Exponentially hard problems are sometimes polynomial, a large deviation analysis of search algorithms for the random satisfiability problem, and its application to stop-and-restart resolutions

Simona Cocco[1] and Rémi Monasson[2]
[1]*CNRS–Laboratoire de Dynamique des Fluides Complexes, 3 rue de l'Université, 67000 Strasbourg, France*
[2]*CNRS–Laboratoire de Physique Théorique de l'ENS, 24 rue Lhomond, 75005 Paris, France*
(Received 20 February 2002; published 19 September 2002)

A large deviation analysis of the solving complexity of random 3-satisfiability instances slightly below threshold is presented. While finding a solution for such instances demands an exponential effort with high probability, we show that an exponentially small fraction of resolutions require a computation scaling linearly in the size of the instance only. This exponentially small probability of easy resolutions is analytically calculated, and the corresponding exponent is shown to be smaller (in absolute value) than the growth exponent of the typical resolution time. Our study therefore gives some theoretical basis to heuristic stop-and-restart solving procedures, and suggests a natural cutoff (the size of the instance) for the restart.

DOI: 10.1103/PhysRevE.66.037101 PACS number(s): 89.20.Ff, 05.20.−y, 89.70.+c, 89.75.Hc

Computational problems are usually divided into two classes. They are easy if there exists a solving procedure whose running time grows at most polynomially with the size of the problem, or hard if no such algorithm is believed to exist, and the best avalaible procedures may require an exponentially growing time [1]. The polynomial vs exponential classification was enriched in the past decades through the derivation of quantitative bounds on resolution complexity, and the study of average performances of various resolution algorithms for computational tasks with model input distributions.

In this paper, we show that, though the polynomial/exponential dichotomy certainly applies to the typical resolution complexity of computational problem, it may not be so for large deviations from typical behavior. Typically exponentially hard problems may sometimes be solved in polynomial time, a phenomenon that we take advantage of to accelerate resolution drastically.

We concentrate here on random 3-Satisfiability (3-SAT), a paradigm of hard combinatorial problems recently studied using statistical physics tools and concepts [2,3] as e.g., number partitioning [4], vertex cover [5], etc. to which computer scientists have devoted a great attention over the past years [6–8]. An instance of random 3-SAT is defined by a set of M constraints (clauses) on N boolean (i.e., true or false) variables. Each clause is the logical OR of three randomly chosen variables, or of their negations. The question is to decide whether there exists a logical assignment of the variables satisfying all the clauses (called solution). The best currently known algorithm to solve 3-SAT is the Davis-Putnam-Loveland-Logemann (DPLL) procedure [6] (Fig. 1). The sequence of assignments of variables made by DPLL in the course of instance solving can be represented as a search tree (Fig. 2), whose size Q (number of nodes) is a convenient measure of the complexity. For very large sizes ($M,N\to\infty$ at fixed ratio $\alpha=M/N$), some static and dynamical phase transitions arise [2,7–10]. Instances with a ratio of clauses per variable $\alpha>\alpha_C\simeq 4.3$ are almost surely unsatisfiable and obtaining proofs of refutation require an exponential effort [3,11]. Below the static threshold α_C, instances are almost surely satisfiable, but finding a solution may be easy or hard, depending on the value of α. A dynamical transition [3,12] takes place at $\alpha_L\simeq 3.003$ (for the heuristic used by DPLL shown in Fig. 1) separating a polynomial regime ($\alpha<\alpha_L: Q\sim N$, search tree A on Fig. 2) [13,14] from an exponential regime ($\alpha>\alpha_L: Q\sim 2^{N\omega}$, search tree B). This pattern of complexity, and the value of $\omega(\alpha)$ were obtained through an analysis of DPLL dynamics, reminiscent of real-space renormalization in statistical physics [3]. DPLL generates some dynamical flow of the instance, whose trajectory lies in the phase diagram of the $2+p$-SAT model [2], an extension of 3-SAT, where $p\leqslant 1$ is the fraction of 3-clauses (Fig. 2).

(1) Choose a variable and its value (T,F) according to some heuristic rule **(Split);**

(2) Analyze the implications of the choice on all the clauses :

a: If all clauses are satisfied, then stop: a solution is found,
b: If a contradiction appears, negate the last chosen variable and go to 2 **(Backtracking),**
If all previously chosen variables have already been negated once, then stop: unsatisfiability is proven,
c: if there is at least one clause with one variable, fix the variable to satisfy the clause and go to 2 **(Unit Propagation),**
d: Else go to 1.

FIG. 1. DPLL algorithm. When a variable has been chosen at step (1), e.g., $x=T$, at step (2) some clauses are satisfied, e.g., $C=(x \text{ OR } y \text{ OR } z)$ and eliminated, others are reduced, e.g., $C=(\text{not } x \text{ OR } y \text{ OR } z)\to C=(y \text{ OR } z)$. If some clauses include one variable only, e.g., $C=y$, the corresponding variable is automatically fixed to satisfy the clause ($y=T$). This unit-clause (UC) propagation (2c) is repeated up to the exhaustion of all UC. Contradictions result from the presence of two opposite UC, e.g., $C=(y), C'=(\text{not } y)$. A solution is found when no clauses are left. The heuristic studied here is the generalized UC (GUC) rule: a variable is chosen at step (1) from one of the 2-clauses, or from a 3-clause if no 2-clause is present, and fixed to satisfy the clause. The search process of DPLL is represented by a tree (Fig. 2) whose nodes correspond to (1), and edges to (2). Branch extremities are marked with contradictions C (2b), or by a solution S (2a).

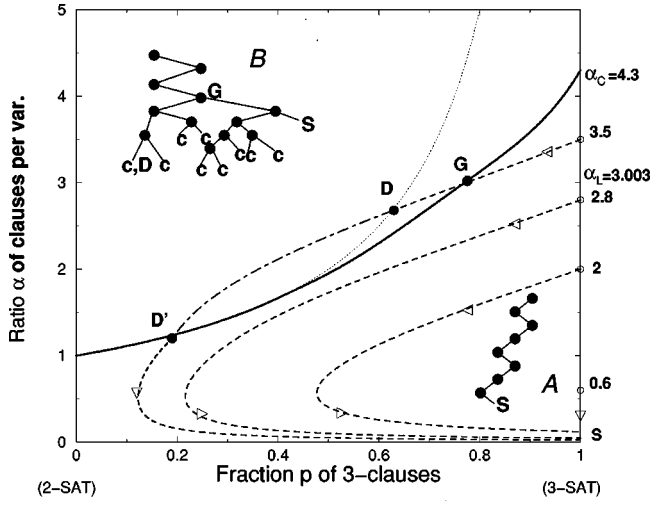

FIG. 2. Phase diagram of $2+p$-SAT and first branch trajectories for satisfiable instances. The threshold line $\alpha_C(p)$ (bold full line) separates SAT (lower part of the plane) from unSAT (upper part) phases. Departure points for DPLL trajectories are located on the 3-SAT vertical axis (empty circles). Arrows indicate the direction of ''motion'' along trajectories (dashed curves) parametrized by the fraction t of variables set by DPLL. For small ratios $\alpha<\alpha_L$ ($\simeq 3.003$ for the GUC heuristic), branch trajectories remain confined in the SAT phase, end in S of coordinates (1,0), where a solution is found (with a search process reported on tree A). For ratios $\alpha_L<\alpha<\alpha_C$, the branch trajectory intersects the threshold line at some point G. A contradiction almost surely arises before the trajectory crosses the dotted curve $\alpha=1/(1-p)$ (point D), and extensive backtracking up to G permits to find a solution (search tree B). With exponentially small probability, the trajectory (dot-dashed curve, full arrow) is able to cross the ''dangerous'' region where contradictions are likely to occur (search tree similar to A); it then exits from this region (point D') and ends up with a solution (lowest dashed trajectory).

We focus hereafter on the large deviations of complexity in the upper SAT phase $\alpha_L<\alpha<\alpha_C$. Using numerical experiments and analytical calculations, we show that, though complexity Q almost surely grows as $2^{N\omega}$, there is a finite, but exponentially small, probability $2^{-N\zeta}$ that Q is bounded from above by N only. In other words, finding solutions to these SAT instances is almost always exponentially hard, and very rarely easy (polynomial time). Taking advantage of the fact that ζ is smaller than ω, we show how systematic restarts of the heuristic may decrease substantially the overall search cost. Our study therefore gives some theoretical basis to stop-and-restart (SR) solving procedures empirically known to be efficient [15], and suggests a natural cutoff for the stop.

Distributions of resolution times Q for $\alpha=3.5$ are reported in Fig. 3. The histogram of $\omega=(\log_2 Q)/N$ essentially exhibits a narrow peak (left side) followed by a wider bump (right side). As N grows, the right peak acquires more and more weight, while the left peak progressively disappears. The center of the right peak gets slightly shifted to the left, but reaches a finite value $\omega^*\simeq 0.035$ as $N\to\infty$ [3]. This right peak thus corresponds to the core of exponentially hard resolutions: resolutions of instances almost surely require a time scaling similar to $2^{N\omega^*}$ as the size N gets large, in agreement with the above discussion.

On the contrary, the abscissa of the maximum of the left peak vanishes as $\log_2 N/N$ when the size N increases, indicating that the left peak accounts for polynomial (linear) resolutions. Its maximum is located at $Q/N\simeq 0.2-0.25$, with weak dependence on N. The cumulative probability P_{lin} to have a complexity Q less than, or equal to N, decreases exponentially: $P_{lin}=2^{-N\zeta}$ with $\zeta\simeq 0.011\pm 0.001$ (Inset of Fig. 4). In the following, we will concentrate on linear resolutions only (an analysis of the distribution of exponential resolutions for the problem of the vertex covering of random graphs [5] can be found in [16]).

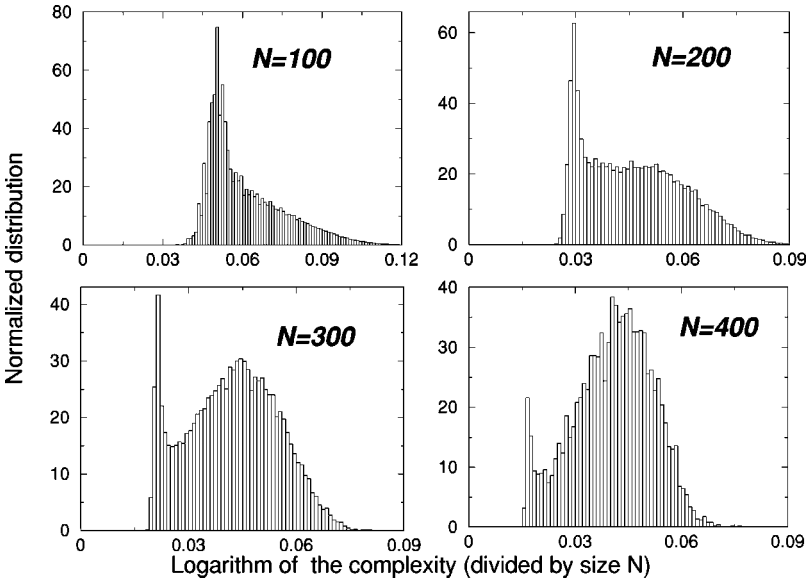

FIG. 3. Histograms of the logarithm ω of the complexity Q (base 2, and divided by N) for $\alpha=3.5$ and different sizes N. Many instances are drawn randomly, and for each sample, DPLL is run until a solution is found (very few unsatisfiable instances can be present and are discarded). Because of the large N limit this instance-to-instance distribution is equivalent to the run-to-run distribution on the same instance, coming from the randomness in the assignments of variables by DPLL; the latter is indeed almost surely independent of the particular instances. See Ref. [7] and the inset of Fig. 4 for a proof of this equivalence.

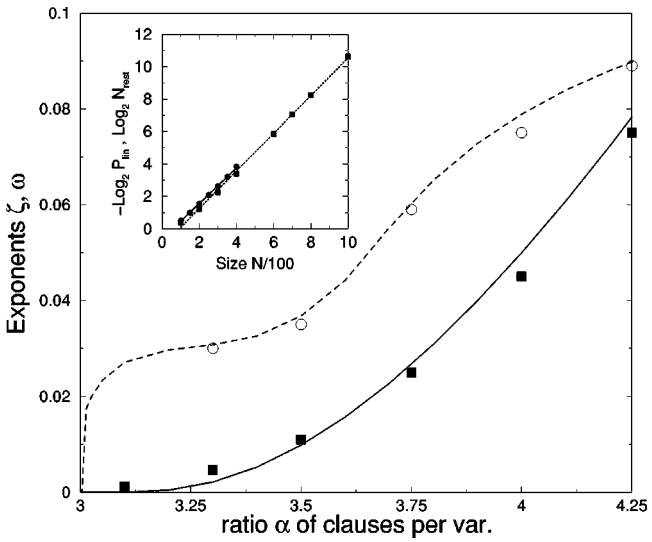

FIG. 4. Log of complexity using DPLL (ω: simulations, circles; theory from [3], dotted line) and SR (ζ: simulations, squares; theory, full line) as a function of ratio α. Inset: minus log of the cumulative probability P_{lin} of complexities $Q \leqslant N$ as a function of the size for $100 \leqslant N \leqslant 400$ (full line); log of the number of restarts N_{rest} necessary to find a solution for $100 \leqslant N \leqslant 1000$ (dotted line) for $\alpha = 3.5$. Slopes are $\zeta = 0.0011$ and $\overline{\zeta} = 0.00115$, respectively.

Further numerical investigations show that, in easy resolutions, the solution is essentially found at the end of the first branch, with a search tree of type *A*, and not *B*, in Fig. 2. Easy resolution trajectories are able to cross the ''dangerous'' region extending beyond point *D* in Fig. 2, contrary to most trajectories which backtrack earlier. Beyond *D*, unit clauses (UC) indeed accumulate. Their number C_1 becomes of the order of N ($C_1/N \simeq 0.022$ for $\alpha = 3.5$), and the probability that the branch survives, i.e., that no two contradictory UC are present, is exponentially small in *N*, in agreement with the scaling of the left peak weight in Fig. 3.

Calculation of ζ requires the analysis of the first descent in the search tree (Fig. 2). Each time DPLL assigns a variable, some clauses are eliminated and others are reduced or left unchanged (Fig. 1). We thus characterize an instance by its state $\mathbf{C} = (C_1, C_2, C_3)$, where C_j is the number of j clauses it includes ($j = 1,2,3$). Initially, $\mathbf{C} = (0,0,\alpha_0 N)$. Let us call $\tilde{P}(\mathbf{C};T)$ the probability that the assignment of T variables has produced no contradiction and an instance with state $\mathbf{C}$. $\tilde{P}$ obeys a Markovian evolution $\tilde{P}(\mathbf{C};T+1) = \Sigma_{\mathbf{C}'} K(\mathbf{C},\mathbf{C}';T)\tilde{P}(\mathbf{C}';T)$ where the entries of the transition matrix K read

$$K(\mathbf{C},\mathbf{C}';T) = B_{p_3}^{C_3',\Delta_3} \sum_{w_3=0}^{\Delta_3} B_{1/2}^{\Delta_3,w_3} \sum_{z_2=0}^{C_2'-v} B_{p_2}^{C_2'-v,z_2} \sum_{w_2=0}^{z_2} B_{1/2}^{z_2,w_2} \times \sum_{z_1=0}^{C_1'-1+v} \frac{1}{2^{z_1}} B_{p_1}^{C_1'-1+v,z_1} \times \delta_{z_2-\Delta_2-w_3+v}\,\delta_{z_1-\Delta_1-w_2+1-v} \quad (1)$$

where δ_C denotes the Kronecker delta function: $\delta_C = 1$ if $C = 0$; 0 otherwise. Variables appearing in Eq. (1) are as follows. $\Delta_j \equiv C_j' - C_j$, $v \equiv \delta_{C_1'}$, z_j (w_j) is the number of j clauses which are satisfied (reduced to $j-1$ clauses) when the $(T+1)$th variable is assigned. These are stochastic variables drawn from several binomial distributions $B_p^{L,K} \equiv \binom{L}{K} p^K (1-p)^{L-K}$. Parameter $p_j = j/(N-T)$ equals the probability that a j clause contains the variable just assigned by DPLL.

The introduction of the generating function $P(\mathbf{y};T) = \Sigma_{\mathbf{C}} e^{\mathbf{y}\cdot\mathbf{C}} \tilde{P}(\mathbf{C},T)$, allows us to express the evolution equation for the state probabilities in a compact manner,

$$P(\mathbf{y};T+1) = e^{-g_1(\mathbf{y})} P(\mathbf{g}(\mathbf{y});T) + (e^{-g_2(\mathbf{y})} - e^{-g_1(\mathbf{y})}) \times P(-\infty, g_2(\mathbf{y}), g_3(\mathbf{y});T), \quad (2)$$

where $g_j(\mathbf{y}) = y_j + \ln[1+\gamma_j(\mathbf{y})/N]$, $\gamma_j(\mathbf{y}) \equiv \gamma_j(y_j, y_{j-1}) = j[e^{-y_j}(1+e^{y_{i-1}})/2 - 1]/(1-t)$ for $j = 1,2,3$ ($y_0 \equiv -\infty$).

From Eq. (1), the C_js undergo $O(1)$ changes each time a variable is fixed. After $T = tN$ assignments, the densities $c_j = C_j/N$ of clauses have been modified by $O(1)$. This translates into large N ansatz for the state probability, $\tilde{P}(\mathbf{C};T) = e^{N\varphi(\mathbf{c};t)}$, and for the generating function, $P(\mathbf{y};T) = e^{N\tilde{\varphi}(\mathbf{y};t)}$, up to non exponential in N terms. φ and $\tilde{\varphi}$ are simply related to each other through a Legendre transform. In particular, $\varphi(\mathbf{0};t)$ is the logarithm of the probability (divided by N) that the first branch has not been hit by any contradiction after a fraction t of variables have been assigned. The most probable values of the densities $c_j(t)$ of j clauses are equal to the partial derivatives of φ in $\mathbf{y} = \mathbf{0}$.

When DPLL starts running on a 3-SAT instance, clauses are reduced and some UC generated. Next they are eliminated through UC propagation, and splits occur frequently (Fig. 1). The number C_1 of UC remains bounded with respect to the instance size *N*, and the density $c_1(t) = \partial\varphi/\partial y_1$ identically vanishes. φ does not depend on y_1, and $\varphi(y_2, y_3; t)$ obeys the following partial differential equation (PDE)

$$\frac{\partial\varphi}{\partial t} = -y_2 + \gamma_2(y_2,y_2;t)\frac{\partial\varphi}{\partial y_2} + \gamma_3(y_2,y_3;t)\frac{\partial\varphi}{\partial y_3}. \quad (3)$$

We have solved analytically PDE (3) with initial condition $\varphi(\mathbf{y};0) = \alpha_0 y_3$. The high probability scenario is obtained for $y_2 = y_3 = 0$: $\varphi(0,0;t) = 0$ indicates that the probability of survival of the branch is not exponentially small in N [14], and the partial derivatives $c_2(t), c_3(t)$ give the typical densities of 2- and 3-clauses, in full agreement with Chao and Franco's result [13]. We plot in Fig. 2 the corresponding resolution trajectories for various initial ratios α_0, using the change of variables $p = c_3/(c_2+c_3)$, $\alpha = (c_2+c_3)/(1-t)$. Furthermore, our calculation provides a complete description of rare deviations of the resolution trajectory from its highly probable locus, giving access to the exponentially small probabilities that p, α differ from their most probable values at ''time'' *t*.

The assumption $C_1 = O(1)$ breaks down for the most probable trajectory at some fraction t_D, e.g., $t_D \simeq 0.308$ for $\alpha_0 = 3.5$ at which the trajectory hits point D on Fig. 2. Beyond D, UC accumulate, and the probability of survival of the first branch becomes exponentially small in N. Variables are almost always assigned through unit propagation: $c_1 > 0$. φ now depends on y_1 and, from Eq. (1), obeys the following PDE:

$$\frac{\partial \varphi}{\partial t} = -y_1 + \sum_{j=1}^{3} \gamma_j(\mathbf{y};t) \frac{\partial \varphi}{\partial y_j}. \tag{4}$$

We have solved PDE (4) through an expansion of φ in powers of $\mathbf{y}$, whose coefficients obey, from Eq. (4), a set of coupled linear ordinary differential equations (ODEs). The initial conditions for the ODEs are chosen to match the expansion of the exact solution of Eq. (3), that is, the typical trajectory and its large deviations, at time t_D. The quality of the approximation improves rapidly with the order k of the expansion, and no difference was found between $k=3$ and $k=4$ results. c_1 first increases, reaches its top value $(c_1)^{max}$, then decreases and vanishes at $t_{D'}$ when the trajectory comes out from the dangerous region where contradictions almost surely occur (Fig. 2). The probability of survival scales as $2^{-N\zeta}$ for large N, with $\zeta = -\varphi(\mathbf{0};t_{D'})/\ln 2$. The calculated values of $\zeta \simeq 0.01, (c_1)^{max} \simeq 0.022$ and $Q/N \simeq 0.21$ for $\alpha = 3.5$ are in very good agreement with numerics. Figure 4 shows the agreement between theory and simulations over the whole range $\alpha_L < \alpha < \alpha_C$.

The existence of rare but easy resolutions suggests the use of a systematic SR procedure to speed up resolution: if a solution is not found before N splits, DPLL is stopped and rerun after some random permutations of the variables and clauses. The expected number N_{rest} of restarts necessary to find a solution being equal to the inverse probability $1/P_{lin}$ of linear resolutions, the resulting complexity should scale as $N 2^{0.011N}$ for $\alpha = 3.5$, with an exponential gain with respect to DPLL one-run complexity, $2^{0.035N}$. Results of SR experiments are reported in Fig. 4. The typical number $N_{rest} = 2^{N\bar{\zeta}}$ of restarts grows indeed exponentially with the size N, with a rate $\bar{\zeta} = 0.0115 \pm 0.001$ equal to ζ. The equality between ζ and $\bar{\zeta}$ confirms the equivalence between sample-to-sample (Fig. 3) and run-to-run (at fixed sample) distributions of complexities for large sizes [8].

Performances are greatly enhanced by the use of SR (see Fig. 4 for comparison between ζ and ω). While with usual DPLL, we were able to solve instances with 500 variables in about one day of CPU for $\alpha = 3.5$, instances with 1000 variables were solved with SR in 15 minutes on the same computer.

Our work therefore provides some theoretical support to the use of SR [15,16], and in addition suggests a natural cutoff at which the search is halted and restarted, the determination of which is usually widely empirical and problem dependent. If a combinatorial problem is efficiently solved (polynomial time) by a search heuristic for some values of the control parameter of the input distribution, there might be an exponentially small probability that the heuristic is still successful (in polynomial time) in the range of parameters where resolution almost surely requires massive backtracking and exponential effort. When the decay rate of the polynomial time resolution probability ζ is smaller than the growth rate ω of the typical exponential resolution time, SR with a cutoff in the search equal to a polynomial of the instance size will lead to an exponential speed up of resolutions.

We thank A. Montanari and R. Zecchina for discussions and communication of their results prior to publication, and the French Ministry of Research for partial financial support through the ACI Jeunes Chercheurs ''Algorithmes d'Optimisation et Systèmes Désordonnés Quantiques.''

[1] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity* (Prentice-Hall, Englewood Cliffs, NJ, 1982).
[2] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky, Nature (London) **400**, 133 (1999).
[3] S. Cocco and R. Monasson, Phys. Rev. Lett. **86**, 1654 (2001); Eur. Phys. J. B **22**, 505 (2001).
[4] S. Mertens, Phys. Rev. Lett. **81**, 4281 (1998); **84**, 1347 (2000).
[5] M. Weigt and A. Hartmann, Phys. Rev. Lett. **84**, 6118 (2000); **86**, 1658 (2001).
[6] J. Gu, P.W. Purdom, J. Franco, and B.W. Wah, in *DIMACS Series on Discrete Mathematics and Theoretical Computer Science* (American Mathematical Society, Providence, 1997), Vol. 35, pp. 19–151.
[7] D. Mitchell, B. Selman, and H. Levesque, in *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)* (AAAI Press, Cambridge, MA, 1992), pp. 440–446.
[8] T. Hogg and C.P. Williams, Artif. Intel. **69**, 359 (1994); I.P. Gent and T. Walsh, *ibid.* **70**, 335 (1994); B. Selman and S. Kirkpatrick, *ibid.* **81**, 273 (1996).
[9] B. Selman and S. Kirkpatrick, Science **264**, 1297 (1994).
[10] E. Friedgut, J. Am. Math. Soc. **12**, 1017 (1999).
[11] V. Chvàtal, E. Szmeredi, J. Assoc. Comput. Mach. **35**, 759 (1988).
[12] C. Coarfa, D.D. Dernopoulos, A. San Miguel Aguirre, D. Subramanian, and M.Y. Vardi, Lect. Notes Comput. Sci. **1894**, 143 (2000).
[13] M.T. Chao and J. Franco, Inf. Sci. (N.Y.) **51**, 289 (1990).
[14] A. Frieze and S. Suen, J. Algorithms **20**, 312 (1996).
[15] O. Dubois, P. André, Y. Boufkhad, and J. Carlier, in *DIMACS Series in Discrete Math and Computer Science* (Ref. [6]), pp. 415–436; C.P. Gomes, B. Selman, N. Crato, and H. Kautz, J. Automated Reasoning **24**, 67 (2000).
[16] A. Montanari, R. Zecchina, Phys. Rev. Lett. **88**, 178701 (2002).