A Simple Model to Generate Hard Satisfiable Instances

Ke Xu[1] **Frédéric Boussemart**[2] **Fred Hemery**[2] **Christophe Lecoutre**[2]

[1]NLSDE Lab, School of Computers
Beihang University
Beijing 100083, China
kexu@nldse.buaa.edu.cn

[2]CRIL-CNRS FRE 2499,
Université d'Artois
Lens, France
⟨lastname⟩@cril.univ-artois.fr

Abstract

In this paper, we try to further demonstrate that the models of random CSP instances proposed by [Xu and Li, 2000; 2003] are of theoretical and practical interest. Indeed, these models, called RB and RD, present several nice features. First, it is quite easy to generate random instances of any arity since no particular structure has to be integrated, or property enforced, in such instances. Then, the existence of an asymptotic phase transition can be guaranteed while applying a limited restriction on domain size and on constraint tightness. In that case, a threshold point can be precisely located and all instances have the guarantee to be hard at the threshold, i.e., to have an exponential tree-resolution complexity. Next, a formal analysis shows that it is possible to generate forced satisfiable instances whose hardness is similar to unforced satisfiable ones. This analysis is supported by some representative results taken from an intensive experimentation that we have carried out, using complete and incomplete search methods.

1 Introduction

Over the past ten years, the study of phase transition phenomena has been one of the most exciting areas in Computer Science and Artificial Intelligence. Numerous studies have established that for many NP-complete problems (e.g., SAT and CSP), the hardest random instances occur, while a control parameter is varied accordingly, between an under-constrained region where all instances are almost surely satisfiable and an over-constrained region where all instances are almost surely unsatisfiable. In the transition region, there is a threshold where half the instances are satisfiable. Generating hard instances is important both for understanding the complexity of the problems and for providing challenging benchmarks [Cook and Mitchell, 1997].

Another remarkable progress in Artificial Intelligence has been the development of incomplete algorithms for various kinds of problems. And, since this progress, one important issue has been to produce hard satisfiable instances in order to evaluate the efficiency of such algorithms, as the approach that involves exploiting a complete algorithm in order to keep random satisfiable instances generated at the threshold can only be used for instances of limited size. Also, it has been shown that generating hard (forced) satisfiable instances is related to some open problems in cryptography such as computing a one-way function [Impagliazzo *et al.*, 1989; Cook and Mitchell, 1997].

In this paper, we mainly focus on random CSP (Constraint Satisfaction Problem) instances. Initially, four "standard" models, denoted A, B, C and D [Smith and Dyer, 1996; Gent *et al.*, 2001], have been introduced to generate random binary CSP instances. However, [Achlioptas *et al.*, 1997] have identified a shortcoming of all these models. Indeed, they prove that random instances generated using these models suffer from (trivial) unsatisfiability as the number of variables increases. To overcome the deficiency of these standard models, several alternatives have been proposed.

On the one hand, [Achlioptas *et al.*, 1997] have proposed a model E and [Molloy, 2003] a generalized model. However, model E does not permit to tune the density of the instances and the generalized model requires an awkward exploitation of probability distributions. Also, other alternatives correspond to incorporating some "structure" in the generated random instances. Roughly speaking, it involves ensuring that the generated instances be arc consistent [Gent *et al.*, 2001] or path consistent [Gao and Culberson, 2004]. The main drawback of all these approaches is that generating random instances is no more quite a natural and easy task.

On the other hand, [Xu and Li, 2000; 2003], [Frieze and Molloy, 2003] and [Smith, 2001] have revisited standard models by controlling the way parameters change as the problem size increases. The alternative model D scheme of [Smith, 2001] guarantees the occurrence of a phase transition when some parameters are controlled and when the constraint tightness is within a certain range. The two revised models, called RB and RD, of [Xu and Li, 2000; 2003] provide the same guarantee by varying one of two control parameters around a critical value that, in addition, can be computed. Also, [Frieze and Molloy, 2003] identify a range of suitable parameter settings in order to exhibit a non-trivial threshold of satisfiability. Their theoretical results apply to binary instances taken from model A and to "symmetric" binary instances from a so-called model B which, not corresponding to the standard one, associates the same relation with every constraint.

The models RB and RD present several nice features:

- it is quite easy to generate random instances of any arity as no particular structure has to be integrated, or property enforced, in such instances.
- the existence of an asymptotic phase transition can be guaranteed while applying a limited restriction on domain size and on constraint tightness. For instances involving constraints of arity k, the domain size is required to be greater than the kth root of the number of variables and the (threshold value of the) constraint tightness is required to be at most $\frac{k-1}{k}$.
- when the asymptotic phase transition exists, a threshold point can be precisely located, and all instances generated following models RB and RD have the guarantee to be hard at the threshold, i.e., to have an exponential tree-resolution complexity.
- it is possible to generate forced satisfiable instances whose hardness is similar to unforced satisfiable ones.

This paper is organized as follows. After introducing models RB and RD, as well as some theoretical results (Section 2), we provide a formal analysis about generating both forced and unforced hard satisfiable instances (Section 3). Then, we present the results of a large series of experiments that we have conducted (Section 4), and, before concluding, we discuss some related work (Section 5).

2 Theoretical background

A constraint network consists of a finite set of variables such that each variable X has an associated domain denoting the set of values allowed for X, and a a finite set of constraints such that each constraint C has an associated relation denoting the set of tuples allowed for the variables involved in C.

A solution is an assignment of values to all the variables such that all the constraints are satisfied. A constraint network is said to be satisfiable (sat, for short) if it admits at least a solution. The Constraint Satisfaction Problem (CSP), whose task is to determine if a given constraint network, also called CSP instance, is satisfiable, is NP-complete.

In this section, we introduce some theoretical results taken from [Xu and Li, 2000; 2003]. First, we introduce a model, denoted RB, that represents an alternative to model B. Note that, unlike model B, model RB allows selecting constraints with repetition. But the main difference of model RB with respect to model B is that the domain size of each variable grows polynomially with the number of variables.

Definition 1 (Model RB) *A class of random CSP instances of model RB is denoted RB(k,n,α,r,p) where:*

- $k \geq 2$ *denotes the arity of each constraint,*
- n *denotes the number of variables,*
- $\alpha > 0$ *determines the domain size* $d = n^\alpha$ *of each variable,*
- $r > 0$ *determines the number* $m = r.n.\ln n$ *of constraints,*
- $1 > p > 0$ *denotes the tightness of each constraint.*

To build one instance $P \in$ RB(k,n,α,r,p), we select with repetition m constraints, each one formed by selecting k distinct variables and $p.d^k$ distinct unallowed tuples (as p denotes a proportion).

When fixed, α and r give an indication about the growth of the domain sizes and of the number of constraints as n increases since $d = n^\alpha$ and $m = rn \ln n$, respectively. It is then possible, for example, to determine the critical value p_{cr} of p where the hardest instances must occur. Indeed, we have $p_{cr} = 1 - e^{-\alpha/r}$ which is equivalent to the expression of p_{cr} given by [Smith and Dyer, 1996].

Another model, denoted model RD, is similar to model RB except that p denotes a probability instead of a proportion. For convenience, in this paper, we will exclusively refer to model RB although all given results hold for both models.

In [Xu and Li, 2000], it is proved that model RB, under certain conditions, not only avoids trivial asymptotic behaviors but also guarantees exact phase transitions. More precisely, with Pr denoting a probability distribution, the following theorems hold.

Theorem 1 *If k, $\alpha > \frac{1}{k}$ and $p \leq \frac{k-1}{k}$ are constants then*

$$\lim_{n\to\infty} Pr[P \in RB(k,n,\alpha,r,p) \text{ is sat }] = \begin{cases} 1 & if\ r < r_{cr} \\ 0 & if\ r > r_{cr} \end{cases}$$

where $r_{cr} = -\frac{\alpha}{\ln(1-p)}$.

Theorem 2 *If k, $\alpha > \frac{1}{k}$ and $p_{cr} \leq \frac{k-1}{k}$ are constants then*

$$\lim_{n\to\infty} Pr[P \in RB(k,n,\alpha,r,p) \text{ is sat }] = \begin{cases} 1 & if\ p < p_{cr} \\ 0 & if\ p > p_{cr} \end{cases}$$

where $p_{cr} = 1 - e^{-\frac{\alpha}{r}}$.

Remark that the condition $p_{cr} \leq \frac{k-1}{k}$ is equivalent to $ke^{-\frac{\alpha}{r}} \geq 1$ given in [Xu and Li, 2000]. Theorems 1 and 2 indicate that a phase transition is guaranteed provided that the domain size is not too small and the constraint tightness or the threshold value of the constraint tightness not too large. As an illustration, for instances involving binary (resp. ternary) constraints, the domain size is required to be greater than the square (resp. cubic) root of the number of variables and the constraint tightness or threshold value of the tightness is required to be at most 50% (resp. $\approx 66\%$).

The following theorem establishes that unsatisfiable instances of model RB almost surely have the guarantee to be hard. A similar result for model A has been obtained by [Frieze and Molloy, 2003] with respect to binary instances.

Theorem 3 *If $P \in$ RB(k,n,α,r,p) and k, α, r and p are constants, then, almost surely[1], P has no tree-like resolution of length less than $2^{\Omega(n)}$.*

The proof, which is based on a strategy following some results of [Ben-Sasson and Wigderson, 2001; Mitchell, 2002], is omitted but can be found in [Xu and Li, 2003].

To summarize, model RB guarantees exact phase transitions and hard instances at the threshold. It then contradicts

[1]We say that a property holds almost surely when this property holds with probability tending to 1 as the number of variables tends to infinity.

the statement of [Gao and Culberson, 2004] about the requirement of an extremely low tightness for all existing random models in order to have non-trivial threshold behaviors and guaranteed hard instances at the threshold.

3 Generating hard satisfiable instances

For CSP and SAT, there is a natural strategy to generate *forced* satisfiable instances, i.e., instances on which a solution is imposed. It suffices to proceed as follows: first generate a random (total) assignment t and then generate a random instance with n variables and m constraints (clauses for SAT) such that any constraint violating t is rejected. t is then a *forced* solution. This strategy, quite simple and easy to implement, allows generating hard forced satisfiable instances of model RB provided that Theorem 1 or 2 holds. Nevertheless, this statement deserves a theoretical analysis.

Assuming that d denotes the domain size (d = 2 for SAT), we have exactly d^n possible (total) assignments, denoted by $t_1, t_2, \cdots, t_{d^n}$, and d^{2n} possible assignment pairs where an *assignment pair* $< t_i, t_j >$ is an ordered pair of two assignments t_i and t_j. We say that $< t_i, t_j >$ satisfies an instance if and only if both t_i and t_j satisfy the instance. Then, the expected (mean) number of solutions $E_f[N]$ for instances that are forced to satisfy an assignment t_i is:

$$E_f[N] = \sum_{j=1}^{d^n} \frac{\Pr[< t_i, t_j >]}{\Pr[< t_i, t_i >]}$$

where $\Pr[< t_i, t_j >]$ denotes the probability that $< t_i, t_j >$ satisfies a random instance. Note that $E_f[N]$ should be independent of the choice of the forced solution t_i. So we have:

$$E_f[N] = \frac{\sum_{1 \le i,j \le d^n} \Pr[< t_i, t_j >]}{d^n \Pr[< t_i, t_i >]} = \frac{E[N^2]}{E[N]}.$$

where $E[N^2]$ and $E[N]$ are, respectively, the second moment and the first moment of the number of solutions for random unforced instances.

For random 3-SAT, it is known that the strategy mentioned above is unsuitable as it produces a biased sampling of instances with many solutions clustered around t [Achlioptas *et al.*, 2000]. Experiments show that forced satisfiable instances are much easier to solve than unforced satisfiable instances. In fact, it is not hard to show that, asymptotically, $E[N^2]$ is exponentially greater than $E^2[N]$. Thus, the expected number of solutions for forced satisfiable instances is exponentially larger than the one for unforced satisfiable instances. It then gives a good theoretical explanation of why, for random 3-SAT, the strategy is highly biased towards generating instances with many solutions.

For model RB, recall that when the exact phase transitions were established [Xu and Li, 2000], it was proved that $E[N^2]/E^2[N]$ is asymptotically equal to 1 below the threshold, where almost all instances are satisfiable, i.e. $E[N^2]/E^2[N] \approx 1$ for $r < r_{cr}$ or $p < p_{cr}$. Hence, the expected number of solutions for forced satisfiable instances below the threshold is asymptotically equal to the one for unforced satisfiable instances, i.e. $E_f[N] = E[N^2]/E[N] \approx E[N]$. In other words, when using model RB, the strategy has almost no effect on the number of solutions and does not lead to a biased sampling of instances with many solutions.

In addition to the analysis above, we can also study the influence of the strategy on the distribution of solutions with respect to the forced solution. We first define the distance $d^f(t_i, t_j)$ between two assignments t_i and t_j as the proportion of variables that have been assigned a different value in t_i and t_j. We have $0 \le d^f(t_i, t_j) \le 1$.

For forced satisfiable instances of model RB, with $E_f^\delta[N]$ denoting the expected number of solutions whose distance from the forced solution (identified as t_i, here) is equal to δ, we obtain by an analysis similar to that in [Xu and Li, 2000]:

$$\begin{aligned} E_f^\delta[N] &= \sum_{j=1}^{d^n} \frac{\Pr[< t_i, t_j >]}{\Pr[< t_i, t_i >]} \quad \text{with } d^f(t_i, t_j) = \delta \\ &= \binom{n}{n\delta} (n^\alpha\text{-}1)^{n\delta} \left[\frac{\binom{n-n\delta}{k}}{\binom{n}{k}} + (1-p)\left(1 - \frac{\binom{n-n\delta}{k}}{\binom{n}{k}}\right) \right]^{rn \ln n} \\ &= \exp\left[n \ln n \left(r \ln \left(1 - p + p(1-\delta)^k\right) + \alpha\delta \right) + O(n) \right]. \end{aligned}$$

It can be shown, from the results in [Xu and Li, 2000] that $E_f^\delta[N]$, for $r < r_{cr}$ or $p < p_{cr}$, is asymptotically maximized when δ takes the largest possible value, i.e. $\delta = 1$.

For unforced satisfiable instances of model RB, with $E^\delta[N]$ denoting the expected number of solutions whose distance from t_i (not necessarily a solution) is equal to δ, we have:

$$\begin{aligned} E^\delta[N] &= \binom{n}{n\delta} (n^\alpha - 1)^{n\delta} (1-p)^{rn \ln n} \\ &= \exp\left[n \ln n \left(r \ln(1-p) + \alpha\delta \right) + O(n) \right]. \end{aligned}$$

It is straightforward to see that the same pattern holds for this case, i.e. $E^\delta[N]$ is asymptotically maximized when $\delta = 1$.

Intuitively, with model RB, both unforced satisfiable instances and instances forced to satisfy an assignment t are such that most of their solutions distribute far from t. This indicates that, for model RB, the strategy has little effect on the distribution of solutions, and is not biased towards generating instances with many solutions around the forced one.

For random 3-SAT, similarly, we can show that as r (the ratio of clauses to variables) approaches 4.25, $E_f^\delta[N]$ and $E^\delta[N]$ are asymptotically maximized when $\delta \approx 0.24$ and $\delta = 0.5$, respectively. This means, in contrast to model RB, that when r is near the threshold, most solutions of forced instances distribute in a place much closer to the forced solution than solutions of unforced satisfiable instances.

4 Experimental results

As all introduced theoretical results hold when $n \to \infty$, the practical exploitation of these results is an issue that must be addressed. In this section, we give some representative experimental results which indicate that practice meets theory even if the number n of variables is small. Note that different values of parameters α and r have been selected in order to illustrate the broad spectrum of applicability of model RB.

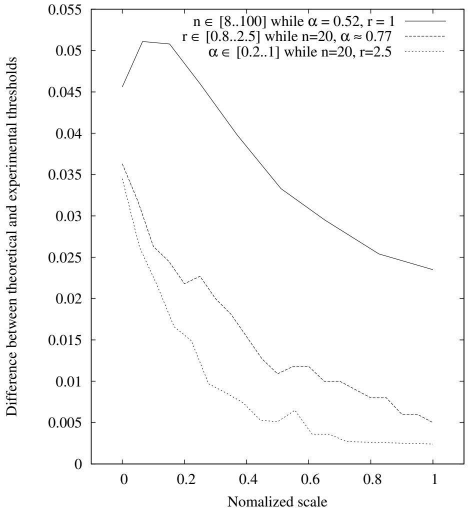

Figure 1: Difference between theoretical and experimental thresholds against α, r and n

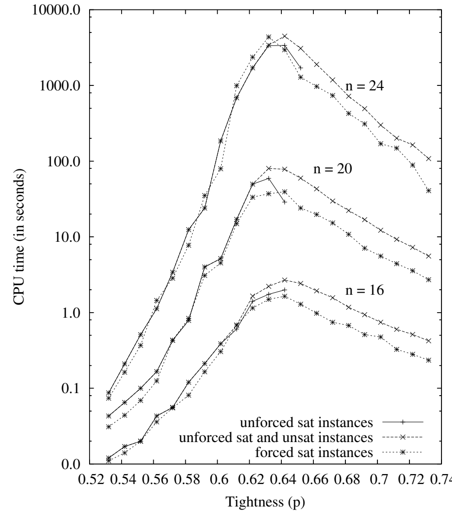

Figure 3: Mean search cost (50 instances) of solving instances in RB(3,$\{16, 20, 24\}$,1,1,p)

First, it is valuable to know in practice, to what extent, Theorems 1 and 2 give precise thresholds according to different values of α, r and n. The experiments that we have run wrt Theorem 2, as depicted in Figure 1, suggest that all other parameters being fixed, the greater the value of α, r or n is, the more precise Theorem 2 is. More precisely, in Figure 1, the difference between the threshold theoretically located and the threshold experimentally determined is plotted against $\alpha \in [0.2, 1]$ ($d \in [2..20]$), against $r \in [0.8, 2.5]$ ($m \in [50..150]$) and against $n \in [8..100]$. Note that the vertical scale refers to the difference in constraint tightness and that the horizontal scale is normalized (value 0 respectively corresponds to $n = 8$, $\alpha = 0.2$ and $r = 0.8$, etc.).

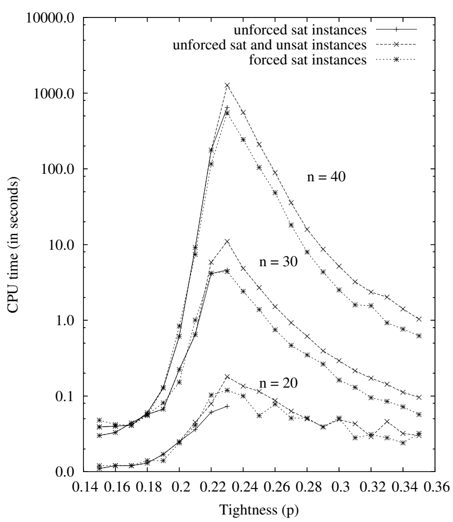

Figure 2: Mean search cost (50 instances) of solving instances in RB(2,$\{20, 30, 40\}$,0.8,3,p)

To solve the random instances generated by model RB, we have used a systematic backtracking search algorithm (MAC) and a local search algorithm (tabu search). Both algorithms have been equipped with a search heuristic that learns from conflicts [Boussemart *et al.*, 2004].

We have studied the difficulty of solving with MAC the binary instances of model RB generated around the theoretical threshold $p_{cr} \approx 0.23$ given by Theorem 2 for $k = 2$, $\alpha = 0.8$, $r = 3$ and $n \in \{20, 30, 40\}$. In Figure 2, it clearly appears that the hardest instances are located quite close to the theoretical threshold and that the difficulty grows exponentially with n (note the use of a log scale). It corresponds to a phase transition (not depicted here, due to lack of space). A similar behavior is observed in Figure 3 with respect to ternary instances generated around the theoretical threshold $p_{cr} \approx 0.63$ for $k = 3$, $\alpha = 1$, $r = 1$ and $n \in \{16, 20, 24\}$.

As the number and the distribution of solutions are the two most important factors determining the cost of solving satisfiable instances, we can expect, from the analysis given in Section 3, that for model RB, the hardness of solving forced satisfiable instances should be similar to that of solving unforced satisfiable ones. This is what is observed in Figure 2.

To confirm this, we have focused our attention to a point just below the threshold as we have then some (asymptotic) guarantee about the difficulty of both unforced and forced instances (see Theorems 5 and 6 in [Xu and Li, 2003]) and the possibility of generating easily unforced satisfiable instances. Figure 4 shows the difficulty of solving with MAC both forced and unforced instances of model RB at $p_{cr} - 0.01 \approx 0.40$ for $k = 2$, $\alpha = 0.8$, $r = 1.5$ and $n \in [20..50]$.

To confirm the inherent difficulty of the (forced and unforced) instances generated at the threshold, we have also studied the runtime distribution produced by a randomized search algorithm on distinct instances [Gomes *et al.*, 2004]. For each instance, we have performed 5000 independent runs. Figure 5 displays the survival function, which corresponds to the probability of a run taking more than x backtracks, of a

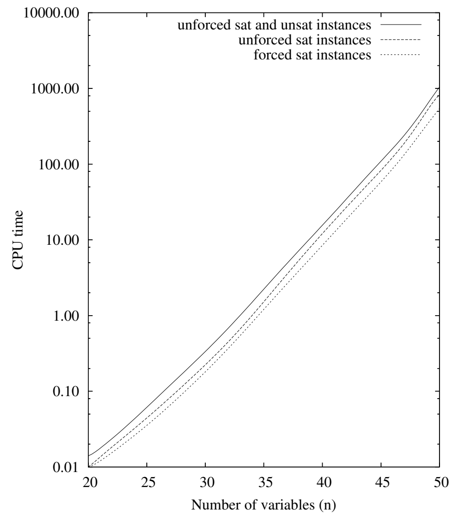

Figure 4: Mean search cost (50 instances) of solving instances in RB(2,[20..50],0.8,1.5,$p_{cr} - 0.01$)

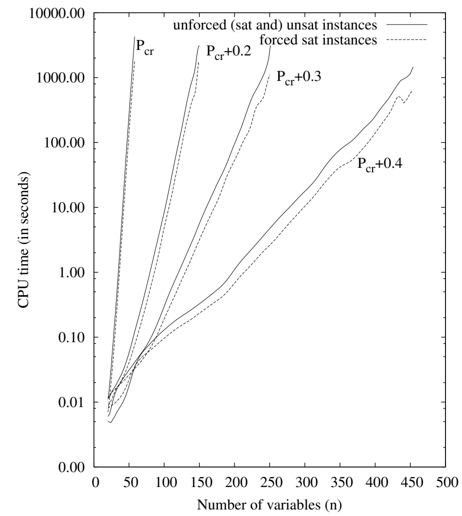

Figure 6: Mean search cost (50 instances) of solving instances in RB(2,[20..450],0.8,1.5,p)

randomized MAC algorithm for two representative instances generated at $p_{cr} \approx 0.41$ for $k = 2$, $\alpha = 0.8$, $r = 1.5$ and $n \in \{40, 45\}$. One can observe that the runtime distribution (a log-log scale is used) do not correspond to an heavy-tailed one, i.e., a distribution characterized by an extremely long tail with some infinite moment. It means that all runs behave homogeneously and, therefore, it suggests that the instances are inherently hard [Gomes *et al.*, 2004].

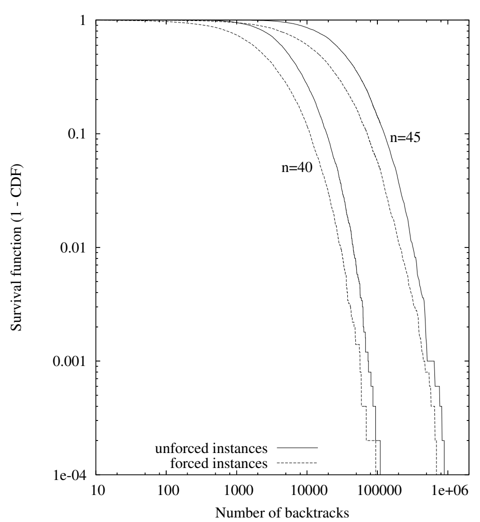

Figure 5: Non heavy-tailed regime for instances in RB(2,$\{40, 45\}$,0.8,1.5,$p_{cr} \approx 0.41$)

Then, we have focused on unforced unsatisfiable instances of model RB as Theorem 3 indicates that such instances have an exponential resolution complexity. We have generated unforced and forced instances with different constraint tightness p above the threshold $p_{cr} \approx 0.41$ for $k = 2$, $\alpha = 0.8$, $r = 1.5$ and $n \in [20..450]$. Figure 6 displays the search effort of a MAC algorithm to solve such instances against the number of variables n. It is interesting to note that the search effort grows exponentially with n, even if the exponent decreases as the tightness increases. Also, although not currently supported by any theoretical result (Theorems 5 and 6 of [Xu and Li, 2003] hold only for forced instances below the threshold) it appears here that forced and unforced instances have a similar hardness.

Finally, Figure 7 shows the results obtained with a tabu search with respect to the binary instances that have been previously considered with MAC (see Figure 2). The search effort is given by a median cost since when using an incomplete method, there is absolutely no guarantee of finding a solution in a given limit of time. Remark that all unsatisfiable (unforced) instances below the threshold have been filtered out in order to make a fair comparison. It appears that both complete and incomplete methods behave similarly. In Figure 7, one can see that the search effort grows exponentially with n and that forced instances are as hard as unforced ones.

5 Related work

As a related work, we can mention the recent progress on generating hard satisfiable SAT instances. [Barthel *et al.*, 2002; Jia *et al.*, 2004] have proposed to build random satisfiable 3-SAT instances on the basis of a spin-glass model from statistical physics. Another approach, quite easy to implement, has also been proposed by [Achlioptas *et al.*, 2004]: any 3-SAT instance is forced to be satisfiable by forbidding the clauses violated by both an assignment and its complement.

Finally, let us mention [Achlioptas *et al.*, 2000] which propose to build random instances with a specific structure, namely, instances of the Quasigroup With Holes (QWH) problem. The hardest instances belong to a new type of phase transition, defined from the number of holes, and coincide with the size of the backbone.

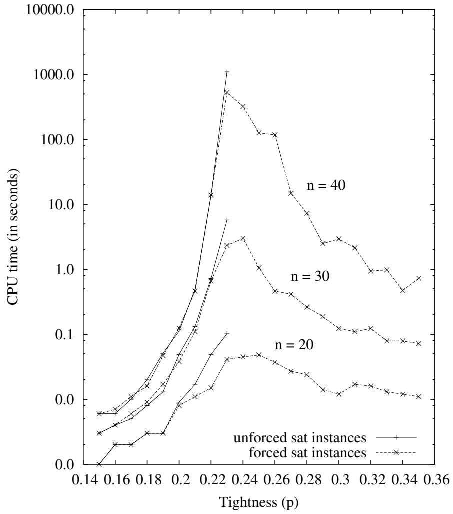

Figure 7: Median search cost (50 instances) of solving instances in RB(2,$\{20, 30, 40\}$,0.8,3,p) using a tabu search

6 Conclusion

In this paper, we have shown, both theoretically and practically, that the models RB (and RD) can be used to produce, very easily, hard random instances. More importantly, the same result holds for instances that are forced to be satisfiable. To perform our experimentation, we have used some of the most efficient complete and incomplete CSP solvers. We have also encoded some forced binary CSP instances of class RB(2,n,0.8,$0.8/ln\frac{4}{3}$,$p = p_{cr} = 0.25$) with n ranging from 40 to 59 into SAT ones (using the direct encoding method) and submitted them to the SAT competition 2004[2]. About 50% of the competing solvers have succeeded in solving the SAT instances corresponding to $n = 40$ ($d = 19$ and $m = 410$) whereas only one solver has been successful for $n = 50$ ($d = 23$ and $m = 544$).

Although there are some other ways to generate hard satisfiable instances, e.g. QWH [Achlioptas *et al.*, 2000] or 2-hidden [Achlioptas *et al.*, 2004] instances, we think that the simple and natural method presented in this paper, based on models with exact phase transitions and many hard instances, should be well worth further investigation.

Acknowledgments

The first author is partially supported by NSFC Grant 60403003 and FANEDD Grant 200241. Other authors have been supported by the CNRS, the "programme COCOA de la Région Nord/Pas-de-Calais" and by the "IUT de Lens".

References

[Achlioptas *et al.*, 1997] D. Achlioptas, L.M. Kirousis, E. Kranakis, D. Krizanc, M.S.O. Molloy, and Y.C. Stamatiou. Random constraint satisfaction: a more accurate picture. In *Proc. of CP'97*, pp. 107–120, 1997.

[Achlioptas *et al.*, 2000] D. Achlioptas, C. Gomes, H. Kautz, and B. Selman. Generating satisfiable problem instances. In *Proc. of AAAI'00*, pp. 256–301, 2000.

[Achlioptas *et al.*, 2004] D. Achlioptas, H. Jia, and C. Moore. Hiding satisfying assignments: two are better than one. In *Proc. of AAAI'04*, pp. 131–136, 2004.

[Barthel *et al.*, 2002] W. Barthel, A.K. Hartmann, M. Leone, F. Ricci-Tersenghi, M. Weigt, and R. Zecchina. Hiding solutions in random satisfiability problems: a statistical mechanics approach. *Physical review letters*, 88(18), 2002.

[Ben-Sasson and Wigderson, 2001] E. Ben-Sasson and A. Wigderson. Short proofs are narrow - resolution made simple. *Journal of the ACM*, 48(2):149–169, 2001.

[Boussemart *et al.*, 2004] F. Boussemart, F. Hemery, C. Lecoutre, and L. Sais. Boosting systematic search by weighting constraints. In *Proc. of ECAI'04*, pp. 146–150, 2004.

[Cook and Mitchell, 1997] S.A. Cook and D.G. Mitchell. Finding hard instances of the satisfiability problem: a survey. *DIMACS Ser. Disc. Math. & TCS*, 35, 1997.

[Frieze and Molloy, 2003] A.M. Frieze and M. Molloy. The satisfiability threshold for randomly generated binary constraint satisfaction problems. In *Proc. of Random'03*, pp. 275–289, 2003.

[Gao and Culberson, 2004] Y. Gao and J. Culberson. Consistency and random constraint satisfaction models with a high constraint tightness. In *CP'04*, pp. 17–31, 2004.

[Gent *et al.*, 2001] I.P. Gent, E. MacIntyre, P. Prosser, B.M. Smith, and T. Walsh. Random constraint satisfaction: flaws and structure. *Constraints*, 6(4):345–372, 2001.

[Gomes *et al.*, 2004] C.P. Gomes, C. Fernández, B. Selman, and C. Bessiere. Statistical regimes across constrainedness regions. In *Proc. of CP'04*, pp. 32–46, 2004.

[Impagliazzo *et al.*, 1989] R. Impagliazzo, L. levin, and M. Luby. Pseudo-random number generation from one-way functions. In *Proc. of STOC'89*, pp. 12–24, 1989.

[Jia *et al.*, 2004] H. Jia, C. Moore, and B. Selman. From spin glasses to hard satisfiable formulas. In *SAT'04*, 2004.

[Mitchell, 2002] D.G. Mitchell. Resolution complexity of random constraints. In *Proc. of CP'02*, pp. 295–309, 2002.

[Molloy, 2003] M. Molloy. Models for random constraint satisfaction problems. *SIAM Journal of computing*, 32(4):935–949, 2003.

[Smith and Dyer, 1996] B.M. Smith and M.E. Dyer. Locating the phase transition in binary constraint satisfaction problems. *Artificial Intelligence*, 81:155–181, 1996.

[Smith, 2001] B.M. Smith. Constructing an asymptotic phase transition in random binary constraint satisfaction problems. *TCS*, 265:265–283, 2001.

[Xu and Li, 2000] K. Xu and W. Li. Exact phase transitions in random constraint satisfaction problems. *Journal of Artificial Intelligence Research*, 12:93–103, 2000.

[Xu and Li, 2003] K. Xu and W. Li. Many hard examples in exact phase transitions with application to generating hard satisfiable instances. CoRR Report cs.CC/0302001, 2003.

[2] `http://www.nlsde.buaa.edu.cn/~kexu/benchmarks/benchmarks.htm`