2

EXAMPLE I: RANDOM BOOLEAN SYSTEMS OF LINEAR EQUATIONS

Does a system of randomly drawn linear equations involving Boolean variables have a solution or not? This question gives us the opportunity to discover and illustrate many notions and techniques that will be developed in greater detail throughout the book. In addition, it provides a striking example of how rich structures can unexpectedly emerge from randomness.

2.1 Boolean systems: definition and resolution.

Linear systems of Boolean equations look very much like their well known counterparts for integer-valued variables, except that equalities are defined modulo two. What does it mean precisely? Consider a set of N Boolean variables x_i with indices $i = 1, \ldots, N$. Any variable shall be False (F) or True (T). The sum of two variables, denoted by $\oplus$, corresponds to the logical exclusive OR between these variables defined through,

$$\begin{aligned} F \oplus T = T \oplus F = T \quad , \\ F \oplus F = T \oplus T = F \quad . \end{aligned} \tag{2.1}$$

In the following we shall use an alternative representation of the above sum rule. Variables will be equal to 0 or 1, instead of F or T respectively. Then, from definition (2.1), the $\oplus$ operation coincides with the addition between integer numbers modulo two.

The following is a linear equation involving three variables,

$$x_1 \oplus x_2 \oplus x_3 = 1 \quad . \tag{2.2}$$

Four among the $2^3 = 8$ assignments of (x_1, x_2, x_3) satisfy the equation: $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$ and $(1, 1, 1)$. A Boolean system of equations is a set of Boolean equations that have to be satisfied together. For instance, the following Boolean system involving four variables

$$\begin{cases} x_1 \oplus x_2 \oplus x_3 = 1 \\ x_2 \oplus x_4 = 0 \\ x_1 \oplus x_4 = 1 \end{cases} \tag{2.3}$$

has two solutions: $(x_1, x_2, x_3, x_4) = (1, 0, 0, 0)$ and $(0, 1, 0, 1)$. A system with one or more solutions is called satisfiable. A trivial example of an unsatisfiable Boolean system is

$$\begin{cases} x_1 \oplus x_2 \oplus x_3 = 1 \\ x_1 \oplus x_2 \oplus x_3 = 0 \end{cases} \quad . \tag{2.4}$$

Determining whether a Boolean system admits an assignment of the Boolean variables satisfying all the equations constitutes the Boolean System Problem (BSP). In the following, we shall restrict for some reasons to be clarified in Section **??** to 3-BSP, a variant of BSP where each Boolean equation include three variables precisely. Examples of 3-BSP inputs are systems (2.2) and (2.4).

3-BSP belongs to the class P of polynomial decision problems defined in Chapter **??**. We present below the well-known Gaussian elimination algorithm which permits to solve this problem, that is, to find a solution or proving there is none in a time growing polynomially with the size of the input Boolean system. In order to calculate this size, some care is necessary to distinguish between N, the cardinality of the set of variables from which a Boolean system is made, and V the number of variables that actually appear in this system. By definition, V is smaller or equal to N. Due to the special structure of the equations we look at, a much better bound on V can be achieved if M is much smaller than $\log N$. Indeed a system of M equations can not involve more than $3M$ distinct variables, and thus $V < 3M$ at a price of a relabelling of variables. Thus we obtain

$$V \leq \min(N, 3M) \quad . \tag{2.5}$$

The number of bits necessary to encode a Boolean system made of M equations and V variables was found in Section **??** to be

$$size = 3\,M \times (E[\log_2 V] + 1) \quad . \tag{2.6}$$

Let us now examine the operation of the Gaussian elimination algorithm on a system with $M = 5$ equations and $V = 6$ variables,

$$\begin{cases} x_1 \oplus x_3 \oplus x_4 = 0 \\ x_1 \oplus x_2 \oplus x_4 = 1 \\ x_3 \oplus x_5 \oplus x_6 = 1 \\ x_2 \oplus x_5 \oplus x_6 = 0 \\ x_1 \oplus x_2 \oplus x_5 = 1 \end{cases} \tag{2.7}$$

The algorithms works by successive taggings of variable and equations. Initially the list L of tagged variables is empty, $L = \emptyset$, and no equation is tagged. At first step, the algorithm tags the first equation and one of the variable it contains e.g. x_1, and adds it to the list: $L = \{x_1\}$. Then, equation 1 is added to the other equations where variable x_1 appears, that is, equations 2 and 5 to obtain

$$\begin{cases} x_1 \oplus x_3 \oplus x_4 = 0 \qquad\qquad (\dagger 1) \\ x_2 \oplus x_3 = 1 \\ x_3 \oplus x_5 \oplus x_6 = 1 \\ x_2 \oplus x_5 \oplus x_6 = 0 \\ x_2 \oplus x_3 \oplus x_4 \oplus x_5 = 1 \end{cases} \tag{2.8}$$

where we have simplified equations using the identity $x \oplus x \oplus y = y$ valid for any Boolean variables x, y from definition (2.1). The ($\dagger$1) symbol points to the first tagged equation. As a result of this step, x_1 now appears only in the tagged equation. Next, we consider another non tagged equation e.g. equation 4, tag it as well as one of its variable, say x_2. Now, $L = \{x_1, x_2\}$. After adding equation 4 to equations 2 and 5 (which include x_2) in (2.8), we get

$$\begin{cases} x_1 \oplus x_3 \oplus x_4 = 0 & (\dagger 1) \\ x_3 \oplus x_5 \oplus x_6 = 1 & \\ x_3 \oplus x_5 \oplus x_6 = 1 & \\ x_2 \oplus x_5 \oplus x_6 = 0 & (\dagger 2) \\ x_3 \oplus x_4 \oplus x_6 = 1 & \end{cases} \tag{2.9}$$

Next, we choose to tag equation 3 and variable x_5. The list of variables is updated to $L = \{x_1, x_2, x_5\}$, and addition of equation 3 to equation 2 results in

$$\begin{cases} x_1 \oplus x_3 \oplus x_4 = 0 & (\dagger 1) \\ 0 = 0 & \\ x_3 \oplus x_5 \oplus x_6 = 1 & (\dagger 3) \\ x_2 \oplus x_5 \oplus x_6 = 0 & (\dagger 2) \\ x_3 \oplus x_4 \oplus x_6 = 1 & \end{cases} \tag{2.10}$$

Equation 2 is trivially satisfied, and removed from the system. We then tag the only not-yet-tagged equation 4 and one of its variables, say, x_6,

$$\begin{cases} x_1 \oplus x_2 \oplus x_3 = 0 & (\dagger 1) \\ x_3 \oplus x_5 \oplus x_6 = 1 & (\dagger 3) \\ x_2 \oplus x_5 \oplus x_6 = 0 & (\dagger 2) \\ x_3 \oplus x_4 \oplus x_6 = 1 & (\dagger 4) \end{cases} \tag{2.11}$$

Since no contradictory equation $0 = 1$ has been generated throughout the tagging process, the system is satisfiable. A solution is explicitely built up by first assigning variables that do not appear in $L = \{x_1, x_2, x_5, x_6\}$ at our convenience e.g. $x_3 = 0, x_4 = 1$. Then, we consider equations and variables, in reverse tagging order. From the last tagged equation *i.e* ($\dagger 4$), x_6 is set to 0. Then equation ($\dagger 3$), ($\dagger 2$), ($\dagger 1$) successively impose that $x_5 = 1$, $x_2 = 1$, and $x_1 = 1$.

If the system has no solution, the Gaussian elimination algorithm will generate a contradictory equation $0 = 1$ at some point of the procedure. An example is provided in Exercise 1. Conversely, if the system is satisfiable, the number of solutions is given by $2^{N-M'}$, where M' is the number of equations at the end of the Gaussian procedure[29]. It is therefore bound from below by 2^{N-M}, an observation which will be useful in Section **??**.

Note that the procedure is such that the systems after and before each tagging are equivalent *i.e.* they are both satisfiable or not satisfiable. This property make

[29] Let k_i be the number of variables that do not appear in equations $j > i$, with $1 \leq i, j \leq M'$. The last equation $i = M'$ admits $2^{k_{M'}-1}$ solutions. Each of these solutions gives rise to $2^{k_{M'-1}-1}$ solutions of equation $i = M' - 1$, and so on. Finally $N' = N - \sum_i k_i$ variables never appear in the equations. The total number of solutions is therefore $2^{N-M'}$.

the outcome of the procedure independent of the choice of the tagged equations of variables, while the number of operations necessary to reach a conclusion depends on this choice. The number of operations (additions) necessary to decide the existence of solutions is easily bounded from above by

$$time \leq \sum_{i=1}^{M} (M-i) \times V = O(M^2 \times V) \quad , \tag{2.12}$$

where i is the tag index running from 1 to M, and $M - i$ is an upper bound to the number of untagged equations to whichis added the newly tagged one at step i. Equations (2.5,2.12) give $time = O(M^3)$; combining with (2.6) we obtain $time < O(size^3)$, hence proving the polynomial nature of Gaussian elimination and of 3-BSP[30].

2.2 When Structure emerges from Randomness.

2.2.1 *Satisfiable vs. unsatisfiable phase transition.*

There are many different ways of generating random Boolean systems. Perhaps the simplest one is the following. To build an equation, we randomly pick up three distinct variables among the N ones, say, x_i, x_j and x_k[31]. This draw is not biased (no variable is more likely to be chosen than the other ones), nor correlated (once one variable has been chosen, the remaining ones are equally likely to be selected). This provides us with the left hand side of the equation,

$$x_i \oplus x_j \oplus x_k \quad . \tag{2.13}$$

The second member, v, is obtained from coin tossing. It is equal to $v = 0$ or $v = 1$ with equal probabilities (one half) and independently of the indices of the variables in the first member. We end up with

$$x_i \oplus x_j \oplus x_k = v \quad . \tag{2.14}$$

Simple combinatorics shows that there are

$$H = \binom{N}{3} \times 2 = \frac{1}{3} N(N-1)(N-2) \tag{2.15}$$

different equations. Any of them can be drawn with equal probability $1/H$. Repeating this process M times, we obtain a system of M equations, independently drawn from each other. This is how system (2.7) was generated on a computer.

The above distribution of instances is, by no means, the only possible one. It however presents several attractive features which make it worth being studied:

[30] Notice that the distinction between V and N is crucial here. Consider for instance the case of a number M of equations of the order of $(\log N)^a$ where a is some positive constant. Then, blind encoding and resolution of the instance would require $size = O(M \times \log N) = O\big((\log N)^{a+1}\big)$ bits and $time = O(M^2 \times N) = O\big(N \times (\log N)^{2a}\big)$ steps respectively, leading to the erroneous conclusion that $time$ is not bounded by a polynomial function of $size$.

[31] To avoid any confusion, only the triplet (i, j, k) of indices of the variables is chosen, not the values of the variables themselves.

1. it is very easy to implement on a computer; large systems, involving thousands of equations and variables are generated in a fraction of second on a commercial computer.
2. it depends on two parameters, N and M, only. In the following, we shall rather use N and the ratio
$$c = \frac{M}{N} = \frac{\text{number of equations}}{\text{number of variables}} \quad . \tag{2.16}$$
3. the statistical independence of equations, as well as the absence of bias make the mathematical study of many properties possible.
4. last but not least, despite an apparent lack of structure (everything is random!), it leads to surprisingly interesting and rich phenomena.

An example of these phenomena is captured by

$$\begin{aligned} P(c,N) = &\text{ Probability that a system of } c\,N \text{ random equations} \\ &\text{ over a set of } N \text{ variables has a solution.} \end{aligned} \tag{2.17}$$

What is the expected behaviour of this probability as c and N vary? Let us fix the number of variables N and vary c defined in eqn (2.16) by changing M. When $c = 0$, there is no equation and the 'empty' system is obviously satisfiable, thus $P(0,N) = 1$. If more and more equations are added to the system, solutions will be less and less likely to exist: on intuitive grounds, $P(c,N)$ should be a decreasing function of c[32]. What happens as N varies at fixed c is less easy to guess, but can be observed from numerical experiments reported in Fig. 2.1. As the numbers of equations, M, and of variables, N, both become very large at fixed ratio c, the probability $P(c,N)$ reaches some simple limit. It tends to either one or zero depending on whether c is, respectively, smaller or larger than some ratio c_S,

$$\lim_{N\to\infty} P(c,N) = \begin{cases} 1 \text{ if } c < c_S \\ 0 \text{ if } c > c_S \end{cases} \quad , \tag{2.18}$$

whose value can be estimated to

$$c_S \simeq 0.92 \quad . \tag{2.19}$$

In other words, very large systems with many equations involving many variables are almost surely satisfiable, respectively unsatisfiable if their ratio of equations per variables is smaller, respetively larger than some threshold ratio, c_S. This statement, plausible at the sight of Fig. 2.1, will be confirmed by a rigorous analysis in Section 2.4.

[32] This argument can be true in a probabilistic way only. Actually, instances with a very small ratio can be unsatisfiable e.g. system (2.4) with two equations with the same first members and opposite second members. Such an instance has vanishing ratio ($c = 2/N$) and probability $1/H$ defined in eqn (2.15) in the large N limit. Similarly, systems with a very large number M of equations may have solutions e.g. when second members are all equal to $v = 0$ or to $v = 1$. Again, this happens with vanishingly small probability, 2^{1-M}, as N is fixed and c gets very large.

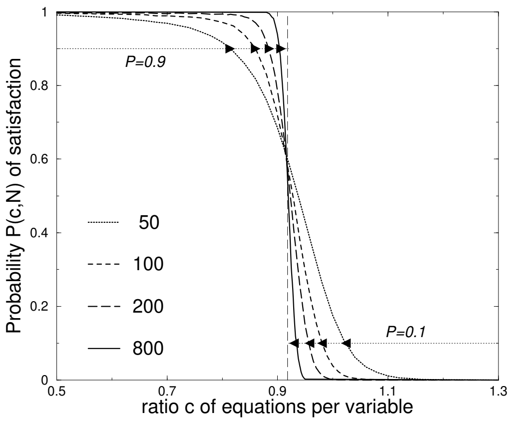

FIG. 2.1. Probability $P(c, N)$ that a system of $c\,N$ equations be satisfiable for different sizes (number of variables) $N = 50, 100, 200, 800$. This probability is obtained from numerical experiments, consisting in randomly generating 50,000 instances of the 3-Boolean System Problem for each values of c and N, and running the Gaussian elimination procedure to measure the fraction of those which admits at least one solution. In the large N limit, the threshold $c_S \simeq 0.92$ (vertical line) separates the satisfiable ($c < c_S$, $P = 1$) and the non satisfiable ($c > c_S$, $P = 0$) phases. Triangles on the horizontal dotted lines at $P = 0.1$ and $P = 0.9$ show how the ratios $c(P, N)$ converge to c_S with increasing size N. The "speed" of convergence define the transition window scaling exponent a (2.20).

To obtain a more accurate picture of this threshold phenomenon, we may address the following question: with what ratio c one has to draw random Boolean systems of large size N in order to escape from the (almost) always satisfiable, or unsatisfiable scenario formalised in eqn (2.18), or quivalently, how fast with N does the probability of satisfaction, $P(c, N)$, converge to its limit (2.18)? Let us call $c(P, N)$ the inverse function of $P(c, N)$ at fixed N; $c(P, N)$ is the ratio of equations per variables at which the probability of a randomly drawn system over N variables has probability P of having at least one solution. Ratios $c(0.9, N)$ and $c(0.1, N)$ are shown on Fig. 2.1 for sizes $N = 50, 100, 200$. For any $0 < P < 1$, $c(P, N)$ gets closer and closer to the threshold as N increases; the relative difference is a vanishing function of N, which apparently decreases as a negative power of N,

$$c(P, N) = c_S + \frac{\bar{c}(P)}{N^a} + o\left(\frac{1}{N^a}\right) \quad . \tag{2.20}$$

The exponent a does not depend on P, while coefficient $\bar{c}$ does: it is negative (respectively positive) for large (resp. small) probabilities P, see Fig. 2.1.

Equation (2.20) is called finite-size scaling hypothesis in physics literature (citation; Kirkpatrick, Selman 1994). It provides us with information about how sharp the transition is, or equivalently, how the transition region, with width $\Delta \equiv c(\epsilon, N) - c(1-\epsilon, N) = \Theta(N^{-a})$, shrinks as N grows (at fixed ϵ). It is therefore expected that, once properly rescaled, $P(c, N)$ admits a non trivial limit,

$$\lim_{N\to\infty} P\left(c_s + \frac{\bar{c}}{N^a}, N\right) = p(\bar{c}) \quad , \tag{2.21}$$

with

$$\lim_{\bar{c}\to-\infty} p(\bar{c}) = 1 \qquad \text{and} \qquad \lim_{\bar{c}\to+\infty} p(\bar{c}) = 0 \quad , \tag{2.22}$$

as required from compatibility with eqn (2.18). $p(\bar{c})$, the inverse function of $\bar{c}(P)$ in eqn (2.20), gives us access to the probability of satisfaction inside the transition window, or more precisely, once the transition window has been magnified by a multiplicative factor N^a to acquire a finite width.

What is the value of the scaling exponent a? We present a heuristic argument supporting the inequality $a \le \frac{1}{2}$; a rigorous proof was established by Wilson (2003). We start from a simple observation, which will turn out to be essential to the rigorous study of the random Boolean system problem exposed in Section 2.4. Consider for instance the Boolean system (2.11). It is made of 4 equations and 6 variables, two of which, x_1 and x_4 appear only once, in equations 1 and 4 respectively. For any assignment of the variables x_2, x_3, x_5, x_6 that appear at least twice, these two equations can be easily satisfied by an adequate choice of x_1, x_4 that do not affect the status of the other equations, namely, 2 and 3. As far as the existence of solution is concerned, system (2.4) is thus equivalent to the subsystem

$$\begin{cases} x_3 \oplus x_5 \oplus x_6 = 1 \\ x_2 \oplus x_5 \oplus x_6 = 0 \end{cases} \tag{2.23}$$

where we have removed equations 1 and 4[33]. This simple remark extends to any system $\mathcal{S}$,

removal from $\mathcal{S}$ of equations including variable(s) with a single occurrence leads to a subsystem $\mathcal{S}' \subset \mathcal{S}$ that is satisfiable if and only if $\mathcal{S}$ is. (2.24)

Let us call $c_1\, N$ the number of clauses that are removed from $\mathcal{S}$. c_1 is a rational number, multiple of $\frac{1}{N}$, and bounded from above by c (reached when all equations are removed and $\mathcal{S}'$ is empty) and 1 (the number of removed equations *i.e.* of variables with a single occurence is smaller than the total number of variables). When $\mathcal{S}$ is randomly drawn, c_1 is a stochastic variable too. In the large N limit, its distribution, $G(c_1|c, N)$, becomes concentrated around some value, $\bar{c}_1$

[33] The removal procedure can be iterated. For instance, x_2 and x_3 appear each twice in system (2.9) and only once in (2.23). We may therefore remove these variables and their attached equations from (2.23); we are left with an empty system, which is clearly satisfiable, proving that (2.9) is, too.

(Creignou, Daudé, Dubois 2001; Dubois, Mandler 2002), calculated in Exercise 4. An explicit calculation shows that c_1 has a vanishing probability to differ from $\bar{c}_1$ by more than $O(1/\sqrt{N})$ (Wilson 2003; Dubois, Mandler 2001). The representation of G is sketched in the left panel of Fig. 2.2. G can be thought as a "dispersion" function, which allows c_1 to fluctuate around $\bar{c}_1$ with a width $W = \Theta(N^{-1/2})$.

We now denote by $P'(c - c_1|c, N)$ the probability that a subsystem $\mathcal{S}'$ with $(c-c_1)\,N$ equations generated through removal of single occurence variables from a system $\mathcal{S}$ with $c\,N$ equations be satisfiable. Given c close c_S, the dependence of P' upon the ratio $c - c_1$ of remaining equations per variable in $\mathcal{S}'$ is expected to look like the central graph of Fig. 2.2. P' jumps down from being almost equal to 1 to a vanishingly small value when crossing the transition region, of width Δ'.

Observation (2.24) tells us that P is essentially the convolution of P' and G,

$$P(c, N) = \sum_{c_1=0,\frac{1}{N},\frac{2}{N},\ldots,\min(c,1)} G(c_1|c, N)\; P'(c - c_1|c, N) \quad . \tag{2.25}$$

It is indeed a straightforward consequence of the above convolution identity that the transition region width of P, Δ, is larger than both widths Δ' and W (see Exercise 3). Even if the transition of P' is very sharp e.g. $\Delta' = \Theta(1/N)$, it gets smeared out by the dispersion of ratios introduced by G, see Fig. 2.2. Thus Δ is at least as large as W, and $a \leq \frac{1}{2}$.

Estimates of the threshold c_S and of the transition width exponent a can be extracted from numerical experiments. Once the threshold has been approximately localized e.g. from Fig. 2.1, data for $P(c, N)$ must be accumulated for various ratios c close to this value and sizes $N = N_1, N_2, ..., N_k$. Then, the curves of $P(\gamma = (c - c_S) \times \big(N_\ell\big)^a, N_\ell)$ as a function of γ are plotted on the same graph for all $\ell = 1, 2, \ldots, k$. By virtue of eqn (2.21), the right values of c_S and a are

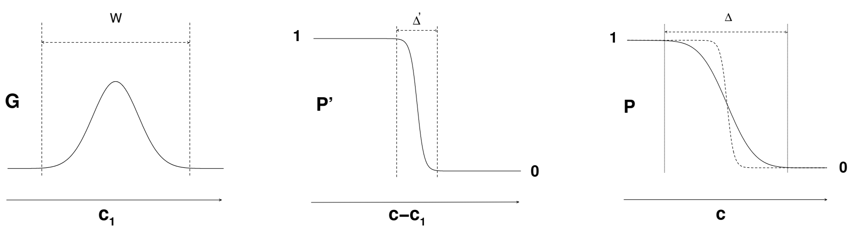

FIG. 2.2. Examples of dispersion function G with width W (left) and probability P' with transition window of width Δ' (middle) smaller than W. Convolution of those gives another probability $P = G * P'$ (right, solid line), whose transition window is wider than the one of P' (dashed line) and close to W. Position along the horizontal axis represents the ratio of removed equations per variable, c_1 (left), of equations per variable, c (right), and the combination $c - c_1$ (middle).

the ones that make the k curves collapse onto each other (Kirkpatrick, Selman 1994). Though very attractive, this procedure is somewhat dangerous. Indeed, control of the accuracy of the predictions is difficult to achieve, and erroneous values for the exponents may result from extrapolation of data at unsufficiently large sizes (citation...). As far as random 3-BSP is concerned, numerical simulations carried out with up to few thousands of variables seem to support scaling hypothesis (2.20) with $a = \frac{1}{2}$ (Ricci-Tersenghi, Weigt, Zecchina 2001). There is no general reason why a should coincide with this upper bound; in fact, the phase transition taking place in random 2-SAT, for which Wilson's upper bound applies is characterized by $a = \frac{1}{3}$ (Bollobàs et al. 2001).

2.2.2 *"Microscopic" description of solutions.*

The random 3-BSP problem exhibit various interesting properties besides the phase transition shown in Fig. 2.1. Sections 2.3.3 and 2.4 will be devoted to the analysis of these properties with tools borrowed from fields as diverse as probability theory, the analysis of algorithms or statistical physics... Hereafter, we present an overview of the output of these analysis. This anomalous presentation will, hopefully, facilitate the lecture of subsequent Sections.

The phase diagram of random 3-BSP is shown in Fig. 2.3. It is composed of a unique axis along which the ratio c of equations per variable increases from zero to infinity. As inferred from numerical experiments, a satisfiability vs. unsatisfiability transition is located at ratio

$$c_S = 0.918... \tag{2.26}$$

The exact value and its derivation will be explicited in Section 2.4. Below c_S, randomly drawn systems are almost surely satisfiable. How many solutions do they have? This number $\mathcal{N}$ is a stochastic variable that vary from system to system. However, in the large N limit, some concentration phenomenon takes place,

$$\lim_{N\to\infty} \text{Prob}\left[2^{N\,\left(s(c)-\epsilon\right)} < \mathcal{N} < 2^{N\,\left(s(c)+\epsilon\right)}\right] = 1 \tag{2.27}$$

for any positive ϵ, where the function $s(c)$, called entropy of solutions hereafter, reads

$$s(c) = 1 - c \quad . \tag{2.28}$$

In other words, the number $\mathcal{N}$ of solutions is very likely to equal $2^{N(1-c)+o(N)}$. The entropy of solutions is shown on Fig. 2.4. Note that it is strictly positive when approaching the threshold from the left: $s(c_S^+) > 0$. In other words, a random system with $(c_S - \epsilon)\,N$ equations has exponentially numerous solutions, which abruptly disappear when adding, say, $2\epsilon N$ randomly drawn equations to the system.

Having counted solutions, we go further and wonder how much those ressemble each other. The notion of ressemblance may be defined without ambiguity through the introduction of a distance, called Hamming distance, between solutions. Consider two assignments $U = \{u_1, u_2, \ldots, u_N\}$ and $V = \{v_1, v_2, \ldots, v_N\}$

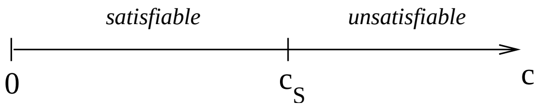

FIG. 2.3. Phase diagram of the random 3-Boolean System Problem. Depending on whether its ratio c of equations per variable is smaller or larger than some threshold $c_S \simeq 0.918$, a very large system has almost surely or almost never solutions respectively.

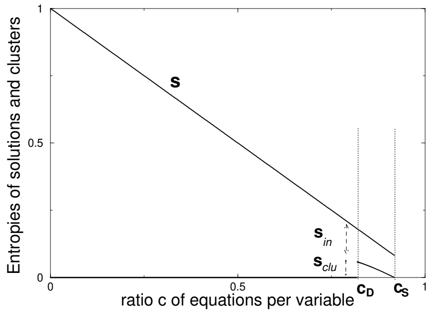

FIG. 2.4. Entropies, that is, logarithms of the numbers (divided by size N) of the numbers of solutions and clusters as a function of the ratio c of equations per variable. The entropy of solutions, s, equals $1-c$ for $c < c_S \simeq 0.918$. For $c < c_D \simeq 0.818$, solutions are uniformely scattered on the N-dimensional hypercube. At c_D, the solution space discontinuously breaks into disjoint clusters. The enropies of clusters, s_{clu}, and of solutions in each cluster, s_{in}, are such that $s_{clu} + s_{in} = s$. At c_S, the number of clusters ceases to be exponentially large ($s_{clu} = 0$). Above c_S, there is almost surely no solution.

of the Boolean variables *i.e.* two sequences of N zeroes or ones. The Hamming distance $d(U,V)$ is equal to the fraction of variables that differ between assignments U and V,

$$d(U,V) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{u_i \neq v_i} = \frac{1}{N}\sum_{i=1}^{N} \big(u_i + v_i - 2\, u_i\, v_i\big) \quad , \qquad (2.29)$$

where $\mathbf{1}_p$ equals 1 if the logical proposition p is true, 0 otherwise. d is indeed a distance: it is positive and symmetric in its arguments; it vanishes if and only if $U = V$, and fulfills the triangular inequality $d(U,V) \leq d(U,Z) + d(Z,V)$ for any Z. The concept of distance irresistibly calls for the introduction of some space where assignments would lie. Indeed, it is natural to think of the 2^N assignments of Boolean variables as the 2^N corners (vertices) of the N-dimensional hypercube; this representation is illustrated in Fig. 2.5 for $N = 2$ and $N = 3$ variables. The

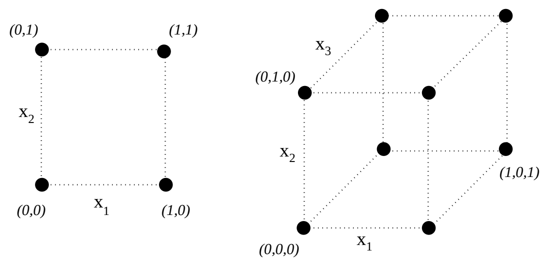

FIG. 2.5. Hypercube representation of Boolean assignments for $N = 2$ (square, left) and $N = 3$ (cube, right) variables. Each vertex of the hypercube encodes an assignment of variables e.g. $(0, 1)$ on the left square. Along an edge of the hypercube, one variable gets negated. The Hamming distance $d(U, V)$ between two vertices U, V is the number of edges separating these vertices divided by N e.g. $d\big((0, 0, 0), (0, 1, 0)\big) = 1/3$ and $d\big((0, 0, 0), (1, 0, 1)\big) = 2/3$ on the right cube.

Hamming distance $d(U, V)$ is simply the minimal number of edges connecting the two vertices associated to U and V on the hypercube (divided by N).

The set of solutions of a system $\mathcal{S}$ is the subset of vertices of the hypercube where all vertices corresponding to unsatisfying assignments have been removed. An appealing picture of this set is that of a (high-dimensional) Swiss cheese, where plain good cheese come from solutions and holes from violating assignments. What is the structure of this Swiss cheese? Are solutions homogeneously spread over the hypercube, or are there regions particularly riched in, or prived of solutions? The answer to these question dramatically depends on the value of the ratio c of equations per variable. A 'geometrical' phase transition, unrelated to the satisfiable vs. unsatisfiable transition of Fig. 2.1 takes place, in the large size limit, at the critical ratio

$$c_D \simeq 0.818... \quad , \tag{2.30}$$

well within the satisfiable phase. The main features of this new phase transition are schematized on the enriched phase diagram of Fig. 2.6 and listed below:

- For ratios $c < c_D$, the Hamming distance between two solutions is, with high probability, equal to $d = 1/2$. Solutions thus differ on $N/2 + o(N)$ variables, as if they were statistically unrelated assignments of the N Boolean variables[34]. In addition, the space of solutions enjoys some connectedness

[34] The concentration of the Hamming distance around some well defined value may at first sight be surprising: how can an exponential number of solutions be, two by two, at the same distance from each other? This is simply an illustration of the "do–not-peel–high–dimensional–apples" phenomenon. Consider a D-dimensional sphere with radius R. The fraction f of the volume located between radius $\alpha\ R$ and R (with $\alpha < 1$) is $f = 1 - \alpha^D$. Therefore, when $D \to \infty$, $f \to 1$: most volume is concentrated close to the surface of the sphere; two randomly chosen points inside the sphere are very likely to be diametrically away from each other. In

property. Any two solutions are linked by a path such that the number of successive holes along this path is bounded: one is not forced to cross a big region prived of solutions when going from one solution to another[35].

- The spatial organization of solutions on the hypercube undergoes a drastic change at c_D (Fig. 2.6). For ratios c above c_D, the space of solutions is not connected any longer. It is made of an exponentially large (in N) number $\mathcal{N}_{clu}$ of connected components, called clusters, each containing an exponentially large number $\mathcal{N}_{in}$ of solutions. Two solutions belonging to different clusters lie apart at a Hamming distance $d_{clu} = 1/2$ while, inside a cluster, the distance is $d_{in} < d_{clu}$. Concentration properties of the type of eqn (2.27) hold for the numbers of clusters and solutions in a cluster: with high probability, $\mathcal{N}_{clu} = 2^{N\, s_{clu} + o(N)}$ and $\mathcal{N}_{in} = 2^{N\, s_{in} + o(N)}$. The entropies of clusters, s_{clu}, and of solutions in a cluster, s_{in}, are plotted in Fig. 2.4 as functions of the ratio c; explicit expressions for $s_{clu}(c), s_{in}(c)$ will be given in Section 2.4. The sum of these entropies obviously gives back the entropy $s(c)$ of solutions (2.28). At c_D, the entropy s of solutions is analytic in c, while s_{clu} and s_{in} undergo discontinuous jumps.

In next Sections, we explain how the above picture of the phase transitions and of the 'microscopic' organization of solutions can be established.

2.3 Getting close to the threshold ...

The probability of satisfaction, $p(c, N)$, defined in (2.17) is instrinsically related to the distribution of the number $\mathcal{N}$ of solutions. The latter is a stochastic variable which, depending on the instance, varies from 0 (if there is no solution) up to 2^N (if all configurations of variables are solutions). Ideally, we would like to calculate

$$p(\mathcal{N}) = \text{probability that an instance has } \mathcal{N} \text{ solutions,} \tag{2.31}$$

at fixed numbers N, M of variables and equations. This is however a very difficult calculation, and we start by examining the first few moments of $\mathcal{N}$,

$$\overline{\mathcal{N}^q} = \sum_{\mathcal{N}=0}^{2^N} p(\mathcal{N})\, \mathcal{N}^q \quad . \tag{2.32}$$

Knowledge of the first two moments, $\overline{\mathcal{N}}$ and $\overline{\mathcal{N}^2}$ will already provide us with useful information on the threshold phenomenon of Fig. 2.1. Indeed, the probability of satisfaction is related to these moments through the inequalities

other words, in three dimensions, the bulk of an apple makes for most of its volume; taking out the skin before eating it results in no big loss. In high dimension, once the apple has been peeled off, nothing is left...

[35]More precisely, for any pair of solutions U, U', there almost surely exists a sequence of solutions, $U_j, j = 0, 1, 2, \ldots, J$, with $U_0 \equiv U$, $U_J \equiv U'$, $J = O(N)$, connecting the two solutions such that $d(U_j, U_{j+1}) = O(1/N)$.

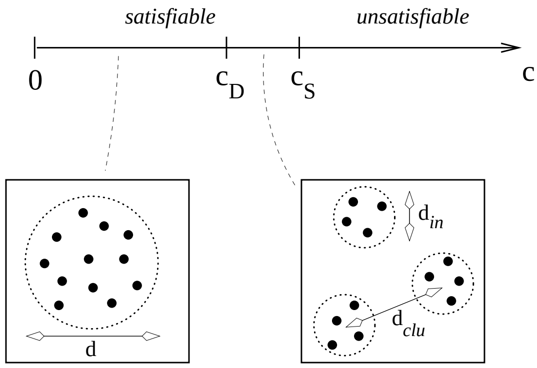

FIG. 2.6. More accurate phase diagram of the random 3-Boolean System Problem compared to Fig. 2.3. A 'geometrical' phase transition takes place in the satisfiable phase at some ratio $c_D \simeq 0.818$. At small ratios $c < c_D$, solutions are uniformely scattered on the N-dimensional hypercube, with a typical normalized Hamming distance $d = 1/2$. At c_D, the solution space discontinuously breaks into disjoint clusters: the Hamming distance $d_{in} \simeq 0.14$ between solutions inside a cluster is much smaller than the typical distance $d_{clu} = 1/2$ between two clusters.

$$\frac{\left(\overline{\mathcal{N}}\right)^2}{\overline{\mathcal{N}^2}} \leq \text{Prob}\big[\mathcal{N} \geq 1\big] \leq \overline{\mathcal{N}} \ . \tag{2.33}$$

The demonstration of these inequalities is proposed in Exercise 2.

2.3.1 *Average number of solutions.*

We start by expressing the number of solutions in a compact algebraic way. Let us define the characteristic function $e_\ell(X)$ of equation number ℓ which associates to any assignment X of the Boolean variables 1 if equation ℓ is satisfied, 0 otherwise. Clearly, the number of solutions can be expressed as

$$\mathcal{N} = \sum_X \prod_{l=1}^{M} e_\ell(X) \quad , \tag{2.34}$$

where the sum runs over all 2^N assignments. The presence of the product of characteristic functions in eqn (2.34) ensures that only solutions to the system of equations contribute to $\mathcal{N}$, each with weight unity.

Calculation of the first moment requires to average eqn (2.34) over the random choices of equations,

$$\overline{\mathcal{N}} = \sum_X \overline{\left[\prod_{\ell=1}^{M} e_\ell(X)\right]} = \sum_X \prod_{\ell=1}^{M} \overline{e_\ell(X)} = \sum_X \left[\overline{e(X)}\right]^M \tag{2.35}$$

where we have made successively use of the facts that the average of a sum of stochastic variables is equal to the sum of their averages, that clauses are independently drawn from each other and with the same distribution regardless of their index ℓ.

To calculate the value of $\overline{e(X)}$, we need to average over the random ingredients entering a clause, that is the choice of the triplet (i,j,k) of the indices of the variables and of the second member v,

$$\overline{e(X)} = \frac{1}{\binom{N}{3}} \sum_{1\le i<j<k\le N} \frac{1}{2} \sum_{v=0,1} \mathbf{1}_{x_i\oplus x_j \oplus x_k = v} \quad . \tag{2.36}$$

Given an assignment X and a triplet (i,j,k) of variables, $x_i \oplus x_j \oplus x_k$ is either equal to 0 or 1; the second sum in eqn (2.36) selects the right value for v, and is simply equal to unity. Thus, we end up with the deceiptively simple result

$$\overline{e(X)} = \frac{1}{2} \quad . \tag{2.37}$$

and, from eqn (2.35),

$$\overline{\mathcal{N}} = 2^{N-M} = 2^{N(1-c)} \quad . \tag{2.38}$$

A simple interpretation of this result is the following. In the absence of equations ($M=0$), all 2^N configurations are solutions. Now, consider the case of $M=1$ equation e.g. equation (2.7). Variables x_i with $4\le i\le N$ are not constrained by this equation and can be freely assigned, while among the eight possible configurations of (x_1,x_2,x_3), four assignments are allowed only. Therefore, the number of solutions equals $2^{N-3}\times 4 = 2^{N-1}$. This reasoning extends to larger values of the number M of equations. Each equation acts on a “filter” on the set of configurations which retain half of the configurations as valid solutions and discard the remaining half ones. As equations are independently drawn from each other, filters act one after the other. After the introduction of M filters (equations), the average number of solutions is divided by 2^M as found in eqn (2.38).

As soon as the ratio c of equations per variable is larger than $c_+ = 1$, the expected number of solutions vanishes at large size N, and so does the probability of satisfaction from inequality (2.33). Therefore, we have found an upper bound to the threshold,

$$c_S \le c_+ = 1 \quad . \tag{2.39}$$

When $c<1$, $\overline{\mathcal{N}}$ is very large and the right inequality (2.33) does not convey any useful information. This may appear suprising at first sight. Take say $c = 0.96$. Then, numerical experiments indicate that a randomly drawn set of $c\times N$ equations has (almost) never any solution for large N, while the above calculation tells us that the average number of solutions is huge ($= 2^{0.04\,N}$)! The explanation to this apparent paradox is simple. While almost none of set of equations have a solution, a very tiny fraction of them have a huge number of solutions, which make the expectation $\overline{\mathcal{N}}$ exponentially large in N.

2.3.2 *Second moment calculation.*

We now turn to the calculation of the average squared number of solutions. Using expression (2.34), we write $\mathcal{N}^2$ as a double sum over all assignements X, Y,

$$\mathcal{N}^2 = \left[\sum_X \prod_{\ell=1}^M e_\ell(X)\right]^2 = \sum_{X,Y} \prod_{\ell=1}^M e_\ell(X) e_\ell(Y) \quad . \tag{2.40}$$

The average of the random choices of equations is straightforward,

$$\overline{\mathcal{N}^2} = \sum_{X,Y} \left[\overline{e(X)e(Y)}\right]^M \quad . \tag{2.41}$$

We are left with the calculation of $\overline{e(X)e(Y)}$, that is, the probability that two assignments X and Y both satisfy the same randomly drawn equation. Intuitively, this probability depends only on how much X and Y ressemble each other. It will decrease from $1/2$ if $Y = X$ from eqn (2.37) down to 0 if $Y =$ Not X due to the odd number (three) of variables in an equation.

The key quantity to consider here is the Hamming distance $d \equiv d(X, Y)$, that is, the probability that, for a randomly chosen index i, variables $x_i \neq y_i$. We then find

$$\begin{aligned}
\overline{e(X)e(Y)} &= \frac{1}{\binom{N}{3}} \sum_{1\le i<j<k\le N} \frac{1}{2} \sum_{v=0,1} \mathbf{1}_{x_i \oplus x_j \oplus x_k = v}\ \mathbf{1}_{y_i \oplus y_j \oplus y_k = v} \\
&= \frac{1}{2\binom{N}{3}} \sum_{1\le i,j,k\le N} \mathbf{1}_{x_i \oplus x_j \oplus x_k = y_i \oplus y_j \oplus y_k} + O\left(\frac{1}{N}\right) \\
&= \frac{1}{2}\left[(1-d)^3 + 3\,d^2\,(1-d)\right] + O\left(\frac{1}{N}\right) \quad .
\end{aligned} \tag{2.42}$$

The two terms $(1-d)^3$ and $3\,d^2\,(1-d)$ in the last bracket are the probabilities that the three pairs of variables coincide *i.e.* $x_i = y_i$, $x_j = y_j$, $x_k = y_k$ and that two of them differ e.g $x_i \neq y_i$, $x_j \neq y_j$, $x_k = y_k$ respectively[36].

Expression (2.41) for the second moment can be rewritten as a sum over all possible values of the Hamming distance d,

$$\overline{\mathcal{N}^2} = \sum_{d=0,\frac{1}{N},\frac{2}{N},\ldots,1} \left[\frac{1}{2}\left((1-d)^3 + 3\,d^2\,(1-d)\right) + O\left(\frac{1}{N}\right)\right]^M \mathcal{M}(d) \quad , \tag{2.43}$$

where $\mathcal{M}(d)$ is the number of pairs X, Y of assignments at distance d. To calculate this multiplicity, we first choose some assignment X; then the assignments Y at distance d from X are obtained by negating $d\,N$ variables in X. There are thus

[36] These expressions are correct up to $O(1/N)$ corrections. For instance, the probability of the joint event $x_i = y_i, x_j = y_j, x_k = y_k$ reduces to the product of single events (e.g. $x_i = y_i$) probabilities exactly expressed in terms of the Hamming distance d in the large N limit only.

$\binom{N}{dN}$ such assignments Y. As this number is independent of X, we then simply multiply by the number of assignments X,

$$\mathcal{M}(d) = 2^N \binom{N}{dN} \quad . \tag{2.44}$$

Use of the Stirling formula[37] for the combinatorial coefficient permits us to establish the following asymptotic expression for the average squared number of solutions,

$$\overline{\mathcal{N}^2} = \sum_{d=0,\frac{1}{N},\frac{2}{N},\ldots,1} 2^{N\,\gamma(d,c)+o(N)} \quad , \tag{2.46}$$

where the function γ appearing in the exponent reads

$$\gamma(d,c) = 1 - c - (1-d)\log_2(1-d) - d\log_2 d + c\log_2\left[(1-d)^3 + 3\,d^2\,(1-d)\right] \; . \tag{2.47}$$

In the large N limit, the sum in eqn (2.46) can be evaluated from the application of the Laplace method, that is, by keeping the largest contribution to the sum and discarding all exponentially smaller terms. We obtain,

$$\overline{\mathcal{N}^2} = 2^{N\,\gamma^*(c)+o(N)} \quad , \tag{2.48}$$

where $\gamma^*(c)$ is the maximum over $d \in [0;1]$ of γ at a fixed c, reached in $d^*(c)$[38].

Function $\gamma(c,d)$ is plotted in Fig. 2.7 for various ratios c. A local maximum is present in $d^*(c) = 1/2$ for any c, with corresponding height $\gamma^*(c) = 2(1-c)$. This is the only local maximum when $c < 0.779$. For $c > 0.779$, another local maximum is present and located in $d^*(c) < 1/2$. The global maximum of γ is reached in $d = 1/2$ as long as $c \le c_- \simeq 0.889$, and in $d^*(c) < 1/2$ above this critical ratio (Fig. 2.7). Comparing this result with the first moment expression (2.38),

$$\lim_{N\to\infty} \frac{\left(\overline{\mathcal{N}}\right)^2}{\overline{\mathcal{N}^2}} = \begin{cases} 1 \text{ if } c < c_- \\ 0 \text{ if } c > c_- \end{cases} \quad , \tag{2.49}$$

and, from inequality (2.33), that c_- is a lower bound to the value of the threshold (Daudé, Dubois, Creignou 2001),

$$c_S > c_- \simeq 0.889 \quad . \tag{2.50}$$

[37]The factorial $N!$ of a large integer N is asymptotically equivalent to

$$N! = \sqrt{2\pi N}\left(\frac{N}{e}\right)^N \quad . \tag{2.45}$$

Systematic corrections to this expression can be calculated and lead to an asymptotic series for $N!$. Here, and most of the time in our large N calculations, only the exponential contribution (second term on the right hand side of eqn (2.45)) matters for the final result.

[38]A direct proof of equation (2.46) is as follows. First, $\overline{\mathcal{N}^2}$ is bounded from above by $(N+1)\times 2^{N\gamma^*(c)}$. Secondly, for large N, the gap between the location $d^*(c)$ of the maximum of γ and the closest value of d, say $d^\dagger$, in the sum (2.47) is $\le 1/N$. Therefore, $\gamma^*(c) - \gamma(d^\dagger, c) = O(1/N)$, and $\overline{\mathcal{N}^2}$ is bounded from below by $2^{N\gamma(d^\dagger,c)} = 2^{N\gamma^*(c)+O(1)}$.

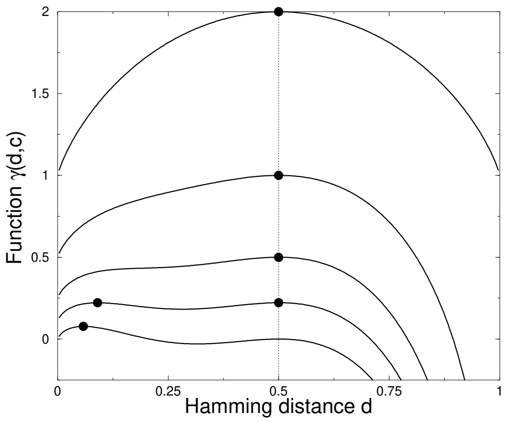

FIG. 2.7. Function $\gamma(d,c)$ defined in eqn (2.47) and appearing in the calculation of the average squared number of solutions $\overline{\mathcal{N}^2}$ for different values of the ratio c of equations per variable (from top to down: $c = 0$, 0.5, 0.75, 0.889 and 1). d is the Hamming distance *i.e.* fraction of different variables in two solutions of a random set of equations. Black dots identify the maxima of γ over d at fixed c. They are located in $d^* = 1/2$ (dotted line), $\gamma^* = 2(1-c)$ for $c < 0.889$, and in $d^* < 1/2, \gamma^* > 2(1-c)$ for $c > 0.889$.

In fact, we have cheated in writing down identity (2.49). The asymptotic equivalence of $(\overline{\mathcal{N}})^2$ and $\overline{\mathcal{N}^2}$ does not allow us to deduce anything about their ratio for $c < c_-$. Proof that the limit of the ratio is one as stated in eqn (2.49) requires the calculation of subleading terms that we do not reproduce here (citation).

There is more to the above calculation than a lower bound to the threshold. The location of the maximum of γ, $d^*(c)$, which emerged from the calculation of $\overline{\mathcal{N}^2}$, has an interesting significance. It is an approximation of the Hamming distance between two solutions. By definition, this Hamming distance, averaged over all pairs of solutions is, for a given satisfiable Boolean system,

$$\langle d \rangle = \frac{1}{\mathcal{N}^2} \sum_{X,Y} d(X,Y) \prod_{l=1}^{M} [e_\ell(X) e_\ell(Y)] \tag{2.51}$$

where the presence of the e_ℓs ensures that the sum runs over pairs X, Y of solutions only. Next, we would like to calculate $\overline{\langle d \rangle}$, that is, the average of $\langle d \rangle$ over the random choices of equations. This is a hard task due to the correlations existing between the denominator, $\mathcal{N}^2$, and the numerator in eqn (2.51). An approximate expression for $\overline{\langle d \rangle}$, called annealed estimate, is found from the ratio of the averages of the numerator and denominator instead of the average of the ratio,

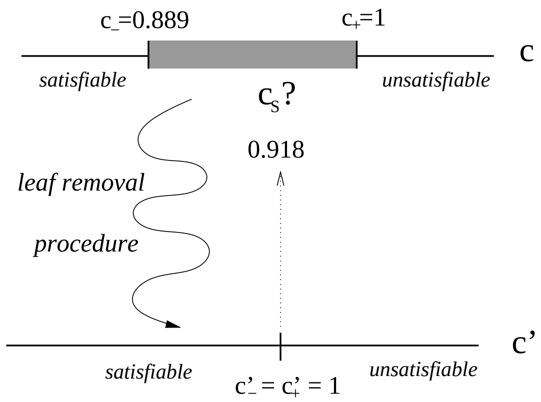

FIG. 2.8. Phase diagram of the random Boolean System Problem. Top axis: original problem parametrized by the ratio c of equations per variable. c_- and c_+ are lower and upper bounds to the value of the threshold c_S determined from the probabilistic inequalisties (2.33). Bottom axis: same as top axis once the leaf removal procedure has eliminated all one-occurence variables and their attached equations; the threshold ratio c' is unity, and inversion of the relation $c \to c'$ allows to determine $c_S \simeq 0.918$.

$$\overline{\langle d \rangle} \simeq \frac{1}{\mathcal{N}^2} \sum_{X,Y} d(X,Y) \overline{\prod_{l=1}^{M} \left[e_\ell(X) e_\ell(Y) \right]} = d^*(c) \ , \tag{2.52}$$

from the second moment calculation. Therefore, we expect some qualitative change of behaviour to take place at c_-. For $c < c_-$, the Hamming distance is, within the annealed approximation, equal to $1/2$. Two solutions are likely to differ over $N/2$ variables, as if they were totally uncorrelated between each other on the hypercube of Fig. 2.5. For $c > c_-$, the Hamming distance between solutions is much smaller than $1/2$: the constraints imposed by the Boolean systems are so strong that solutions ressemble each other very much. It is worth noticing that the transition between the two regimes (uncorrelated solutions at low c, strong correlations above c_-) is abrupt: $d^*(c)$ discontinuously jumps from $1/2$ down to 0.09 when c passes through c_-. This scenario is in qualitative agreement with the exact picture of the the geometrical phase transition taking place at c_D and the microscopic organization of solutions in the range $c_D < c < c_S$. The use of the annealed approximation explains the quantitative discrepancy between, from the one hand, c_- and d^*, and from the other hand, c_D and d_1.

2.3.3 *Higher moments and large deviations.*

The theoretical results we have obtained so far are summed up in the upper part of Fig. 2.8. For $c < c_-$ (respectively $c > c_+$), a system of $c\,N$ randomly drawn equations from a set of N variables has, when $N \to \infty$, almost always (respectively never) a solution. The precise determination of the threshold separating these satisfiable and unsatisfiable phases remains out-of-reach with our probabilistic tool (2.33).

The reason for this partial failure was unveiled at the end of the first moment calculation. This, and higher moments, is affected by the presence of rare instances with a very large number of solutions, and does not fairly reproduce the value of $\mathcal{N}$ characterizing the "typical" behaviour of the random 3-BSP problem. A simple illustration of the difference between average and typical behaviours is provided by the following example. Consider a stochastic variable Z with a bimodal distribution: $Z = 2^{N\,a}$ with probability $1 - 2^{-b\,N}$, and $Z = 2^{N\,a'}$ with probability $2^{-b\,N}$ where $a < a'$ and b is a strictly positive parameter. Z is equal to $2^{N\,g_0}$ with $g_0 = a$, but for an exponentially small fractions of events: this defines the typical value. The average value, $\overline{Z} = 2^{N\,g_1 + o(N)}$, with $g_1 = \max(a, a' - b)$ may be exponentially larger than the typical value if $a < a' - b$. The average of Z may therefore not be a good representative quantity of its typical value. A hint at what could be a good indicator is provided by the calculation of the q^{th} moment, $\overline{Z^q} = 2^{N\,q\,g_q + o(N)}$, where $g_q = \max(a, a' - b/q)$. We see that the typical value of Z can be deduced from moments of low order $q < q^* = b/(a' - a)$ only: $g_0 = g_q$ in this range. Higher moments do not faithfully reflect the 'true' value of the stochastic variable Z. Unfortunately, in practice, the value of q^* is not known *a priori*, but is certainly positive. Therefore, we are led to think that the solution to our problem of determining the typical value of Z could come from the calculation of its q^{th} moment when $q \to 0$ ...

This conclusion is at the base of the statistical physics approach presented in Section **??**. Consider a satisfiable system of equations. We define its entropy s as the logarithm of the number of solutions,

$$\mathcal{N} = 2^{N\,s} \quad . \tag{2.53}$$

By definition, s is comprised between 0 (unique solution) and 1 (if there is no equation to be satisfied). The entropy s is a stochastic variable with associated probability, from definition (2.31),

$$\text{Prob}[s] = P\left(\mathcal{N} = 2^{N\,s}\right) = 2^{N\,\omega(s,c;N)} \quad . \tag{2.54}$$

The above equation is merely a definition of $\omega(s, c; N)$. We explain later why this quantity is expected to have a well defined limit for large sizes,

$$\omega(s, c) = \lim_{N\to\infty} \omega(s, c; N) \quad . \tag{2.55}$$

This limit is the rate function of the entropy, see Section 1.3.3.3. The first and second moment calculations are easily interpreted in terms of the rate function $\omega(s, c)$. Using definitions (2.53) and (2.54), we rewrite the q^{th} moment (2.32) of the number of solutions as

$$\overline{\mathcal{N}^q} = \sum_s 2^{N\,\left(\omega(s,N,c) + q\,s\right)} = 2^{N\,g(q,c) + o(N)} \ , \tag{2.56}$$

where the sum runs over the discrete set of possible values for s, $s_{\mathcal{N}} = \log_2(\mathcal{N})/N$ with $1 \leq \mathcal{N} \leq 2^N$, and

$$g(q,c) = \max_{0 \leq s \leq 1} \left[\omega(s,c) + q\, s \right] \quad . \tag{2.57}$$

The value s_q of s which maximises the r.h.s. of eqn (2.57) is the root of the extremisation condition

$$\frac{\partial \omega}{\partial s}(s_q, c) = -q \quad . \tag{2.58}$$

In other words, s_q is the abscissa of the point A_q at which the representative curve of ω has slope $-q$. Let us call ω_q the height of A_q. The situation is sketched for the first values of q in Fig. 2.9 [39]. The top of the ω curve is point A_0, with height $\omega_0 = 0$ in the satisfiable phase and abscissa s_0 (depending on c). With high probability, a randomly drawn instance of 3-BSP will have an entropy of solutions equal to s_0. Though smaller or larger entropies are exponentially unlikely, they may be encountered and are indeed very important to the calculation of the moments. Fig. **??** shows that the q^{th} moment is dominated by very rare instances ($\omega_q < 0$) having very large entropies ($s_q > s_0$). The larger q, the further A_q gets from the typical point A_0, and the more atypical the corresponding instances look like.

We now understand why the first and second moment calculation cannot, in general, provide us with the true value of the threshold. However, there is some hope we can escape from this dead end. While moments of order $q \geq 1$ cannot give us reliable information on the typical entropy s_0, the latter can be identified from the location of the maximum of ω or, equivalently, from the $q \to 0$ limit, see (2.58). This is the purpose of the replica method presented in Section **??**.

To end up this section, let us mention a simple geometrical way to interpret the first and second moment inequalities (2.33). g defined in (2.56) is a convex function of its argument q[40], from which we deduce

$$g(1) \leq \frac{g(0) + g(2)}{2} \quad . \tag{2.60}$$

In addition, the derivative of $g(q)$ in $q = 0$ is equal to the typical entropy s_0, a positive number. Convexity implies that g is an increasing function of $q \geq 0$, and

$$g(0) \leq g(1) \quad . \tag{2.61}$$

[39] The representation of the rate function ω given here is schematic, the exact value of ω as a function of s and c will be calculated in Section **??** and shown in Fig. **??**.

[40] Call $G(q) = \ln(\overline{\mathcal{N}^q})$; G depends on the system size N (the dependence upon ratio c is omitted to simplify notations). Differentiating twice G with respect to q, we obtain

$$G''(q) = \frac{\overline{\mathcal{N}^q (\ln \mathcal{N})^2}}{\overline{\mathcal{N}^q}} - \left(\frac{\overline{\mathcal{N}^q \ln \mathcal{N}}}{\overline{\mathcal{N}^q}} \right)^2 = \frac{1}{\overline{\mathcal{N}^q}} \overline{\left[\mathcal{N}^q \left(\ln \mathcal{N} - \frac{\overline{\mathcal{N}^q \ln \mathcal{N}}}{\overline{\mathcal{N}^q}} \right)^2 \right]} \geq 0 \ . \tag{2.59}$$

Thus $g(q,N) = G(q)/N$ is convex. This property holds in the $N \to \infty$ limit, though $g(q)$ may not be twice differentiable.

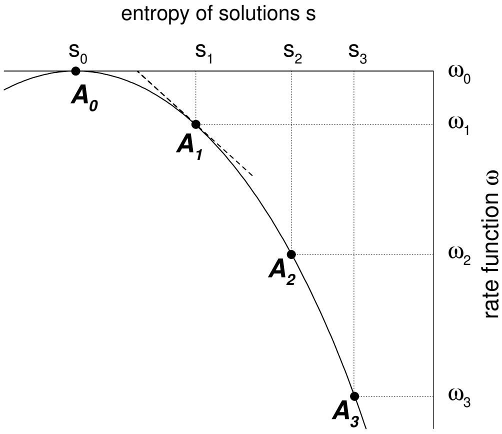

FIG. 2.9. Sketch of the rate function ω for random 3-BSP as a function of the entropy s of the number of solutions and for a fixed ratio c. A_q is the point on the representative curve of ω whose tangent has slope equal to $-q$ (materialised for $q = 1$ by the dashed line). The abscissa s_q represents the entropy of solutions of a typical instance contributing to the q^{th} moment e.g. $s_1 = 1 - c, s_2 = \gamma^*(c)$ for the first two moments, see Section 2.3 . The height ω_q is negative for all $q \geq 1$ showing that these moments reflect the properties of highly unlikely instances. A randomly drawn instance has with high probability entropy s_0, the abscissa of point A_0 with maximal height ω_0.

The geometrical significance of the above identities is illustrated by the schematic plot of g proposed in Fig. 2.10. Multiplication by N and exponentiation of equations (2.60,2.61) leads to

$$\text{Prob}[\mathcal{N} \geq 1] \leq \overline{\mathcal{N}} \leq \sqrt{\text{Prob}[\mathcal{N} \geq 1] \times \overline{\mathcal{N}^2}}\ , \tag{2.62}$$

which gives back (2.33) as announced.

2.4 Mathematical analysis.

We now sketch how the 3-BSP problem may be analysed and the results announced in Section 2.2.2 found rigorously. The techniques used are borrowed from the analysis of algorithms, and probability theory (first and second methods exposed in Section 2.3).

Let $\mathcal{S}$ be a randomly drawn full system. We have seen in Section 2.2.1 that removal of an equation containing a single-occurence variable preserves the satisfiability, or unsatisfiability of $\mathcal{S}$. This procedure can be iterated to further simplify $\mathcal{S}$ until no single-occurence variable is left. The output of the procedure

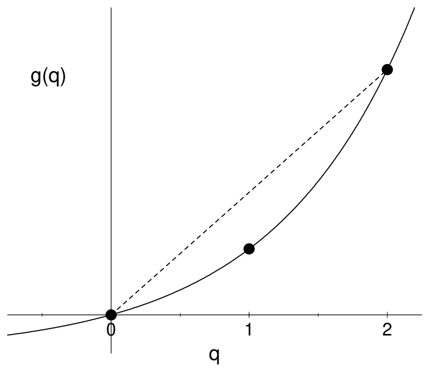

FIG. 2.10. Sketch of the logarithm $g(q)$ of the q^{th} moment of the number of solutions defined in (2.56) as a function of q and for a fixed ratio c. The three dots represent the points $q = 0, 1, 2$, with tangential slopes s_0, s_1, s_2, see Fig. 2.9. The representative curve of g is convex, and therefore lies below the chord over the range $0 \leq q \leq 2$ (dashed line). In addition, as the derivative of g in $q = 0$ is positive and g is convex, $g(q)$ is larger than its value in $q = 0$ for all positive arguments.

is $\mathcal{S}'$, the largest subsystem of $\mathcal{S}$ where variables appear at least twice[41]. In the following, we analyse:

- the procedure which allows us to extract system $\mathcal{S}'$ from $\mathcal{S}$;
- the statistical properties of $\mathcal{S}'$, and their consequences in terms of satisfiability for $\mathcal{S}$;
- how the solutions of the original system $\mathcal{S}$ can be reconstructed from the solutions of $\mathcal{S}'$, and how clustering emerges from this reconstruction process.

2.4.1 *A leaf removal procedure.*

Intuition on single-occurence variable removal is made easier once we introduce the graphical representation of Boolean systems shown in Fig. 2.11. A Boolean system is represented as a hypergraph[42] over a set of N vertices, each of which accounts for a variable. A hyperedge symbolises an equation and is shown as a plaquette joining three vertices. To each plaquette is attached a $0, 1$ label equal to the second member of the equation. Variables absent from the set of equations are isolated vertices, while variables with a single occurence are leaves of the hypergraph, that is, vertices belonging to one plaquette.

[41] From this definition, it is clear that $\mathcal{S}'$ is unique and independent of the order in which equations with single-occurence variable are removed.

[42] A hypergraph is an extension of the concept of graph. In a graph, edges join pairs of vertices. In a hypergraph, each hyperedge connect more than two (here, three) vertices together.

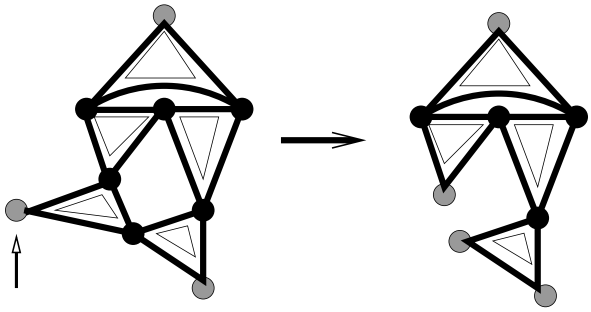

FIG. 2.11. Graph representation of a Boolean system and the operation of the leaf removal procedure. Variables and equations are represented by vertices and plaquettes joining triplets of vertices respectively. The value of the second member attached to each plaquette is not shown here. A step of the leaf removal procedure consists in listing all leaves (corresponding to single–occurence variables, and symbolised by gray vertices), choosing randomly one of them (gray vertex pointed by the arrow), and eliminating this leaf and its attached plaquette. New leaves may appear. Removal is repeated until no leaf is left.

Removal of a single-occurence variable and of its attached equation from the system $\mathcal{S}$ is equivalent to removal of a leaf and its plaquette from the hypergraph. As shown in Fig. 2.11, removal of a leaf may 'uncover' vertices and produce new leaves. The process may therefore be iterated well after all leaves present in the original hypergraph have been removed. How can we quantitatively track the reduction of the hypergraph as removal goes on? Let us call step of the procedure the action of choosing a leaf and removing it together with its plaquette, and ℓ–vertex a vertex which appears in ℓ distinct plaquettes (with $\ell \geq 0$). The number of ℓ–vertices after T steps is a stochastic variable, depending upon the system $\mathcal{S}$ and the sequence of leaves removed by the procedure, denoted by $N_\ell(T)$. Obviously, for all T,

$$\sum_{\ell \geq 0} N_\ell(T) = N \quad \text{and} \quad \sum_{\ell \geq 0} \ell \, N_\ell(T) = 3\,(M - T) \quad , \tag{2.63}$$

as a result of the conservation of the total number of vertices and plaquettes respectively. Removal goes on as long as $N_1(T) \geq 1$. Denote $\mathsf{N}(T)$, and call population the set $\{N_\ell(T), \ell \geq 0\}$ of all numbers of ℓ–vertices. Knowledge of the population is generally not sufficient to unambiguously determine the hypergraph produced by T steps of the removal procedure. Indeed, many hypergraphs have the same population $\mathsf{N}(T)$ An essential observation is, however, that the output of T steps of the removal procedure is equally distributed among the set of all hypergraphs having population $\mathsf{N}(T)$. In other words, a complete statistical information about hypergraphs is obtained from the knowledge of the population.

Let us now see how this population $\mathsf{N}(T)$ is modified during step $T \to T+1$. The variations of the N_ℓs are stochastic variables due to the randomness in the system $\mathcal{S}$ and the choice of the leaf to be removed, with conditional expectations with respect to $\mathsf{N}(T)$ given by

$$\mathbf{E}\big[N_\ell(T+1) - N_\ell(T)|\mathsf{N}(T)\big] = -\mathbf{1}_{\ell=1} + \mathbf{1}_{\ell=0} + 2\,p_{\ell+1}(T) - 2\,p_\ell(T) \quad . \tag{2.64}$$

When a plaquette is removed, a 1-vertex disappears ($-\mathbf{1}_{\ell=1}$ term in (2.64)) to become a 0-vertex ($\mathbf{1}_{\ell=0}$). This plaquette is attached to two other vertices. The numbers ℓ, ℓ' of occurrences of each of these two vertices are distributed with probability

$$p_\ell(T) = \frac{\ell\, N_\ell(T)}{3\,(M-T)} \quad , \tag{2.65}$$

and are diminished by one once the plaquette is taken away[43]. From (2.64), the expectation values of N_ℓ vary by a quantity of the order of unity at each time step, and the expected densities of ℓ-vertices, defined through

$$n_\ell(T) = \frac{\mathbf{E}\big[N_\ell(T)\big]}{N} \quad , \tag{2.66}$$

undergo changes of the order of $1/N$ only. We are naturally led to conclude that densities are function of a much longer 'time scale', $t = T/N$. In other words, given $t \in [0;1]$, $n_ell([tN])$ and $n_ell([tN]+o(N))$ are equal to within $o(1)$ terms as N tends to infinity, and we denote by $n_\ell(t)$ their common limit. Assuming that the densities $n_\ell(t)$ are differentiable functions of this time, we obtain from evolution equation (2.64) a set of coupled first order differential equations[44],

$$\frac{dn_\ell}{dt}(t) = -\mathbf{1}_{\ell=1} + \mathbf{1}_{\ell=0} + \frac{2}{3(c-t)}\big[(\ell+1)\,n_{\ell+1}(t) - \ell\, n_\ell(t)\big] \ . \tag{2.67}$$

Resolution of these equations require to stipulate the initial conditions. At time $t=0$, that is, prior to any removal of plaquette, densities are Poisson distributed with parameter $3\,c$,

$$n_\ell(0) = e^{-3\,c}\,\frac{(3\,c)^\ell}{\ell!} \quad , \forall\ \ell\ . \tag{2.68}$$

Demonstration of the above identities is left to the care of the reader in Exercise ??.

Solutions of equations (2.67) with initial conditions (2.68) read

$$n_0(t) = e^{-3\,c\,b(t)^2} + 3\,c\,b(t)^2\,\big(1-b(t)\big) \ ,$$

[43]The probability p_1 of picking up a variable, say, x_1, is equal to the number ℓ_1 of its occurrences divided by the total number of variables in the system, $3(M-T)$. The probability of picking any variable with occurrence ℓ is p_1 multiplied by the number $N_\ell(T)$ of such variables, hence (2.65). Equation (2.63) ensures that p_ℓ is properly normalized.

[44]The left hand side of (2.64) reads $N[n_\ell(t+1/N) - n_\ell(t)] = dn_\ell/dt + O(1/N)$ through a Taylor expansion.

$$n_1(t) = 3\,c\,b(t)^2 \left(e^{-3\,c\,b(t)^2} + b(t) - 1\right) \ ,$$
$$n_\ell(t) = e^{-3\,c\,b(t)^2}\,\frac{\left(3\,c\,b(t)^2\right)^\ell}{\ell!} \ , \quad \forall\, \ell \geq 2 \ , \tag{2.69}$$

where

$$b(t) \equiv \left(1 - \frac{t}{c}\right)^{1/3} \quad . \tag{2.70}$$

These equations are valid as long as n_1 is positive, since the procedure stops when no leaf is left. The density $n_1(t)$ of 1–vertices is showed on Fig. 2.12 for various initial ratios c of equations per variable. The shape of the curve reflects the competition between two opposite effects: the annihilation of 1–vertices due to leaf removal and their creation as a result of reduction of 2-vertices tp 1-vertices. At small ratios e.g. $c = 0.7$, the former mechanism dominates and n_1 is a decreasing function of time. For slightly larger ratios, the density n_1 does not monotonously decrease with time any longer, but still vanishes at time $t^* = c$ *i.e.* when no plaquette is left.

According to the solution (2.69) of the equation of motion, this statement holds for ratios c for which the equation

$$b = 1 - e^{-3\,c\,b^2} \quad , \tag{2.71}$$

has no strictly positive root. Numerical resolution of (2.71) is shown in Fig. 2.13. When c is larger than

$$c_D = \min_{0<b<1} \left[- \frac{\ln(1-b)}{3\,b^2} \right] \simeq 0.818469... \quad , \tag{2.72}$$

there exist non zero real solutions to (2.71), and we denote by b^* the largest one. This root is plotted as a function of the ratio c in Fig. 2.13. As shown in Fig. 2.12, the density of 1–vertices decreases and vanishes at time $t^* < c$ where $b(t^*)$ (2.70) reaches b^*. The output of the procedure is a non-empty subset $\mathcal{S}'$ of $\mathcal{S}$ with $M' = N\,(c - t^*)$ equations, such that any of the

$$N' = N \sum_{\ell \geq 2} n_\ell(t^*) \tag{2.73}$$

variables present in $\mathcal{S}'$ appears at least twice. The ratio of equations per variable in $\mathcal{S}'$ is given by

$$c' = \frac{M'}{N'} = \frac{c\,b^{*\,2}}{1 - 3\,c\,b^*\,(1-b^*)} \quad , \tag{2.74}$$

and is shown in Fig. 2.13.

Let us briefly sum up the outcome of the above analysis. There exists a critical ratio c_D (2.72) for the original system $\mathcal{S}$ such that

- if $c < c_D$, the leaf removal procedure succeeds in eliminating all variables and equations, thus $\mathcal{S}'$ is empty;

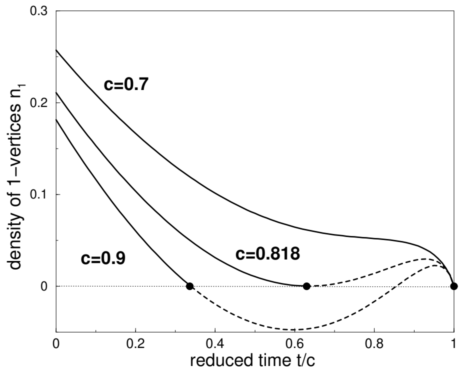

FIG. 2.12. Evolution of the density of 1-vertices, $n_1(t)$, under the operation of the leaf removal procedure as a function of the reduced time t/c. For $c < c_D \simeq 0.818$, $n_1(t)$ remains positive until all the plaquettes are eliminated at $t^* = c$. For $c > c_D$ the procedure stops at the time t^* for which n_1 vanishes (black dots), and becomes negative for $t > t^*$ (dashed part of the curves). Notice that t^* discontinuously jumps down at $c = c_D$.

- if $c > c_D$, the output of the procedure is a non-empty subset $\mathcal{S}'$, with ratio of equation per variables equal to c' (2.74).

The reader could feel concerned about fluctuations around those average results. Equation of motion (2.67) is indeed true for the average density of vertices only. Fortunately, one can show that, for large sizes N, the numbers of ℓ–vertices are highly concentrated around their average values,

$$N_\ell(T) = N\, n_\ell\left(\frac{T}{N}\right) + o(N) \quad , \tag{2.75}$$

whatever the initial instance $\mathcal{S}$ and the sequence of choices done by the removal procedure. Therefore, the evolution of the population cannot deviate from the average behaviour when $N \to \infty$. A more detailed explanation of this concentration property will be provided in Chapter 7.

2.4.2 *Clustering and sat–unsat transitions.*

As before, we consider a system $\mathcal{S}$ with ratio c and apply the leaf removal procedure. The output is the subsystem $\mathcal{S}'$. Can we reconstruct the solutions of $\mathcal{S}$ from the ones of $\mathcal{S}'$? The answer is positive, and the reconstruction process schematized in Fig. 2.14 permits us to characterize the structure of the set of solutions in an accurate way.

Assume that we have, in the course of the removal procedure, store all removed equations on top of each other in a stack. Let X' be a solution of $\mathcal{S}'$[45].

[45] We do not care here about how this solution is found, but one could, for instance, apply the Gaussian elimination algorithm of Section 2.1 to $\mathcal{S}'$.

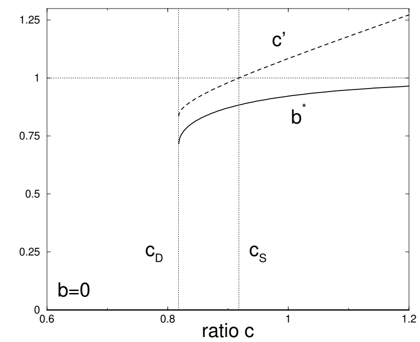

FIG. 2.13. Backbone size b^* (2.71) -full curve- and ratio c' (2.74) for the reduced system $\mathcal{S}'$ -dashed curve- as functions of c. b^* and c' take non zero values right above $c = c_D \simeq 0.818$ (leftmost vertical dotted line). The threshold $c' = 1$ (horizontal dotted line) for the disappearance of solution is reached when $c = c_S \simeq 0.918$ (rightmost vertical dotted line).

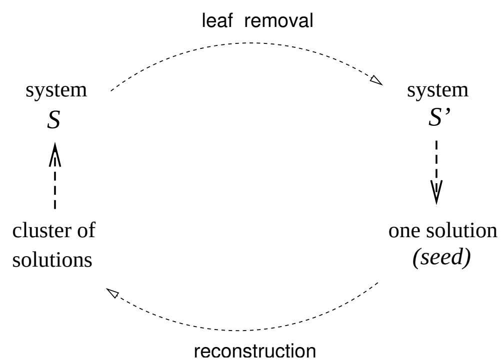

FIG. 2.14. Reconstructing solutions. The leaf removal procedure extract a subsystem $\mathcal{S}'$ (right) from the original system $\mathcal{S}$ (left). From each solution of $\mathcal{S}'$, we reconstruct solutions of $\mathcal{S}$ by reinserting removed equations one after the other. Some freedom is possible to assign newly reintroduced variables and thus a single solution of $\mathcal{S}'$ (seed) allows us to reconstruct a whole cluster of solutions for $\mathcal{S}$. Notice that, when $c < c_D$, the subsystem $\mathcal{S}'$ is almost surely empty, as is the unique seed.

The length of X', that is, the number of variables it includes is N' defined in (2.73). Through a relabelling of variables we can always assume that X' specifies the values of the first N' variables x_i. We are now going to reinsert in $\mathcal{S}'$ the equations that were removed from $\mathcal{S}$ following a Last–In–First–Out unstacking order.

Consider the first reinserted *i.e.* on top of stack equation,

$$x_i \oplus x_j \oplus x_k = v \quad , \tag{2.76}$$

where $v = 0, 1$ is the second member, and (i, j, k) a triplet of distinct integers comprised between 1 and N. Call ν_1 the number of these integers strictly larger than N'. That this equation was eliminated by the leaf removal procedure and was the last one to be so tells us that $\nu_1 \geq 1$: at least one of the three variables in (2.76) is not assigned by the particular solution X'. If ν is precisely equal to unity, we have, say, $i, j \leq N'$ and $k \geq N' + 1$. Then the values of x_i, x_j are known for X', and there is a unique way to assign x_k to satisfy (2.76). If $\nu_1 = 2$, we know, say, x_i from X', and have two possible combinations of x_j, x_k fulfilling (2.76). Therefore we can construct two distinct solutions of the system $\mathcal{S}_1' = \mathcal{S}' +$ reinserted equation. The reasoning is straightforwardly extended to the last $\nu_1 = 3$ case, with the general result that $2^{\nu_1 - 1}$ distinct solutions can be reconstructed from X'. Then, after reinsertion of the second-to-top equation, we reconstruct $2^{\nu_2 - 1}$ solutions from any of the solutions of $\mathcal{S}_1'$ where ν_2 is the number of new variables *i.e.* present in the reinserted equation and not fixed by S_1'. Iterating this procedures up to the reinsertion of all $M - M'$ equations permits us to obtain

$$\mathcal{N}_{rec} = \prod_{T=1}^{M-M'} 2^{\nu_T - 1} \tag{2.77}$$

solutions of $\mathcal{S}'_{M-M'} = \mathcal{S}' +$ all equations. Notice that solutions of $\mathcal{S}'_{M-M'}$ do not quite coincide with the the ones of the original system $\mathcal{S}$. Variables that were absent from equations in $\mathcal{S}$ are still undetermined from $\mathcal{S}'_{M-M'}$, and can be freely chosen, giving an extra multiplicative factor to the total number of solutions of $\mathcal{S}$ that can be reconstructed from X',

$$\mathcal{N}_{in} = 2^{N_0}\, \mathcal{N}_{rec} = 2^{N_0} \prod_{T=1}^{M-M'} 2^{\nu_T - 1} \quad , \tag{2.78}$$

where N_0 is the number of 0–occurrence variables. For reasons which will be clarified later on this set of solutions and X' are respectively called cluster of solutions and seed of the cluster (Fig. 2.14). Taking the logarithm of (2.77) and dividing by N, we obtain the following expression for the entropy of the reconstructed from seed X',

$$s_{in} = \frac{N_0}{N} + \frac{1}{N} \sum_{T=1}^{M-M'} (\nu_T - 1) \quad . \tag{2.79}$$

The average value of this entropy can be calculated from the knowledge of the average values of the ν_Ts. As equations were carefully introduced in opposite order to their removal, the expectation value of $\nu_T - 1$ (conditioned to population $\mathsf{N}(T)$) can be estimated from the analysis of Section 2.4.1 and is equal to $2\, p_1(M' + 1 - T)$ (2.65), and we end up with

$$s_{in} = e^{-3\,c} + \int_0^{t^*} dt\, \frac{2\, n_1(t)}{3\,(c-t)} \ , \tag{2.80}$$

where the first term comes from the contribution $n_0(0)$ (2.68) of absent variables, and t^* is the time at which the leaf removal procedure halts.

2.4.2.1 *Case $c < c_D$.* For small ratios, $\mathcal{S}'$ is empty, and so is X'. The halt time $t^* = c$ and integration of (2.80) with the help of (2.69) leads to the simple result $s_{in}(c) = 1 - c$. As we have reconstructed all possible solutions of $\mathcal{S}$ from the empty X', s_{in} coincides with the total entropy s of solutions, proving the announced result (2.28)[46].

2.4.2.2 *Case $c > c_D$.* In this range of ratios, the leaf removal procedure stops at $t^* < c$, and has not succeeded in eliminating all equations and variables. The average entropy associated to a cluster reconstructed from one seed is

$$s_{in}(c) = 1 - c - b^* - c\, b^{*\,2}(2\, b^* - 3) \quad , \tag{2.81}$$

where b^* is, as before, the largest positive root of (2.71).

To complete our description of clusters, some statistical knowledge about their seeds is required. The number $\mathcal{N}'$ of solutions of $\mathcal{S}'$ can be analyzed by means of the first and second moments method (2.33), giving respectively some upper and lower bound to the probability Prob$[\mathcal{N}' \geq 1]$ of existence of solutions. The first moment is easy to calculate along the lines of Section 2.3.1; as the second members of the various equations are uncorrelated, we find

$$\overline{\mathcal{N}'} = 2^{N'} \times \left(\frac{1}{2}\right)^{M'} = 2^{N'\left(1-c'\right)} \quad . \tag{2.82}$$

The overbar denotes here the unbiased expectation value over all instances with N' variables, M' equations such that any variable appears at least twice. The second moment calculation is made more difficult by the existence of contraints on the minimal number (two) of occurrences of variables in $\mathcal{S}'$. It requires a combinatorial analysis of the number of systems $\mathcal{S}'$ *i.e.* ways of choosing equations, having a given pair of configurations for solutions. This calculation is beyond the scope of this chapter and was done by Dubois and Mandler (2002). The outcome is that, at large N, the second moment $\overline{(\mathcal{N}')^2}$ is asymptotically equal to the first moment squared, $\left(\overline{\mathcal{N}'}\right)^2$, when $c' < 1$ and exponentially larger when $c' > 1$. This result has two important consequences.

First, from (2.33), we conclude that $c' = 1$ is at the same time an upper and a lower bound to the exact value of the threshold for $\mathcal{S}'$. Therefore $c'_S = 1$ is the

[46] Once more, we make use of the concentration of measure mentioned at the end of Section 2.4.1. The entropy of solutions inside a cluster, s_{in}, is with high probability equal to its average value when $N \to \infty$.

location of the sat-unsat threhold for reduced subsystems. Using (2.74), we see that $c' = 1$ is reached for an initial ratio equal to

$$c_S \simeq 0.917936... \quad , \tag{2.83}$$

as shown in Fig. 2.13. Remarkably, while the moments method directly applied to $\mathcal{S}$ gave lower and upper bounds to the threshold separated by a finite gap only, the outcome is much better when applied to $\mathcal{S}'$ (Fig. 2.8). The concentration of $\mathcal{N}'$ constrasts with the (instance–to-instance) fluctuations exhibited by $\mathcal{N}$, and suggests that the latter thus essentially comes from fluctuations in the numbers N_0 and N_1 of 0- and 1-vertices (whose plaquettes form the dangling ends of the graph on Fig. 2.11) eliminated by the leaf removal algorithm. This comes as no surprise since variations of N_0 and N_1 induce drastic changes on the number of solutions e.g. the presence of a 0-vertex multiply the number of GS by two. Conversely, in $\mathcal{S}'$, variables appear at least twice and are more interconnected, giving rise to weaker fluctuations for $\mathcal{N}'$.

Secondly, when $c' < 1$, the number of solutions of $\mathcal{S}'$ is, with high probability, given by (2.82)[47]. This is the number of seeds each of which gives a cluster of solutions for $\mathcal{S}$. The number of clusters of solutions is therefore

$$\mathcal{N}_{clu} = 2^{N'(1-c')} = 2^{N\, s_{clu}} \quad , \tag{2.84}$$

where the entropy of clusters is defined as

$$s_{clu}(c) = \frac{N'}{N}\,(1-c') = b^* - 3\,c\,b^{*\,2} + 2\,c\,b^{*\,3} \quad . \tag{2.85}$$

From the reconstruction process it is clear that two different seeds cannot give twice the same solutions. Therefore the total entropy of solutions of $\mathcal{S}$ is

$$s(c) = s_{clu}(c) + s_{in}(c) = 1 - c \quad , \tag{2.86}$$

as obtained when summing (2.85) and (2.81). We have now established (2.28) for all ratios $c < c_S$. The entropies of solutions in a cluster, s_{in}, and of clusters, s_{clu}, are shown in Fig. 2.4.

What we are left with is merely justifying the cluster denomination used above. The reconstruction process allows a complete characterization of solutions, in terms of an extensive number of (possibly overlapping) blocks made of few variables, each block being allowed to flip as a whole from a solution to another. When $c < c_D$, with high probability, two randomly picked solutions differ over a fraction $d = 1/2$ of variables, but are connected through a sequence of $O(N)$ successive solutions differing over $O(1)$ variables only. For $c_D < c < c_S$, flippable blocks are juxtaposed to a set of seed-dependent frozen variables. For picture 2.6 to be true, we must establish that different clusters do not overlap

[47] This statement comes from the fact that the second moment is equal to the squared first moment, and therefore fluctuations of $\mathcal{N}'$ around the average value are extremely rare.

or, more precisely, that any two solutions belonging to two different clusters are far away from each other. A lower bound to the Hamming distance d between these two solutions is the Hamming distance d_s between the seeds of their respective clusters. A lower bound d_s, as a function of c, can in turn be estimated using again the first moment method *i.e.* from the vanishing condition of the expectation number $\overline{\mathcal{N}'^2(d_s)}$ of solutions of $\mathcal{S}'$ lying apart at distance $\leq d_s$. An explicit calculation shows that $d_s > 0$ for all $c > c_D$ (Cocco et al. 2003; Mezard, Ricci-Tersenghi, Zecchina 2003).

2.4.2.3 *On backbones.* The reconstruction procedure helps us to interpret equation (2.71) as a self-consistent equation for the size of the backbone of a cluster of solutions. Let X' be a solution of $\mathcal{S}'$, and $\mathcal{B}$ the set of variables taking the same value in all solutions of the cluster associated to seed X'. The cardinality of $\mathcal{B}$ divided by N is the backbone b. Obviously, the variables assigned by X' are elements of $\mathcal{B}$, thus

$$b \geq \sum_{\ell \geq 2} n_\ell(t^*) = b^*(1 - 3\,c\,b^*(1-b^*)) \quad , \tag{2.87}$$

showing that the backbone is strictly positive. So far no proof of the equality between b and b^* has been obtained in literature, though this could, in principle, be inferred from the analysis of the removal and reconstruction procedures described above. The following is a hand-waving argument supporting the interpretation of b^* as the cluster backbone (Ricci-Tersinghi, Weigt, Zecchina 2001).

We want to estimate the probability that a variable, say, x_1 does not belong to the backbone *i.e.* that x_1 takes value 0 on some solutions and 1 on others. x_1 appears in ℓ equations, where ℓ is a Poisson variable with parameter $3\,c$. Consider one of these equations,

$$x_1 \oplus x_j \oplus x_k = v \quad . \tag{2.88}$$

If either x_j or x_k does not belong to $\mathcal{B}$, neither does x_1. Thus the probability that the above equation does not constrain x_1 to take the same value on all solutions is $1 - b^2$. This expression assumes that the events '$x_j \in \mathcal{B}$ ' and '$x_k \in \mathcal{B}$' are independent, a literally wrong assumption which, however, apparently becomes asymptotically true for large system sizes. Similarly, under the assumption that the above events are independent from equation to equation, we obtain

$$\text{Prob}[x_1 \in \mathcal{B}|\ell] = 1 - (1-b^2)^\ell \quad . \tag{2.89}$$

Summing over ℓ, we have

$$\text{Prob}[x_1 \in \mathcal{B}] = \sum_{\ell=0}^{\infty} e^{-3\,c} \frac{(3\,c)^\ell}{\ell!} \left[1 - (1-b^2)^\ell\right] = 1 - e^{-3\,c\,b^2} \; . \tag{2.90}$$

As the l.h.s. of the above equation is precisely the size of the backbone, b, we see that b fulfills (2.71) and is equal to b^* as announced.

2.5 A testground for resolution procedures.

The deep understanding of the statistical properties of 3-BSP makes this problem a valuable benchmark for assessing the performances of various combinatorial search algorithms. At first sight, the idea seems rather odd. 3-BSP is a polynomial problem, and we may expect that common search procedures will not perform better than the polynomial algorithm exposed in Section 2.1. Actually most of them will have exponentially long average running times! Yet such results are interesting in that they show that exponentiality does not result from the intrinsic complexity of the problem but from the poor quality of the algorithms. After all, we cannot discard the (remote) possibility that the same situation happens for NP-complete problems e.g. SAT.

2.5.1 *Gradient descent.*

The first algorithm we consider is gradient descent (GD). GD is defined as follows:

- Start from a randomly chosen configuration of the variables. Call energy the number E of unsatisfied equations.
- While $E \geq 1$;
  - * let V be the set of variables x_i that make the energy decrease when negated ($x_i \to \bar{x}_i$);
  - * if V is empty, print 'Final energy = E', and stop;
  - * otherwise, pick up uniformly at random one variable in V, negate it, update E;
- Print 'Satisfiable', and stop.

GD is the simplest procedure allowing one to catch the gross structure of the energy landscape associated to 3-BSP. If the landscape is simple (Fig. 1.9A), GD should be able to find solutions in linear time. Conversely, if the landscape is plagued with multiple local minima (Fig. 1.9B&C), GD will get stuck and never reach the bottom energy (= 0 in presence of a solution). Numerical experiments indicate that GD is generally unable to find solutions, and Fig. 1.9B or C are the correct pictures for the landscape. This statement can be proven as follows.

Let us apply GD to an instance of 3-BSP. The system of equations has a graph representation sketched in Fig. 2.15, already introduced in Section 2.4.1. Variables are represented by vertices, and equations by plaquette joinings three variables. Once a configuration of the variables is chosen, each plaquette may be labelled by its status, S or U, whether the associated equation is respectively satisfied or unsatisfied (Fig. 2.15). A fundamental property of 3-BSP is that each time a variable is changed, *i.e.* its value is negated, the equations it belongs to change status too.

This property makes the analysis of some properties of GD easy. Consider the system made of 15 vertices and 7 plaquettes in Fig. 2.16, and suppose the central plaquette is violated (U) while all other plaquettes are satisfied (S) (leftmost graph). The number of unsatisfied equations is $E = 1$. Now run GD on this special instance of BSP. Two cases arise, symbolized in Fig. 2.16, whether the vertex attached to the variable to be flipped belongs, or not, to the central plaquette. It is an easy check that, in both cases, $E' = 2$ and the change is not permitted by GD. The system of Fig. 2.16 will be called hereafter island. When the central plaquette has status U f and all the other plaquettes have status S, the island is called blocked. Though the system of 3-BSP encoded by a blocked island is obviously satisfiable (think of negating at the same time one variable attached to a vertex V of the central plaquette and one variable in each of the two peripherical plaquettes joining the central plaquette at V), GD will never be able to find a solution and will be blocked forever in the local minimum with height $E = 1$.

This situation typically happens for randomly drawn systems. Cocco et al. (2004) have shown that the number of blocked islands in a randomly drawn system is, with high probability when $N \to \infty$, equal to $(\Psi_1(c) + o(1))\, N$ where

$$\Psi_1(c) = \frac{729}{1024}\, c^7\, e^{-45\, c} \quad . \tag{2.91}$$

Since each blocked island has one unsatisfied equation, Ψ_1 is also a lower bound to the number of violated equations per variable. Hence, when GD stops, the number of unsatisfied equations left will be of the order of N.

The argument is easily generalizable to gradient descent with more than one look ahead. To extend the notion of blocked island to the case where GD is allowed to invert R, and not only 1, variables at a time, it is sufficient to have $R + 1$, and not 2, peripherical plaquettes attached to each vertex of the central plaquette. Enumeration of these blocked islands leads to a strictly positive lower bound $\Psi_R(\alpha)$ for the final number of violated equations (divided by N). As a consequence, GD, even with R simultaneous flips permitting to overcome local barriers, remains almost surely trapped at an extensive (in N) level of violated equations for any finite R. Actually the lower bound $\Psi_R(c)N$ tends to zero only if R is of the order of $\log N$[48].

2.5.2 *Random Walk procedure.*

The second procedure we consider is the Random Walk (RW) algorithm introduced by Papadimitriou (1992). RW is based on the observation that a violated equation can be satisfied through negation of one of its variables:

- Start from a randomly chosen configuration of the variables. Call energy the number E of unsatisfied equations.
- While $E \geq 1$;
  - ∗ pick up at random and uniformly one of the E unsatisfied equations;
  - ∗ pick up at random and uniformly one its 3 variables;
  - ∗ negate the value of this variable, update E;

[48]A similar conclusion about the height of the barriers was reached by Montanari and Semerjian through an analysis of the formula belonging to the giant cluster...

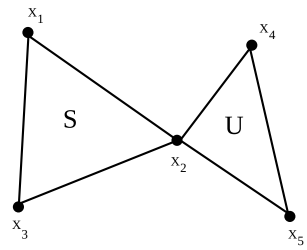

FIG. 2.15. Graphical representation of a system with equations $x_1 \oplus x_2 \oplus x_3 = v_1$ and $x_2 \oplus x_4 \oplus x_5 = v_2$. Each equation is represented by a plaquette whose vertices are the attached variables (the second members v_1, v_2 are not represented). When the variables are assigned some values, the equations can be satisfied (S) or unsatisfied (U).

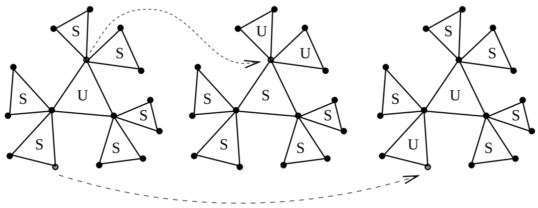

FIG. 2.16. A blocked island (left) is a system of 7 equations (1 central, 6 peripherical) with variables such that the central plaquette is unsatisfied and all peripherical plaquettes are satisfied. Inversion of any variable (grey vertex) increases the number of unsatisfied clauses by 1, be it attached to the central (middle) or to a peripherical (right) plaquette.

- PRINT 'SATISFIABLE', AND HALT.

Notice that, as a result of the negation, some equations that were satisfied may become violated. Therefore the energy is not guaranteed to decrease with the number of steps of the algorithm. From one hand, contrary to GD, RW is able to escape from local minima of the energy landscape, and is *a priori* capable of better performances. From the other hand, RW may run forever... A major question is how long should the algorithm be running before we stop thinking that the studied system has solutions hard to find and get some confidence that there is really no solution. The theoretical methods developed for answering this question and, more generally, assessing the performances of RW, together with the results obtained so far will be presented in Chapter **??**. Here we limit ourselves to a heuristic presentation of the scenario issued from numerical experiments.

Experiments indicate that there is critical value of the ratio of equations per variables, $c_E \simeq 0.33$, hereafter referred to as dynamical threshold, separating two regimes:

- for $c < c_E$, RW generally finds a solution very quickly, namely with a number of flips growing linearly with the number of variables N. Figure 2.17

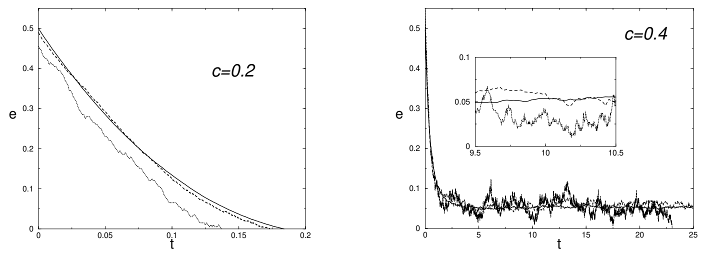

FIG. 2.17. Fraction e of unsatisfied equations as a function of time t (number of steps divided by N) during the operation of RW on a randomly drawn system of 3-BSP with ratio $c = 0.2$ (left) and $c = 0.4$ (right) with $N = 10^3$ (dotted), 10^4 (dashed), 10^5 (full curve) variables. Note the difference of horizontal scales between the two figures. Inset: blow up of the $t \in [9.5; 10.5]$ region; the amplitude of fluctuations around the plateau decreases with increasing size N.

shows the plot of the fraction e of unsatisfied clauses as a function of the time (number of steps) T for one randomly drawn system with ratio $c = 0.2$ and $N = 500$ variables. The curve shows a fast decrease from the initial value ($e(T = 0) = \frac{1}{2}$ independently of c for large values of N, but deviations can be found at small sizes, see Fig. 2.17) down to zero on a time scale $\theta(N)$[49]. The resolution time T_{res} depends both on the system of equations under consideration and the choices of the algorithm; its expectation value scales as

$$\mathbf{E}\big[T_{res}\big] = N\, t_{res} + o(N) \ . \tag{2.92}$$

where the coefficient t_{res} is an increasing function of c[50].

- for systems with ratios of equations per variable in the $c_E < c < c_S$ range, the initial relaxation regime taking place on the $T = \theta(N)$ time scale does not allow RW to reach a solution (Fig. 2.17B). The fraction e of unsat equations then fluctuates around some plateau value $e_{plateau}$ for a very long time. Fluctuations are smaller and smaller (and the height of the plateau better and better defined) as the size N increases. As a result of fluctuations, the fraction e of unsatisfied equations may temporarily either increase or decrease. When a fluctuation happens to drive RW to $e = 0$, a solution is found and the algorithm stops. The corresponding resolution time, T_{res}, is stochastic; numerical experiments for different sizes N indicate that its expectation value scale as

[49] This decrease characterises the overall operation of RW. A precise look at the $e(T)$ curve reveals that the energy e may occasionally increase.

[50] On intuitive grounds, as a step of the algorithm can satisfy $\theta(1)$ equations at a time, we expect the average value of T_{res} to be of the order of the number M of equations at least. Thus t_{res} should grow at least linearly with c. Experiments shows that the growth is in fact more than linear, see Section **??** for a theoretical study.

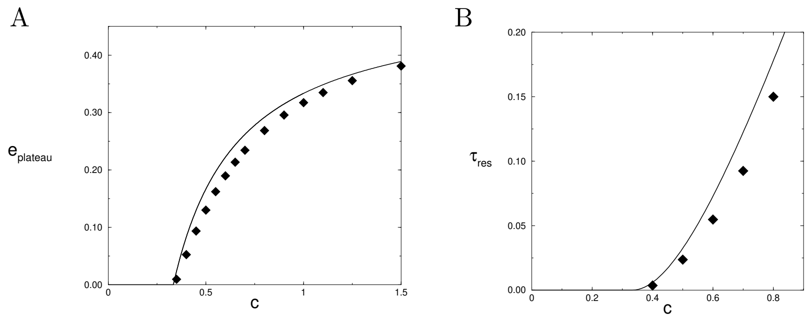

FIG. 2.18. Fraction $e_{plateau}$ of unsatisfied equations on the plateau (**A**) and logarithm τ_{res} of the average resolution time divided by N (**B**) as a function of the ratio c of equations per variable. Diamonds are the output of numerical experiments, and have been obtained through average of data from simulations over 1,000 systems and runs of RW for various sizes N, and extrapolation to $N \to \infty$ (Semerjian, Monasson 2003; Barthel, Hartmann, Weigt 2003). Full lines are theoretical approximations from Section **??**.

$$\mathbf{E}\big[T_{res}\big] = e^{N\,\tau_{res}+o(N)} \ . \tag{2.93}$$

where the coefficient τ_{res} is an increasing function of c. The plateau energy $e_{plateau}$ and the logarithm τ_{res} of the resolution time are shown in Fig. 2.18.

Notice that the dynamical threshold c_E above which the plateau energy is positive (RW stops finding a solution in $\theta(N)$ steps) is strictly smaller than the critical threshold $c_S \simeq 0.918$, where systems go from satisfiable with high probability to unsatisfiable with high probability. In the intermediate range $c_E < c < c_S$, systems are almost surely satisfiable but RW needs an exponentially large time to prove so.

In conclusion, the average resolution time of RW appears to undergo a polynomial/exponential, more precisely linear/exponential transition at some well defined value c_E of the ratio of equations per variable. The study of GD has taught us that the energy landscape of 3-BSP shows many local minima for any positive c. Remarkably, RW is capable to overcome these minima in a time growing linearly with the number of variables only when $c < c_E$. Above c_E, RW remains trapped at a high energy level (plateau energy of Fig. 2.18) for an exponentially large time. We shall see in the next Chapter that this trapping phenomenon is common in statistical physics, and is the basis for the important notion of metastable state.

2.5.3 *Backtrack search.*

The third and last procedure we study is the Davis-Putnam-Loveland-Logemann (DPLL) algorithm. Contrary to GD or RW, DPLL can prove the absence of solutions to a given system S, and for this reason is called complete algorithm.

The procedure, widely used in practice, is based on two principles: backtracking and unit-propagation. During the course of the algorithm, some values are assigned to the variables e.g. $x_1 = 1$, and equations updated accordingly e.g. $x_1 \oplus x_2 \oplus x_3 = 0 \to x_2 \oplus x_3 = 1$. If some equation turns out to be violated, DPLL backtracks to the last assigned variable, negates it e.g. $x_1 = 0$, and the search process resumes. The procedure halts either when a sequence of assignments have succeeded in satisfying all equations (a solution is then found), or when all possible values for the variables have been tried in vane and found to be contradictory with the system (a proof of unsatisfiability is then obtained). An important aspect of the procedure regards how to assign variables. Priority is given to variables that appear in unit-equations e, that is, reduced to single variable equations e.g. $e : x_2 = 0$ as a result of prior assignments. The unit-propagation principle states that one should rather satisfy these equations before doing any other variable assignment. Unit-propagation allows DPLL to draw logical consequences from previous assignments and makes the procedure much more efficient than a pure, truth-table-like procedure. If no unit-equation is present, a variable is assigned according to some heuristic rule specified below, an operation called variable splitting. The major difference between splitting and unit-propagation is that the latter is simply a consequence of past choices, while the former operation is arbitrary and both values for the split variable are *a priori* admissible. If, unluckily, the value guessed by the heuristic finally leads to a contradiction, backtracking will be unavoidable, making the search process much longer.

DPLL can therefore be described as a recursive function of the assignment A of the variables. Initially, DPLL is called with the empty assignment $A = \emptyset$ since no variable has been set[51]:

PROCEDURE DPLL[A]

- LET S_A BE WHAT IS LEFT FROM S GIVEN VARIABLE ASSIGNMENT A;
- IF S_A IS EMPTY, PRINT 'SATISFIABLE'; HALT;
- IF S_A CONTAINS A VIOLATED EQUATION, PRINT 'CONTRADICTION', RETURN; *(backtracking)*
- OTHERWISE, LET U BE THE SET OF UNIT-EQUATIONS IN S_A;
  * IF $U \neq \emptyset$, PICK-UP ONE OF THE EQUATIONS IN U, SAY, e, AND CALL DPLL[A$\cup\{e\}$]; *(unit-propagation)*
  * IF $U = \emptyset$, CHOOSE A NOT-YET-ASSIGNED VARIABLE, SAY, x, AND ITS VALUE v ACCORDING TO SOME HEURISTIC RULE, AND CALL DPLL[A$\cup\{x = v\}$], THEN DPLL[A$\cup\{x = \bar{v}\}$]; *(variable splitting)*

Rules for assigning variables in the absence of unit-equations are heuristic in that they aim at doing good assumptions *i.e.* diminishing as much as possible the search process to come from a limited information about the current system

[51] A more complete description of the algorithm for the SAT and COL problems will be given in Chapter **??**.

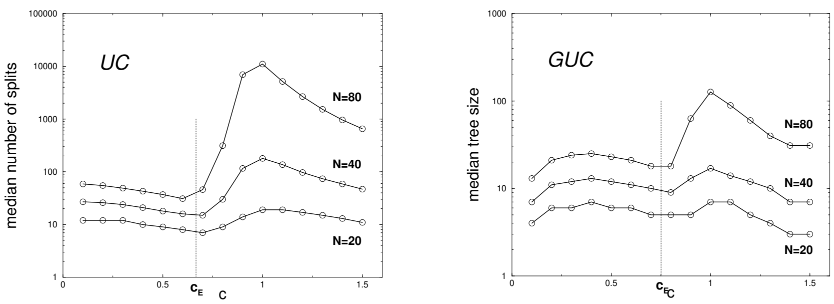

FIG. 2.19. Median number of splits required by the DPLL procedure with the UC (left) and GUC (right) heuristics as a function of the ratio c of equations per variable, and for $N = 20, 40, 60$ variables (from bottom to top). Data have been extracted from the resolution of 10,000 randomly drawn systems; continuous lines are guidelines for the eye. Note the difference of (logarithmic) scale between UC and GUC curves showing that DPLL-GUC is much more efficient than DPLL-UC. The polynomial/exponential transition is located at ratios $c_E = \frac{2}{3}$ and $c_E = 0.7507...$ for UC and GUC respectively.

of equations. Of course, perfect heuristic do exist: trying all possible values for not-yet-assigned variables would ensure that no wrong guess is ever done! But the time required would be exponentially long. In practice, heuristics have to make their decision in polynomial time. Two simple splitting heuristics are[52]:

- ◇ UC: choose at random and uniformly any unset variable, and assign it to 0 or 1 with equal probabilities ($\frac{1}{2}$).
- ◇ GUC: choose at random and uniformly any equation with minimal length *i.e.* involving 2 variables if any, or 3 variables otherwise. Pick up at random and uniformly one its variable, and assign it to 0 or 1 with equal probabilities ($\frac{1}{2}$).

UC amounts to make a random guess and is the simplest possible heuristic. GUC is more clever since each time a split is done from a 2-variable equation, this equation is turned into a unit-equation, and a forced assignment through unit-propagation is subsequently gained. In the following, we call DPLL-UC and DPLL-GUC the variants of DPLL based on the UC anf GUC heuristics respectively.

A computer-independent measure of the computational effort required by DPLL is the number T_{split} of variable splittings. This number varies from system to system (at fixed number N of variables and ratio c), and from run to run of DPLL due to the stochasticity introduced by the heuristic rule. The outcome of numerical experiments for the median number of splits[53] is shown in Fig 2.19.

[52]Denominations will be made clearer in Section ??.

[53]The median is more representative of the typical value of the number of splits than the expectation value, since the latter may be dominated by huge and unlikely samples, see discus-

Not surprisingly, the computational effort is an increasing function of the size N. The dependence upon ratio c is non monotonous with a maximum located around $c \simeq c_S$. As is the case for RW algorithm, numerical data supports the existence of a dynamical threshold, c_E, separating computational efforts scaling linearly and exponentially with N,

$$\mathbf{E}\big[T_{split}\big] = \begin{cases} N\, t_{split} + o(N) & \text{if } c < c_E \\ \exp(N\,\tau_{split} + o(N)) & \text{if } c > c_E \end{cases} , \tag{2.94}$$

where t_{split} and τ_{split} are functions of the ratio c. The value of the dynamical threshold can be derived from theoretical calculations shown in Section ... and is equal to $c_E = \frac{2}{3}$ and $c_E \simeq 0.7507...$ for UC and GUC heuristics respectively. Three dynamical regimes are therefored identified (Connamacher 2004):

- Linear & satisfiable phase ($c < c_E$): systems with small ratios are solved with essentially no backtracking. A solution is found after $\theta(N)$ splits.
- Exponential& satisfiable phase ($c_E < c < c_S$): systems with ratios slightly below threshold have solutions, but DPLL generally requires an exponential number of splits to find one of them. An explanation for this drastic breakdown of performances will be given in Chapter ...
- Exponential & unsatisfiable phase ($c > c_S$): finally, finding a proof of unsatisfiability typically requires an exponentially large number of splits. Note that, as c gets higher and higher, each split affects more and more equations (of the order of c), and contradictions are detected earlier and earlier. Rigorous calculations show that $\tau_{split} = \theta(1/c)$, and the computational effort decreases with increasing c (Fig. 2.19). The median number of splits is considerably smaller for DPLL-GUC than for DPLL-UC, a result expected from the advantages of GUC against UC discussed above.

As a conclusion, we have seen that the polynomial/exponential transition first encoutered in the analysis of RW also takes place with DPLL. Not surprisingly, the location of the dynamical threshold is higher for the backtracking algorithm than for the simpler random walk procedure.

An interesting question is whether the dynamical transition is related to the onset of clustering taking place at $c_D \simeq 0.818$ as could be thought. Indeed, the energy landscape is somewhat more complex in the presence than if the absence of clustering, a fact that could hinder the performances of search algorithms. The relationship between the structure of the energy landscape and dynamical properties will be investigated in Chapters ...

2.6 Conclusion

In this Chapter, we have investigated a computational problem, namely, linear systems over Boolean variables with a simple distribution of random equations. Unexpectedly rich features (satisfiable vs. unsatisfiable transition, clustering,

sion of Section 2.3.3 for a similar phenomenon regarding the number of solutions of a random system.