Popularity and Similarity in SAT

Jordi Levy
IIIA, CSIC, Barcelona, Spain

Joint work with
Carlos Ansótegui, Maria Luisa Bonet and Jesús Giráldez

MACFANG'17, Barcelona

SAT

Set of variables $\Sigma = \{a, b, c, \dots\}$
Literals: variables either affirmed a or negated $\neg a$
Clause: disjunction of literals $a \vee \neg b \vee \neg c$
Formula: conjunction of clauses $\{a \vee b \ , \ \neg a \vee c\}$
SAT: find an assignment $I : \Sigma \to \{0, 1\}$ such that
at least one literal of **every** clause is set to 1.
Note that $I(a) + I(\neg a) = 1$

Set of variables $\Sigma = \{a, b, c, \dots\}$
Literals: variables either affirmed a or negated $\neg a$
Clause: disjunction of literals $a \vee \neg b \vee \neg c$
Formula: conjunction of clauses $\{a \vee b\ ,\ \neg a \vee c\}$
SAT: find an assignment $I : \Sigma \to \{0, 1\}$ such that
at least one literal of **every** clause is set to 1.
Note that $I(a) + I(\neg a) = 1$

$$
\begin{array}{l}
a \vee \neg b \vee \neg c \\
\neg a \vee b \\
a \vee c \\
\neg a \vee c
\end{array}
$$

SAT

Set of variables $\Sigma = \{a, b, c, \dots\}$
Literals: variables either affirmed a or negated $\neg a$
Clause: disjunction of literals $a \vee \neg b \vee \neg c$
Formula: conjunction of clauses $\{a \vee b\ ,\ \neg a \vee c\}$
SAT: find an assignment $I : \Sigma \to \{0, 1\}$ such that
at least one literal of **every** clause is set to 1.
Note that $I(a) + I(\neg a) = 1$

$$a \vee \neg b \vee \neg c$$
$$\neg a \vee b$$
$$a \vee c$$
$$\neg a \vee c$$

$$a \to 0$$
$$b \to 0$$
$$c \to 1$$

SAT

Set of variables $\Sigma = \{a, b, c, \dots\}$
Literals: variables either affirmed a or negated $\neg a$
Clause: disjunction of literals $a \vee \neg b \vee \neg c$
Formula: conjunction of clauses $\{a \vee b \;,\; \neg a \vee c\}$
SAT: find an assignment $I : \Sigma \to \{0, 1\}$ such that
at least one literal of **every** clause is set to 1.
Note that $I(a) + I(\neg a) = 1$

$$a \vee \boxed{\neg b} \vee \neg c$$
$$\boxed{\neg a} \vee b$$
$$a \vee \boxed{\mathrm{c}}$$
$$\boxed{\neg a} \vee \boxed{\mathrm{c}}$$

$a \to 0$
$b \to 0$
$c \to 1$

Solving SAT in the 60's

$$a \vee \neg b \vee \neg c$$
$$\neg a \vee b$$
$$a \vee c$$
$$\neg a \vee c$$

$a \vee \neg b \vee \neg c$

$\neg a \vee b$

$a \vee c$

$\neg a \vee c$

$c = 0$ decision

$a \vee \neg b \vee \neg c$

$\neg a \vee b$

$a \vee c$

$\neg a \vee c$

$$a \vee \neg b \vee \neg c$$
$$\neg a \vee b$$
$$a \vee c$$
$$\neg a \vee c$$

$c = 0$ decision

$$a \vee \neg b \vee \neg c$$
$$\neg a \vee b$$
$$a \vee c$$
$$\neg a \vee c$$

$\downarrow$ $a = 1$ unit propagation

$$a \vee \neg b \vee \neg c$$
$$\neg a \vee b$$
$$a \vee c$$
$\neg a \vee c \leftarrow$ Empty clause

Solving SAT in the 60's

$$a \vee \neg b \vee \neg c$$
$$\neg a \vee b$$
$$a \vee c$$
$$\neg a \vee c$$

$c = 0$ decision | $c = 1$ backtracking

$c = 0$	$c = 1$
$a \vee \neg b \vee \neg c$	$a \vee \neg b \vee \neg c$
$\neg a \vee b$	$\neg a \vee b$
$a \vee c$	$a \vee c$
$\neg a \vee c$	$\neg a \vee c$

$\downarrow$ $a = 1$ unit propagation

$$a \vee \neg b \vee \neg c$$
$$\neg a \vee b$$
$$a \vee c$$
$$\neg a \vee c$$

Solving SAT in the 60's

$a \vee \neg b \vee \neg c$

$\neg a \vee b$

$a \vee c$

$\neg a \vee c$

$c = 0$ decision

$a \vee \neg b \vee \neg c$

$\neg a \vee b$

$a \vee c$

$\neg a \vee c$

$\downarrow$ $a = 1$ unit propagation

$a \vee \neg b \vee \neg c$

$\neg a \vee b$

$a \vee c$

$\neg a \vee c$

$c = 1$ backtracking

$a \vee \neg b \vee \neg c$

$\neg a \vee b$

$a \vee c$

$\neg a \vee c$

$\downarrow$ $a = 0$ decision

$a \vee \neg b \vee \neg c$

$\neg a \vee b$

$a \vee c$

$\neg a \vee c$

- SAT is NP-complete (whether P=NP is one of the Millennium Prize Problems)
- P=NP iff there exists a proof-system that can prove every tautology in polynomial size

  Resolution:

  $$\frac{\begin{array}{c} x \vee A \\ \neg x \vee B \end{array}}{A \vee B}$$

  The pigeon-hole principle require exponential resolution proofs
- Random SAT formulas require exponential tree-like refutations

SAT in Theory

- SAT is NP-complete (whether P=NP is one of the Millennium Prize Problems)
- P=NP iff there exists a proof-system that can prove every tautology in polynomial size

  Resolution:

$$\frac{\begin{array}{c} x \vee A \\ \neg x \vee B \end{array}}{A \vee B}$$

  The pigeon-hole principle require exponential resolution proofs

- Random SAT formulas require exponential tree-like refutations

SAT in Theory

- SAT is NP-complete (whether P=NP is one of the Millennium Prize Problems)
- P=NP iff there exists a proof-system that can prove every tautology in polynomial size
  Resolution:

$$\frac{\begin{array}{c} x \vee A \\ \neg x \vee B \end{array}}{A \vee B}$$

  The pigeon-hole principle require exponential resolution proofs
- Random SAT formulas require exponential tree-like refutations

SAT in Practice

- Despite negative theoretical results...
  SAT solvers are able to solve industrial instances with thousands of variables and millions of clauses in few seconds
  ...not so random instances
- Every year we celebrate a SAT solver competition and we have hundreds of real-world SAT instances coming from industrial applications:

SAT in Practice

- Despite negative theoretical results...
  SAT solvers are able to solve industrial instances with thousands of variables and millions of clauses in few seconds
  ...not so random instances
- Every year we celebrate a SAT solver competition and we have hundreds of real-world SAT instances coming from industrial applications:
  - Hardware verification
  - Software verification
  - Planning
  - ...

SAT in Practice

- Despite negative theoretical results...
  SAT solvers are able to solve industrial instances with thousands of variables and millions of clauses in few seconds
  ...not so random instances
- Every year we celebrate a SAT solver competition and we have hundreds of real-world SAT instances coming from industrial applications:

Objectives

- Study the structural properties of real-world SAT instances
- Propose new models of random formulas
- Exploit this knowledge to improve SAT solvers specialized in those kind of formulas

"Modern" SAT Solvers at a Glance

- Learning: Every time we find a conflict, generate a learnt clause. This clause is redundant. We try to detect the conflict with less decisions next time
- VSID heuristics: Decide on those variables more frequently involved in recent conflicts
- Clause deletion: From time to time, remove some learnt clauses according to their "quality"
- Restarts: From time to time, forget the partially computed assignment keeping only learnt clauses

"Modern" SAT Solvers at a Glance

- Learning: Every time we find a conflict, generate a learnt clause. This clause is redundant. We try to detect the conflict with less decisions next time
- VSID heuristics: Decide on those variables more frequently involved in recent conflicts
- Clause deletion: From time to time, remove some learnt clauses according to their "quality"
- Restarts: From time to time, forget the partially computed assignment keeping only learnt clauses

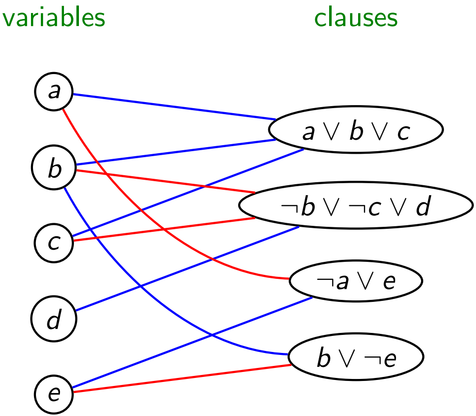
"Modern" SAT Solvers at a Glance

- Learning: Every time we find a conflict, generate a learnt clause. This clause is redundant. We try to detect the conflict with less decisions next time
- VSID heuristics: Decide on those variables more frequently involved in recent conflicts
- Clause deletion: From time to time, remove some learnt clauses according to their "quality"
- Restarts: From time to time, forget the partially computed assignment keeping only learnt clauses

"Modern" SAT Solvers at a Glance

- Learning: Every time we find a conflict, generate a learnt clause. This clause is redundant. We try to detect the conflict with less decisions next time
- VSID heuristics: Decide on those variables more frequently involved in recent conflicts
- Clause deletion: From time to time, remove some learnt clauses according to their "quality"
- Restarts: From time to time, forget the partially computed assignment keeping only learnt clauses

SAT Instances as Networks (or Graphs)



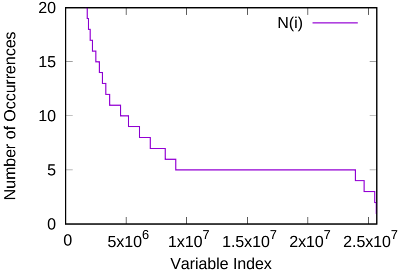
Analysis of Industrial Formulas

Consider all variables of the SAT'08 Competition and sort them

$N(i)$ = number of occurrences of i-th most frequent variable

Most have 5 occurrences, although the average is 13.6

A few have millions of occurrences!!!



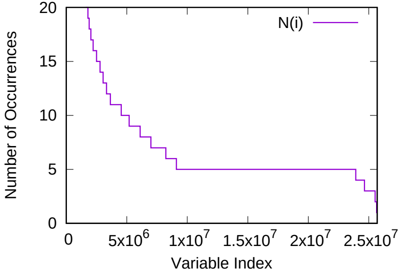
Analysis of Industrial Formulas

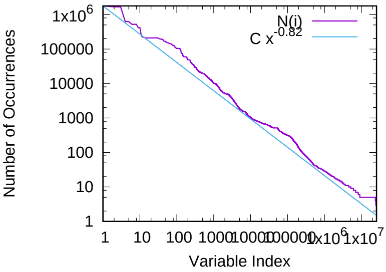
Consider all variables of the SAT'08 Competition and sort them

$N(i)$ = number of occurrences of i-th most frequent variable

Most have 5 occurrences, although the average is 13.6

A few have millions of occurrences!!!



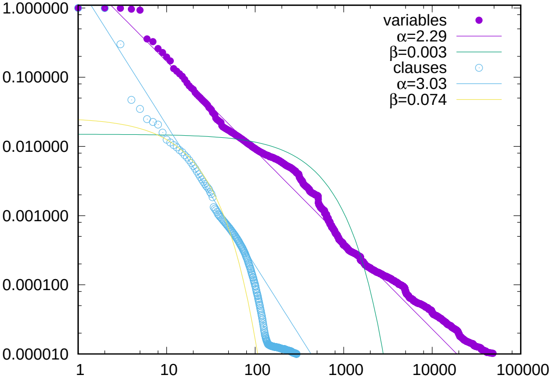


Expected number of occurrences of i-th most frequent variable and degree distribution

$$N(i) \sim i^{-0.82} \qquad P(k) \sim k^{-2.22}$$

Analysis of Industrial Formulas

Seen as a graph, industrial SAT formulas are scale-free on variables and on clauses degree distributions

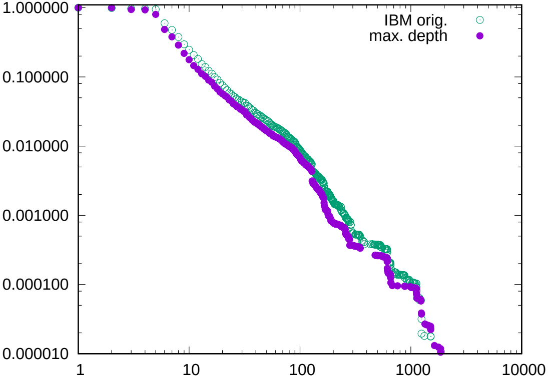


“Destroying” (Solving) SAT Formulas

- Instantiating 5% (even 1%) of most frequent variables randomly make most formulas unsatisfiable.
  SAT solvers do not instantiate so many
- Scale-free structure is not affected
  Notice that after instantiating we remove clauses and variables

"Destroying" (Solving) SAT Formulas

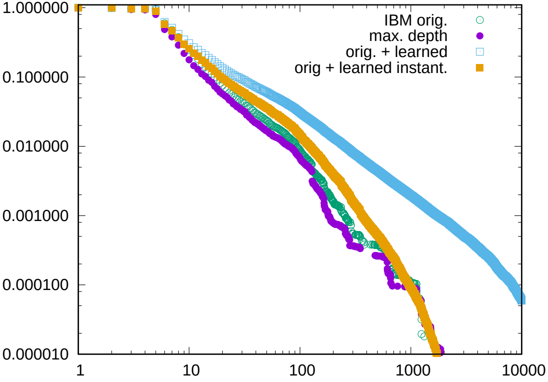
The effect of SAT solvers on an industrial formula:



Instantiation does not "destroy" scale-free structure.

“Destroying” (Solving) SAT Formulas

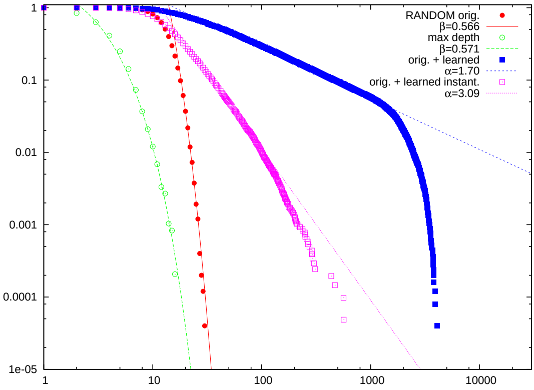
The effect of SAT solvers on an industrial formula:



Instantiation does not “destroy” scale-free structure.
Learning seems to reinforce scale-free structure.

"Destroying" (Solving) SAT Formulas

The effect on a random (Erdös-Rényi) formula:



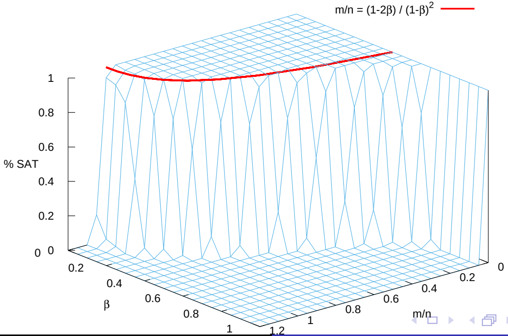
Learning acts as preferential attachment!

Random Scale-Free Formulas

- Theoretical SAT researchers: We were wrong! Classical random SAT formulas (Erdös-Rényi model) are not appropriate, use (simple) random scale-free formulas:
  Sample variable x_i with $P(x_i) \sim i^{-\beta}$, to get $P(k) \sim k^{1+1/\beta}$
- Random formulas exhibit a linear SAT-UNSAT phase transition
  In scale-free formulas (for big β) this is sub-linear
  Proofs are small!
- Since formulas are scale-free, the best variable branching heuristics is assigning most frequent variables
  We tried it in the past, BUT we have better heuristics
- Scale-free formulas are too easy on practice
- Fortunately... I met Dmitri (thanks Dmitri)... and he told me about similarity

Random Scale-Free Formulas

- Theoretical SAT researchers: We were wrong! Classical random SAT formulas (Erdös-Rényi model) are not appropriate, use (simple) random scale-free formulas:
  Sample variable x_i with $P(x_i) \sim i^{-\beta}$, to get $P(k) \sim k^{1+1/\beta}$
- Random formulas exhibit a linear SAT-UNSAT phase transition
  In scale-free formulas (for big β) this is sub-linear
  Proofs are small!



Random Scale-Free Formulas

- Theoretical SAT researchers: We were wrong! Classical random SAT formulas (Erdös-Rényi model) are not appropriate, use (simple) random scale-free formulas:
  Sample variable x_i with $P(x_i) \sim i^{-\beta}$, to get $P(k) \sim k^{1+1/\beta}$
- Random formulas exhibit a linear SAT-UNSAT phase transition
  In scale-free formulas (for big β) this is sub-linear
  Proofs are small!
- Since formulas are scale-free, the best variable branching heuristics is assigning most frequent variables
  We tried it in the past, BUT we have better heuristics
- Scale-free formulas are too easy on practice
- Fortunately... I met Dmitri (thanks Dmitri)... and he told me about similarity

Random Scale-Free Formulas

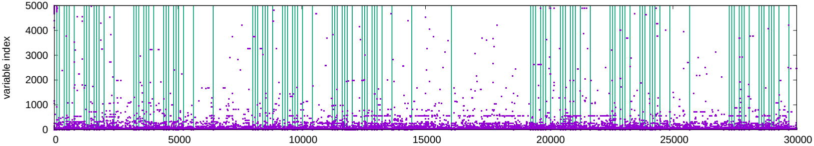
- Theoretical SAT researchers: We were wrong! Classical random SAT formulas (Erdös-Rényi model) are not appropriate, use (simple) random scale-free formulas:
  Sample variable x_i with $P(x_i) \sim i^{-\beta}$, to get $P(k) \sim k^{1+1/\beta}$
- Random formulas exhibit a linear SAT-UNSAT phase transition
  In scale-free formulas (for big β) this is sub-linear
  Proofs are small!
- Since formulas are scale-free, the best variable branching heuristics is assigning most frequent variables
  We tried it in the past, BUT we have better heuristics
- Scale-free formulas are too easy on practice
- Fortunately... I met Dmitri (thanks Dmitri)... and he told me about similarity

Random Scale-Free Formulas

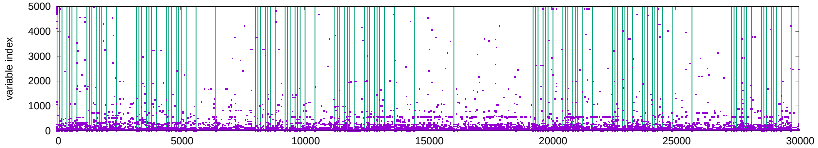
- Theoretical SAT researchers: We were wrong! Classical random SAT formulas (Erdös-Rényi model) are not appropriate, use (simple) random scale-free formulas:
  Sample variable x_i with $P(x_i) \sim i^{-\beta}$, to get $P(k) \sim k^{1+1/\beta}$
- Random formulas exhibit a linear SAT-UNSAT phase transition
  In scale-free formulas (for big β) this is sub-linear
  Proofs are small!
- Since formulas are scale-free, the best variable branching heuristics is assigning most frequent variables
  We tried it in the past, BUT we have better heuristics
- Scale-free formulas are too easy on practice
- Fortunately... I met Dmitri (thanks Dmitri)... and he told me about similarity

Random Formulas with Popularity and Similarity

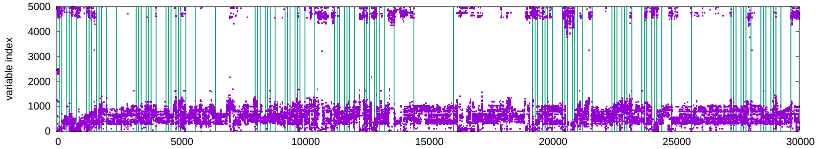
$\beta = 0.8$ $T = 1.5$ (vars. ordered by popularity)



$\beta = 0.8$ $T = 1.5$ (vars. ordered by popularity)

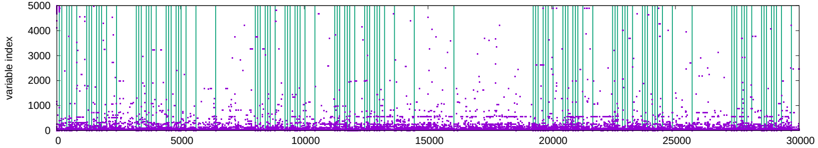


$\beta = 0.1$ $T = 0.75$ (vars. ordered by similarity)

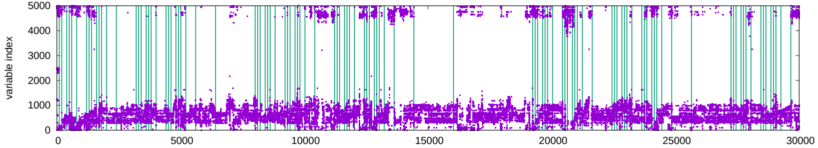


Random Formulas with Popularity and Similarity

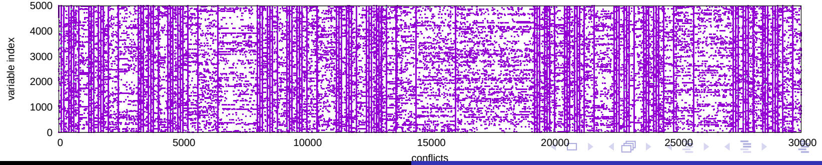
$\beta = 0.8$ $T = 1.5$ (vars. ordered by popularity)



$\beta = 0.1$ $T = 0.75$ (vars. ordered by similarity)



$T = 100$ ($\beta = 0.1$ and vars. ordered by similarity)



Conclusions and Further Work

- An equilibrium between the forces of popularity and similarity defines the structure of industrial SAT instances
- Modern SAT solvers exploit both properties
- Explicit computation of variable coordinates may lead to better branching heuristics
- Analysis of the temperature of formulas may characterize their difficulty