Algorithmic and Complexity Issues Concerning Phase-Transition Phenomena in Combinatorial Problems*

Paul E. Dunne, Alan Gibbons, Michele Zito

Department of Computer Science
University of Liverpool
Liverpool L69 7ZF
United Kingdom
{ped,amg,michele}@csc.liv.ac.uk

Abstract. The study of phase-transition phenomena from a mathematical perspective can be traced back at least as far as work of Laplace in the 18th Century. While much of the initial motivation arose from analyses of physical models used in statistical mechanics, there has been extensive work developing from the late 1950s into such phenomena within the models used in Percolation Theory. Even more recently, the study of such phenomena has been shown to raise important questions in a computational context, e.g. in the average-case efficiency of algorithms for hard combinatorial problems. In this paper we present an introduction to the idea of phase-transition phenomena reviewing both the classical mathematical background and discussing how these ideas relate to the context of combinatorial algorithms. We highlight, in particular, some open issues whose solution may provide useful insights into the development of more effective methods for dealing with hard combinatorial search problems.

1 Introduction

A *phase-transition* may, in very informal terms, be described as that behaviour whereby a 'small' variation in some parameter of a system under observation occasions a 'significant' change in some property of the system. Undoubtedly the most familiar exemplar of such mechanisms is the change in physical states of matter as the temperature of a substance is increased, e.g. ice (solid) melting into water (liquid) then boiling to become steam (gas). Mathematical models of such effects have a long history, the first being proposed by Laplace at the end of the 18th Century. These models, and others arising in the domain of statistical mechanics such as Ising [26] and Potts [37], make use of so-called '*effective field*' assumptions, i.e. informally, the assumption when effects on a system are modelled by the interactions between a number of distinct 'sites',

* The work reported in this paper was carried out with support from EPSRC grant GR/L/77089.

that such interactions are independent (in some sense) of the distances separating pairs of sites.

The mathematical models of physical systems, outlined above, may seem rather remote, and thereby of little relevance, to those questions concerning combinatorial problems with which this paper is concerned. In fact, this appearance is justified only so far as the superficial context of the specific direct applications, but not, as we intend to make clear, as regards the generality of ideas underpinning such models.

Our aim in this paper is to present a review of the classical analytic contexts in which the mathematical study of phase-transition phenomena have been developed: specifically, Percolation Theory and Combinatorial Graph Theory. We then outline how concepts of phase-transition were observed in experimental studies within the field of Artificial Intelligence concerning the behaviour of search mechanisms, and comment on the related ideas developed independently concerning efficient average-case algorithms for 'hard' combinatorial problems. More extensive treatments of combinatorial analyses and algorithmic issues are given in Sections 2 and 3 respectively, with a review of the principal experimental studies also presented in Section 3. Some open questions are discussed in the final section.

1.1 Analytic Study: Percolation and Combinatorial Graph Theory

In one sense the basic model assumed in the study of Percolation could be viewed as a very general embodiment of the effective field assumption referred to earlier. The Percolation Model is simply described together with the associated Percolation Problem.

Let $d \in \mathbb{N}$ with $d \geq 2$. $L_d = (\mathbb{Z}_d, E_d)$ denotes the infinite d-dimensional hypercubic lattice with $\mathbb{Z}_d$ being the vertices (or *sites*) and E_d the edges. For a probability $p \in [0,1]$ edges, $e \in E_d$, are *open* with probability p and *closed* with probability $1-p$, all choices being made independently.

The *Percolation Model* was introduced in 1957 by Broadbent and Hammersely [7] as a means of studying flow problems in porous media. One of the principal problems addressed within this model concerns what can be said regarding the value of p at which an infinite cluster containing the origin exists with probability bounded away from 0, i.e. an infinite length path of open edges containing the vertex $\mathbf{0}$. More formally suppose $\theta_d(p)$ denotes the probability that L_d has such a cluster when edges are open with probability p. It can be shown that for all $d \geq 2$ there is a *critical probability*, p_c (depending on d), such that $\theta_d(p) = 0$ $(p < p_c)$ and $\theta_d(p) > 0$ $(p > p_c)$. Determination of exact values of p_c for all but a few values of d remains an open issue in this area.

We shall return to consider this problem briefly in Section 2, however, the interested reader is referred to Grimmett [22] for a thorough development of this and other issues in Percolation Theory. For our purposes it is useful to note the important features of this model from the perspective of the general review being presented. Taking our informal definition as a starting point, we see that the 'system' is the infinite hypercubic lattice, L_d; the 'parameter' being modified that of the probability, p, for an edge to be open; and the 'property' being observed is the presence (or otherwise) of an infinite open cluster. The percolation model formalises the important idea of a *critical threshold probability*, p_c, determining whether the observed system has the property of interest, cf. the analogy with 'melting' and 'boiling' points in the physical example used.

Percolation Theory defines one important domain for mathematical analyses of phase-transition phenomena. Another field - Evolutionary Graph Theory - exhibits some similarities (as well as some notable differences) to this, and was pioneered in the work of Erdös and Rényi [15]. Consider a random n-vertex (undirected) graph $G(V,E)$ in which edges are present with probability $p(n)$ and absent with probability $1-p(n)$, again all choices being made independently. If one considers a property, Π, of n-vertex graphs (formally, Π may be considered as a specific subset of all n-vertex graphs, over all values of n), then the probability $\mu_p(\Pi,n)$ that a random n-vertex graph G_p has property Π is well-defined.

Connectivity is one fundamental graph property and was investigated in the random graph model by Erdös and Rényi who not only proved the existence of a critical threshold probability for this property but also determined its exact value. Thus for G_p a random graph using the model defined,

Theorem 1. *Let Π be the property of being a connected graph and $p_c(n) = \log_e n/n$.*

$$\begin{array}{l} \lim_{n\to\infty} \mu_p(\Pi,n) = 0 \;\; if \;\; \lim_{n\to\infty} \; p(n)/p_c(n) < 1 \\ \lim_{n\to\infty} \mu_p(\Pi,n) = 1 \;\; if \;\; \lim_{n\to\infty} \; p(n)/p_c(n) > 1 \end{array} \tag{1}$$

Comparing this with the informal view given earlier, the 'system' is now an undirected n-vertex graph; the parameter being varied is the probability that an edge is present; and the property being observed is that of the random graph being connected. A further comparison of this theorem with that of the critical percolation probability model reveals some points of interest. In both cases one has the existence of a given property in a random structure determined in terms of some critical parameter. The manner in which these appear, however, exhibits some differences. The percolation problem is set in an infinite domain (L_d) so that, d being fixed, the notion of examining a behaviour in the limit, i.e. with respect to the 'size' of the system increasing, is redundant. [1] In addition the likelihood of an infinite cluster when edges are open with probability p is set by the size of p relative to p_c. Neither of these are true of the graph-theoretic example. A consideration of limiting probabilities is (self-evidently) necessary: there being a finite number of n-vertex graphs, each of which, if $p \in (0,1)$ has a positive (albeit, perhaps extremely small) probability of being generated. Thus,

[1] It should be noted that in the Percolation model, p_c is such that $p > p_c$ ensures a greater than 0 probability of an infinite cluster, rather than the technically stronger condition of such a cluster existing with probability 1. The possibility of a jump discontinuity in $\theta_d(p)$ at $p = p_c$ has not been ruled out leaving open the question of whether this stronger condition actually does hold.

in order meaningfully to describe the phenomenon of interest it becomes necessary to concentrate on the *proportion* of n-vertex graphs possessing it, together with the likelihood of these being coincident with a random G_p.

A more subtle difference arises in the context of how $p(n)$ relates to the critical threshold $p_c(n)$ of the theorem statement. One might question why the conditions $p(n) < p_c(n)$ and/or $p(n) > p_c(n)$ do not suffice. Dealing with this raises the important concept of *threshold interval length* which we return to in the next section. In brief, because the combinatorial graph model must deal with limit properties of finite sets, there will be functions, $\chi(n) > 0$ say, such that for $p(n) < p_c(n) - \chi(n)$ the limit probability is 0 and for $p(n) > p_c(n) + \chi(n)$ this limit is 1. For $p_c(n) - \chi(n) \leq p(n) \leq p_c(n) + \chi(n)$, however, it may not be possible to claim either. Thus a new issue raised in the combinatorial graph model concerns the definition and existence of *sharp* threshold, such being dependent (to take one approach) on the precise relationship between $p_c(n)$ and $\chi(n)$. For a treatment of classical random graph theory, the monograph of Bollobás [4] is a good source.

In concluding this sub-section we observe that many of the differences in approach and presentation between Percolation Theory and Evolutionary Graph Theory stem from the contrast between infinite and finite perspectives relevant to each. There are several instances in the former theory, however, where results are derived through analysing the appearance of particular finite structures of size n. For example, one proof that the critical threshold probability, p_c, governing the existence of infinite clusters in L_d, is bounded away from 0 is obtained by counting the *expected* number of open paths containing exactly n edges starting at the origin. Thus, in L_d, this number is trivially bounded above by $2d(2d-1)^{n-1}$, and if edges are open with probability p, the expected number of these present is certainly no greater than $2d(2d-1)^{n-1}p^n$. It follows that if $p < 1/(2d-1)$, then the expected number of paths of length n is less than 1, i.e. $\theta_d(p) = 0$, for $p \in [0, 1/(2d-1))$. It is plausible that ideas from Percolation models may be usefully specialised to give some insight into other domains in which a finite perspective is appropriate, the most immediate of which, in the context of this article, are those involving algorithmic performance, the aspect of phase-transition phenomena which we now examine.

1.2 Experimental Study and Algorithmic Issues

Research concerning phase-transition phenomena from a computational perspective has, in the main, focussed on algorithmic issues in 'hard' combinatorial search problems, although von Neumann [42] in 1956 had derived fundamental results concerning threshold effects arising in the design of reliable hardware systems containing unreliable components.

At first sight it is unclear as to how the phase-transition effects discussed above might appear in an algorithmic context. In this sub-section we outline the nature of the experimental evidence for such, originating from Artificial Intelligence research, and the, rather different, evidence accruing from a select number of algorithmic results.

It is useful to focus on some canonical problems in order to refer to these subsequently.

Definition 1. 3-COL *is the property comprising all* n*-vertex undirected graphs* $G(V, E)$ *for which one of three colours can be assigned to each vertex of* G *in such a way that no two vertices joined by an edge are allocated the same colour.*

Let X_n *denote a set of* n *Boolean variables. A* CNF *formula,* F*, over* X_n*, has the form* $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$*, (*$\wedge$ *denoting logical conjunction) where each* clause C_i *is a non-trivial disjunction of* literals *from* X_n*. A literal being either a variable* x *or its negation, non-trivial indicating that both forms do not occur in the same clause.* SAT *is the property comprising all* CNF *formulae,* F*, for which there is an instantiation of the variables* X_n *under which* F *takes the value* **true**. 3-SAT *is the subset of* SAT *comprising* CNF *formulae in which there are exactly* 3 *literals in each clause.*

A substantial proportion of work in Artificial Intelligence has been directed towards the performance of methods for the *Satisfiability Problem*, usually although not invariably, in its restriction to instances in 3-CNF (3-SAT). By considering randomly generated n-variable, cn-clause instances with the qualifying constant c varying, it has been observed [8,11,32] that as c increases the probability of satisfiable instances being generated declines sharply over a very small range and that this range coincides with a peak in the observed average run-time of the search methods used to find satisfying assignments. Although most such studies employ variations on the Davis-Putnam method [12], it appears that the latter behaviour persists irrespective of the specific satisfiability checker employed.

The sharp drop in the likelihood of a satisfiable instance being generated (observed to occur when c is around 4.24) is, arguably, unsurprising: even very basic analytic arguments suffice to show that this occurs at a value of c no greater than 5.19 [9]. The coincidence of this with a peak in average search time (together with the observed 'lower' average times in instances with significantly fewer/more clauses, i.e. almost always satisfiable respectively unsatisfiable cases) seems to be less clear cut. While some tentative explanations have been put forward, e.g. 'constrainedness' [20], there is little in the way of strong analytic support for these. Even such basic questions as the precise quantifiable change, e.g. is this simply in the degree of some $f(n)$ with an actual average run-time of $2^{f(n)}$, or whether such changes are always present provided a critical threshold probability for the existence of solutions is present, remain unanswered. Some directions for tackling questions such as these have been outlined in Dunne, Gibbons, and Zito [13].

Although specific examples are few, rather stronger support for the presence of extreme changes in expected run-time has emerged from studies of some NP-hard search problems. Thus, Angluin and Valiant [3] describe $O(n \log^2 n)$ average time methods for finding Hamiltonian cycles in graphs whose edge density (this can be viewed as related to p in the G_p model of random graphs introduced above) is large enough to make it almost certain that such cycles exist. Similarly, using a particular model of random SAT instances (the 'constant density' model introduced in Goldberg [21]) polynomial on average randomised methods for finding satisfying assignments have been discovered, the most general of these being that of Wu and Tang [43].

Other discussions of phase-transition effects in the context of search may be found in [25,28,35].

2 Analytic Methods for Studying Critical Probabilities

It has already been noted that the Percolation model considers random structures of open edges in the d-dimensional infinite hypercubic lattice $L_d = (\mathbb{Z}_d, E_d)$, so that the probability space defined from all settings of edges as open or closed (a single such instantiation being referred to as a *configuration*) forms a product space. It is useful, in describing a general setting for finite combinatorial problems, to introduce the following approach.

X_n is the set of binary strings of length n. For a probability, p, (that may depend on n) $X_{n,p}$ is the probability space in which a random n-bit X_p is chosen by fixing each bit to 1 with probability p, and to 0 with probability $q = 1-p$; all choices being made independently. Thus, using $||X||$ to denote the number of 1s in X, the probability that the random string X_p equals X is $p^{||X||}(1-p)^{n-||X||}$. A *property*, Π of binary strings is just a subset of all finite length strings. Letting I_Π denote the indicator function for Π, i.e. the function such that $I_\Pi(x) = 1$ if $x \in \Pi$, and 0 otherwise, allows the probability that $X_p \in \Pi$ to be expressed as

$$\mu_p(\Pi, n) \; = \; \sum_{X \in X_n} p^{||X||} \; (1-p)^{n-||X||} \; I_\Pi(X) \tag{2}$$

In addition it is useful, at this point, to introduce a further formulation to allow consideration of different classes of computational problem.

For specific problems an input binary string is interpreted as denoting some element of a particular class of combinatorial structures, e.g. a string of $n(n-1)/2$ bits encodes the edges of an n-vertex undirected graph; a string of $8\binom{n}{3}$ bits the clauses present in an n-variable 3-CNF. In such interpretations the context in which a combinatorial problem is set can be viewed as comprising,

a) A set of problem *instances*, I.
b) A set of candidate solutions or *witnesses*, W.
c) A binary relation $R \subseteq I \times W$.

For example, with graph connectivity: I is the set of n-vertex graphs; W could be chosen as the set of n-vertex trees; and $(G, T) \in R$ if T is a spanning tree of G.

In such settings there are many different classes of computational problem that might be defined. In the present context the *decision version* is most relevant. For a combinatorial problem (I, W, R), the decision problem, D_R is: given an instance $x \in I$, decide if there *exists* any $w \in W$ for which $(x, w) \in R$. We notice that this is easily related to our earlier concept of *property*, by defining the set Π_D of binary strings

$$\Pi_D \;=\; \{\, x \in I \;:\; \exists\, w \in W \textit{ such that } (x,w) \in R\} \tag{3}$$

Returning to the measure $\mu_p(\Pi, n)$ in the context of undirected graph properties, we have already referred to Erdös and Rényi's discovery of a critical threshold for connectivity. A number of other graph properties have subsequently been examined: Koršunov [31] shows that the connectivity threshold is also a critical threshold for the existence of a Hamiltonian cycle; Dunne and Zito [14], Achlioptas and Molloy [2] narrow the range in which the critical threshold for 3-colourability lies: Achlioptas and Friedgut [1] establishing that a sharp threshold does exist for this property.

The 3-Colourability results mentioned, highlight a particular aspect evident in analytic work on phase-transition thresholds for combinatorial properties: there is an *existence* result. as well as lower and upper bounds on its exact location. Leaving aside existence results for the moment, in common with similar results for other properties, the lower bound is obtained via analysis of the expected outcome of a specific algorithm, i.e. it follows by proving that the algorithm almost certainly finds a colouring in sufficiently sparse graphs ($<$ $1.98n$ edges for [2]). In contrast the upper bound ($> 2.61n$ edges in [14]) is non-algorithmic and is, in essence, obtained through a 'first moment' argument: obtain an upper estimate on the expected number of 'legitimate' colourings for a random graph of cn edges; then choose c to be sufficiently large to let this expected value go to 0.

While analytic determination of the exact value of critical thresholds poses several problems, establishing the *existence* of these has recently been demonstrated for a large class of combinatorial decision problems. Suppose one focuses on *monotone* properties, i.e. those such that $x \in \Pi$ implies that $x^{(i)} \in \Pi$ for all $1 \leq i \leq n$, where $x^{(i)}$ denotes the binary string x with its ith bit set to 1: properties such as connectivity, hamiltonicity, *un*satisfiability, and *non-k*-colourability are all monotone. These graph properties are, furthermore, *symmetric*, i.e. invariant under permutations of the vertex labelling hence isomorphic graphs are both in Π or neither is. It is known that monotonicity alone suffices for the presence of a 'coarse' threshold [5]. Friedgut and Kalai [17] present analyses showing that monotone symmetric properties induce sharp thresholds. Before outlining this work, it is necessary to distinguish what constitutes a *sharp* critical threshold. The key idea in capturing this concept is the *rate* at which $\mu_p(\Pi, n)$ increases as a function of p, once p is sufficiently large to bound μ_p away from 0. Suppose the $\epsilon(n) \in (0, 1/2]$, that $p_0(n)$ denotes the value at which $\mu_{p_0}(\Pi, n) = \epsilon(n)$, and $p_1(n)$ the value for which $\mu_{p_1}(\Pi, n) = 1 - \epsilon(n)$. Then the sharpness of a threshold property depends upon the size of the gap between p_0 and p_1 *relative to* the probability, p_c such that $\mu_{p_c}(\Pi, n) = 0.5$. Thus, the smaller the value $\delta(n) = p_1(n) - p_0(n)$ the more steeply $\mu_p(\Pi, n)$ increases from values 'close to' 0 to values 'close to' 1. Friedgut [18] defines sharp threshold behaviour for a property Π as that for which $\lim_{n\to\infty}\ \delta(n)/p_c(n) = 0$ and 'coarse' thresholds as those for which $\lim_{n\to\infty} \ \delta(n)/p_c(n) > 0$. With such an approach it is clear that the behaviour of the probability $\mu_p(\Pi, n)$, or more accurately its derivative with respect to p, is crucial in determining the exact category of threshold behaviour pertaining to Π. Central to the analysis of this derivative is the following fundamental result of Russo and Margulis [34,38] which also has an important significance in the Percolation Model.

Theorem 2. *For any monotone property, Π,*

$$\frac{d\mu_p(\Pi, n)}{dp} \;=\; E_p[|\{i : x \notin \Pi \text{ and } x^{(i)} \in \Pi\}|] \tag{4}$$

$$= \sum_{x\in X_n} \sum_{i=1}^{n} (I_\Pi(x^{(i)}) - I_\Pi(x))p^{||x||}(1-p)^{n-||x||} \tag{5}$$

The proof of this theorem is straightforward, given the definition of $\mu_p(\Pi, n)$. The importance of *symmetric* properties lies in the contributions of the form $\sum_{x\in X_N}(I_\Pi(x^{(i)}) - I_\Pi(x))p^{||x||}(1-p)^{n-||x||}$ which in graph-theoretic properties with $N = \binom{n}{2}$ the number of edges, is that the Russo-Margulis theorem may be restated as

$$\frac{d\mu_p(\Pi, N)}{dp} \;=\; N\mu_p(\Pi^{\{1,2\}}, N) \tag{6}$$

where $\Pi^{\{1,2\}} \subset \Pi$ is the set of graphs in Π in which the edge $\{1,2\}$ is present but such that, removing this edge yields a graph not belonging to Π.

In order to exploit this relationship one further idea is needed. Clearly, (6) provides a means for determining the rate of increase for $\mu_p(\Pi, n)$. The difficulty in applying this concerns finding a lower estimate for $\mu_p(\Pi^{\{1,2\}})$ (in the symmetric graph property context). Notice that this can be interpreted as the probability that the presence of a single *named* edge influences whether a graph does or does not belong to Π. Using this interpretation a bound on $\mu_p(\Pi^{\{1,2\}})$ follows easily from the following powerful theorem proved in [6].

Theorem 3. *For* any *property, Π, if p is such that $\mu_p(\Pi, n) = t$ (with $t \leq 0.5$) then there is a bit position, k, for which*

$$\mu_p(\Pi^{(k)}, n) \;\geq\; \frac{ct \log n}{n} \tag{7}$$

(c) being a constant)

In combination with the reformulation of Russo-Margulis in (6) we can deduce that $d\mu_p(\Pi, n)/dp \geq ct \log n$ for all $t \in (0, 0.5]$. Although sufficient to ensure that once $\mu_p > \epsilon$ (constant) a small increase in p yields $\mu_{p+\zeta} > 1-\epsilon$, a technical complication arises when t is dependent on n. The further developed required in order to consider such cases is presented by Friedgut and Kalai [17]. In a subsequent paper, [18], Friedgut presents a general approach for establishing the presence of sharp thresholds. One consequence of the latter work is to settle an open question arising in the widely studied *Satisfiability Threshold Conjecture*.

Consider the probability space, $F_{n,p}$, defined over 3-CNF formulae on n variables, in which a random formula, F_p is constructed by including each clause (of the $N = 8\binom{n}{3}$ possible) independently with probability p. The property '*F is unsatisfiable*' is monotone, and the problem of locating its exact critical threshold probability has been open since experimental studies suggested its location in the region of $p = 4.24n/N$. While considerable effort has been invested in narrowing the range of possible values through analytic techniques, prior to [18] it had not even been established that a sharp threshold existed. At present the range of values for c in the transition probability cn/N has been reduced to $c > 3.003$ by Frieze and Suen [19], and recently $c < 4.574$, by Zito [44] building on techniques developed earlier in [29,30].

In contrast the existence of critical threshold probabilities for percolation in L_d, is relatively easily established, see e.g [22]. Progress on identifying the pertinent values for arbitrary d has, however, proved to be very difficult. For the square lattice, L_2, the classic paper of Kesten [27] establishes $p_c = 0.5$, but apart from asymptotic estimates, valid for large d, obtained by Hara and Slade [23,24], no exact values have been discovered for higher dimensions.

3 Algorithms, Complexity and Experiment

We recall our setting of combinatorial problems as a binary relation $R \subseteq I \times W$, I being a set of problem instances, and W a set of a candidate solutions for each instance. We have considered one variant, the decision version D_R, whereby for a given instance $x \in I$, all that is asked is if there is any witness, $w \in W$ for which $(x, w) \in R$. Such decision problems provide a convenient formalism for complexity-theoretic issues, in particular if we make the assumption that for all $(x, w) \in R$, $|w|$ is bounded by a polynomial function of $|x|$ ($|z|$ denoting the length of the binary string z), we can introduce the *complexity classes*, P and NP: P is the set of relations, R, for which there is a polynomial bound $t(n)$ and decision procedure solving D_R that takes at most $t(|x|)$ steps to return an answer for any $x \in I$; NP is the class of relations, R, for which there is a polynomial bound, $t(n)$, and a decision procedure that given $x \in I$ and $w \in W$, determines if $(x, w) \in R$ in at most $t(|x|)$ steps. Thus, informally, P is the set of relations, for which the question of whether a solution *exists* for a given instance can be solved efficiently; NP is the class of relations for which the question of whether a *specific* solution, w, is valid with respect to a given instance can be answered by an efficient algorithm.

A major open question within computational complexity theory is whether the sets P and NP are identical. Strong circumstantial evidence suggests that there are problems in NP that do not belong to P, i.e. that checking a potential solution is correct is 'easier' than determining if a valid solution exists. The subclass of NP formed by the $NP - complete$ problems, whose study was initiated in the seminal paper of Cook in 1971 [10], presents a set of relations that are the 'least likely' to belong to P. This is by virtue of the fact that if R is NP-complete and $R \in P$ then $P = NP$.

Of course, an 'obvious' algorithm for solving D_R, for a combinatorial problem, R, is merely exhaustively to enumerate all potential solutions in W for a given instance x (recall that we assume $|w|$ is bounded by a polynomial in $|x|$ for any $(x, w) \in R$). Unfortunately, except in rare and (normally) trivial cases, this process is unlikely to be polynomially bounded. Consider settings of 3-SAT and 3-COL in this respect.

For 3-SAT, I is the set of formulae in 3-CNF over n variables. These can be encoded as a binary string of length $8\binom{n}{3}$; W is the set of all Boolean assignments to n variables, any one of which can be encoded by an n-bit string.

For 3-COL, I is the set of n-vertex undirected graphs. These can be encoded as binary strings of length $\binom{n}{2}$ and W the set of all possible assignments of one of three colours to n vertices. A single colouring can be encoded in $n \log_2 3$ bits.

For F an instance of 3-SAT, $|F| = 8\binom{n}{3}$, but the number of potential witnesses is 2^n; for G an instance of 3-COL, $|G| = \binom{n}{2}$ but the number of potential witnesses is 3^n.

One advantage of the obvious enumeration process is that as well as *deciding* whether a given instance has a particular property, in the event of a positive answer, it also identifies a witness to this effect, i.e. an appropriate assignment in 3-SAT, a proper colouring in 3-COL. In practice, rather than a simple decision an actual witness is required, thus in addition to D_R the decision version of a problem, one may consider the *search* version, S_R, which given an instance $x \in I$ asks for a $w \in W$ such that $(x, w) \in R$, if such a witness exists.

A major research area within both Artificial Intelligence and Algorithmics concerns the development of 'good' strategies for solving the search version of hard combinatorial problems, 'hard' being in the sense that the decision version if NP-complete. It was in this context that experimental studies of average run-time were developed. The pioneering study of [8] identified noticeable variations in average run-time over random instances of problems such as 3-SAT and 3-COL. Explanatory models for this behaviour have been sought in analogies of phase-transition effects taken from statistical mechanics, e.g. [36] so that the search space of candidate solutions is viewed as a highly disordered system which can be limited relative to the properties of specific instances. Alternative approaches utilise *quasi* information-theoretic ideas, [20], seeking to account for phase-transition effects and thresholds in terms of parameters defined from distributions of random instances.

Another topic of interest concerns the fact that in experimental studies of, say 3-SAT, random instances entailing significant search cost can appear at locations which lies in a nominally 'easy' region, e.g. with $\ll 4.24n$ clauses. An example of work investigating this aspect is [40]. In complexity-theoretic terms, in one sense, such behaviour is to be expected: [13] has shown that 3-SAT remains NP-complete even when restricted to instances containing exactly m clauses for all $n/3 \leq m \leq (7 - \epsilon)\binom{n}{3}$ (and all constant $\epsilon > 0$). Thus, *in the worst-case* one would expect such hard instances to arise. The experimental studies of Vlasie

[41] suggest, using arguments from Kolmogorov Complexity, that such instances are extremely rare. [2]

We have mentioned a few cases where supporting analytic evidence for a phase-transition in run-time derives from provably efficient on average methods. We conclude this review of algorithmic issues by considering one such approach: the randomised SAT algorithm of Wu and Tang [43]. The basis for this is the following: random instances of SAT are generated by fixing the number of variables (n); the number of clauses (m), and then for each clause including the literal y with probability p. The decisions are made independently over all clauses and $2n$ possible literals. This defines the so-called *constant density* model of random formulae proposed by Goldberg [21]. In earlier work on this model, Franco [16] has shown that if $p \geq 1/m^{1/2}$ then a random formula is almost certainly satisfiable by almost all instantiations of its n variables. In the same paper, it is also shown that for $p \leq \alpha \log_e m/2m$ (for an appropriate constant choice of α), a random formula almost surely contains an empty clause (one with no literals) and hence is unsatisfiable. Wu and Tang present an analysis of the remaining interval. Their algorithm takes a SAT instance and a failure probability ϵ as input and proceeds by first checking for empty clauses, and then eliminating trivial clauses from F. If t is the number of (non-trivial) clauses remaining, the algorithm then proceeds to generate (at most) k random instantiations of the variables, testing each of these in turn. The key point in the average run-time is in the choice of k which is fixed at $\log_e \epsilon / \log_e(1-(1-1/t)^t)$. For p in the relevant range it is shown that the probability of failing to find a satisfying instantiation by this process does not exceed ϵ.

Two points worth noting about such approaches are, firstly, unless an empty clause is present in the initial instance, the method does not identify *unsatisfiable* formulae (there is no distinguishing between the case that the random generation of assignments has been 'unlucky' and the instance being genuinely unsatisfiable). Secondly, there is a superficial similarity between this method and mechanisms such as $GSAT$ [39], in which a number of instantiations are generated repeatedly. $GSAT$, however, starts with a single random instantiation and manipulates this in a controlled manner for a set number of iterations. It is interesting to observe that $GSAT$ has proved successful in some practical scenarios, even to the extent that reducing other NP-hard search problems to SAT

[2] Vlasie, actually suggests *finite* in number. Suppose instances contain N bits. His argument stems from the fact if A is a 'fast on average' decision process for an NP-complete problem, then assuming P does not equal NP, the worst-case instances for A are those whose description can be compressed into $\ll N$ bits (this follows from Li and Vitanyi's results in [33]). Since A is fast on average, the set of instances on which A exceeds a run-time of T (for large enough T) is relatively small, and, in the sense of Kolmogorov complexity, not 'truly random'. The 'finite' argument is presented by observing that, for any constant k, only a finite number of such cases cannot be compressed by k bits. What is unclear in this setting is that if k is a function of n with $k(n) \to \infty$, then there can be infinitely many instances which exceed the 'fast on average' run-time to a significant degree.

has been mooted as a suitable method of solving these. In the next section, we discuss potential drawbacks to this approach.

4 Selected Open Questions

In this concluding section we present, briefly, a few open questions covering topics mainly drawn from algorithmics issues pertaining to phase-transition phenomena in combinatorial search problems.

4.1 Analytic Support for Experimental Results

With the exception of a select number of efficient on average algorithms whose performance is dependent on density properties of random instances, there is almost no strong analytic foundation supporting the variations in expected run-time that have been observed in experimental studies of search processes on random instances. One possible avenue of attack on this question may be by concentrating on classes of *self-reducible* problems: those whose decision properties may be expressed in terms of deciding 'smaller' instances of the same problem, e.g SAT since $F \in SAT$ if $F^{x_1=0} \in SAT$ or $F^{x_1=1} \in SAT$. One straightforward development of the Russo-Margulis Theorem may provide a starting point here. Recall that $X^{(k)}$ denotes the binary string X with its kth bit set to 1, and similarly define $X_{(k)}$ as X with its kth bit set to 0. Suppose $X_{n,p}$ is the probability space introduced earlier, and that $T : X_{n,p} \to \mathbb{N}$ is a random variable. Then

$$\frac{dE_p(T(X))}{dp} = \sum_{k=1}^{n} E_p(\delta_k(T)) \tag{8}$$

where $\delta_k(T) = T(X^{(k)}) - T(X_{(k)})$. If T is taken as the run-time of some search process on instances X, then this relationship shows that the rate of change of expected run-time (over $X_{n,p}$) can be expressed in terms of the total change in run-time that occurs when single bits of an instance are altered.

4.2 Necessary Conditions for Critical Thresholds

Bollobás and Thomason [5], Friedgut and Kalai [17], and Friedgut [18] indicate that monotonicity is sufficient to ensure a property has (at least) a coarse threshold. Dunne, Gibbons, and Zito [13] demonstrate that it is not a necessary condition (even for a sharp threshold). The example property is, however, 'almost' monotone in the sense there are a vanishingly small fraction of instances, X, for which $X \in \Pi$ but $X^{(k)} \notin \Pi$. An interesting question is to what extent monotonicity can be relaxed (even when restricting to symmetric graph properties) before even coarse thresholds do not appear.

4.3 Is Reducing to SAT a 'Good' Idea?

We noted earlier that the success of systems such as $GSAT$ [39], have led to some researchers mooting the possiblility of transforming instances of NP-hard search problems into instances of SAT as a mechanism for solving these. While it may well be the case that such an approach occasionally proves fruitful, there are reasons why one may wish to be cautious about its general utility. If one considers reductions from, say, graph-theoretic problems to 3-SAT, these will map an n-vertex, m-edge graph onto an $f(n,m)$-variable, $g(n,m)$-clause instance of 3-SAT. This mapping is, however, rarely bijective, i.e. an arbitrary $f(n,m)$-variable, $g(n,m)$-clause instance of 3-SAT may not be the image of any n-vertex, m-edge graph under the transformation, or could be the image of several (non-isomorphic) such graphs. Thus, one reason why one might doubt the efficacy of transforming into SAT in order to make use of a given satisfiability checker, is that the basis for using the latter might only derive from its experimental performance on *randomly* generated instances of SAT: this might provide no indication of the method's efficiency on specific subsets of a given size, in particular the subset corresponding to those instances mapped onto by a reduction. In fact, it could be the case that far from assisting with solving a particular search problem, depending on the nature of the initial graph instances and the satisfiability checker employed, it would be 'better' to employ some heuristic method on the original graph instance *directly* rather than reducing to SAT, i.e. can one provide experimental support for the hypothesis that some 'natural' reductions may 'on average' transform 'easily' solvable instances of one search problem into typically hard (for the SAT method being used) instances of SAT?

Experimental studies in progress of the present authors are concerned with testing this hypothesis that it is possible to construct random instances of 3-SAT which have high search cost for widely-used SAT checking systems such as optimised versions of [12] or [39], even though the instances originate from 'easy' examples. This uses a standard transformation from 3-COL to 3-SAT. Appropriate settings of parameters take a graph in the 'easy' positive region for 3-COL into the 'easy' but negative region for SAT. Since, as 3-SAT instances, these are very highly structured, two facts are evident: first they are generated at random with vanishingly small probability; and, secondly, the instances have significantly lower Kolmogorov complexity: at worst $\binom{n}{2}$ bits compared to the $O(n^3)$ for a 'random' 3-SAT instance. In principle, via Li and Vitanyi [33] such instances should be among those which tax efficient on average processes.

5 Conclusion

Phase-transition phenomena have been extensively studied in a number of well established mathematical domains. In this paper we have given a brief overview of some of these. The ideas underpinning phase-transition effects raise many interesting issues of both combinatorial and algorithmic importance, a few of these have been outlined above.

References

1. Achlioptas, D., Friedgut, E.: A threshold for k-colorability. To appear *Random Structures and Algorithms*, 14, 1999
2. Achlioptas, D., Molloy, M.: The analysis of a list colouring algorithm on a random graph, *Proc. 38th IEEE Symp. on FOCS*, (1997), 204-212
3. Angluin, D., Valiant, L.G.: Fast probabilistic algorithms for Hamiltonian Circuits and Matchings. *Jnl. of Comp. and System Sci.*, 18 (1979), 82-93
4. Bollobás, B.: Random Graphs, Academic Press, London, 1985
5. Bollobás, B., Thomason, A.: Threshold Functions, *Combinatorica*, 7, (1986), 35-38
6. Bourgain, J., Kahn, J., Kalai, G., Katznelson, Y., Linial, N.: The influence of variables in product spaces. *Israel Jnl. Math.*, 77, (1992), 55-64
7. Broadbent, S.R., Hammersley, J.M.: Percolation processes I. Crystals and mazes. *Proc. Camb. Philos. Soc.*, 53, (1957), 629-641
8. Cheeseman, P., Kanefsky, B., Taylor, W.: Where the really hard problems are. *Proc. of the 12th IJCAI*, (1991), 331-337
9. Chvátal, V., Szemerédi, E.: Many hard examples for resolution. *Jnl. of the ACM*, 35, (1988), 759-768
10. Cook, S.: The complexity of theorem proving procedures, *Proc. 3rd ACM Symp. on Theory of Computing*, ACM Press, New York (1971), 151-158
11. Crawford, J., Auton, L.D.: Experimental results on the cross-over point in satisfiability problems. *Proc. AAAI-93*, Washington DC, (1993), 21-27
12. Davis, M., Putnam, H.: A computing procedure for quantification theory. *Jnl. of the ACM*, 7, (1960), 201-215
13. Dunne, P.E, Gibbons, A.M., Zito, M.: Complexity-Theoretic Models of Phase-Transitions in Search Problems. to appear: *Theoretical Computer Science*
14. Dunne, P.E., Zito, M.: An improved upper bound on the non-3-colourability threshold. *Inf. Proc. Lett.*, 65, (1998), 17-23
15. Erdös, P., Rényi, A.: On the evolution of random graphs. *Mat Kutato Int. Kozl*, 5, (1960), 17-60
16. Franco, J.: On the probabilistic performance of algorithms for the satisfiability problem. *Inf. Proc. Lett.*, 23, (1986), 103-106
17. Friedgut, E., Kalai, G.: Every monotone graph property has a sharp threshold. *Proc. Amer. Mathl. Soc.*, 124, (1996), 2993-3002
18. Friedgut, E.: Sharp thresholds of graph properties, and the k-sat problem. (to appear)
19. Frieze, A.M., Suen, S.: Analysis of two simple heuristics on a random instance of k-*SAT*. *Jnl. of Algorithms*, 20, (1996), 312-355
20. Gent, I., MacIntyre, E., Prosser, P., Walsh, T.: The constrainedness of search. *Proc. 13th National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, AAAI Press/ MIT Press, (1996), 246-252,
21. Goldberg, A.: Average case complexity of the satisfiability problem. *Proc. Fourth Workshop on Automated Deduction*, (1979), 1-6
22. Grimmett, G.: Percolation. Springer, New York (1989)
23. Hara, T., Slade, G.: Mean-field critical behaviour for percolation in high dimensions. *Comm. in Math. Physics*, 128, (1990), 333-391
24. Hara, T., Slade, G.: Mean-field behaviour and the lace expansion. *Probability and Phase Transition*, (Ed: G.R. Gritmmett), Kluwer, Dordrecht, (1994), 87-122

25. Hogg, T., Huberman, B.A., Williams, C.: Phase transitions and the search problem. *Artificial Intelligence*, 81, (1996), 1-15
26. Ising, E.: Beitrag zur theorie des ferromagnetismus. *Zeitschrift fur Physik*, 31, (1925), 253-258
27. Kesten, H.: The critical probability of bond percolation on the square lattice equals $\frac{1}{2}$. *Comm. in Math. Physics*, 74 (1980), 41-59
28. Kirkpatrick, S., Selman, B.: Critical behavior in the satisfiability of random Boolean expressions. *Science*, 264, (1994), 1297-1301
29. Kirousis, L.M., Kranakis, E., Krizanc, D.: A better upper bound for the unsatisfiability threshold. *Proc. DIMACS Workshop on the Satisfiability Problem: Theorey and Applications*, (March 1996)
30. Kirousis, L.M., Kranakis, E., Krizanc, D.: Approximating the unsatisfiability threshold of random formulas. *Proc. 4th Annual European Symposium on Algorithms, ESA '96*, (September 1996)
31. Koršunov, A.D.: Solution of a problem of Erdös and Rényi on Hamiltonian cycles in nonoriented graphs. *Soviet Math. Doklady*, 17, (1976), 760-764
32. Larrabee, T., Tsuji, Y.: Evidence of a satisfiability threshold for random 3CNF formulas. Techn. Report UCSC-CRL-92-42, Univ. of Calif., Santa Cruz, (1992)
33. Li, M., Vitanyi, P.: Average case complexity under the universal distribution equals worst-case complexity. *Inf. Proc. Lett.*, 42, (1992), 145-149
34. Margulis, G.: Probabilistic characteristics of graphs with large connectivity. *Problems of Inf. Transmission*, 10, (1974), 174-179
35. Mitchell, D.G., Selman, B., Levesque, H.J.: Hard and easy distributions of SAT problems. *Proc. AAAI-92*, AAAI Press/ MIT Press, (1992), 459-465
36. Monasson, R., Zecchina, R., Kirkpatrick, S., Selman, B., Troyanksy, L.: Phase transition and search cost in the $(2+p)$-SAT problem. *4th Workshop on Physics and Computation*, Boston Univ., (1996), 229-232
37. Potts, R.D.: Some generalized order-disorder transformations. *Proc. Camb. Philos. Soc.*, 48, (1952), 106-109
38. Russo, L.: On the critical percolation probabilities. *Z. Wahrsch. Verw. Gebiete*, 43, (1978), 39-48
39. Selman, B., Levesque, H., Mitchell, D.: A new method for solving hard satisfiability problems. *Proc. 10th National Conf. on Art. Intellig.*, (1992), 440-446
40. Smith, B.M., Grant, S.A.: Sparse constraint graphs and exceptionally hard problems. *Proc. IJCAI95*, (1995), 646-651
41. Vlasie, D.R.: The very particular structure of the very hard instances, *Proc. 13th National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, AAAI Press/ MIT Press, (1996), 266-270
42. von Neumann, J.: Probabilistic organisms and the synthesis of reliable organisms from unreliable components. *Automata Studies* (Ed: C.E. Shannon, J. McCarthy), Princeton Univ. Press, (1956), 43-98
43. Wu, L.C., Tang, C.Y.: Solving the satisfiability problem by using randomized approach. *Inf. Proc. Lett.*, 41, (1992), 187-190
44. Zito, M.: Ph.D. Dissertation, Dept. of Comp. Sci., Univ. of Warwick, 1999