## LETTER TO THE EDITOR

# Percolation and the complexity of games†

Constantin P Bachas‡§ and Wilfried F Wolff∥¶

‡ Stanford Linear Accelerator Center, Stanford University, Stanford, CA 94305, USA
∥ Department of Applied Physics, Stanford University, Stanford, CA 94305, USA

**Abstract.** We establish a connection between a facet of the complexity of games and the problem of percolation on the underlying tree of winning strategies. We show that the complexity is minimal for both regular and totally random trees and identify a class of trees for which it is maximal.

A large number of natural and artificial systems have an exact or approximate hierarchical organisation. A useful tool in analysing them has been the renormalisation group and the concept of self-similarity. It is, however, precisely the absence of self-similarity which characterises the most complex and interesting systems. In an effort to classify hierarchical structures Huberman and Hogg (1986) introduced a notion of complexity which measures the degree of non-isomorphism at all levels of the hierarchy and which was numerically shown to be small not only for regular (uniformly multifurcating) but also for totally random trees. A physical manifestation of such a notion of complexity is the speed of relaxation of hierarchical structures (Bachas and Huberman 1986): regular ones relax fastest and structural noise is irrelevant. This behaviour should be contrasted to the Shannon entropy or detailed (rather than coarse-grained) information content which is minimised for regular, but maximised for random trees. Similar ideas in a different context, that of forecasting the itinerary sequence in a chaotic system, have also been advocated for by Grassberger (1986).

In this letter we will consider yet another manifestation of this notion of complexity, namely in the context of percolation on arbitrary trees and show how it is related to the complexity of games and the time requirement of search algorithms.

Consider a game whose duration is, for simplicity, taken to be exactly $n$ moves. At each move the player is faced with several options, some of which lead to a forced loss assuming an infinitely intelligent opponent. The remaining non-losing strategies form a tree with $n$ levels of hierarchy which we will refer to, by slight abuse of notation, as the winning tree of the game. Regular, i.e. uniformly multifurcating, winning trees correspond to simple games: an example would be a game in which the player is at each step presented with a choice of four colours (red, green, blue and white) and has

to pick either blue or white to avoid losing immediately. This is to be contrasted with complex games such as, for instance, chess with a fixed number of total moves, for which the tree of non-losing strategies is certainly not regular since certain bad moves might, e.g., allow your opponent to checkmate you for the rest of the game, thereby limiting your options considerably. A little thought will convince the reader that more generally hard games are characterised by a sensitive history dependence, which is manifested as a lack of self-similarity of the winning tree, whose subtrees are not identical to each other and to the whole. Put differently, the checkmate positions commonly found in newspapers are not all equally hard.

To make these ideas more precise, consider a monkey which at each step of the game decides at random to pursue every one of the available options with probability $p$. We of course allow the monkey to pursue more than one option at any given time, i.e. to play simultaneously on several boards, since otherwise its chances of winning against an infinitely smart opponent would always be vanishingly small. We will say that the monkey does not lose if, throughout the game, it has pursued at least one of the non-losing strategies. When the number of moves becomes arbitrarily large this can only happen if $p$ is greater than the threshold $p_c$ for percolation from the root to a leaf of the winning tree. The smaller $p_c$, the simpler the game, since the randomly playing monkey may pursue fewer options and still not lose; we may thus define $p_c$ as a measure for the complexity of the game.

An important remark on semantics is in order here: for any particular game, $p_c$ is really a measure of the lack of fault tolerance, rather than of the algorithmic complexity, defined as the effort necessary for programming a machine to play the game successfully. This is because, irrespective of the structure of the winning tree, one particular winning strategy might be easy to follow with a simple set of instructions. With this caveat in mind, we will nevertheless continue to refer to $p_c$ as a complexity since ($a$) it is the only sensible such measure one can assign to the unlabelled tree of winning strategies (i.e. to the ensemble of games with the same topological winning tree) and ($b$) the lack of fault tolerance is, after all, one aspect of the complexity of real games.

To study percolation on general trees we will use the fact that the probability $Q(t)$ that no path percolates from the root to a leaf of a subtree $t$ satisfies the recursion relation

$$Q(t) = (1 - p + pQ(t_1)) \ldots (1 - p + pQ(t_b)) \tag{1}$$

where $b$ is the number of branches emanating from the root of $t$ and $t_1, \ldots, t_b$ label the corresponding subtrees (see figure 1).

We first consider the simplest case, namely percolation on regular or uniform trees, which have the same branching ratio $b$ at each node of every generation. Variations of this problem have already been considered in studying the mean-field theory of percolation (Fisher and Essam 1961, Essam 1980, Stauffer 1985) and more recently in
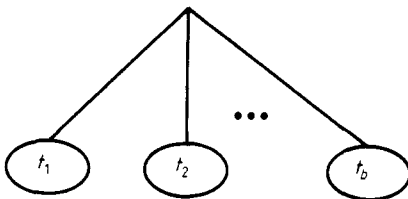


**Figure 1.** A tree $t$ and its subtrees $t_1, \ldots, t_b$ at the first hierarchy level used in the recurrence relation (1).

the context of spreading activation in computer networks (Hogg and Huberman 1986) and also in studying the time requirements of certain heuristic searches (Karp and Pearl 1983, Stone and Sipala 1986). The probability $Q_n$ that in a uniform tree with $n$ generations no path percolates then satisfies the recursion relation

$$Q_n = (1 - p + pQ_{n-1})^b. \tag{2}$$

Equation (2) has the fixed point $Q = 1$, corresponding to no percolation. This fixed point becomes unstable for $p > p_c^{uniform} \equiv 1/b$, signalling the appearance with a finite probability of a percolating cluster from the root to the bottom of the tree.

Clearly, the fatter the tree, i.e. the larger the branching ratio $b$, the smaller the percolation threshold, since fat trees are harder to cut. Although one could argue that the complexity of a game should increase whenever some winning strategies are discarded (e.g. keeping only white as a winning choice in the aforementioned game) this is a rather trivial effect. Therefore we will limit ourselves in the following to games whose trees have a fixed average branching ratio $b$ per generation or a fixed multiplication rate of non-losing strategies per move. This is reminiscent of ultradiffusion where fatter trees relax faster and where one must fix the tree silhouette in order to study the effect of the tree structure on the dynamics (Bachas and Huberman 1986).

We shall now show that $p_c$ is indeed minimised for regular trees as it should be since these trees correspond to the simplest games.

*Theorem 1.* The critical threshold for percolation $p_c$ in a tree with average branching ratio $b$ satisfies

$$p_c \geq p_c^{uniform} = 1/b.$$

*Proof.* Consider an arbitrary tree $t$ with average branching ratio $b$; this means that the total number of nodes at the first hierarchy level is $b$, at the second $b^2$, at the third $b^3$, etc, but the branching rate need not be evenly distributed among all members of a given generation. The probability $Q(t)$ for no percolation on $t$ satisfies

$$Q(t) = (1 - p + pQ(t_1)) \ldots (1 - p + pQ(t_b)) \geq \{1 - p + p[Q(t_1) \ldots Q(t_b)]^{1/b}\}^b \tag{3}$$

where $t_1, \ldots, t_b$ are the (not necessarily identical) subtrees with roots at the first hierarchy level. The inequality follows from the well known fact that the arithmetic mean is always bigger or equal to the geometric mean. At the second step we have

$$Q(t_1) \ldots Q(t_b) \geq \{1 - p + p[Q(\tilde{t}_1) \ldots Q(\tilde{t}_{b^2})]^{1/b^2}\}^{b^2} \tag{4}$$

where $\tilde{t}_1, \ldots, \tilde{t}_{b^2}$ are the $b^2$ subtrees with roots at the second hierarchy level; this follows from the recursion relation (1) and the inequality (3). Combining inequalities (3) and (4) we obtain

$$Q(t) \geq (1 - p + p\{1 - p + p[Q(\tilde{t}_1) \ldots Q(\tilde{t}_{b^2})]^{1/b^2}\}^b)^b.$$

Repeating this procedure $n$ times we finally get

$$Q(t) \geq f^{(n)}(0)$$

where $f^{(n)}$ is the $n$th iterate of the function $f(x) = (1 - p + px)^b$. The right-hand side is the probability for no percolation on a regular tree, which completes the proof.

Next, we consider the case of random trees which are constructed by allowing the branching ratio $x$ at every node to be an independent random variable with distribution $P(x)$ and average value $\langle x \rangle = b$. After deleting bonds with probability $1 - p$ the cluster connected to the root is another random tree with modified probability distribution

$$\tilde{P}(y) = \sum_{x=1}^{\infty} P(x) \binom{x}{y} p^y (1-p)^{x-y}$$

and average branching ratio $\langle y \rangle = pb$. Here $\binom{x}{y}$ is the binomial cofficient. The probability for percolation on the original tree is equal to the probability that the modified random tree survives for an infinite number of generations. From the theory of branching processes (Harris 1963) it is known that the latter probability is finite when the average branching ratio is bigger than one and zero when it is smaller than one, from which we deduce that the critical probability for percolation on the original random tree is given by $\langle y \rangle = p_c b = 1$.

The above argument shows that random trees with average branching ratio $\langle x \rangle = b$ have the same percolation threshold as regular trees with branching ratio $b$ and consequently, in view of theorem 1, correspond to games of minimal complexity. This is in agreement with the measure of complexity introduced by Huberman and Hogg (1986) as well as with the dynamic exponent that characterises ultradiffusion with long-range hopping (Bachas and Huberman 1986). Intuitively, random trees have minimal complexity because they can be made more and more regular by coarse-graining, i.e. by defining a new elementary move as a series of consecutive old moves.

We have thus demonstrated that the winning trees of truly complex games (for which $p_c > 1/b$) form a set of measure zero. We now proceed to identify a class of such trees, describing the winning strategies of maximally complex games (for which $p_c = 1$). These are constructed by letting the left-half nodes of every generation give rise to $2b - 1$ offsprings, while the rest continue as unbranched paths to the bottom, as shown in figure 2, so that the average branching ratio is $b$. To see why the percolation threshold is equal to one for such a tree, note that if $p < 1$ then the leftmost path as seen in figure 2, from the root to the bottom is cut with probability one at a finite hierarchy level $m$ away from the root; the remaining $b^m - 1$ nodes generate subtrees which $(2b - 1)$-furcate for a finite number of steps and then continue as unbranched paths thereafter. Clearly these subtrees cannot percolate if $p < 1$ so that $p_c = 1$ as claimed before.

An intuitive explanation why this game is of maximal complexity is that there is zero tolerance for errors since the player must either follow the strategy corresponding
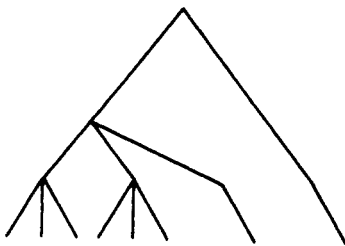


**Figure 2.** An example of a maximally complex tree; the left-half nodes of every generation give rise to three offspring, while the rest continue as unbranched paths to the bottom. The average branching ratio is $b = 2$.

to the leftmost branch or else will eventually be led to a situation where non-losing moves are unique. This same class of trees also leads to slowest relaxation in ultra-diffusion (Bachas and Huberman 1986). It would be interesting to see if there exists a quantitative relationship between the dynamical critical exponent and the percolation threshold $p_c$.

Finally we would like to point out another possible interpretation of our results, namely in the context of heuristic searches. Following Karp and Pearl (1983) one can consider the problem of finding an optimal path on a tree whose edges are assigned the cost 1 with probability $p$ and 0 with probability $1 - p$. A 'uniform cost' algorithm that performs this task is one that, starting from its leftmost node, expands the front of nodes reachable with a given cost from the root of the tree until it finds a leaf at the bottom of the tree. Karp and Pearl show that for regular trees this algorithm runs efficiently (linear time requirement) if the 0-cost edges have a finite probability to percolate and has an exponential time requirement otherwise. Since, as shown before, these 0-cost edges can never percolate on a sufficiently complex tree we would then expect the running requirement for these trees to always be exponential. It would be very interesting to analyse this problem in more detail but this is beyond the scope of this letter.

**References**

Bachas C P and Huberman B A 1986 *Phys. Rev. Lett.* **57** 1965
Essam J W 1980 *Rep. Prog. Phys.* **43** 833
Fisher M E and Essam J W 1961 *J. Math. Phys.* **2** 609
Grassberger P 1986 *Proc. 16th Statistical Physics Conf., Boston*
Harris T 1963 *The Theory of Branching Processes* (Berlin: Springer)
Hogg T H and Huberman B A 1986 *Art. Intell. J.* to be published
Huberman B A and Hogg T H 1986 *Physica* D **22** 376
Karp R M and Pearl J 1983 *Art. Intell. J.* **21** 99
Stauffer D 1985 *Introduction to Percolation Theory* (London: Taylor and Francis)
Stone H S and Sipala P 1986 *IBM J. Res. Dev.* **30** 242