Phase Transitions and Complexity in Computer Science: an overview of the statistical physics approach to the random satisfiability problem.

Giulio Biroli [1] *, Simona Cocco [2] † and Rémi Monasson [3,4] ‡

[1] *Center for Material Theory, Department of Physics and Astronomy, Rutgers University, Piscataway, NJ 08854 USA,*
[2] *Department of Physics, The University of Illinois at Chicago, 845 W. Taylor St., Chicago IL 60607, USA,*
[3] *CNRS-Laboratoire de Physique Théorique de l'ENS, 24 rue Lhomond, 75005 Paris, France,*
[4] *The James Franck Institute, The University of Chicago, 5640 S. Ellis Av., Chicago IL 60637, USA.*

Phase transitions, ubiquitous in condensed matter physics, are encountered in computer science too. The existence of critical phenomena has deep consequences on computational complexity, that is the resolution times of various optimization or decision problems. Concepts and methods borrowed from the statistical physics of disordered and out–of–equilibrium systems shed new light on the dynamical operation of solving algorithms.

I. INTRODUCTION

"You are chief of protocol for the embassy ball. The crown prince instructs you either to invite Peru or to exclude Qatar. The queen asks you to invite either Qatar or Romania or both. The king, in a spiteful mood, wants to snub either Romania or Peru or both. Is there a guest list that will satisfy the whims of the entire royal family?"

This example, quoted from Brian Hayes' excellent review article [1], illustrates the so-called satisfiability (SAT) problem, of central importance in computer science from both theoretical and practical points of view [2]. An instance of the problem SAT is a set of logical constraints involving Boolean variables (having only two possible values, true or false). The question is to know whether there exists at least one solution (configuration of the variables) satisfying the set of constraints. From a formal standpoint, the chief of protocol must solve a SAT instance comprising three Boolean variables p, q, r (true, respectively false, if the ambassador of the corresponding country will be present, resp. absent at the ball) and three constraints, also called clauses, that can be written as $(p \vee \bar{q})$, $(q \vee r)$, $(\bar{p} \vee \bar{r})$. The overbar denotes the logical negation; for instance $\bar{p}$ is true if the Peruvian ambassador is not invited. The symbol $\vee$ stands for the logical inclusive OR operator. A few seconds of reflection are sufficient in the present case to scan the $8 = 2^3$ lists of guests and select some satisfactory combination. But no efficient procedure is known to solve large SAT instances. Even with the best available algorithms, the time of resolution may dramatically (exponentially) grow with the size of the instance, *i.e.* the number of variables and clauses. SAT is a paradigm of hard computational problem and is thus at the root of complexity theory in computer science [2,3], see Appendix A. Understanding why SAT is hard and improving the performances of its solving procedures would also benefit to the resolution of the optimization tasks frequently associated to industrial applications.

II. RANDOM 3-SAT AND ITS THRESHOLD.

With such objectives in mind, computer scientists started a few years ago to study a particularly interesting class of instances of the SAT problem, called random 3-SAT [4]. This is a simplified, model class of instances which depend on two parameters only: the number N of variables and the ratio α of clauses per variable. Each clause contains three variables or their logical negations. The αN clauses are randomly drawn, independently of each other. The demands of the royal family correspond to an instance of the 2-SAT problem, with $N = 3$ variables and ratio $\alpha = 1$.

Intuitively, α is the ratio of the number of constraints divided by the number of degrees of freedom. It is a sensible guess that, at fixed N, a randomly drawn instance of 3-SAT has solutions if α is small (underconstrained situation), and has none if α is large (overconstrained situation). The probability $P(\alpha)$ that an instance be satisfiable as a function of α, and for different sizes N is shown Figure 1. In addition to being a decreasing function of α as expected from above, a striking phenomenon happens as N grows. An abrupt decrease of P takes place at a critical value $\alpha_c \simeq 4.3$ of the ratio. Instances with less than $\alpha_c N$ clauses are almost surely satisfiable, whereas the ones with more than $\alpha_c N$ clauses have almost never any solution.

This transition is accompanied by an outburst of resolution hardness (computational complexity). Figure 2 shows the median time necessary to solve a random 3-SAT instance, that is to find a solution (below the threshold), or check there is none (above the threshold). The curves of Figure 2 obviously depend upon the specific solving procedures used. However the general pattern of the complexity as a function of α, namely the presence of successive regions corresponding to, roughly speaking, easy, hard, and less hard resolutions is a generic feature valid for commonly used solving algorithms.

Why should physicists get interested in SAT, a strictly mathematic problem by definition? First of all, 3-SAT is complex. Mathematicians have been able to establish rigorous bounds on the threshold, $3.14 \leq \alpha_c \leq 4.51$, but the

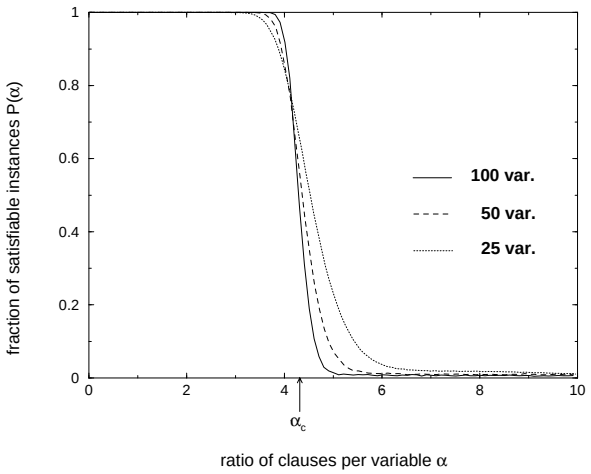

FIG. 1. Probability $P(\alpha)$ that a set of αN clauses be satisfiable for different sizes N (number of variables). This probability is obtained from numerical experiments, consisting in generating randomly 10,000 instances of the 3-SAT problem for each values of α and N, and measuring the fraction of them being satisfiable. In the large N limit, the threshold $\alpha_c \simeq 4.3$ separates the satisfiable ($\alpha < \alpha_c$, $P = 1$) and the non satisfiable ($\alpha > \alpha_c$, $P = 0$) phases. A similar transition also takes place for random 2-SAT instances, where all clauses involve two variables; the threshold is then equal to $\alpha_c = 1$.

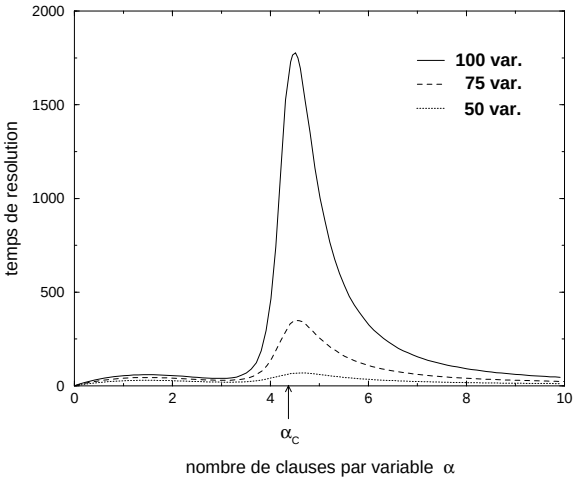

FIG. 2. Resolution time of 3-SAT instances as a function of the ratio of clauses per variable α and for three different sizes. The time is measured from the size of the search tree generated by the solving procedure (Figure 6 and 7). Data correspond to the median resolution time of 10,000 instances; the average time may be somewhat larger due to the presence of rare, exceptionally hard instances. The computational complexity is maximal at α_c. It is exponential in the vicinity of the threshold and in the unsatisfiable phase, but less and less as α increases.

Dictionary

K-SAT	*statistical physics*
Boolean variable x = True , False	*Ising spin* $s_x = +1, -1$
Clauses	*Couplings and fields acting on spins*
Number of clauses violated by a logical configuration	*Energy E of the spins configuration*
some examples:	
2 SAT x or $\bar{y}$	$E = \frac{1}{4}(1-s_x)(1+s_y)$
(x or $\bar{y}$) and ($\bar{x}$ or z)	$E = \frac{1}{4}(1-s_x)(1+s_y) + \frac{1}{4}(1+s_x)(1-s_y)$
3 SAT x or $\bar{y}$ or z	$E = \frac{1}{8}(1-s_x)(1+s_y)(1-s_z)$
Minimal number of violated clauses	*Ground state energy*
The problem is satisfiable	*Ground state energy = 0*
The problem is not satisfiable	*Ground state energy > 0*

FIG. 3. Dictionary translating the 3-SAT problem (left) into statistical physics language (right).

exact value seems out of reach with the available probabilistic techniques. Approximation methods developed by physicists in the course of the study of phase transitions allow not only to estimate the value of the threshold but also to unveil the microscopic structure of the solutions and the mechanism leading to their disappearance. Concepts as phase diagrams, dynamical renormalization flows, Markovian evolution also prove to be useful to understand the operation of algorithms. Statistical physics therefore offers precious intuitions, a few of which have already been confirmed by mathematicians or have paved the way to the derivation of new rigorous results.

III. A SPIN GLASS AT ZERO TEMPERATURE

In order to unveil the relationship between random satisfiability problems and statistical physics, let us translate in Fig. 3 the 3-SAT problem and its ingredients (Boolean variables, clauses, ...) in a language more familiar to physicists (Ising spins, interactions, ...). The main idea is to introduce an energy function, which is merely a cost function equal to the number of unsatisfied clauses for each variables-spins configuration, and study the ground state properties. The satisfiability of the 3-SAT problem is therefore equivalent to the vanishing of ground state energy.

Form a physical point of view, the 3-SAT energy function is similar to the Hamiltonian of spin glasses. These systems, characterized by a frozen-in structural *disorder*,

have been intensively studied in the last twenty years. Spin glasses are materials weakly diluted with magnetic ions. The random positions of the ions induce random (in sign and strength) magnetic interactions. The lack of homogeneity results in an extremely complex energy landscape. In particular the enormous number of metastable states makes the low temperature behavior very unusual and interesting from a fundamental point of view. In the 3-SAT case, the disorder is induced by the random clauses which make the problem more and more frustrated as α increases (See Appendix B for an introduction to frustration).

The physical scenario we obtained for random 3-SAT is the following. For $\alpha > \alpha_c$ the ground state energy becomes positive with probability one, whereas it vanishes for $\alpha < \alpha_c$. The exact solution of random 3-SAT remains an open (and very hard) problem yet. However, combining some exact results with approximated techniques we were able to obtain $\alpha_c \simeq 4.48$ which is just 5% larger than the numerical value. Some aspects of the transition are indeed surprising. For α slightly lower than α_c the number of solutions remains enormous $\sim 2^{0.14\,N}$ [8], and then vanishes abruptly at α_c. Hence, increasing slightly the number of clauses is enough to make all solutions disappear simultaneously! Moreover, immediately beyond the transition, a finite fraction of over-constrained variables assuming the same value in each optimal configuration abruptly emerges. This backbone of variables plays the role of an order parameter: it vanishes for $\alpha < \alpha_c$ and jumps discontinuously to a value $\sim 15\%$ at the transition. Hence, the 3-SAT transition may be interpreted as a first-order transition.

This abrupt transition is clearly different from the one displayed by 2-SAT, a less interesting problem from a computational point of view, since it can be solved in a polynomial time (Appendix A). The 2-SAT transition is of the second order, and the onset of the backbone is continuous. To get a better understanding of the relationship between computational complexity and the features of the phase transition, we have introduced a mixed model, called 2+p-SAT, which interpolates between 2 and 3-SAT. This model is defined by taking $(1-p)\alpha N$ random clauses of length 2 (*i.e.* with 2 variables) and $p\alpha N$ clauses of length 3. For $p = 0$ and for $p = 1$ one recovers exactly 2-SAT and 3-SAT respectively. A tricritical point at $p_0 \simeq 0.4$ appears, which separates the phase diagram (p, α) in two parts: a 3-SAT-like region ($p_0 < p < 1$) with a discontinuous jump in the backbone and a 2-SAT-like region with a continuous transition ($0 < p < p_0$), see Fig. 4.

Before discussing how the phase diagram of 2+p-SAT allows one to understand the computation complexity of 3-SAT, let us stress that random 3-SAT displays a very interesting physical behavior already in the satisfiable phase. In fact, for $\alpha > \alpha_s \simeq 3.96$ (but always lower than α_c), the space of solutions is no more homogeneous. Instead, it breaks up in an enormous number of disjoint groups of solutions (Fig. 5) [9]. These groups have a

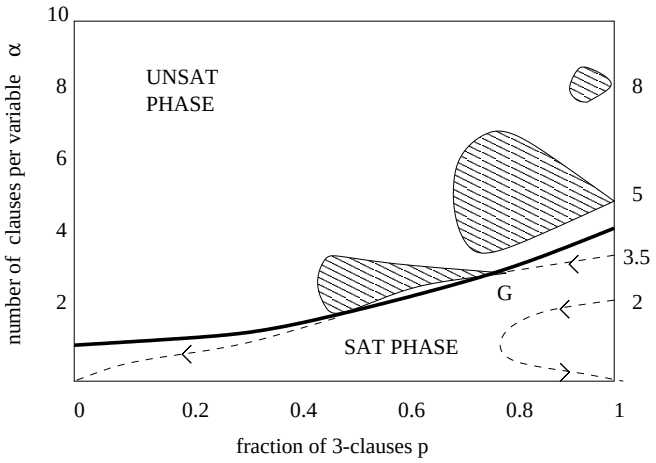

FIG. 4. Phase diagram of 2+p-SAT and dynamical trajectories of DPLL. The threshold line $\alpha_c(p)$ (bold full line) separates sat (lower part of the plane) from unsat (upper part) phases. Extremities lie on the vertical 2-SAT (left) and 3-SAT (right) axis at coordinates $(p = 0, \alpha_c = 1)$ and $(p = 1, \alpha_c \simeq 4.3)$ respectively. Departure points for DPLL trajectories are located on the 3-SAT vertical axis and the corresponding values of α are explicitly given. Hatched regions symbolize tree trajectories in the unsat region, and dashed curve represent branch trajectories in the sat phase. Arrows indicate the direction of "motion" along branch trajectories parameterized by the fraction t of variables set by DPLL. For small ratios $\alpha < \alpha_L$, branch trajectories remain confined in the sat phase and end in S of coordinates $(1, 0)$, where a solution is found. At $\alpha_L \simeq 3.003$, the single branch trajectory hits tangentially the threshold line in T of coordinates $(2/5, 5/3)$ (T depends *a priori* on the heuristics used, but lies very close to the tricritical point). In the intermediate range $\alpha_L < \alpha < \alpha_c$, the branch trajectory intersects the threshold line at some point G (that depends on α). A dense tree then grows in the unsat phase, as happens when 3-SAT departure ratios are above threshold $\alpha > \alpha_c \simeq 4.3$. DPLL then reaches back the highest backtracking node in the search tree, that is, the first node when $\alpha > \alpha_C$, or node G for $\alpha_L < \alpha < \alpha_C$. In the latter case, a solution can be reached from a new descending branch while, in the former case, unsatisfiability is proven, see Figure 7.

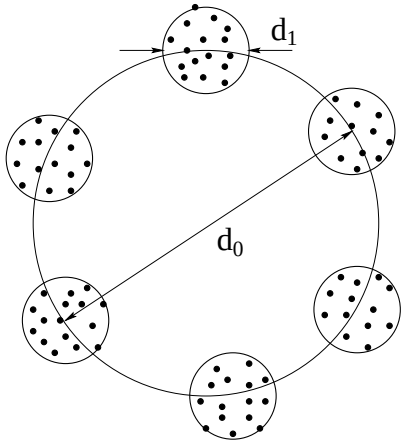

FIG. 5. Schematic representation of the structure of the solutions space for the random 3-SAT at $\alpha > \alpha_s$. The space of optimal configurations is divided in an exponential number of disjoint groups. The typical distance (measured as the fraction of different spins) between two optimal configurations is equal to d_1 inside the same group, and to d_0 between two groups.

multifractal distribution that can be computed analytically. As a consequence, the statistical physics approach allows one to analyze in detail the structure of the space of solutions and not only the global properties of 3-SAT.

IV. SOLVING ALGORITHM AND TRAJECTORIES.

Why is the complexity of 3-SAT solving affected by the phase transition? To answer this question, let us first briefly expose the ubiquitous solving procedure of Davis–Putnam-Loveland–Logemann (DPLL). DPLL is an exhaustive search procedure operating by trials and errors, the sequence of which can be graphically represented by a search tree (Figure 6). Briefly, **(1)** a node in the tree corresponds to a choice of a variable. An outgoing branch, attached to the value the latter (true or false), grows from the node. Thus a node gives birth to two branches at most. **(2)** Along a branch, the logical implications of the last choice are analyzed. **(3)** If a contradiction (violated clause) arises, the last choice is modified (backtracking of the tree) and the procedure goes on along a new branch (step 2); if all clauses are satisfied, a solution is found and the search process is over; otherwise, the algorithm resumes to step 1.

Computational complexity, the amount of operation necessary to solve the instance, is given by the size of the search tree, *i.e.* the number of nodes it contains. Performances can be improved by designing sophisticated heuristic rules for choosing variables (step 1).

The DPLL algorithm gives rise to a complex, non Markovian dynamical process, differing largely from the common modelization of the evolution of physical systems e.g. Monte Carlo dynamics. Our study of the operation of DPLL is based on the following, elementary observation (Figure 5). The recursive procedure of DPLL turns the initial instance of the 3-SAT problem into a mixed instance with clauses of length two and three (clauses of length unity are eliminated at step 2 of the procedure, see above). Therefore, if the initial instance is symbolized by a point of coordinates $p = 1, \alpha$ in the phase diagram of Figure 4, this representative point will evolve under the dynamical action of the algorithm and define a trajectory. Three ranges of initial ratios α must be distinguished on the vertical axis $p = 1$ of the 3-SAT problem.

When $\alpha < \alpha_L = 3.003$, the search tree essentially reduces to a unique branch (Figure 7A), which ends up with a solution [11]. The trajectory corresponding to this branch depends on the heuristic rule implemented in DPLL (Figure 5) and is schematically represented on Figure 4. It first heads to the left and then reverses to the right until reaching a point on the 3-SAT axis at a small ratio without ever leaving the sat region. Further action of DPLL leads to a rapid elimination of the remaining clauses and the trajectory ends up at the right lower corner $p = 1, \alpha = 0$, where a solution is achieved. Thus, in the range of ratios $\alpha < \alpha_L \simeq 3.003$, 3-SAT is easy to solve: the computational complexity scales linearly with the size N (Figure 2).

For ratios above threshold ($\alpha > \alpha_C \simeq 4.3$), instances almost never have a solution but a considerable amount of backtracking is necessary before proving that clauses are incompatible. As shown in Figure 7B, a generic unsat tree includes many branches. The number of nodes grows exponentially with N, and is conveniently expressed as $2^{N\omega}$. The sequence of points (p, α) characterizing the evolution of the 2+p-SAT instance solved by DPLL does not define a line any longer, but rather a patch, or cloud of points with a finite extension in the phase diagram of Figure 4.

The value of ω can be calculated analytically as a function of the initial ratio, see Appendix C. The main idea is to monitor the growth of the dominant, that is, most numerous branches (Appendix C) in the tree until their extinction resulting from the onset of contradictions. Results are in excellent quantitative agreement with the numerical experiments of Figure 2 [10]. Main qualitative features are: ω is positive in the whole unsat phase $\alpha > \alpha_c$, indicating that computational complexity is exponential in this region; ω is maximal at the threshold and gets smaller and smaller as α increases (as the instance is more and more over-constrained, it becomes easier and easier to detect a group of mutually incompatible clauses).

The intermediate region $\alpha_L < \alpha < \alpha_C$ juxtaposes the two previous behaviors, see tree Figure 7C. The branch trajectory, started from the point $(p = 1, \alpha)$ corresponding to the initial (and satisfiable) 3-SAT instance, crosses the critical line $\alpha_c(p)$ at some point G (Figure 4). The algorithm then enters the unsat phase and generates 2+p-

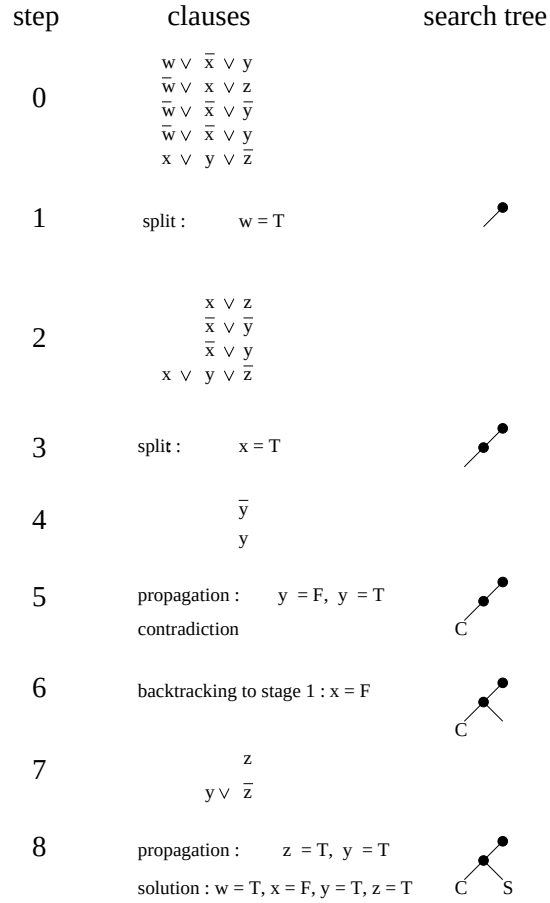

FIG. 6. Example of 3–SAT instance and Davis–Putnam-Loveland–Logemann resolution. **Step 0.** The instance consists of $M = 5$ clauses involving $N = 4$ variables x, y, w, z, which can be assigned to true (T) or false (F). $\bar{w}$ means (NOT w) and v denotes the logical OR. The search tree is empty. **1.** DPLL randomly selects a variable among the shortest clauses and assigns it to satisfy the clause it belongs to, e.g. w =T (splitting with the Generalized Unit Clause –GUC– heuristic) [6,9]. A node and an edge symbolizing respectively the variable chosen (w) and its value (T) are added to the tree. **2.** The logical implications of the last choice are extracted: clauses containing w are satisfied and eliminated, clauses including $\bar{w}$ are simplified and the remaining ones are left unchanged. If no unitary clause (*i.e.* with a single variable) is present, a new choice of variable has to be made. **3.** Splitting takes over. Another node and another edge are added to the tree. **4.** Same as step 2 but now unitary clauses are present. The variables they contain have to be fixed accordingly. **5.** The propagation of the unitary clauses results in a contradiction. The current branch dies out and gets marked with C. **6.** DPLL backtracks to the last split variable (x), inverts it (F) and creates a new edge. **7.** Same as step 4. **8.** The propagation of the unitary clauses eliminates all the clauses. A solution S is found and the instance is satisfiable. For an unsatisfiable instance, unsatisfiability is proven when backtracking (see step 6) is not possible anymore since all split variables have already been inverted. In this case, all the nodes in the final search tree have two descendent edges and all branches terminate by a contradiction C.

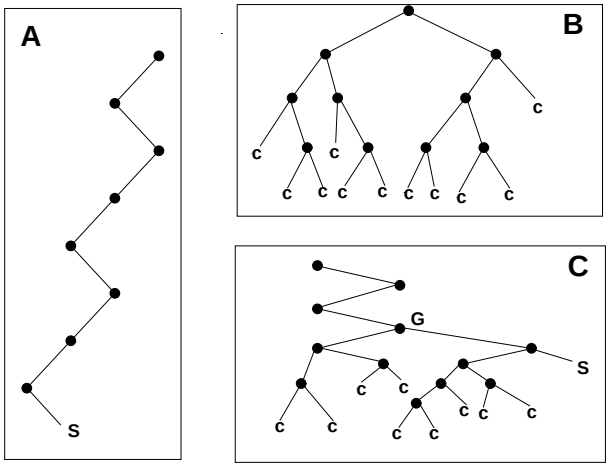

FIG. 7. Types of search trees generated by the DPLL solving procedure. Nodes (black dots) stand for the choices of variables made by the heuristic, and edges between nodes denote the elimination of unitary clauses. **A.** *simple branch:* the algorithm finds easily a solution without ever backtracking. **B.** *dense tree:* in the absence of solution, the algorithm builds a "bushy" tree, with many branches of various lengths, before stopping. **C.** *mixed case, branch + tree:* if many contradictions arise before reaching a solution, the resulting search tree can be decomposed in a single branch followed by a dense tree. The junction G is the highest backtracking node reached back by DPLL.

SAT instances with no solution. A dense subtree, that DPLL has to go through entirely, forms beyond G (Figure 4). There is no need for DPLL to backtrack above G (Figure 7C), since nodes immediately above G are located in the sat phase and carry 2+p-SAT instances with solutions. DPLL will eventually reach a solution (point S in Figure 7C). The corresponding branch shown Figure 4 is highly non typical and does not contribute to the complexity. The resolution complexity of the initial 3-SAT instance thus reduces, to the exponential order in N, to the complexity of the critical unsatisfiable 2+p-SAT instance located in G, and can be analytically predicted from our theory (Appendix C). In this intermediary region, random 3-SAT instances are exponentially hard to solve.

V. CONCLUSION AND PERSPECTIVES.

The existence of a phase transition and its influence on the computational complexity is not unique to random 3-SAT but rather constitutes a common property shared by many optimization problems on random structures, e.g. the vertex cover problem [12] or the coloring of random graphs, ...

The statistical physics approach allows for a non rigorous but precious understanding of these critical phenomena. It also emphasizes the relevance of the concept

of average, or typical-case complexity (occurring with high probability), in contradistinction with the theory of complexity in computer science essentially based on worst case analysis. As an illustration, the theory of NP-completeness [2] strongly suggests that no algorithm exists capable of solving *all 3-SAT instances* in a polynomial time but we have seen that DPLL solves *almost all random 3-SAT instances* with ratios $\alpha < \alpha_L$ in a linear time!

The goal of the present study, which is already partially concretized if one refers to the success of optimization techniques issued from physics e.g. the simulated annealing, would be to take advantage of this new understanding to improve the current algorithms. Figure 4 suggests for instance some general line to improve the heuristics rules to assign variables: computational complexity would not be exponential below the threshold if the heuristics trajectories were able to avoid crossing the critical line. Recently a brand new heuristic using the concept of backbone has permitted to solve successfully instances with 700 variables at the threshold [13], while previous heuristics could not solve instances with sizes larger than 500 (a huge improvement, indeed, since resolution times increases exponentially at this point).

In turn, the objects and issues studied in computer science prove to be an original and fruitful source of theoretical problems whose unusual dynamical or statistical aspects may entice physicists to deepen their understanding of the frontiers of the statistical physics of disordered and off-equilibrium systems.

Acknowledgments: The results reported in this article are the fruit of various joint works with our collaborators, S. Kirkpatrick, B. Selman, M. Weigt, R. Zecchina whom we are grateful to.

APPENDIX A: NOTION OF COMPUTATIONAL COMPLEXITY.

Computational tasks are classified according to their complexity, *i.e.* the number of elementary operations (additions, substitutions, comparisons,...) that an algorithm has to effectuate in order to solve them. Usually a problem whose complexity grows polynomially with its size (the number of variables and clauses used to define the problem) is considered to be easy, whereas exponential growths correspond to hard problems.

Usually, to show that a problem is easy, one has to find explicitly a resolution algorithm that requires a polynomial number of operations. A practical and important example is the sorting problem which consists in classifying N numbers according to their absolute value. The best algorithms are able to execute this task in $\propto N \log N$ operations for any initial list. It has been proved that 2-Satisfiability, where each clause contain two variables, belongs to this class of polynomial (easy) problems.

This seems not to be the case of 3-Satisfiability. As for hundreds of optimization problems, e.g. the famous traveling salesman problem (which consists in finding the shortest tour going through N cities), no polynomial algorithm has ever been found for 3-SAT in spite of an intense research in the last decades. In practice, all the algorithms used to solve these problems may require an exponential running time. Moreover, it has been proved that if one of these problems reveals eventually to be easy then all the other ones would be easy too. This further diminishes the possibility to find such a polynomial algorithm for one of these problems, which are called in the literature NP-complete.

However, from a rigorous point of view, proving the impossibility to solve easily (polynomially) NP-complete problems has been claimed to be one of the seven most fundamental questions in mathematics at the Clay congress held recently in Paris. Hence, the existence of an extremely ingenious algorithm solving easily 3-SAT is not ruled out so far.

APPENDIX B: SOLUTIONS OF SAT AND GROUND STATES OF SPIN GLASSES.

The number of unsatisfied clauses for the problem of the embassy ball can be written as (see dictionary in figure 3):

$$E = -\frac{1}{4}\left(s_p\, s_q - s_q\, s_r - s_p\, s_r - 3\right) \quad .$$

The cost function E defines a Hamiltonian for the three Ising spins s_i placed on a triangle with vertices $i = p, q, r$. This model is a spin glass prototype. First of all, the couplings do not have the same sign as in the Ising model case. Secondly, these signs are quenched random variables depending on the particular realization of the clauses. Finally, even if in the case of embassy ball the three couplings can be simultaneously satisfied (for this one has to take a spin configuration such that the product of the two spins on the vertices of a bond equals the sign of their coupling), this is not generally true. Consider for example the energy function obtained by changing the sign of one of the three couplings. In this case there exists no spin configuration which satisfies simultaneously all the couplings and the system is *frustrated*. Disorder and frustration are the two fundamental characteristics of spin glasses.

Actually, very interesting spin glass models has been defined and studied in order to analyze the SAT problems by statistical physics tools. In these models each spin interacts with a finite and random number of other spins. Hence, the underlying lattice structure is random graph characterized by local random connectivities which depend on the particular realization of the clauses. Moreover, 3-SAT gives rise to a pretty unusual disordered

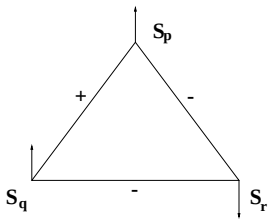

FIG. 8. Spins and couplings for the embassy ball problem.

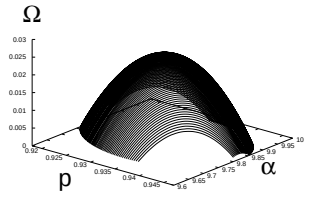

FIG. 9. Snapshot of the histogram-surface $\Omega(p,\alpha)$, at a given instant "t".

model, since the spins interact three by three, and not via a usual pairwise interaction as for 2-SAT (Figure 3).

In spin glass systems, thermodynamic observables (free energy, energy,...) depends on the particular disorder realization. However, they become self-averaging in the thermodynamic limit, *i.e.* do not fluctuate from sample to sample. As a consequence, one can obtain the free energy density by computing the average of the logarithm of the partition function over the disorder distribution.

This is in general a difficult computation that requires the replica method, widely used to study disordered systems. By this technique we have computed, through some approximations, the critical value α_c, and we have analyzed certain microscopic properties of the solutions (fraction of fixed variables, number of different spins from a solution to the other,...).

APPENDIX C: GROWTH OF THE SEARCH TREE AND RESOLUTION TIME.

DPLL solving procedure builds up the search tree in a sequential way, adding more and more nodes, branches up to the completion of the tree. As shown Figure 5, each node corresponds to an instance of the 2+p-SAT model, with defining parameters p, α. We have imagined a different building up, that results in the same complete tree but can be mathematically analyzed: the tree grows in parallel, layer after layer, with two branches simultaneously growing out of each node. Each branch obeys DPLL evolution rules (choice of variables, simplification of 1-clauses, and halt if a contradiction arises), such that the final tree is identical to the obtained through the original, sequential process. At each instant t of the building process, the tree consists of many branches, the extremities of which have distinct characteristic parameters. Let us consider the histogram $\mathcal{B}(p,\alpha;t)$ of the values of these parameters. This histogram grows exponentially with the size N of the instance. It is thus convenient to consider its logarithm $\Omega(p,\alpha;t) = \frac{1}{N}\log_2 \mathcal{B}(p,\alpha;t)$, which defines a surface in the three-dimensional space p, α, Ω (Figure 8).

The addition of a new layer to the search tree changes the characteristic parameters of the existing branches, and leads to the appearance, or halt of other branches. As a result, the search tree evolution can be recast in terms of a out-of-equilibrium growth process of a bidimensional surface Ω, where the "time" t is simply the depth of the tree. The equation of growth reads,

$$\frac{\partial \Omega}{\partial t} = \mathcal{H}\left[\frac{\partial \Omega}{\partial p}, \frac{\partial \Omega}{\partial \alpha}, p, \alpha, t\right] \quad , \tag{C1}$$

where $\mathcal{H}$ depends upon the heuristic rule used by DPLL. The number of branches $\mathcal{B}(t)$ at depth t is equal, to the dominant order in N, to $2^{N\,\Omega^*(t)}$ where $\Omega^*(t)$ is the height of the top of the surface. The growth is interrupted at depth t_a, when contradictions appear at the extremities of all dominant branches. The final number of branches, *i.e.* of leaves C (Figure 7) is $\mathcal{B}(t_a) = 2^{N\,\Omega^*(t_a)}$. It can be checked on Figure 7B that, for any complete tree, the number of leaves is simply equal to the number of nodes plus one. In the large N limit, the knowledge of $\Omega^*(t_a)$ gives a direct access to ω.

[1] B. Hayes, Can't get satisfaction, *American Scientist* **85**, 108 (1997), available from http://www.amsci.org/amsci/

[2] M. Garey, D.S. Johnson, *Computers and Intractability; A guide to the theory of NP–completeness*, W.H. Freeman and Co., San Francisco (1979).

[3] S. Mertens, Computational complexity for physicists, Lecture given at the DPG school on "Quantum Computing" (2001), available from http://arXiv.org/abs/cond-mat/0012185.

[4] Hogg T., Huberman, B.A., Williams, C. Artificial Intelligence 81I & II1996

[5] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, L. Troyansky, Determining computational complexity from characteristic 'phase transitions'. *Nature* **400**, 133 (1999).

[6] M. Mezard, G. Parisi, M. Virasoro, *Spin-glasses and beyond*, World Scientific, Singapore (1987).

[7] O. Martin, R. Monasson, R. Zecchina, Statistical mechanics methods and phase transitions in optimization problems, to appear in *Theoretical Computer Science* (2001), available from http://arXiv.org/abs/cond-mat/00104428.

[8] R. Monasson, R. Zecchina, Entropy of the K-Satisfiability problem, *Phys. Rev. Lett.* **76**, 3881 (1996).

[9] G. Biroli, M. Weigt, R. Monasson, A variational description of the ground state structure in random satisfiability problems, *Eur. Phys. J. B* **14**, 551 (2001).

[10] S. Cocco, R. Monasson, Trajectories in phase diagrams, growth processes and computational complexity: how search algorithms solve the 3-Satisfiability problem, *Phys. Rev. Lett.* **86**, 1654 (2001); Analysis of the computational complexity of solving random satisfiability problems using branch and bound search algorithms, *to appear in European Physical Journal B* (2001), available from http://arXiv.org/abs/cond-mat/0012191.

[11] M.T. Chao, J. Franco, Probabilistic analysis of two heuristics for the 3-satisfiability problem, *SIAM Journal on Computing* **15**, 1106-1118 (1986). M.T. Chao, J. Franco, Probabilistic analysis of a generalization of the unit-clause literal selection heuristics for the k-satisfiability problem, *Information Science* **51**, 289–314 (1990).

[12] A. Hartmann, M. Weigt, Typical solution time for a vertex-covering algorithm on finite-connectivity random graphs, *Phys. Rev. Lett.* **86**, 1658 (2001).

[13] O. Dubois, G. Dequen, A backbone–search heuristic for efficient solving of hard 3-SAT formulae, *IJCAI meeting* (2001).