Journal on Satisfiability, Boolean Modeling and Computation 1 (2006) 123-141

Research Note

Impurity: Another Phase Transition of SAT

Eliezer L. Lozinskii lozinski@cs.huji.ac.il
School of Computer Science and Engineering
The Hebrew University, Jerusalem 91904, Israel

Abstract

It is well known that satisfiability of random sets of propositional clauses undergoes phase transition while the clause-to-variable ratio of the sets increases. We introduce another parameter of sets of clauses, *impurity*, and show that the satisfiability undergoes a phase transition as a function of impurity. This phenomenon supports a conjecture that various properties (such as random graph connectivity, perfect integer partition) exhibit phase transition under control of several different syntactic parameters.

Keywords: *SAT, phase transition, impurity*

Submitted January 2006; revised August 2006; published December 2006

1. Introduction: Propositional satisfiability

Various objects of different nature undergo a radical change of their qualitative properties while their quantitative parameters change values. If an evolution of objects inverts their property, it is said that the property undergoes a *phase transition* [17, 16, 19]. A remarkable feature of phase transition is that it happens as a rapid change of the corresponding property while the parameters controlling this change vary in a relatively narrow range of values, and the narrower this range, the *sharper* the phase transition.

Consider a set of n propositional *variables* $\{v_1, \ldots, v_n\}$, and a set of $2n$ *literals* such that each literal l is either a variable v_i or its negation $\neg v_i$. A *CNF* propositional formula F is a conjunction of a set of *clauses* such that each clause is a disjunction of a subset of literals. If each variable is assigned a value of *true* or *false*, then a clause is true if it contains a true literal, and F is true if all its clauses are true. An assignment of truth values to the variables making F true is a *model* of F, and if a model exists then F is *satisfiable* (otherwise, *unsatisfiable*).

The problem *SAT* of checking satisfiability of a CNF propositional formula is one of most important and difficult computational problems. It is the classic *NP-complete* problem [7] providing the basis for the theory of computational complexity. Many hard problems can be encoded as SAT. SAT is also a major part of many proof and deduction procedures, as a formula ϕ is a logical consequence of F iff $F \wedge \neg\phi$ is unsatisfiable. No wonder that SAT became the topic of numerous theoretical and experimental studies.

Let $\mathbf{F}(n, m, k)$ denote a set of all CNF propositional formulae $F(n, m, k)$ over n variables containing m clauses each with k literals such that each clause is chosen randomly and uniformly out of the set of all $\binom{n}{k}2^k$ possible k-literal clauses. Then k-SAT is the problem of checking satisfiability of $F(n,m,k)$. Let $psat(n,m,k)$ denote the probability that a formula $F(n,m,k) \in \mathbf{F}(n,m,k)$ is satisfiable.

Consider satisfiability of $F(n,m,k)$ in the process of its growth by acquiring more and more clauses. While $m < 2^k$ all the formulae of $\mathbf{F}(n,m,k)$ are satisfiable, so, $psat(n, m < 2^k, k) = 1$. As m grows some formulae of $\mathbf{F}(n,m,k)$ become unsatisfiable, and $psat(n,m,k)$ decreases becoming eventually zero, since a set of all $\binom{n}{k}2^k$ possible clauses is obviously unsatisfiable. Many theoretical and experimental studies indicated that for all n, k, $psat(n,m,k)$ is a monotone decreasing function of the clause-to-variable ratio $r = m/n$, and there exists a clauses-to-variables ratio r_k depending on k such that in the vicinity of r_k the value of $psat(n,m,k)$ changes rapidly from 1 to 0, so, the property of satisfiability of $F(n,m,k)$ undergoes a phase transition.

For $k = 2$, Chvátal and Reed [6], and Goerdt [15] proved that $r_2 = 1$, and the satisfiability of $F(n,m,2)$ undergoes a *sharp* phase transition:

$$\lim_{n\to\infty} psat(n,m,2) = \begin{cases} 1 & \text{if} \quad m/n < 1 \\ 0 & \text{if} \quad m/n > 1 \end{cases} \tag{1}$$

For $k \geq 2$ Friedgut [13] proved that there exists a sequence $r_k(n)$ such that for all $\epsilon > 0$,

$$\lim_{n\to\infty} psat(n,m,k) = \begin{cases} 1 & \text{if} \quad m/n = r_k(n) - \epsilon \\ 0 & \text{if} \quad m/n = r_k(n) + \epsilon \end{cases} \tag{2}$$

For $k \geq 3$ it is unknown yet whether the sequence $r_k(n)$ converges. However, the value of $r_3(n)$ is bounded by 3.26 from below [1] and 4.506 from above [10]. Achlioptas and Moore [2] proved for $k \geq 2$ that $r_k > 2^{k-1}\ln 2 - d_k$, where $d_k \to (1 + \ln 2)/2$. Numerous computer experiments provide an evidence that $r_3 \approx 4.25$ [21, 8] $r_4 \approx 9.76$ [14].

So, clauses-to-variables ratio controls satisfiability of sets of clauses in the rather remarkable way. However, is this ratio the only control parameter of this important property? Is there another syntactic parameter of a set of clauses that affects its satisfiability? The next sections provide an affirmative answer to the latter question.

2. Impurity

The clause-to-variable ratio r of a set of clauses provides an information regarding its satisfiability as $psat(r)$[1.] determines the uncertainty associated with the outcome of a satisfiability test. This uncertainty reaches its maximum in a close vicinity of r_k where $psat = 0.5$. We are looking for another measure μ of a set of clauses (different from r, and also effectively computable) that would provide an additional information of its satisfiability by reducing the uncertainty. For every value of r (in a wide range of values[2.]) there are both satisfiable and unsatisfiable sets, so, if the probability $psat(r, \mu)$ that a set with measures r, μ is satisfiable is monotone in μ, then $psat(r, \mu)$ may undergo a phase transition under control of μ.

1. We use $psat(r)$ or $psat$ instead of $psat(n, rn, k)$ if this abbreviation does not cause ambiguity.
2. Any set with $m > 2^k$ and r as small as $1/k$ may contain an unsatisfiable subset of 2^k k-literal clauses over k variables. At the other extreme of r, there are satisfiable sets of $\binom{n}{k}(2^k - 1)$ clauses containing no clause with all negated (or all unnegated) literals.

Let $pos(v), neg(v)$ denote, respectively, the number of unnegated and negated occurrences of a variable v in a set of clauses F. If v occurs in F either only unnegated or only negated (i.e., $pos(v) = 0$ or $neg(v) = 0$) then v is a *pure* variable in F. An important observation implemented in many Davis-Putnam-Logemann-Loveland style algorithms [9] is that satisfiability of F does not change if all pure unnegated and negated variables in F are assigned the values *true* and *false*, respectively. That is, F is satisfiable iff F' is so, where F' is a result of deleting from F all clauses containing pure variables. So, pure variables don't affect satisfiability of F, and even allow its simplification as F' is a subset of F. In particular, if all variables of F are pure then F is trivially satisfiable.

If both $pos(v)$ and $neg(v)$ are non-zero in F, let v be called *impure variable*, and denote $max(v) = \max(pos(v), neg(v))$, $min(v) = \min(pos(v), neg(v))$. Any assignment of truth value to v removes from F at most $max(v)$ clauses, but deletes literals of v from at least $min(v)$ clauses making them "shorter" (let the length of a clause be the number of its literals). Deleting a literal from a clause reduces the probability that an assignment to its variables satisfies the clause since this probability is monotone in the clause length.

Let $imp(v) = min(v)/max(v)$ be the *impurity* of v, and $imp(F)$ stand for the impurity of F that is the average impurity of its variable:

$$imp(F) = \frac{1}{n} \sum_{i=1}^{n} min(v_i)/max(v_i) \tag{3}$$

$$0 \leq imp(F) \leq 1. \tag{4}$$

So, for all n, r, k, all sets with zero impurity are satisfiable, $psat(n, r, k, imp = 0) = 1$, and the argument above suggests that this probability decreases while imp increases from 0 to 1. The following sections show that $psat$ undergoes phase transition as a function of imp.

3. Experiments: Generating sets with a given value of impurity

To study sets of clauses with different impurity we have implemented three different models of generating random sets of m k-literal clauses over n variables with $0 \leq imp < 1.0$. These models differ in characteristics of the produced sets of clauses, and in several aspects complement each other.

Experiments were run with programs that generate random sets of clauses (corresponding to each model), check their satisfiability, and count the number of truth value assignments to variables required for the check (the latter measure is discussed in Section 5). About ten million 3-literal sets were generated with the following parameters: $n = 30 - 200$; $r = 3.0 - 1000$; $pul = 0.05 - 0.50$, $imp = 0.0 - 1.0$.

3.1 Model P

Almost all experimental and theoretical studies of propositional satisfiability consider sets of clauses generated by the following procedure. To produce a new literal choose randomly and uniformly one of n variables, then insert an unnegated literal of the variable with a constant probability pul; insert k literals of different variables into each clause; generate m clauses for a set. Sets generated by this procedure with $pul = 0.5$ conform to those produced by choosing randomly and uniformly m clauses out of the set of all different k-literal clauses over n variables.

The effect of relative number of negated and unnegated literals in a set on characteristics of the set has been studied recently. Bayardo and Schrag [3] found that higher structural regularity of sets increases the mean difficulty of SAT. In order to produce very hard SAT instances they generated sets of clauses in such a way that all variables occur in the set almost the same number of times, and there is almost the same number of negated and unnegated literals.

Dubois, Boufkhad and Mandler [10] considered syntactic structure of a typical random 3-SAT formula produced with $pul = 0.5$. They defined an asymptotic distribution of the number of occurrences of variables in a formula, and of the number of unnegated literals; then by analyzing this distribution derived an improved upper bound of the satisfiability threshold, equal 4.506.

Sinopalnikov [22] studied *skewed* random k-SAT: sets generated with $pul < 0.5$. His study showed that the smaller pul, the larger the threshold value of r at which $psat$ undergoes phase transition.

Model P produces sets of clauses with a constant pul. The value of pul determines distribution of impurity of the sets of clauses generated by the model. For a constant value of r, sets with different impurity can be generated by varying pul between 0 and 0.5. Sets generated with $pul = 0$ have $imp = 0$. As pul grows, the difference between the number of negated and unnegated literals of each variable decreases, so the mean impurity of the sets increases.

Figure 1 displays probability of satisfiability $psat$ as a function of both clause-to-variable ratio r and impurity imp. For all fixed values of $r \geq 3.8$ ($n = 100$) the probability of satisfiability exhibits the typical behaviour of phase transition as a function of imp: it starts with $psat = 1.0$ at $imp = 0$, while imp grows $psat$ decreases slowly to a value $1 - \epsilon$ for a small $\epsilon > 0$, then steeply drops to $psat = \epsilon$ within a narrow range of imp called *scaling window* [4], and then continues to decrease. The two curves marked with triangles and squares correspond to sets of clauses generated with pul of 0.5 and 0.3, respectively. Figure 1 displays the projection of these curves on the (r, imp)-plane (thin curves), and shows that the value of average impurity of sets of clauses depends on both pul and r, growing rather slowly with r.

Table 1: Impurity of sets of clauses generated with constant $pul = 0.5$
$mean$ - average impurity of sets
$\sigma(set)$ - standard deviation of impurity of sets

imp	r					
	3.00	4.26	6.00	8.00	10.0	20.0
$mean$	.607	.657	.702	.736	.760	.821
$\sigma(set)$	.024	.022	.019	.017	.016	.012

Figure 1 and Table 1 (that presents the mean and standard deviation of impurity of sets produced with the maximum value of $pul = 0.5$) show that Model P does not produce sets with large value of impurity. Besides, if the standard deviation of impurity of sets, $\sigma(set)$, could be made smaller, the larger proportion of the generated sets would have the target value of impurity.

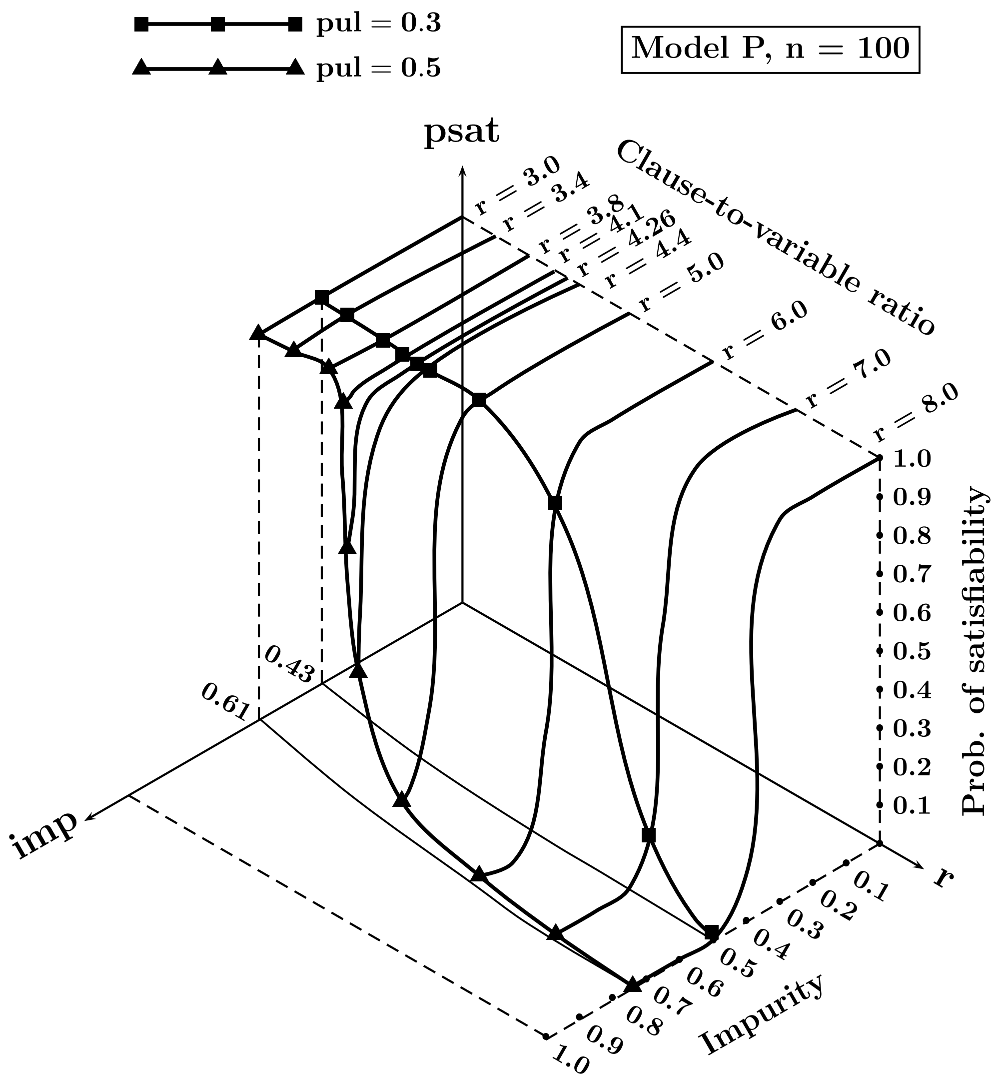

Figure 1: Phase transition of *psat(r, imp)* of sets of clauses generated with constant *pul* (model P)

3.2 Model S

To generate sets with impurity approaching 1.0, and a small variance, Model S employs the following algorithm.

Given a target impurity α, the algorithm chooses for each new literal a variable v randomly and uniformly, then before inserting a new literal of the variable in a clause the algorithm calculates the current impurity $\tilde{\alpha}$ of the set under generation, and inserts v or $\neg v$ to minimize the new current difference $\Delta = |\tilde{\alpha} - \alpha|$.

Let l, $\tilde{n}_l$, $\tilde{\alpha}_l$ denote, respectively, the number of literals, the number of different variables currently appearing in the set, and its current impurity. Without loss of generality[3], the algorithm maintains for each variable w, $neg(w) \leq pos(w)$ in the following way. If a variable v chosen for a new literal does not yet appear in the set, then a literal v is inserted such that $pos(v) = 1$, $neg(v) = 0$, $imp(v) = 0$, $\tilde{n}_{l+1} = \tilde{n}_l + 1$. At any step of the algorithm, if for any variable w $neg(w) < pos(w)$, then any next assignment to w keeps $neg(w) \leq pos(w)$, if $neg(w) = pos(w)$, then the next assignment will be w.

So, after the first assignment to any variable v,

$$\tilde{\alpha}_{l+1} = \tilde{\alpha}_l \left(1 - \frac{1}{\tilde{n}_l + 1}\right) \tag{5}$$

If v already appears in the set then there are two possibilities. If the new literal is v then

$$\tilde{\alpha}^{v}_{l+1} = \tilde{\alpha}_l - \frac{neg(v)}{\tilde{n}_l \; pos(v) \; (pos(v) + 1)} \tag{6}$$

If the new literal is $\neg v$ then

$$\tilde{\alpha}^{\neg v}_{l+1} = \tilde{\alpha}_l + \frac{1}{\tilde{n}_l \; pos(v)} \tag{7}$$

So,

$$\Delta_{l+1} = \min\{|\tilde{\alpha}^{v}_{l+1} - \alpha|, |\tilde{\alpha}^{\neg v}_{l+1} - \alpha|\} \tag{8}$$

It follows that after an insertion of each new literal of a variable v

$$\Delta_{l+1} < \Delta_l \quad \text{or} \quad \Delta_{l+1} < \tfrac{2}{\tilde{n}_{l+1} \; lit(v)},$$

where $lit(v)$ is the number of occurrences of v currently in the set.

The number of literals in a set of m k-literal clauses is nrk, the average number of occurrences of a variable in the set is rk, at the start of a set generating $\Delta = \alpha$. So, the larger nrk and smaller α the closer the impurity of a set generated by the algorithm to the target value α.

Table 2 presents percentage of sets generated by the algorithm with a target impurity $\alpha = 0.1 - 0.9$ for which $\Delta \leq 0.005$.

The maximum impurity of 1.0 is practically unattainable, as the impurity of any variable with an odd number of occurrences in the set is strictly less than 1. Hence, only the very rare sets, in which every variable occurs an even number of times, can have $imp = 1$. Table 3

3. The impurity of a set of clauses and of any its variable v do not change if all literals of v are reversed, so a set with $neg(v) \leq pos(v)$ for all its n variables represents an equivalence class of 2^n sets with the same impurity of all sets and all corresponding variables. If the variables are numbered, this class of sets with an equivalent impurity can be extended by permutation of their numbers. So, for any set of clauses F, there exists a set F' over the same number of variables such that $imp(F) = imp(F')$, for every variable v of F there is a variable w of F', and vice versa, such that $imp(v) = imp(w)$, and for all w $neg(w) \leq pos(w)$.

Table 2: Percentage of sets with a target impurity α produced by Model S ($n = 100$, $k = 3$)

α	r						
	3.0	4.0	4.26	5.0	6.0	7.0	8.0
0.1 - 0.5	100	100	100	100	100	100	100
0.6	99.5	100	100	100	100	100	100
0.7	96.8	99.6	99.8	100	100	100	100
0.8	90.0	97.4	98.1	99.8	99.9	100	100
0.9	28.2	69.9	76.2	88.4	98.2	99.4	99.9

presents for sets generated with the target impurity $\alpha = 1.0$ their average impurity $\overline{imp}$, and the percentage of sets with $\overline{imp} \pm 0.005$. Values of $\sigma(set)$ of sets generated by Model S appear in Tables 4 and 5, and will be discussed in the sequel.

Table 3: Sets generated by Model S with $\alpha = 1.0$ ($n = 100$, $k = 3$)

r	3.0	4.0	4.26	5.0	6.0	7.0	8.0
$\overline{imp}$	0.89	0.91	0.92	0.93	0.94	0.95	0.96
% of sets	33.1	41.2	45.4	48.3	50.3	62.4	70.7

Figure 2 displays probability of satisfiability $psat$ as a function of r and imp. As in Figure 1 $psat$ exhibits the typical behaviour of phase transition as a function of imp. The values of $psat$ differ slightly from those of Figure 1 (for the same n, r, imp), and for $r = 4.26, 6.00$ are presented in Table 4 that compares the generating models. For $r \geq 3.8$, $psat = 0$ for all sets produced by Model S with maximum values of impurity shown in Table 3, so, the corresponding curves in Figure 2 are prolonged up to $imp = 1.0$ with $psat = 0$.

Table 4 shows (for $r = 4.26, 6.0$) that the standard deviation $\sigma(set)$ of impurity of sets generated by Model S is more than 10 times smaller than that of sets generated by Model P (this ratio holds also for all $r \leq 50$ checked in the experiments). The small value of $\sigma(set)$ is due to the fact that in Model S every assignment to a variable minimizes the difference between the current impurity of the generated set and the target one. So, every assignment to an individual variable must adjust the impurity of the entire set, but this leads to a relatively large standard deviation $\sigma(var)$ of impurity of variables within a set. Indeed, in Table 4 $\sigma(var)$ of sets generated by Model S is larger than that of sets produced by Model P.

3.3 Model V

To study impurity in a wider class of random sets of clauses we used a model that generates sets with a relatively small variance of impurity of each individual variable. Model V implements the following procedure. Given a target impurity α, choose randomly and

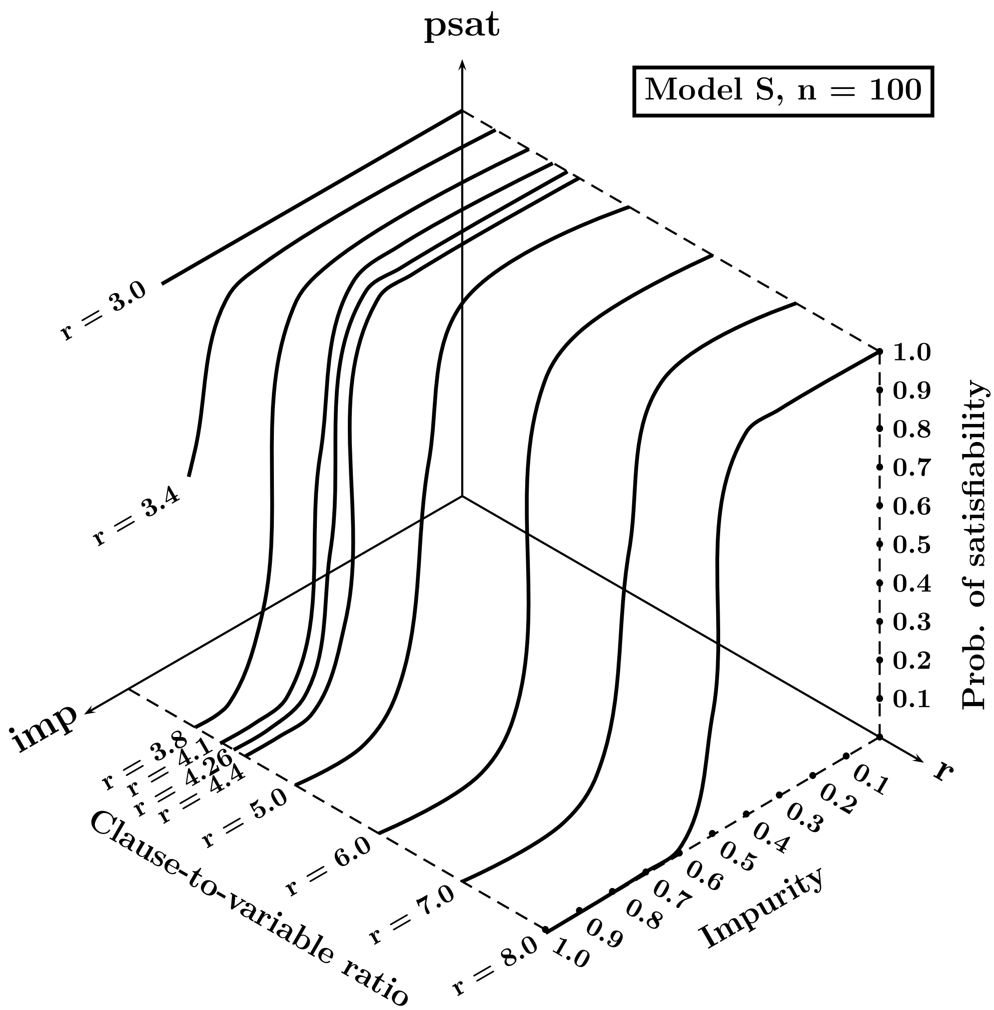

Figure 2: Phase transition of *psat(r, imp)* of sets of clauses generated with a constant impurity of each set (Model S)

uniformly one of n variables. Then before inserting a new literal of the variable in a clause calculate the current impurity $\tilde{\alpha}$ of this variable, and insert v or $\neg v$ to minimize the new current difference $\Delta = |\alpha - \tilde{\alpha}|$.

Let α, λ, pos_λ, neg_λ, α_λ, L_λ denote, respectively, the target impurity of sets, the number of literals of a variable v currently in the set under generation, the number of its unnegated, negated literals, the current value of its impurity, the last λ-th literal of the variable inserted in the set. Then

$$pos_\lambda + neg_\lambda = \lambda \tag{9}$$

$$\alpha_\lambda = \min(pos_\lambda, neg_\lambda)/\max(pos_\lambda, neg_\lambda) \tag{10}$$

$$\Delta_\lambda = |\alpha - \alpha_\lambda| \tag{11}$$

If a variable v chosen for a new literal does not yet appear in the set, then a literal $L_1 = v$ is inserted such that $\lambda = 1$, $pos_1 = 1$, $neg_1 = 0$, $\alpha_1 = 0$, $\Delta_1 = \alpha$. For all λ, if

$pos_\lambda = neg_\lambda$ then $L_{\lambda+1} = v$, so, for all λ, $neg_\lambda \leq pos_\lambda$ [4]. Each L_λ is assigned to minimize Δ_λ, so for all $\lambda > 1$ and all variable v, if

$$\left|\alpha - \frac{neg_\lambda}{pos_\lambda + 1}\right| \leq \left|\alpha - \frac{neg_\lambda + 1}{pos_\lambda}\right| \tag{12}$$

then $L_{\lambda+1} = v$, otherwise $L_{\lambda+1} = \neg v$.

For any variable v, any series of consecutive insertions of its unnegated literal is bounded. It can be shown that if for some λ $L_\lambda = v$, that is

$$\left|\alpha - \frac{neg_{\lambda-1}}{pos_{\lambda-1} + 1}\right| \leq \left|\alpha - \frac{neg_{\lambda-1} + 1}{pos_{\lambda-1}}\right| \tag{13}$$

then there exists a finite value $t \geq 1$ such that for all $\lambda \leq x < \lambda + t$, $L_x = v$, but $L_{\lambda+t} = \neg v$, that is

$$\left|\alpha - \frac{neg_{\lambda-1}}{pos_{\lambda-1} + t + 1}\right| > \left|\alpha - \frac{neg_{\lambda-1} + 1}{pos_{\lambda-1} + t}\right|. \tag{14}$$

In the same way it can be shown that for any variable, any series of consecutive insertions of its negated literal is bounded. Because both pos_λ and neg_λ are monotone in λ, the changes of α_λ caused by each new assignment decrease with growing λ while α_λ approaches α. Since the average number of literals of a variable in a set is rk, both $\sigma(var)$ and $\sigma(set)$ decrease with growing r, as presented in Tables 4, 5. Figure 3 shows impurity of a variable for the first 24 assignments to its literals for the target values of $\alpha = 0.1, 0.3, 1.0$.

For Model V Figure 4 displays $psat$ as a function of r and imp. It shows that for different values of r, $psat$ undergoes a phase transition while imp grows from 0 to 1. The curves are very similar to those of Figures 1, 2 although with slightly different values of $psat$ (for the same n, r, imp) presented in Table 4. In this Table, indeed as intended, $\sigma(var)$ of sets of Model V is smaller than that of sets of Models P and S, since by Model V, every assignment to a variable adjusts the value of imp of this particular variable.

4. Comparison of models

Tables 4, 5 allow comparison of the models. For the three models Table 4 presents values of $psat$, $\sigma(var)$, $\sigma(set)$ for $imp = 0.1 - 0.94$ and $r = 4.26, 6.0$. The models are arranged in the order of decreasing $psat$ such that $psat_S \geq psat_P \geq psat_V$. It can be noticed that the corresponding values of $\sigma(var)$ follow the same order: $\sigma_S(var) > \sigma_P(var) > \sigma_V(var)$, while $\sigma_P(set) > \sigma_V(set) > \sigma_S(set)$. Table 5 shows a steady concentration of impurity of sets generated by the models around the mean value 0.5 while the size of the sets grows. The value of $\sigma(set)$ of Model P decreases slower than that of Models S and V (the last column and three last rows of Table 5).

Let $W_\epsilon(imp)$ denote the scaling window in which $psat$ (for a constant r) undergoes phase transition as a function of imp, and $lower(\epsilon)$, $upper(\epsilon)$, $middle$ stand for boundaries of $W_\epsilon(imp)$ and its middle such that

$$lower(\epsilon) = \inf\{imp | psat(imp) \leq 1 - \epsilon\}, \tag{15}$$

$$upper(\epsilon) = \sup\{imp | psat(imp) \geq \epsilon\}, \tag{16}$$

$$psat(middle) = 0.5. \tag{17}$$

4. This does not cause a loss of generality, as shown in Subsection 3.2

The *width* of scaling window:

$$|W_\epsilon(imp)| = \frac{upper(\epsilon) - lower(\epsilon)}{middle}. \tag{18}$$

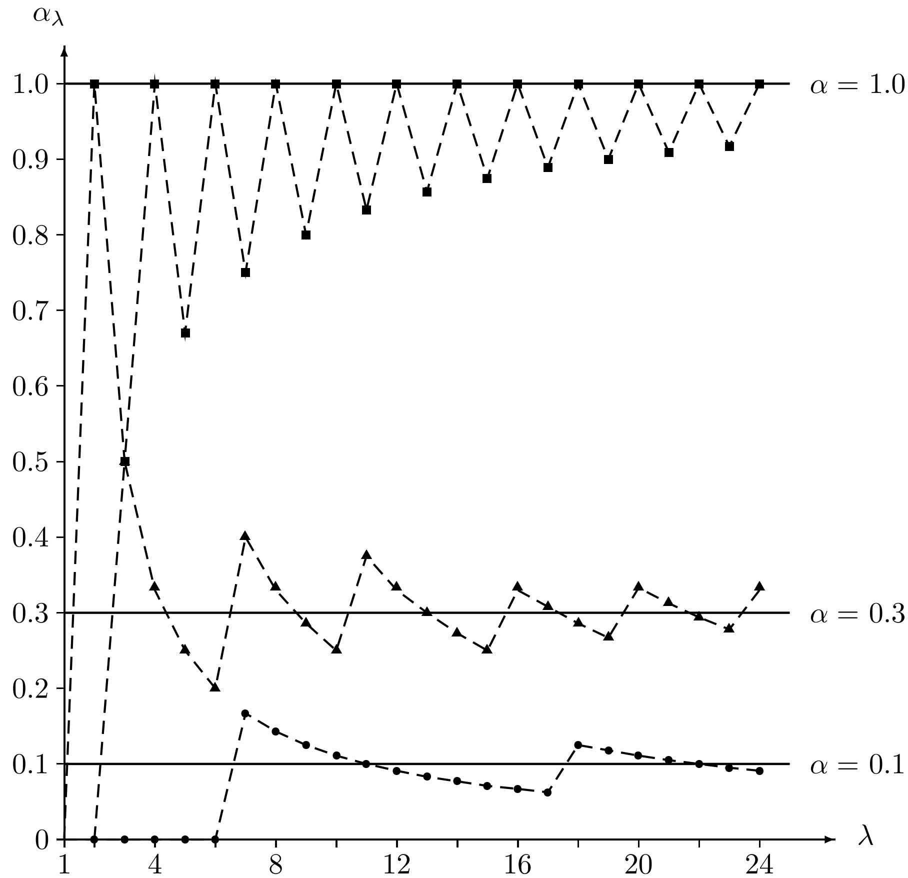

Figure 3: Impurity α_λ of any variable for the first $\lambda \leq 24$ assignments to its literals (target $\alpha = 0.1, 0.3, 1.0$)

Table 6 shows (for Model S) the movement of scaling windows $W_\epsilon(imp)$ toward low values of impurity while the clause-to-variable ratio grows from 3.8 to 1000 ($n = 100$, $k = 3$). This movement is explained by the fact that for $m \leq \binom{n}{k}$ there are sets of m clauses with zero impurity and $psat = 1$; on the other hand, for $0 < imp$ and all $r > r'$, $psat(r, imp) \leq psat(r', imp)$. This means also that if $psat(imp)$ undergoes a phase transition for some ratio r', then a phase transition must take place for at least all ratios $r' \leq r \leq \binom{n}{k}/n$.

Models P and V produce a similar movement of the corresponding scaling windows slightly shifted toward smaller values of impurity, since (as shown partly in Table 4) for the same values of n, r, imp it holds $psat_S \geq psat_P \geq psat_V$.

To show phase transition of $psat$ in more details, Figure 5 displays for $r = 4.26$ (Model S) $psat$ vs. impurity, and its scaling windows shrinking while n grows from 30 to 200.

Expression (2) is equivalent to a statement that for $k \geq 2$ and all $\epsilon(r) > 0$, $\lim_{n\to\infty} |W_\epsilon| = 0$. So, to estimate the sharpness of phase transition of $psat$ as a function of impurity we

Table 4: Comparison of the three generating models ($n = 100$)

P - constant *pul*, S - constant impurity of sets of clauses
V - constant impurity of every variable in a set of clauses
$\sigma(var)$ - average standard deviation of impurity of variables in a set of clauses
$\sigma(set)$ - standard deviation of impurity of sets of clauses

r	model		*mean imp*									
			.100	.200	.300	.400	.500	.600	.660	.800	.900	.940
4.26	S	*psat*	1.0	1.0	1.0	1.0	1.0	.997	.779	.004	.000	.000
		$\sigma(var)$	.133	.198	.239	.261	.267	.256	.240	.178	.106	.084
		$\sigma(set)$	.0003	.0004	.0005	.0006	.0007	.0008	.0010	.0015	.0033	.0082
	P	*psat*	1.0	1.0	1.0	1.0	1.0	.960	.524			
		$\sigma(var)$	.109	.165	.207	.235	.246	.238	.221			
		$\sigma(set)$	.0112	.0165	.0199	.0230	.0250	.0240	.0220			
	V	*psat*	1.0	1.0	1.0	1.0	.961	.478	.022	.000	.000	
		$\sigma(var)$	.030	.036	.044	.050	.055	.065	.074	.084	.085	
		$\sigma(set)$	.0028	.0036	.0043	.0049	.0057	.0066	.0074	.0088	.0089	
6.00	S	*psat*	1.0	1.0	1.0	1.0	.928	.088	.000	.000	.000	.000
		$\sigma(var)$	.121	.181	.221	.244	.250	.241	.214	.168	.098	.059
		$\sigma(set)$	.0002	.0003	.0003	.0004	.0005	.0005	.0006	.0008	.0014	.0058
	P	*psat*	1.0	1.0	1.0	1.0	.266	.006	.000			
		$\sigma(var)$	.086	.134	.175	.207	.224	.221	.202			
		$\sigma(set)$	.0089	.0133	.0174	.0211	.0228	.0224	.0190			
	V	*psat*	1.0	1.0	1.0	.421	.000	.000	.000	.000	.000	
		$\sigma(var)$	.021	.025	.030	.034	.037	.044	.050	.057	.066	
		$\sigma(set)$	.0019	.0026	.0029	.0034	.0039	.0046	.0053	.0056	.0064	

compare the width of scaling windows $W_\epsilon(imp)$ of *psat* vs. *imp* with $W_\epsilon(r)$ of *psat* controlled by r as presented in Table 7. The third row gives values of $|W_\epsilon(r)|$ for random sets of 3-literal clauses with r changing from 3 to 5, while the last row gives $|W_\epsilon(imp)|$ for sets with $r = 4.26$ and impurity changing from 0.1 to 1.0. It shows that for $n = 30 - 200$ scaling windows controlled by impurity are narrower and shrinking faster (the last column of Table 7) than those determined by clause-to-variable ratio. This fact provides an evidence for a conjecture that the phase transition of satisfiability as a function of impurity is sharp for a wide range of r.

Conjecture 1. *There exists a sequence* $imp^*(n, r, k)$ *such that for all* $3.8 \leq r \leq \binom{n}{k}/n$ *and* $\epsilon > 0$,

$$\lim_{n\to\infty} psat(n, r, k, imp) = \begin{cases} 1 & \text{if} \quad imp = imp^*(n, r, k) - \epsilon \\ 0 & \text{if} \quad imp = imp^*(n, r, k) + \epsilon \end{cases} \tag{19}$$

The models used in this study produce sets of clauses with different distributions of impurity of the sets, and of variables in the sets (Table 4). Nevertheless, the sets exhibit a similar phase transition of *psat* as a function of r and *imp* (Figures 1, 2, 4). This fact suggests that impurity is a significant parameter controlling propositional satisfiability.

Although Model P exhibits a pattern of phase transition similar to that of Models S and V, there are differences between the notions of impurity and probability of unnegated literal: (i) Given a set of clauses, it is hard to determine the value of *pul* that generated the set, while impurity is easily computable a parameter of the set characterizing its expected satisfiability; (ii) The notion of impurity suggests a conjecture (Section 6) that for many

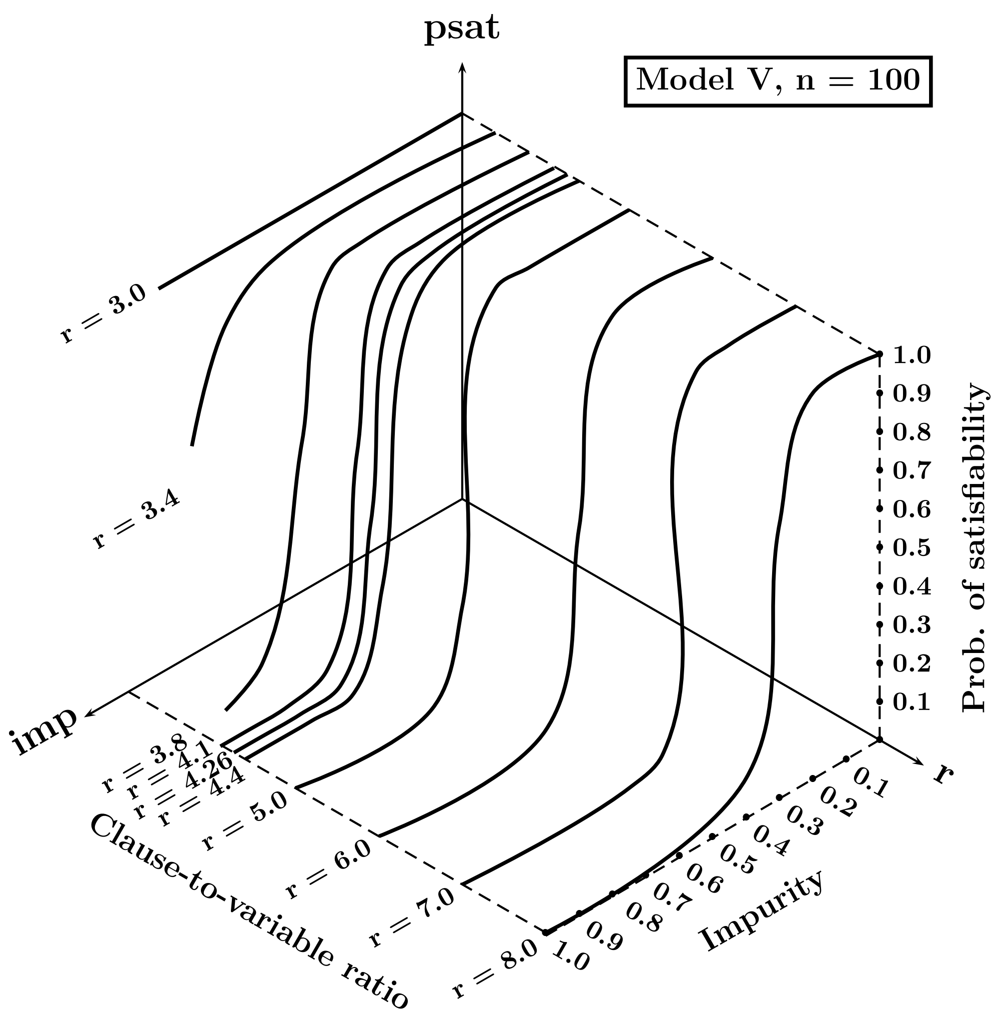

Figure 4: Phase transition of *psat(r, imp)* of sets of clauses generated with constant impurity of each variable (model V)

processes undergoing a phase transition controlled by some parameter, there exist different parameters causing a phase transition of the process; (iii) Aiming at a certain impurity of a set of clauses allows producing sets with a richer diversity than it is possible with a constant pul.

The phenomenon discovered by Sinopalnikov [22] that the smaller pul, the larger the value of r at which $psat$ undergoes phase transition , can be explained by the fact that the smaller pul, the smaller impurity of the sets with the same r, and therefore the higher the probability of their satisfiability. An increase of r compensates for the effect of decreased pul by decreasing $psat$, and causing its phase transition. The curves with squares and triangles in Figure 1 illustrate this phenomenon.

Table 5: Concentration: standard deviation $\sigma(set)$ of impurity of growing sets of clauses ($imp = 0.5$).

P - constant pul S - constant impurity of sets of clauses
V - constant impurity of every variable in a set of clauses

n	model	r						$\frac{\sigma(set)_{r=3}}{\sigma(set)_{r=10}}$
		3.0	3.8	4.26	5.0	7.0	10.0	
30	S	.0032	.0026	.0021	.0019	.0012	.0008	4.00
	V	.0167	.0117	.0103	.0084	.0057	.0038	4.39
	P	.0483	.0462	.0450	.0424	.0380	.0334	1.45
100	S	.0010	.0008	.0007	.0006	.0004	.0003	3.33
	V	.0088	.0066	.0057	.0045	.0031	.0022	4.00
	P	.0273	.0257	.0250	.0240	.0215	.0197	1.39
200	S	.0005	.0004	.0003	.0003	.0002	.0001	5.00
	V	.0046	.0036	.0034	.0029	.0023	.0016	2.88
	P	.0186	.0182	.0179	.0170	.0161	.0145	1.28
$\frac{\sigma(set)_{n=30}}{\sigma(set)_{n=200}}$	S	6.40	6.50	7.00	6.33	6.00	8.00	
	V	3.63	3.25	3.03	2.90	2.48	2.38	
	P	2.60	2.54	2.51	2.49	2.36	2.30	

Table 6: Moving scaling windows $W_\epsilon(imp)$, Model S ($n = 100$)

r	3.8	4.26	5	10	50	100	500	1000
$lower(0.01)$	.69	.61	.54	.38	.21	.19	.18	.18
$lower(0.05)$	.71	.63	.56	.39	.25	.23	.20	.19
$middle$	.77	.69	.61	.45	.32	.30	.28	.28
$upper(0.05)$	.85	.74	.67	.50	.39	.38	.37	.37
$upper(0.01)$	.93	.77	.69	.53	.42	.41	.40	.40

5. Hard sets

To estimate the computational hardness of satisfiability testing of various sets of clauses (and make this estimation machine independent) programs in the experiments counted for each set the number of truth assignments to variables that is proportional to the run time. Figure 6 shows for Model S the average number of assignments for unsatisfiable sets, satisfiable ones, and all sets (curves marked with black and hollow circles, and black triangles, respectively) as a function of impurity for $n = 100$ and $r = 4.26$. The curves are very similar to those depicting run time as a function of r. The run time of unsatisfiable sets decreases monotonously with growing impurity, while the run time of satisfiable sets is bounded by that of unsatisfiable ones, and approaches the latter with a maximum value by the middle of the scaling window. As it can be expected, the hardest unsatisfiable

Table 7: Shrinking of scaling windows $W_\epsilon(r)$, $W_\epsilon(imp)$, Model S

	$n = 30$		$n = 100$		$n = 200$		$\frac{\lvert W_{.01}(n=30)\rvert}{\lvert W_{.01}(n=200)\rvert}$
ϵ	0.01	0.05	0.01	0.05	0.01	0.05	
$\lvert W_\epsilon(r)\rvert$	0.562	0.381	0.251	0.174	0.203	0.154	2.768
$\lvert W_\epsilon(imp)\rvert$	0.528	0.375	0.232	0.159	0.162	0.088	3.259

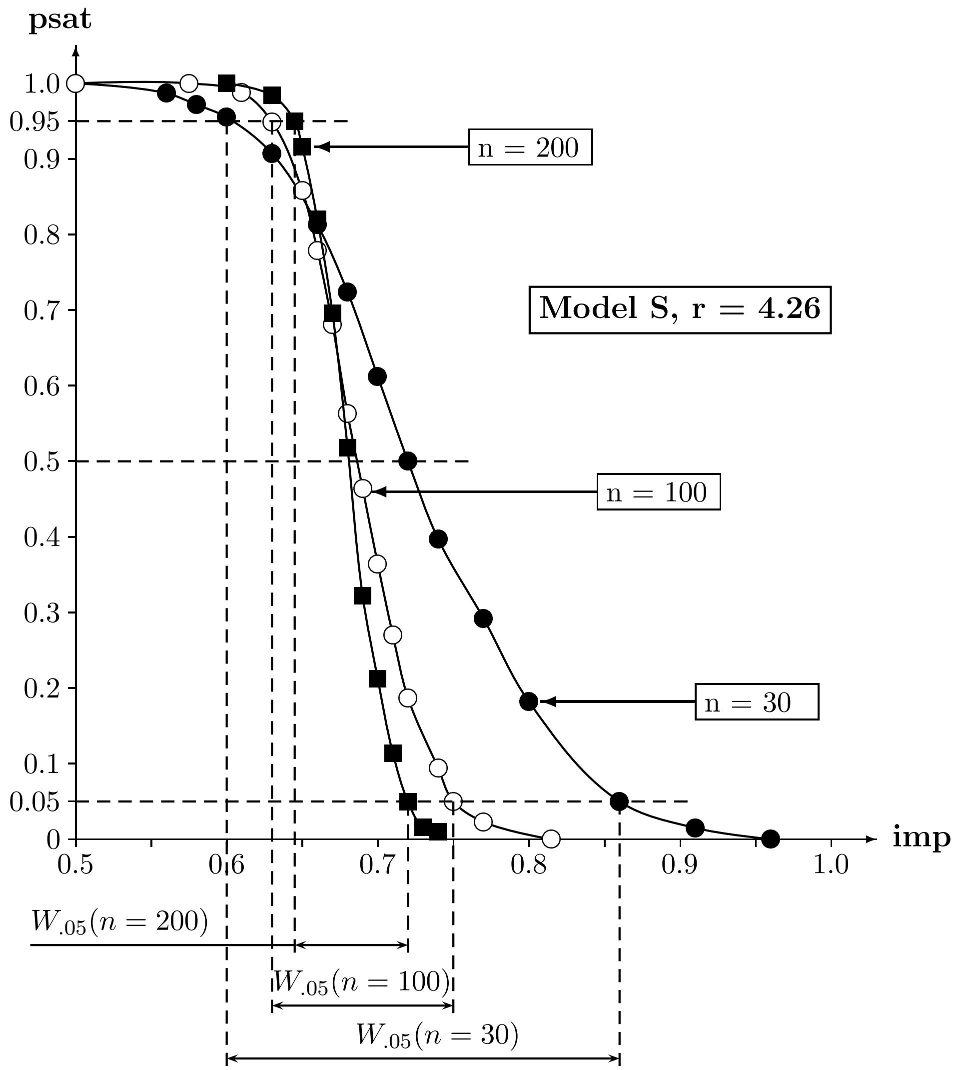

Figure 5: Phase transition of $psat(imp)$, and its scaling windows (Model S)

instances appear at the lower edge of a scaling window (marked with a large black dot) where unsatisfiable sets emerge with a low probability.

The average impurity of sets generated with a constant *pul* is apart from the lower edge of the corresponding scaling windows controlled by *imp* for most values of r, so, many hardest unsatisfiable instances are omitted in the common experiments with SAT. Table 8 compares for different values of r and n the run time (in thousands of assignments) of the hardest unsatisfiable sets generated by Model V with that of unsatisfiable sets produced by Model P with $pul = 0.5$. The last columns of the table present parameters a and b of the function $a2^{n/b}$ that approximates run time with errors within 6%. (For $r = 3.1$ Model P does not produce a significant proportion of unsatisfiable sets). The table gives values of impurity of sets with the corresponding run time.

In the table, for the same values of n and r, the run time of the hardest unsatisfiable sets produced by Model V is consistently larger than that produced by Model P. This result accords with the study by Bayardo and Schrag [3] who showed that the larger the structural regularity of a set, the harder checking its satisfiability. Indeed, Table 4 shows that the standard deviation of impurity of variables $\sigma(var)$ in sets of Model V is smaller than that of Model P.

6. Conclusion: More parameters controlling phase transition

The previous sections show that besides the clause-to-variable ratio r of sets of clauses that controls the phase transition of their satisfiability there is a different parameter, impurity, that controls the satisfiability very similarly to r. By this virtue both r and *imp* provide partial information regarding the satisfiability of a given set. Unless $\mathbf{P} = \mathbf{NP}$ no syntactic measure can determine satisfiability of a set of clauses with certainty. So, there is room for more information of the satisfiability that may be provided by another parameter of the sets.

At the present state of the art, checking satisfiability of a set of clauses may require time exponential in the size of the set. So, if the available machine resources and time limitation do not facilitate solving SAT for a given set of clauses, it would be helpful to exploit parameters of the set that can be computed effectively, and provide an estimation of the probability that it is satisfiable. Let us call such a parameter *a witness* to satisfiability. The clause-to-variable ratio r, and impurity are such witnesses. Table 4 shows that a larger $\sigma(var)$ indicates a larger *psat*, so $\sigma(var)$ is a candidate for a witness. Besides, this means that an increase of the structural regularity of a set of clauses causes not only an increase of its expected hardness [3], but also a decrease of probability of its satisfiability.

Consider a set of clauses $F(n, m, k)$. As $imp(F)$ is the average impurity of its variables, there are variables in F having impurity larger or smaller than $imp(F)$. By the argument of Section 2 supported by the experiments, the larger $imp(F)$ the smaller the probability that F is satisfiable. However, the impact of each variable v_i of F on its satisfiability should depend not only on $imp(v_i)$, but also on the number $lit(v_i)$ of clauses in which v_i occurs. It is reasonable to assume that for a constant $imp(F)$, the larger the number of occurrences of variables with larger values of $imp(v_i)$ (and smaller that of variables with smaller impurity), the smaller the probability of satisfiability of F. Consider the following measure of a set of

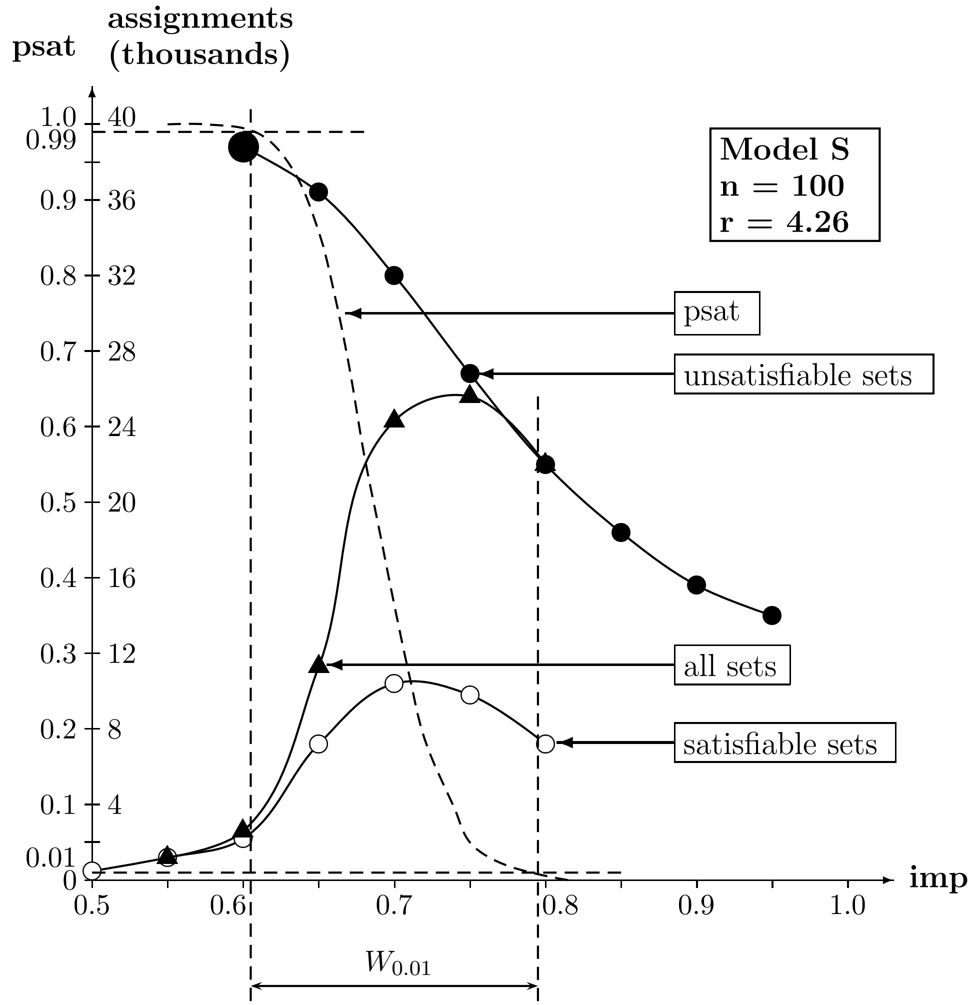

Figure 6: Run time as a function of *imp* (in number of assignments)

clauses $F(n, m, k)$ – *balance*:

$$bal(F) = \frac{1}{nrk} \sum_{i=1}^{n} lit(v_i) * imp(v_i), \tag{20}$$

Since nrk is the number of all literals in F, and for all v_i, $0 \leq imp(v_i) \leq 1$, it holds

$$0 \leq bal(F) \leq 1. \tag{21}$$

Conjecture 2. *Let* $\mathbf{S}(n, r, k, imp)$ *denote a set of all random sets of clauses with constant values of* n, r, k *and constant impurity* $\delta < imp < 1 - \delta$ *for a small* $\delta > 0$*. Then* $psat(S)$ *for a set* $S \in \mathbf{S}(n, r, k, imp)$ *is a monotone decreasing function of* $bal(S)$*, and undergoes a phase transition while* $bal(S)$ *increases.*

Table 8: Hardest unsatisfiable sets (run time in thousands of assignments)

r			n					a	b
			40	60	80	100	120		
3.1	Model V	*run time*	0.695	3.739	20.30	110.2		21.38	8.10
		imp	0.83	0.85	0.88	0.88			
4.26	Model V	*run time*	0.691	3.689	19.11	97.91	506.6	24.31	8.33
		imp	0.45	0.45	0.49	0.49	0.49		
	Model P	*run time*	0.618	2.697	11.89	53.48	235.7	29.35	9.22
	$pul = 0.5$	*imp*	0.57	0.59	0.59	0.60	0.61		
6.0	Model V	*run time*	0.649	2.741	11.91	51.96	223.4	34.15	9.44
		imp	0.31	0.32	0.34	0.34	0.35		
	Model P	*run time*	0.416	1.332	4.351	14.21	47.40	37.42	11.61
	$pul = 0.5$	*imp*	0.59	0.61	0.63	0.63	0.64		

There are famous and extensively investigated cases of phase transition.

The probability that a graph $G(n, m)$ chosen randomly and uniformly out of the set of all undirected graphs with n vertices and m edges is connected increases monotonously with m, and undergoes a sharp phase transition at the value of $m = (n \ln n)/2$ [11, 12].

The probability that a bag of integers (a multi-set containing possibly identical integers) $B(n, m)$ containing m integers randomly and uniformly chosen from a range $[1, \ldots, 2^n]$ can be divided into two distinct parts with equal sums of elements (a *perfect partition*) is a monotone function of m, and has a sharp phase transition at $m = n$ [14, 18, 5, 20]

Conjecture 3. *Various properties (such as graph connectivity and perfect integer partition) that undergo phase transition vs. a syntactic parameter, have phase transition under control of several different syntactic measures.*

Candidates for these secondary measures are standard parameters of distribution of degree of vertices within a $G(n, m)$, and of distribution of integers in a $B(n, m)$.

Acknowledgments

I wish to thank the anonymous reviewers for their most inspiring comments.

References

[1] Achlioptas, D., and Sorkin, G., 2000, Optimal myopic algorithms for random 3-SAT, In *Proceedings, 41st IEEE Symp. on Foundations of Computer Science.*

[2] Achlioptas, D., and Moore, C., 2002, The asymptotic order of the random k-SAT threshold, In *Proceedings, 43rd Annual Symp. on Foundations of Computer Science*, pp. 779-787.

[3] Bayardo Jr., R. J, and Schrag, R., 1996, Using CSP look-back techniques to solve exceptionally hard SAT instances. In *Proceedings, Second International Conference on Principles and Practice of Constraint Programming*, pp. 46-60.

[4] Bollobás, B., Borgs, C., Chayes, J. T., Kim, J. H., and Wilson, D. B., 2001, The scaling window of the 2-SAT transition. *Random Structures and Algorithms*, **18** (3): 201-256.

[5] Borgs, C., Chayes, J. and Pittel, B., Phase transition and finite-size scaling for the integer partitioning problem, *Random Structures and Algorithms*, 18, 2001, pp. 247-288.

[6] Chvátal, V., and Reed, B., 1992, Mick gets some (the odds are on his side), In *Proceedings, 33rd Annual Symp. on Foundations of Computer Science*, Pittsburgh, PA, pp. 620-627.

[7] Cook, S., 1971, The complexity of theorem proving procedures, In *Proceedings, 3rd ACM STOC*, pp. 151-158.

[8] Crawford, J., and Auton, L., 1996, Experimental results on the crossover point in random 3-SAT, *Artificial Intelligence*, **57**: 31-81.

[9] Davis, M., Logemann, G., and Loveland, D., 1962, A machine program for theorem proving, *Communications of the ACM*, **5** (7): 394-397.

[10] Dubois, O., Boufkhad, Y., and Mandler, J., 2000, Typical random 3-SAT formulae and the satisfiability threshold, In *Proceedings, 11th ACM-SIAM Symp. on Discrete Algorithms*, San Francisco, CA, pp. 126-127.

[11] Erdös, P. and Rényi, A., On random graphs I, *Publicationes Mathematicae*, Debrecen, 6, 1959, pp. 290-297.

[12] Erdös, P. and Rényi, A., On the evolution of random graphs, *A Magyar Tudománios Akadémia Mathematicae Kutató Intézetének Közleményei*, 5, 1960, pp. 17-61.

[13] Friedgut, E., 1999, Sharp thresholds of graph properties, and the k-SAT problem, *J. Amer. Math. Soc.*, **12** (4): 1017-1054.

[14] Gent, I., and Walsh, T., 1994, The SAT phase transition, In *Proceedings, ECAI-94*, pp. 105-109.

[15] Goerdt, A., 1996, A threshold for unsatisfiability, *J. Comput. System Sci.*, **53** (3): 469-486.

[16] Hogg, T., Huberman, B., and Williams, C., 1996, Phase transitions and the search problem, *Artificial Intelligence* **81**: 1-15.

[17] Huberman, B., and Hogg, T., 1987, Phase transitions in Artificial Intelligence systems, *Artificial Intelligence* **33**: 155-171.

[18] Korf, R., A complete anytime algorithm for number partitioning, *Artificial Intelligence*, 106, 1998, 181-203.

[19] Martin, O., Monasson, R., and Zecchina, R., 2001, Statistical mechanics methods and phase transitions in optimization problems, *Theoretical Computer Science* **265**: 3-67.

[20] Mertens, S., A physicist's approach to number partitioning, *Theoretical Computer Science*, 265, 2001, pp. 79-108.

[21] Mitchell, D., Selman, B., and Levesque, H., 1992, Hard and easy distributions of SAT problems, In *Proceedings, AAAI-92*, pp. 459-465.

[22] Sinopalnikov, D. A., Satisfiability threshold of the skewed random k-SAT, In *Proceedings, 7th Intl. Conf. on Theory and Applications of Satisfiability Testing*, Vancouver, BC, Canada, 2004.