ variables then there is a linear number of variables of degree <= 1. (Chebyshev's Inequality)