On the Structure of Industrial SAT Instances*

Carlos Ansótegui[1], María Luisa Bonet[2], and Jordi Levy[3]

[1] Universitat de Lleida (DIEI, UdL)
[2] Universitat Politècnica de Catalunya (LSI, UPC)
[3] Artificial Intelligence Research Institute (IIIA, CSIC)

Abstract. During this decade, it has been observed that many real-world graphs, like the web and some social and metabolic networks, have a *scale-free* structure. These graphs are characterized by a big variability in the arity of nodes, that seems to follow a power-law distribution. This came as a big surprise to researchers steeped in the tradition of classical random networks.

SAT instances can also be seen as (bi-partite) graphs. In this paper we study many families of industrial SAT instances used in SAT competitions, and show that most of them also present this scale-free structure. On the contrary, random SAT instances, viewed as graphs, are closer to the classical random graph model, where arity of nodes follows a Poisson distribution with small variability. This would explain their distinct nature.

We also analyze what happens when we instantiate a fraction of the variables, at random or using some heuristics, and how the scale-free structure is modified by these instantiations. Finally, we study how the structure is modified during the execution of a SAT solver, concluding that the scale-free structure is preserved.

1 Introduction

The Satisfiability problem (SAT) is central in Computer Science. It was the first problem to be proven NP-Complete, and it is used extensively to encode many other problems into it. Therefore, finding good algorithms to solve SAT is of practical use in many areas of Computer Science. Even though the general SAT problem is NP-Complete, many very large industrial instances can be solved efficiently by modern solvers. The aim of this work is to study the body of industrial instances to detect general properties that are shared by the majority of instances. We focus on the structure of the instances viewed as bi-partite graphs, where nodes represent variables and clauses, and edges represent the presence of a variable in a clause. In particular, we try to detect the distribution on the frequencies of the variables and of the sizes of the clauses, in SAT instances used in the latest SAT Competitions and SAT Races. Our work was inspired by [BDIS05], where they suggest that industrial instances, as many other real-world graphs could have a scale-free structure.

* Research partially supported by the projects TIN2007-68005-C04-{01,03,04} and TIN2006-15662-C02-02 funded by the MEC.

I.P. Gent (Ed.): CP 2009, LNCS 5732, pp. 127–141, 2009.

The classical random graph model [ER59] was one of the best studied during the last century, and set the basis of graph theory. In [WS98], a new model of random graphs is proposed, called *small-world* to describe the structure of some social collectivities. In [AJB99], they show that the world wide web, viewed as a graph, has a structure than cannot be described by the classical random graph model. They propose a new model called *scale-free*. The name comes from the fact that, in this new model, the arity of nodes follows a power-law distribution $p(k) \sim k^{-\alpha}$, and these distributions are scale-free. However, the name also suggests that these graphs present some kind of self-similarity. In recent years it has been observed that many other real-world graphs, like some social and metabolic networks, also have a scale-free structure.

Power-law (zeta and Pareto) distributions are characterized by a big variability, consequence of a polynomially decreasing tail. A small fraction of the individuals is responsible for most of the average, in what is popularly known as the *80:20 rule* (i.e. 80% of the land is owned by the 20% of the population). Many other heterogeneous distribution are also called power-law or *heavy-tailed* when their tail decreases polynomially, in contrast with other classical distributions, like normal, Poisson, or binomial that have a exponentially decreasing tail. Experience tells us that power-law distributions are as frequent in nature, if not more frequent, as exponentially decreasing distributions. For instance, the CPU time of the different executions (with different random variable selection) of a solver on a formula follow a power-law distribution [GFSB04].

The topology of graphs have a major impact on the cost of solving search problems on these graphs. Gent et al. [GHPW99] analyze the impact of a small-world topology on the cost of coloring graphs, and Walsh [Wal01] does the same in the case of scale-free graphs. Therefore, we can expect that SAT solving, viewed as a search process of on a graph (the formula), will be affected by the topology of this graph.

It is well-known in the SAT community that classical random k-CNF formulas and industrial (or real-world) formulas have a distinct nature. This makes SAT solvers to specialize in one or the other kind of formulas. In the SAT competition there is a special track for each kind of formulas, whereas in the SAT Race competition, only industrial formulas are used to test the solvers. Random k-CNF formulas, as graphs, follow the Erdös-Rényi model. In the phase transition point for $k = 3$, for instance, most of the variables have a number of occurrences very close to 12.75.[1] In this paper, we show that most industrial instances are better modeled as scale-free graphs.

We think that the present study provides a step towards a theoretical explanation of why some SAT solvers perform better on industrial instances, and others on random SAT instances. Moreover, the better understanding of real-world instances could lead to the improvement of existing SAT solvers.

The paper can also serve as basis for new random SAT generation models that produce instances closer to real-world ones. This problem is distinguished as one

[1] The number 12.75 comes from multiplying the size of the clauses $k = 3$ by the clause/variable ratio $m/n = 4.25$ at the phase transition point.

of the 10 challenge problems in SAT [SKM97, Sel00, KS03, KS07]. Recently, in [ABL09], we have proposed some random SAT instance generators that produce formulas with variable frequencies following a power-law distribution. We show that solvers specialized on industrial instances perform better in these *random industrial-like instances* than solvers specialized on random formulas.

Another application of the study could be to evaluate which is the best family of solvers to use on a particular instance, by analyzing the distribution of variable frequencies or clause sizes. In particular, this could be use as one more selection criteria in a portfolio approach [XHHLB08].

The paper proceeds as follows. In Section 2, we present the study of the distributions that best represent the frequencies of variable occurrences and clause sizes. Also we describe the statistical techniques we use in our work. In Section 3, we study whether the scale-free nature is preserved under partial instantiations of variables. In Section 4, we analyze the structure of the formulas during the execution of complete SAT solver of different nature. We conclude in Section 5.

2 Analysis of Industrial SAT Instances

2.1 Methodological Background

Every SAT instance can be seen as a bi-partite graph, with a set of nodes $V \cup C$, where V represents the variables and C represents the clauses. The edges are the pairs $(v, c) \in V \times C$ such that variable v appears in clause c. In what follows, $n = |V|$ and $m = |C|$. In order to analyze if a bi-partite graph is scale-free, we have to study the arity of the nodes. Notice that the arity of a node $v \in V$ is the number of occurrences of the variable v, and the arity of $c \in C$ is the size of the clause c.

For every bi-partite graph we can compute $f_v^{real}(k)$ as the number of variables that have a number of occurrences equal to k, divided by n, and similarly, $f_c^{real}(k)$ is the number of clauses of size k divided by m. We add the label *real* to emphasize that these functions come from empirical data. We can also define the accumulative versions of these functions as $F_v^{real}(k) = \sum_{i \geq k} f_v^{real}(i)$ and $F_c^{real}(k) = \sum_{i \geq k} f_c^{real}(i)$. Notice that, assuming that there are no empty clauses and all variables occur somewhere, $F_v^{real}(1) = F_c^{real}(1) = 1$.

In the scale-free model, the arity of nodes is characterized by a random variable K that follows a power-law distribution $f^{pow}(k) = P(K{=}k) = c\,k^{-\alpha}$. The exponent α has typically values inside $[2, 3]$. This distribution diverges at zero, and there is a lower bound k_{min} for the values of k from where we get the power-law behavior or *heavy tail*. In the discrete case (the one that concerns us), the normalizing constant is $c = 1/\zeta(\alpha, k_{min}) = 1/\sum_{i=0}^{\infty}(i + k_{min})^{-\alpha}$, where ζ is the Hurwitz zeta function. For big values of k_{min} we can approximate this distribution using the continuous version. In this case the probability density function is $f^{pow}(k) = \frac{\alpha-1}{k_{min}} \left(\frac{k}{k_{min}}\right)^{-\alpha}$, and the cumulative function is $F^{pow}(k) = \left(\frac{k}{k_{min}}\right)^{-\alpha+1}$.

There is not a proper (formal) definition of what a scale-free graph is, but one of their basic properties –usually taken as a definition– is that the arity of nodes *seems* to follow a power-law distribution. Therefore, we must check if, for some values of α_v and α_c, we have $f_v^{real}(k) \approx c\,k^{-\alpha_v}$ and $f_c^{real}(k) \approx c\,k^{-\alpha_c}$. Notice that, applying logarithms to both sides, we get $\log f(k) = \log c - \alpha \log k$. Therefore, if $f_v^{real}(k)$ and $f_c^{real}(k)$ are power-law, representing them as a function of k with double-logarithmic axes, we should get closed to a straight line with slope $-\alpha$.

In some papers, the value α is calculated by linear regression of $\log f(k)$ as a function of $\log k$. In [LADW05, section 2.1.3] there is a discussion of why it is better to plot the cumulative logarithm $\log F(k)$, instead of $\log f(k)$, to compute the regression. But, in this case, the slope is $-\alpha + 1$. Following this argument, in Figure 1 we represent $F_v(k)$ and $F_c(k)$ versus k with double-logarithmic axes, for some families of industrial formulas.

We will follow the maximum likelihood method for computing an estimation of α, as described in [CSN07]. To estimate the value of α for a collection of empirical data $k_1, \ldots, k_n$, we compute the value of α that maximizes the probability that the data were drawn from the model:

$$P(k_1, \ldots, k_n \,|\, \alpha) = \prod_{i=1}^{n} \frac{\alpha - 1}{k_{min}} \left(\frac{k_i}{k_{min}}\right)^{-\alpha}$$

We take logarithms, since the maximum will be in the same place, then we take derivatives and make the function equal to zero:

$$\begin{aligned} \frac{\partial}{\partial \alpha} \log P(x_1, \ldots, x_n \,|\, \alpha) = \\ = \frac{\partial}{\partial \alpha} \left(n \, \log \frac{\alpha - 1}{k_{min}} - \alpha \sum_{i=1}^{n} \log \frac{k_i}{k_{min}} \right) = \\ = \frac{n}{\alpha - 1} - \sum_{i=1}^{n} \log \frac{k_i}{k_{min}} = 0 \end{aligned}$$

we get

$$\hat{\alpha} = 1 + \frac{n}{\sum_{i=1}^{n} \log(k_i / k_{min})}$$

For the discrete case, a good approximation for big values of k_{min} is

$$\hat{\alpha} = 1 + \frac{n}{\sum_{i=1}^{n} \log \frac{k_i}{k_{min} - 1/2}}$$

Notice that the estimated α depends on k_{min}. To compute the value of $\hat{k_{min}}$, we try to minimize the distance between the (experimental) cumulative distribution function $F^{real}(x)$ and the (theoretical) cumulative distribution function $F^{pow}(x; \alpha, k_{min})$. The distance between both distributions is calculated as the maximal difference between both functions. Then, we compute the value of k_{min} that minimizes this distance:

$$d = \min_{k_{min} \geq 1} \left\{ \max_{k \geq k_{min}} \left\{ \left| \frac{F^{real}(k)}{F^{real}(k_{min})} - F^{pow}(k; \hat{\alpha}, k_{min}) \right| \right\} \right\}$$

We get so the value of k_{min} and of d. The value of this distance d is an indicator of the fitness of the estimation.

When we say that arity of nodes *seems* to follow a power-law distribution, we emphasize the *seems* because it is obvious that SAT formulas, as well as the WWW and other scale-free graphs, are not randomly generated. Therefore, we do not expect the arity of nodes to follow exactly any distribution. However, we want to check if some distribution fits the data better than others. In particular, we have tried to fit, apart from a power-law distribution, an exponential distribution.

The probability density function for an exponential distribution has the form $c\,e^{-\beta x}$. Calculating the constant, for the discrete case, we get $f^{exp}(k;\beta,k_{min}) = (1-e^{-\beta})\,e^{-\beta\,(k-k_{min})}$ and its cumulative function $F^{exp}(k) = e^{-\beta(k-k_{min})}$. In this case the estimation of the β parameter by the method of maximum likelihood gives:

$$\begin{aligned}\frac{\partial}{\partial\beta}\log P(k_1,\ldots,k_n\,|\,\beta) &= \\ &= \frac{\partial}{\partial\beta}\left(n\,\log(1-e^{-\beta}) - \beta\,\sum_{i=1}^{n}(k_i-k_{min})\right) = \\ &= \frac{n\,e^{-\beta}}{1-e^{-\beta}} - \sum_{i=1}^{n}(k_i-k_{min}) = 0\end{aligned}$$

Hence,

$$\hat{\beta} = \log\left(\frac{n}{\sum_{i=1}^{n}(k_i-k_{min})}+1\right)$$

The value of k_{min} is calculated as in the case of the power-law distribution.

For distinct families of industrial formulas, we have calculated $f_v^{real}(k)$and $f_c^{real}(k)$, as well as their cumulative functions. First, we have studied instances independently in each family, observing that they all have the same nature. Thus, we decide to group them by families, assuming that all formulas of the same family follow the same probability distribution. Therefore, for a family, $f_v^{real}(k)$ is the sum for every formula of the number of variables that have k occurrences, and similarly for $f_c^{real}(k)$. Notice that, under this assumption, the arity of a variable, independently of in which formula of the family it occurs, is an independent realization of the same random variable. Therefore, we can do this addition. Later, we have fitted a power-law distribution and an exponential distribution, and we have calculated the distance d^{pow} between $F_v^{real}(k)$ and the estimated $F_v^{pow}(k;\alpha,k_{min})$, and the distance d^{exp} between $F_v^{real}(k)$ and the estimated $F_v^{exp}(k;\beta,k_{min})$. When $d^{pow} < d^{exp}$, we say that the power-law distribution fits better than the exponential distribution. We use this criteria to state that a family of formulas has a scale-free structure. It is also important to compare the value of k_{min} obtained in each estimation, noted k_{min}^{pow} and k_{min}^{exp}. A big value of k_{min} means that we need to discard a lot of values of $F^{real}(k)$ to fit the distribution, and it must be taken as a point against the fitted distribution. Also a value of α far away from the interval $[2,3]$ must be read as a point against the scale-free structure.

Table 1. Most likelihood values of α and β estimated for a power-low and an exponential distribution. In bold we remark the smallest distance between the real and the fitted distributions. We also report the total number of variable occurrences n, mean $E[V]$ and variance $Var[V]$, and the respective values for clause sizes.

Variables (V)										
					Power-law			Exponential		
Family	#inst	n	$E[V]$	$Var[V]$	α	k_{min}^{pow}	d^{pow}	β	k_{min}^{exp}	d^{exp}
cmu	3	16678	7.95	12.11	3.49	5	**0.072**	0.224	4	0.176
een	12	739744	7.60	13.26	2.67	10	**0.043**	0.054	15	0.136
fuhs	2	73486	9.05	14.56	2.79	62	**0.075**	0.181	4	0.158
goldb	11	114038	21.02	88.71	2.05	21	**0.042**	0.003	100	0.204
grieu	9	6914	364.21	42.23	1.77	100	0.577	0.004	100	0.538
ibm	38	4985723	10.75	23.97	2.63	7	**0.027**	0.017	45	0.083
manol	59	7827736	6.93	16.24	2.95	57	0.059	0.017	76	**0.033**
mizh	13	725644	12.49	148.12	4.09	15	0.172	0.034	22	0.247
narai	6	9642548	9.72	16.38	3.85	5	0.152	0.109	1	0.347
palac	2	298266	10.82	60.55	1.84	20	0.087	0.003	100	0.074
post	10	12906872	7.90	44.15	2.57	12	0.132	0.135	1	0.334
schup	7	2196731	8.09	12.40	2.59	41	0.120	0.063	9	0.182
simon	12	798804	7.78	11.96	2.53	14	**0.028**	0.022	50	0.065
uts	10	1420464	13.01	74.70	1.76	69	0.111	0.003	75	**0.088**
velev	60	8442829	88.31	379.04	1.82	13	**0.030**	0.003	87	0.287
random	40	400000	12.75	3.57	18.65	24	0.019	0.777	25	**0.008**
SAT'08	100	27964721	13.30	113.48	2.29	12	**0.051**	0.003	73	0.254

Clauses (C)										
					Power-law			Exponential		
Family	#inst	m	$E[C]$	$Var[C]$	α	k_{min}^{pow}	d^{pow}	β	k_{min}^{exp}	d^{exp}
cmu	3	53769	2.46	1.21	5.35	3	0.126	1.778	3	**0.048**
een	12	2278059	2.47	0.69	3.80	4	**0.044**	2.420	3	0.046
fuhs	2	256742	2.59	0.82	4.89	5	0.041	2.182	3	**0.020**
goldb	11	710559	3.37	1.46	10.48	5	0.158	4.803	5	**0.008**
grieu	9	961030	2.62	0.76	8.54	26	0.108	3.878	3	**0.020**
ibm	38	21084555	2.54	1.57	3.77	6	**0.023**	0.375	4	0.032
manol	59	23244626	2.33	0.47						
mizh	13	3036234	2.98	0.91	1.58	1	0.328	0.408	1	0.334
narai	6	37639556	2.49	2.05	3.33	2	**0.088**	1.113	2	0.090
palac	2	1274356	2.53	9.33	1.71	4	0.116	1.055	2	0.116
post	10	42441234	2.40	1.39	3.33	2	0.143	2.884	33	**0.053**
schup	7	6947242	2.56	1.36	4.30	4	0.093	2.585	3	**0.046**
simon	12	2675233	2.32	0.90	3.76	4	0.033	0.498	5	**0.026**
uts	10	7101806	2.60	11.56	3.63	2	0.114	0.004	35	0.116
velev	60	253221473	2.94	9.01	3.35	72	0.042	0.021	28	**0.040**
random	40	1700000	3.00	0.00						
SAT'08	100	140942860	2.64	5.68	3.03	17	**0.054**	0.074	10	0.068

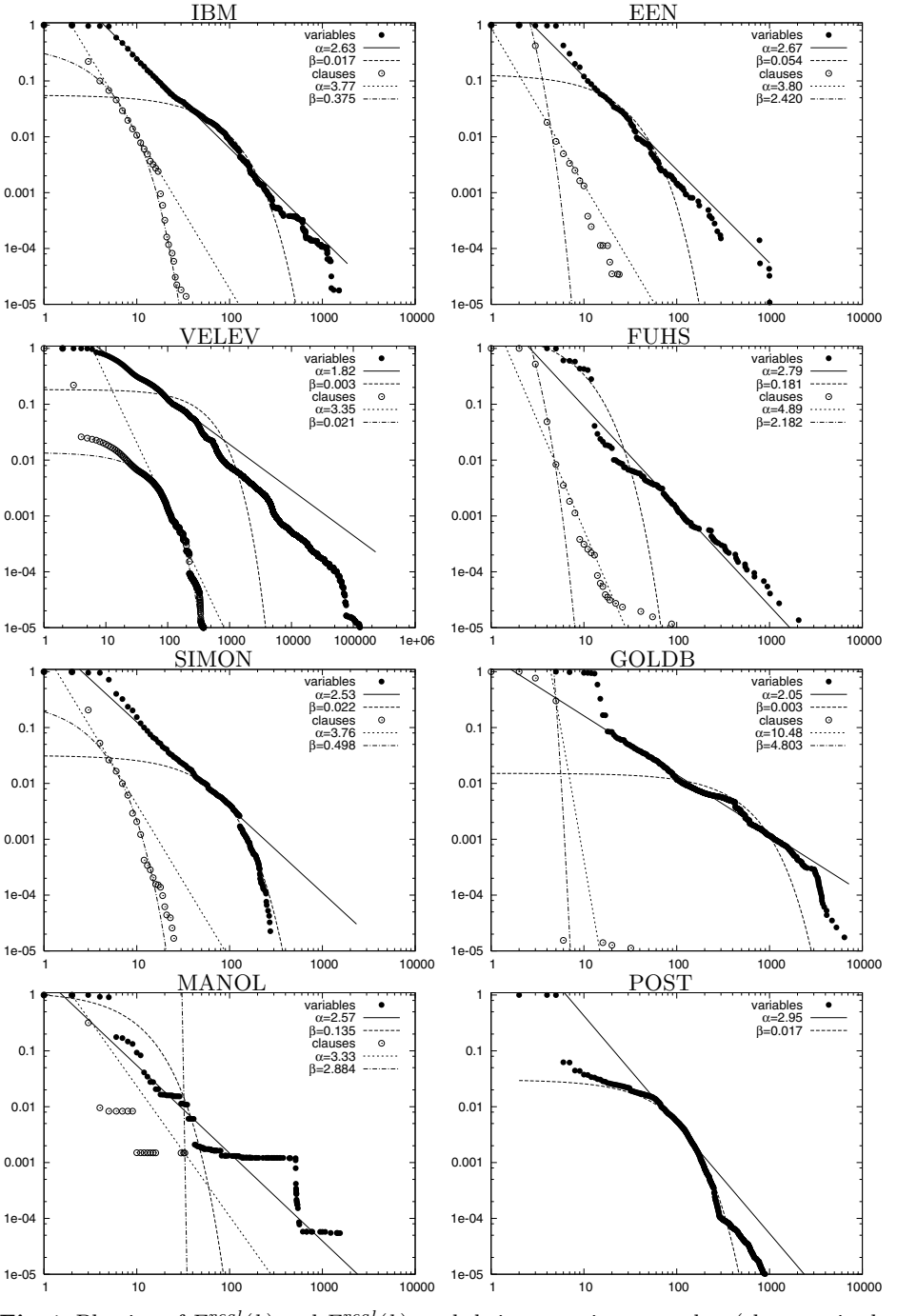

Fig. 1. Plotting of $F_v^{real}(k)$ and $F_c^{real}(k)$, and their respective power-law (characterized by α) and exponential (characterized by β) estimations, for some families of formulas. In families where all clauses are small, we have avoided the representation of $F_c^{real}(k)$.

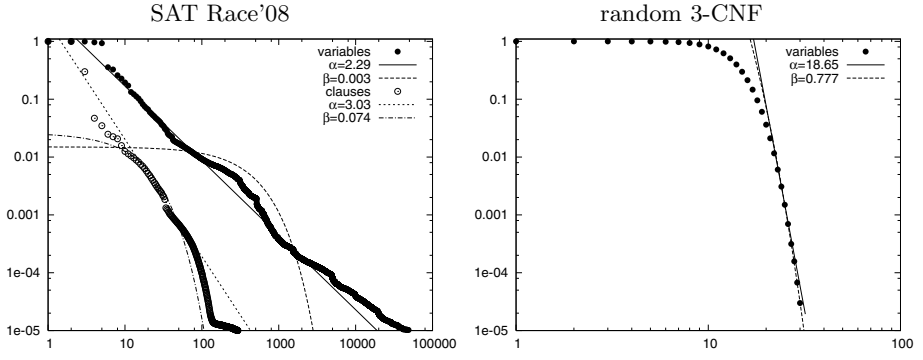

Fig. 2. Plotting of $F_v^{real}(k)$ and $F_c^{real}(k)$, and their respective power-law and exponential estimation, for the formulas of the SAT'08 Race and random 3CNF

2.2 Results of the Analysis

We have selected a set of families of formulas from the industrial category of the 2002–2005 and 2007 SAT Competitions, and the 2006 and 2008 SAT Races. For these families, Table 1 presents the estimations of the parameters of the distributions power-law and exponential for variables occurrences and clause sizes. We have also extended the study to a family of 40 random 3-CNF instances of 10^4 variables in the phase transition point; and to the *heterogeneous* family composed by the 100 instances used in the latest SAT Race 2008 competition. In Table 1 we also include information about the sum of the number of variables and clauses of all formulas of the family, and the average number of occurrences of variables and sizes of clauses, as well as their variance. For the computation of k_{min} (the value where the data starts to fit the distribution) we impose a limit value of 100. We consider that, if the distance d between the observed data and the distribution is smaller than 0.1, then it is plausible that the data follows that distribution. To conclude that the family follows a power-law distribution we also require that $d^{pow} < d^{exp}$ and the value of k_{min} to be small. For the families where all clauses have size at most 3, we obviate the study for the distribution of clause size.

In Figure 1 we plot the distributions of some families, as well as the estimated power-law and exponential distributions that best fit them. In Figure 2 we also plot the distributions for the heterogeneous family of the SAT Race 2008, and the random 3-CNF formulas.

We can conclude that for the families: CMU, EEN, FUHS, GOLDB, IBM, SIMON and VELEV, the number of variable occurrences follow a power-law distribution. In the case of clause size, only the families EEN, IBM and NARAI seem to follow a power-law distribution. Therefore, in general, the variable occurrences follows a power-law distribution in more families than the clause size. The value of α for variables is also smaller than the α for clauses, that tends to

fall out of the interval $[2, 3]$. We think that the explanation for this phenomena is that, when the formulas are encoded, people try to avoid the use of very big clauses, since they weak the propagation power in SAT solvers. We also observe that some families, like MANOL, do not seem to follow a particular distribution.

In the random 3-CNF formulas, the exponential distribution fits better than the power-law, although the distance d^{pow} is surprisingly small. If we plot the distribution for each formula of the family, we see that it is very homogeneous, without the typical peeks that we find in industrial data. Moreover, the value of $\alpha = 18.65$ is big enough to discard a power-law distribution.

Looking at the plot of the SAT Race'08 heterogeneous family, we see that the data fits better the power-law distribution than other homogeneous families. In this case, we have to take into account that the addition of so many instances, by a kind of law of the big numbers, tends to make distributions smoother. The values of α that we get are $\alpha = 2.29$ for variable occurrence and $\alpha = 3.03$ for clause size. As in some homogeneous families, we observe that the value of α in the case of clause size is bigger than the value of α for variable occurrence, and falls in the limit of the interval $[2, 3]$.

3 Instantiating Variables in Industrial Instances

Albert, Jeong & Barabási [AJB00] studied the effect of *failure* and *attack* actions in the diameter of an Erdös-Rényi graph and of a scale-free graph. The diameter is the average minimum distance between two nodes, failure consists in removing a certain percentage of randomly selected nodes, and attack consists in removing the nodes following a certain heuristic (e.g. those nodes with higher arity). They observed that failure and attack have the same effect on Erdös-Rényi graphs (after removing 5% of the nodes, the diameter increases in the same way independently of how nodes are chosen). However, while failure almost does not change the diameter of scale-free graphs, attack increases the diameter even more than in the case of an Erdös-Rényi graphs. Considering that Internet is a scale-free graph, they conclude that it is robust against random failures of the servers, but it is specially susceptible to *terrorism* attacks.

In the case of SAT solvers, the instantiation of variables removes nodes in the bi-partite graph representing the formula (e.g. the instantiation $v = true$ removes the variable-node v, and all those clause-nodes c, where c contains the literal v). Since classical random SAT instances are similar to Erdös-Rényi graphs, we can expect the same behavior on random formulas, when we instantiate variables randomly, as when we use some heuristics. However, in scale-free industrial instances, we expect a very different effect.

We have conducted a series of experiments where we instantiate up to 10% of the variables of some families of formulas, and we analyze the decrease in size of the formula. Notice that we only instantiate variables, i.e., we do not apply any local inference like unit propagation, and we do not discard the obtained subformula, even if it contains the empty clause. In Section 4 we perform a similar experiment using real SAT solvers.

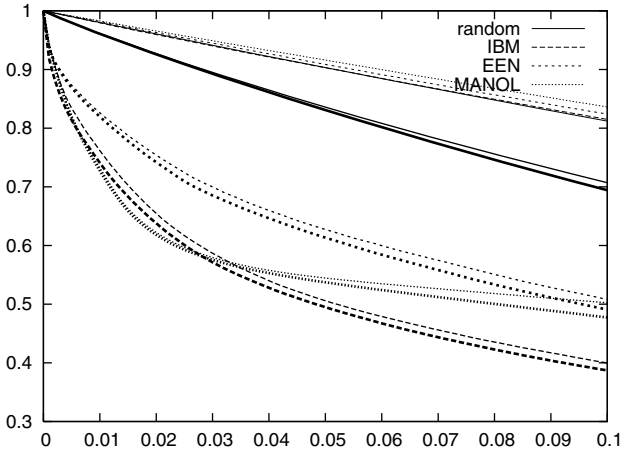

Fig. 3. Percentage of the formula-size decrease as a function of the percentage of instantiated variables. For the 3 lines of each family, the upper one corresponds to the random strategy, the middle to the Jeroslow-Wang, and the lower to the most-frequent strategy.

We experiment with the IBM and the EEN families –the ones with a more clear scale-free structure–, with the MANOL family –that does not seem to follow a neat distribution–, and with the random 3CNF set –that we know have an absolutely different structure–. Apart from the random selection of variables, we have analyzed the use of the most-frequent variable[2] and of the Jeroslow-Wang heuristics [JW90]. Results are shown in Figure 3. We observe that instantiating randomly selected variables has the same effect in all families: after instantiating 10% of the variables, the size of the formula decreases between 16% and 19%. The size-decrement seems to be proportional to the percentage of instantiated variables, i.e. the slope seems to be constant and the same in all families.

For the other two heuristics (most-frequent variable and Jeroslow-Wang), the size-decrease in random formulas is bigger, but not so much as in the industrial formulas: in random formulas, after instantiating 10% of the variables, the decrease is around 30%, whereas in industrial formulas the decrease is around 50%. Moreover, the size-decrease seems to be constant in the case of random formulas, whereas in industrial formulas, the use of these heuristics speeds up the size-decrease, but at a certain point, when we have instantiated around 1% or 2% of the variables, the slope decreases substantially. Both heuristics seem to have the same effect, although the most-frequent heuristic is always a little better (bigger decrease) than the Jeroslow-Wang heuristic.

The natural question is now: after instantiating a significant part of the variables, is the formula still scale-free? We have studied the formulas that we get after instantiating some variables of the IBM formulas following the three

[2] The most-frequent heuristic consists in selecting the variable with a higher number of occurrences, and the polarity with it appears more times.

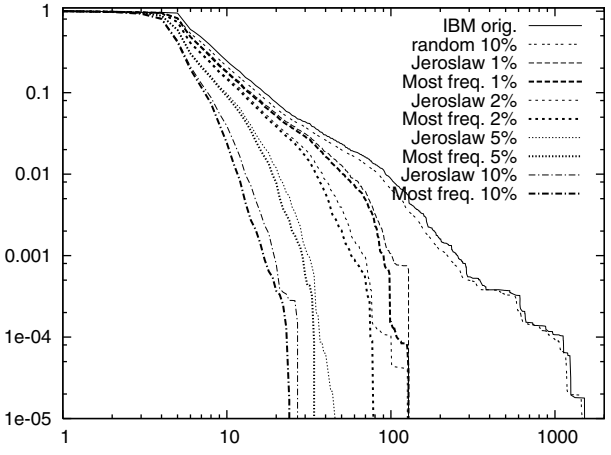

Fig. 4. Function $F_v(k)$ for IBM formulas where 1%, 2%, 5% and 10% of the variables have been instantiated using the random, Jeroslow-Wang and most-frequent strategies

Table 2. Analysis of the partially instantiated IBM formulas

	random				Jeroslow-Wang				Most freq.			
	Power-law		Exponential		Power-law		Exponential		Power-law		Exponential	
	α	d^{pow}	β	d^{exp}	α	d^{pow}	β	d^{exp}	α	d^{pow}	β	d^{exp}
0%	2.63	**0.027**	0.017	0.083	2.63	**0.027**	0.017	0.083	2.63	**0.027**	0.017	0.083
1%	2.56	**0.027**	0.017	0.078	2.72	**0.017**	0.046	0.036	2.79	**0.020**	0.052	0.034
2%	2.57	**0.025**	0.017	0.076	2.82	**0.015**	0.083	0.026	2.89	**0.012**	0.093	0.030
5%	2.59	**0.020**	0.018	0.075	3.27	0.029	0.218	**0.019**	3.39	0.029	0.250	**0.014**
10%	2.62	**0.021**	0.019	0.077	5.79	0.023	0.407	**0.014**	5.90	0.023	0.510	**0.021**

heuristics. Results are shown in Figure 4. As we can see, the random instantiation of variables has almost no effect on the probability distribution of variable occurrences $f_v(k)$. However, heuristics tend to remove variables with high number of occurrences. As a consequence, after partially instantiating around 5% of the variables, the formula looses its scale-free property, and seems to follow an exponential distribution (see Table 2).

4 Formulas during SAT Solvers Search

We want to answer the question of what kind of formula a state-of-the-art SAT solver sees during the search. The question is important because, if we implement solvers specialized in industrial instances (assuming that they are scale-free) during the execution of the solver, when some variables are already instantiated, we can be dealing with a not scale-free formula anymore. This means that, when

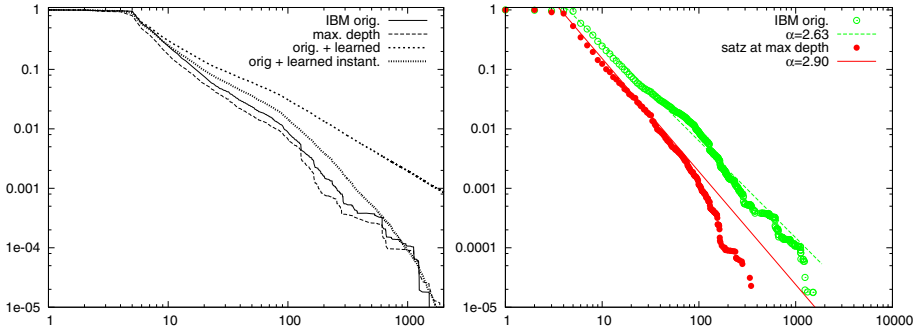

Fig. 5. Study of IBM formulas during the search. Left: with minisat, right: with satz.

a significant part of the variables are instantiated, the solver would do better changing its strategy.

Any complete SAT solver will backtrack immediately once it checks the current partial assignment is not consistent (in contrast to the setting in section 3), and second, state-of-the-art SAT solvers specialized on industrial instances augment the formula during search by adding new clauses, due to the learning mechanism they incorporate.

We have modified two very different SAT solvers, minisat [ES03]and satz [LA97]. Apart from the different heuristics and data structures these solvers incorporate, minisat applies a learning mechanism while satz does not.

We conducted experiments to answer the previous stated question, executing the solvers on each instance of the IBM family. We selected this family as the representative of scale-free formulas and random formulas as non scale-free formulas. The results reflect the average behavior of the family.

First, we study the formula under the longest partial assignment after 1000 seconds of search. Second we study, both the formula under the current partial assignment and the complete formula (original formula plus learned clauses) after 200000 decisions.

Figure 5 (left), shows the results of our experimentation on the IBM instances with minisat. As we can see, the scale-free structure is preserved in all cases. At maximal depth the distribution of frequencies is almost the same as in the original formulas. This seems to contradict the effect of partial assignments described in previous section but we have to remark that here the partial assignment is consistent. Moreover, it seems that the effect of the learned clauses makes the α exponent decrease.

We have repeated the same experiment with the same IBM formulas but after at most one hour of execution time of satz. Recall that here apart from applying a different heuristic we have not learned clauses. In Figure 5 (right), we can see that at the deepest assignments the formulas are still scale-free, although the exponent has been increased.

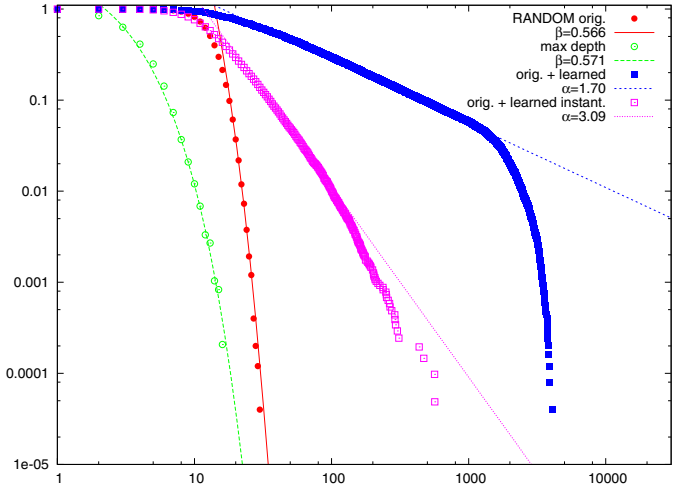

Fig. 6. Study of random formulas during the search

Therefore, very different SAT solvers seem to preserve the scale-free nature of formulas during their execution. Now the question is, what happens if we start with a random formula? For our experiment we have generated 50 random 3-CNF formulas of 500 variables at the phase transition point. Figure 6 shows the results. At the deepest decisions, after 1000 seconds, we see that the formulas still show an exponential decay with the same β as in the original formulas. However, after $2 \cdot 10^6$ decisions the formulas show a clear scale-free structure due to the addition of the learned clauses. As in the first experiment with the IBM family, the exponent α is smaller for the uninstantiated formula. To explain this phenomenon recall that the solvers like minisat, decide on the most active variables in learned clauses and learn clauses that contain decided variables. This creates an effect of *rich get richer* that has been proposed as a mechanism for creation of scale-free networks [BA99].

5 Conclusions

We have shown that most of the industrial formulas have a scale-free structure whereas random formulas have an Erdös-Rényi graph structure. This difference makes heuristics to perform better in industrial formulas than in random formulas.

We have observed that heuristically guided partial assignments (without guaranteeing consistency) make frequency distributions decay faster, destroying the power-law tail after instantiating 5% of the variables. However, if the assignments are consistent, as during the search in a SAT solver, we can instantiate up to 70% variables preserving the power-law tail (although increasing the exponent).

Finally, we have observed that the learning mechanism incorporated in modern SAT solvers tends to preserve the power-law distribution and even decrease its exponent.

References

[ABL09] Ansótegui, C., Bonet, M.L., Levy, J.: Towards industrial-like random SAT instances. In: Proc. of the 21st Int. Joint Conf. on Artificial Intelligence, IJCAI 2009 (2009)

[AJB99] Albert, R., Jeong, H., Barabási, A.-L.: The diameter of the www. Nature 401, 130–131 (1999)

[AJB00] Albert, R., Jeong, H., Barabási, A.-L.: Error and attack tolerance of complex networks. Nature 406, 378–482 (2000)

[BA99] Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)

[BDIS05] Boufkhad, Y., Dubois, O., Interian, Y., Selman, B.: Regular random k-sat: Properties of balanced formulas. J. of Automated Reasoning 35, 181–200 (2005)

[CSN07] Clauset, A., Shalizi, C.R., Newman, M.E.J.: Power-law distributions in empirical data. Arxiv, 0706.1062 (2007)

[ER59] Erdós, P., Rényi, A.: On random graphs. Publicationes Mathematicae 6, 290–297 (1959)

[ES03] Eén, N., Sörensson, N.: An extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) SAT 2003. LNCS, vol. 2919, pp. 502–518. Springer, Heidelberg (2004)

[GFSB04] Gomes, C.P., Fernández, C., Selman, B., Bessière, C.: Statistical regimes across constrainedness regions. In: Wallace, M. (ed.) CP 2004. LNCS, vol. 3258, pp. 32–46. Springer, Heidelberg (2004)

[GHPW99] Gent, I.P., Hoos, H.H., Prosser, P., Walsh, T.: Morphing: Combining structure and randomness. In: Proc. of the 16th Nat. Conf. on Artificial Intelligence, AAAI 1999, pp. 654–660 (1999)

[JW90] Jeroslow, R.G., Wang, J.: Solving propositional satisfiability problems. Annals of Mathematics and Artificial Intelligence 1, 167–187 (1990)

[KS03] Kautz, H.A., Selman, B.: Ten challenges redux: Recent progress in propositional reasoning and search. In: Rossi, F. (ed.) CP 2003. LNCS, vol. 2833, pp. 1–18. Springer, Heidelberg (2003)

[KS07] Kautz, H.A., Selman, B.: The state of SAT. Discrete Applied Mathematics 155(12), 1514–1524 (2007)

[LA97] Li, C.M., Anbulagan: Look-ahead versus look-back for satisfiability problems. In: Smolka, G. (ed.) CP 1997. LNCS, vol. 1330, pp. 341–355. Springer, Heidelberg (1997)

[LADW05] Li, L., Alderson, D., Doyle, J.C., Willinger, W.: Towards a theory of scale-free graphs: Definition, properties, and implications. Internet Mathematics 2(4), 431–523 (2005)

[Sel00] Selman, B.: Satisfiability testing: Recent developments and challenge problems. In: Proc. of the 15th Annual IEEE Symposium on Logic in Computer Science, LICS 2000, p. 178 (2000)

[SKM97] Selman, B., Kautz, H.A., McAllester, D.A.: Ten challenges in propositional reasoning and search. In: Proc. of the 15th Int. Joint Conf. on Artificial Intelligence, IJCAI 1997, pp. 50–54 (1997)

[Wal01] Walsh, T.: Search on high degree graphs. In: Proc. of the 17th Int. Joint Conf. on Artificial Intelligence, IJCAI 2001, pp. 266–274 (2001)

[WS98] Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393, 440–442 (1998)

[XHHLB08] Xu, L., Hutter, F., Hoos, H., Leyton-Brown, K.: SATzilla: Portfolio-based algorithm selection for SAT. J. of Artificial Intelligence Research 32, 565–606 (2008)