Computability of percolation thresholds

Olle Häggström*

June 6, 2007

Abstract

The critical value for Bernoulli percolation on the $\mathbf{Z}^d$ lattice in any dimension d is shown to be a computable number in the sense of the Church–Turing thesis.

1 Introduction

In 2004, at the 8th Brazilian School of Probability, I gave a lecture series entitled *Percolation theory: the number of infinite clusters*, based mainly on a draft version of Häggström and Jonasson [6]. This was a highly rewarding experience, not only because of the beautiful location but also because of one of the most stimulating audiences I have ever had. During one of the breaks, and later at an open problems session, Andrei Toom asked whether the critical value p_c for Bernoulli percolation on the $\mathbf{Z}^d$ lattice is computable in the sense of the Church–Turing thesis for all d, and described (probably somewhat tongue-in-cheek) the lack of a known answer to this question as a serious shortcoming of the subject of percolation theory. The purpose of this note is to show how an affirmative answer to Toom's question can be deduced relatively easily from some of the percolation technology that had been developed for other purposes in the 1980's and 1990's. To prove the desired computability result from scratch is a different story, and I suspect it would be quite involved.

The rest of this section is devoted to describing the setup and stating the main result. Then, in Section 2, the algorithm that is used to establish the result is described, and finally in Section 3 the algorithm is shown to halt in finite time with the desired output.

Percolation theory (see Grimmett [4] for an introduction) deals with connectivity properties of random media, and the most basic setup, known as Bernoulli percolation, is as follows. Let $G = (V, E)$ be a finite or infinite but locally finite graph with vertex set V and edge set E, fix $p \in [0, 1]$, and remove each edge $e \in E$ independently with probability $1-p$, thus keeping it with probability p, and consider the resulting subgraph of G. It will sometimes be convenient to represent an outcome of this percolation process as an element of $\{0, 1\}^E$, where a 0 denotes the removal of an edge, and a 1 its retention. In the following, we will conform to standard terminology by speaking of retained edges as open, and deleted edges as closed. For $x, y \in V$, we write $x \leftrightarrow y$ for the event that there exists a path of retained edges between x and y.

What we have defined here, and will be concerned with in the following unless otherwise stated, is bond percolation. Alternatively, one may consider site percolation,

*Research supported by the Swedish Research Council and by the Göran Gustafsson Foundation for Research in the Natural Sciences and Medicine.

where it is the vertices rather than the edges that are retained (declared open) or removed (declared closed) independently with retention probability p.

When G is infinite, it is natural to ask for the probability of the existence of at least one infinite connected component of the resulting subgraph. Kolmogorov's 0-1-law implies that this probability $\psi(p)$ is either 0 or 1 for any p, and a simple coupling argument shows that $\psi(p)$ is nondecreasing in p. Combining these two observations yields the existence of a critical value $p_c = p_c(G) \in [0,1]$ such that

$$\psi(p) = \begin{cases} 0 & \text{for } p < p_c \\ 1 & \text{for } p > p_c\,. \end{cases}$$

Much of percolation theory deals specifically with the case where G is the $\mathbf{Z}^d$ lattice, i.e., the graph with vertex set $\mathbf{Z}^d$ and with edge set $E_{\mathbf{Z}^d}$ consisting of edges connecting vertices at Euclidean distance 1 from each other. The case $d = 1$ is fairly trivial with $p_c = 1$, but already the case $d = 2$ turns out to be extremely intricate, with Kesten's [7] 1980 result that $p_c(\mathbf{Z}^2) = \frac{1}{2}$ standing out as one of the classical achievements in the subject. For higher dimensions $d \geq 3$ no exact expressions for $p_c(\mathbf{Z}^d)$ are known. This makes it natural to ask for upper and lower bounds for p_c as well as properties such as the computability considered here.

Beginning in 1936, a number of formal models of computing – the most well-known ones being *Turing machines* and λ-*calculus* – were introduced that were later shown to yield equivalent notions of computability. The *Church–Turing thesis* states that the set of functions $f : \mathbf{Z} \to \mathbf{Z}$ computable according to one (hence all) of these models exhausts the set of functions that would naturally be regarded as computable. Due to the vagueness of the statement, the thesis cannot be formally proven, but it is held in high esteem among computer sceintists, and certainly any function that allows computation by a program written in (pseudo-)Pascal or other standard programming languages can also be computed on a Turing machine. See, e.g., Knuth [8] or Blass and Gurevich [3] for more on this topic. Actually programming a Turing machine is an extremely tedious task, and we will instead follow tradition by reverting to describing our algorithms in a more informal language, yet specifically enough to make it evident that they can be implemented on a computer.

The notion of computability of a function is easily extended to that of a real number x with binary expansion $x = \sum_i x_i 2^{-i}$: we say that x is computable if $f(i) = x_i$ is a computable function (and it is easy to see that this computability property is unchanged if we switch to, e.g., base 3 or any other integer base). We can now state our main result:

Theorem 1.1 *The critical value $p_c = p_c(\mathbf{Z}^d)$ for Bernoulli bond percolation on $\mathbf{Z}^d$ is computable for any d.*

Our choice to state and prove the result only for bond percolation is just a matter of convenience: the result and its proof allow almost verbatim translation to the site percolation setting.

One peculiarity of our proof of Theorem 1.1 is a slight lack of constructiveness: Either p_c is dyadic (i.e., equals $j2^{-i}$ for some integers i and j), or it is not. If it is, then obviously it is also computable, while if it is not, then the algorithm in Section 2 will compute p_c. We are thus unable to point at a single algorithm and with confidence say that *this* algorithm computes p_c. This peculiarity is an artifact of the precise choice of definition of computability of a real number x. If instead (as some authors prefer) we choose the equivalent definition of saying that x is computable if there exists an algorithm which given any i produces an interval of length 2^{-i} that contains x, then a minor variation of the algorithm in Section 2 will suffice to achieve this regardless of whather x is dyadic or not; see Remark 2.3.

The algorithm outlined in Section 2 – or more precisely the variant given in Remark 2.3 – improves a (randomized) scheme for estimating p_c due to Meester and Steif [11]. Their scheme produces a sequence of estimates $\hat{p}_c^{(1)}, \hat{p}_c^{(2)}, \ldots$ that converges (almost surely) to p_c, but at no stage is there any guarantee that $\hat{p}_c^{(i)}$ is within a given distance ε from p_c. In contrast, our algorithm yields a sequence $\hat{p}_c^{(1)}, \hat{p}_c^{(2)}, \ldots$ for which we know that $|\hat{p}_c^{(i)} - p_c| \leq 2^{-i}$ for each i.

How far – i.e., to which lattices and graphs – can Theorem 1.1 be extended? Certainly not to all graphs, because, as observed by van den Berg [2], there exists for any $p \in [0,1]$ a graph G with $p_c(G) = p$. It might be tempting to hope that $p_c(G)$ is computable for every *transitive* graph, but I suspect that even this is false, in view of Leader and Markström's [9] construction of uncountable families of non-isomorphic transitive graphs.

2 The algorithm and some basic properties

Fix the dimension d. For N a multiple of 8, define

$$\Lambda_N = \left\{ x = (x_1, \ldots, x_d) \in \mathbf{Z}^d : -\tfrac{5N}{8} \leq x_j \leq \tfrac{5N}{8} \text{ for } j = 1, \ldots, d \right\},$$

and define E_{Λ_N} as the set of edges in the $\mathbf{Z}^d$ lattice whose endpoints are both in Λ_N. For Bernoulli bond percolation on $\mathbf{Z}^d$ consider the two events A_N and B_N defined in terms of the edges in E_{Λ_N} as follows.

- A_N is the event that at least one connected component of the set of open edges in E_{Λ_N} contains two vertices at Euclidean distance more than $N/10$ from each other.
- B_N is the event that the set of open edges in E_{Λ_N} contains a connected component intersecting all the $2d$ sides of the cube Λ_N, but that no other connected component contains two vertices at Euclidean distance more than $N/10$ from each other.

For $p \in [0,1]$, define $\mathbf{P}_p$ as the probability measure on $\{0,1\}^{E_{\mathbf{Z}^d}}$ corresponding to Bernoulli bond percolation on $\mathbf{Z}^d$. Note that for rational p, the probability $\mathbf{P}_p(A_N)$ is easy to compute: just go through all the $2^{d(\frac{5N}{4})^d}$ different configurations $\omega \in \{0,1\}^{E_{\Lambda_N}}$, check for each of them whether A_N happens, and sum

$$p^{n(\omega)}(1-p)^{d(\frac{5N}{4})^d - n(\omega)}$$

over those ω's for which A_N happens; here $n(\omega)$ is the number of 1's in ω. By the same token, we can compute $\mathbf{P}_p(B_N)$.

Our algorithm which, given i, produces the i first binary digits of p_c – or equivalently, gives an interval of the form $[j2^{-i}, (j+1)2^{-i})$ containing p_c – is as follows.

(I) Set $N = 8$.

(II) Compute $\mathbf{P}_{j2^{-i}}(A_N)$ and $\mathbf{P}_{j2^{-i}}(B_N)$ for $j = 0, 1, \ldots, 2^i$.

(III) If for some $j \in \{0, \ldots, 2^i - 1\}$ we have
$$\mathbf{P}_{j2^{-i}}(A_N) < (2d-1)^{-3^d}$$
and
$$\mathbf{P}_{(j+1)2^{-i}}(B_N) > 1 - 8^{-9},$$
then let j' be the smallest such j, output the interval $[j'2^{-i}, (j'+1)2^{-i})$ and stop. Otherwise increase N by 8 and continue with (II).

We need to show, under the assumption that p_c is non-dyadic, that the algorithm terminates after some finite number of cycles, and that the interval $[j2^{-i}, (j+1)2^{-i})$ it outputs satisfies
$$p_c \in [j2^{-i}, (j+1)2^{-i}). \tag{1}$$
The termination property follows from Proposition 2.1 below applied to $p = j2^{-i}$ and $p^* = (j+1)2^{-i}$ where $j = \max\{j' : j'2^{-i} < p_c\}$, and property (1) follows from Proposition 2.2. Hence, once the two propositions are proved, we know that non-dyadicity of p_c implies its computability, and as explained in Section 1 this implies Theorem 1.1. We defer the proofs of the propositions to Section 3.

Proposition 2.1

(a) *For any $p < p_c$, we have* $\lim_{N\to\infty} \mathbf{P}_p(A_N) = 0$.

(b) *For any $p > p_c$, we have* $\lim_{N\to\infty} \mathbf{P}_p(B_N) = 1$.

Proposition 2.2

(a) *For no $p < p_c$ and no $N \in 8, 16, 24, \ldots$ do we have* $\mathbf{P}_p(B_N) > 1 - 8^{-9}$.

(b) *For no $p > p_c$ and no $N \in 8, 16, 24, \ldots$ do we have* $\mathbf{P}_p(A_N) < (2d-1)^{-3^d}$.

Remark 2.3 The reason why the above algorithm doesn't necessarily work in case p_c is dyadic is that if $j2^{-i} = p_c$, then we may end up having $\mathbf{P}_{j2^{-i}}(A_N) > (2d-1)^{-3^d}$ and $\mathbf{P}_{j2^{-i}}(B_N) < 1 - 8^{-9}$ for all N, causing the algorithm to keep running without ever terminating. If we are content with an interval of width 2^{-i+1} (which can still be made as small as we wish) containing p_c, then regardless of dyadicity the algorithm will terminate and produce an interval containing p_c if we simply replace step (III) above by

(III') If for some $j \in \{0, \ldots, 2^i - 1\}$ we have $\mathbf{P}_{j2^{-i}}(A_n) < (2d-1)^{-3^d}$ and $\mathbf{P}_{(j+2)2^{-i}}(B_n) > 1 - 8^{-9}$, then take j' to be the smallest such j, output the interval $[j'2^{-i}, (j'+2)2^{-i})$ and stop. Otherwise increase N by 8 and continue with (II).

Remark 2.4 The proposed algorithm is obviously incredibly slow – so slow that nobody in her right mind would use it in practice to estimate p_c. It might nevertheless be of some theoretical interest to find bounds for its running time. Such bounds can presumably be obtained by inspecting the proofs of the limit theorems from [12] and [1] used in the next section, and extracting convergence rates (though it might require hard work). For the algorithm in Remark 2.3 the bounds can probably be obtained independently of any detailed information about p_c, whereas for the original algorithm p_c would have to enter the bound in one way or another. To see this, suppose for instance that $p_c \in (\frac{3}{8}, \frac{3}{8} + 10^{-1000})$. To get the third binary digit in place we would have to decide whether $\frac{3}{8}$ is sub- or supercritical, and since $\frac{3}{8}$ is so close to p_c that would require a stupendously large N – and the running time is exponential in N.

3 Proofs

The definitions of the events A_N and B_N are tailored to fit into known percolation technology to make the proof of especially Proposition 2.1 as streamlined as possible. In particular, the proof of Proposition 2.1 (a) is based on the famous exponential decay result for subcritical percolation, which was proved by Menshikov [12] and is explained at greater length by Grimmett [4]. Writing D for the radius of the connected component containing the origin 0, i.e.

$$D = \sup\{\mathrm{dist}(0,x) : x \in \mathbf{Z}^d, x \leftrightarrow 0\}$$

where dist denotes Euclidean distance, the result states that for any $p < p_c$ there exists a $C = C(p) > 0$ such that

$$\mathbf{P}_p(D > n) < e^{-Cn} \tag{2}$$

for all n.

Proof of Proposition 2.1 (a). Fix $p < p_c$ and choose $C > 0$ in such a way that (2) holds for all n. Write Z_N for the number of vertices $x \in \Lambda_N$ that are connected to at least one vertex at distance at least $N/10$ away. Since Λ_N contains $\left(\frac{5N}{4}+1\right)^d$ vertices, (2) implies that the expected value of Z_N satisfies

$$\mathbf{E}_p[Z_N] < \left(\tfrac{5N}{4}+1\right)^d e^{-CN/10}$$

so that

$$\begin{aligned}
\lim_{N\to\infty} \mathbf{P}_p(A_N) &\leq \lim_{N\to\infty} \mathbf{P}_p(Z_N \geq 1) \\
&\leq \lim_{N\to\infty} \mathbf{E}_p[Z_N] \\
&\leq \lim_{N\to\infty} \left(\tfrac{5N}{4}+1\right)^d e^{-CN/10} = 0,
\end{aligned}$$

as desired. □

The proof of Proposition 2.1 (b) consists in a reference to a result of Antal and Pisztora [1]. The choice of using this particular result is somewhat arbitrary, and could be replaced by any of a number of similar results from the renormalization technology pioneered by Grimmett and Marstrand [5] and discussed at a gentler pace by Grimmett [4].

Proof of Proposition 2.1 (b). This is Antal and Pisztora [1, Prop. 2.1]. □

In order to prove Proposition 2.2, we need the notion of *1-dependent site percolation.* This is a generalization of ordinary (Bernoulli) site percolation where the independence assumption is weakened as follows. The L_∞-distance between two vertices $(x,y) \in \mathbf{Z}^d$ with coordinates $x = (x_1,\ldots,x_d)$ and $y = (y_1,\ldots,y_d)$ is defined to be $\max_i\{|x_i - y_i|\}$.

Definition 3.1 *A $\{0,1\}^{\mathbf{Z}^d}$-valued random object X is said to be 1-dependent if for any finite collection of vertices $x_1,\ldots,x_k \in \mathbf{Z}^d$ such that no two of them are within L_∞-distance 1 from each other we have that $X(x_1),\ldots,X(x_k)$ are independent.*

The key to proving Proposition 2.2 is the following lemma about 1-dependent site percolation. By a cluster, we mean a maximal connected component of open vertices.

Lemma 3.2 *Fix $\alpha \in [0,1]$, and let $X \in \{0,1\}^{\mathbf{Z}^d}$ be a translation invariant and 1-dependent site percolation process such that for each $x \in \mathbf{Z}^d$ we have $\mathbf{P}(X(x) = 1) = \alpha$.*

(a) *If $\alpha > 1 - 8^{-9}$, then a.s. X contains an infinite cluster.*

(b) *If $\alpha < (2d-1)^{-3^d}$, then a.s. X contains no infinite cluster.*

One way to prove this result (with the contants $1-8^{-9}$ and $(2d-1)^{-3^d}$ inconsequentially replaced by other constants in $(0,1)$) is to invoke the stochastic domination results of Liggett, Schonmann and Stacey [10]. Here we opt, instead, for a simple modification of the classical path and contour counting arguments for proving $p_c \in (0,1)$ in standard Bernoulli percolation, as outlined, e.g., in the introductory chapter of Grimmett [4]. We begin with part (b) of the lemma.

Proof of Lemma 3.2 (b). For any n, the number of non-selfintersecting paths in the $\mathbf{Z}^d$ lattice from the origin is at most $2d(2d-1)^{n-1}$; this is because the first vertex to go to can be chosen in $2d$ ways, and from then on there are at most $2d-1$ vertices to choose from in each step. In each such path R, we can find a subset S of its vertices that has cardinality at least $n/3^d$ and such that no two $x, y \in S$ are L_∞-neighbors; such a subset can be found by picking vertices in R sequentially and deleting all their L_∞-neighbors. By the definition of 1-dependence, we get that each such R has probability at most $\alpha^{n/3^d}$ of being open, in the sense that all its vertices are open. The expected number of such open paths of length n from the origin is therefore bounded by $2d(2d-1)^{n-1}\alpha^{n/3^d}$, which when $\alpha < (2d-1)^{-3^d}$ tends to 0 as $n \to \infty$. Hence, the origin has probability 0 of being in an infinite cluster, and the same argument applies with the origin replaced by any $x \in \mathbf{Z}^d$, so the existence of an infinite cluster has probability 0. $\square$

Proof of Lemma 3.2 (a). We proceed similarly as in part (b). First note that by restricting to a two-dimensional hyperplane in $\mathbf{Z}^d$, we reduce the problem to only having to consider the case $d = 2$. Define a $*$-path as a sequence of vertices with consecutive vertices at L_∞-distance 1 from each other, and a $*$-circuit as one that ends within L_∞-distance 1 from its starting point. Next, define a *$*$-contour* in $\mathbf{Z}^2$ as a non-selfintersecting $*$-circuit that surrounds the origin, and a *closed $*$-contour* as one whose vertices are all closed. Similarly as in (b), we get that the number of countours of length n is bounded by $n8^n$, and that each of them has probability at most $(1-\alpha)^{n/9}$ of being closed. The expected number of closed contours is thus bounded by

$$\sum_{n=1}^{\infty} n8^n(1-\alpha)^{n/9}$$

which is finite when $\alpha > 1 - 8^{-9}$. For such α the number of closed contours is therefore a.s. finite, whence either the origin itself or some open vertex just outside the "outermost" contour is in an infinite cluster. $\square$

Proposition 2.2 will now be proved using two simple renormalization representations (X and Y below) of Bernoulli bond percolation. Given $N \in \{8, 16, 24, \ldots\}$ and $p \in [0,1]$, consider a Bernoulli bond percolation process $X \in \{0,1\}^{E_{\mathbf{Z}^d}}$ with retention parameter p. For $x \in \mathbf{Z}^d$, define the box

$$\Lambda_{N,x} = \{y \in \mathbf{Z}^d : y - Nx \in \Lambda_N\}$$

and define the events $A_{N,x}$ and $B_{N,x}$ analogously to A_N and B_N but pertaining to the edges in $\Lambda_{N,x}$ rather than in Λ_N. Define two site percolation processes $Y, Z \in \{0,1\}^{\mathbf{Z}^d}$ by setting, for each $x \in \mathbf{Z}^d$,

$$Y(x) = \begin{cases} 1 & \text{if } A_{N,x} \\ 0 & \text{otherwise} \end{cases}$$

and

$$Z(x) = \begin{cases} 1 & \text{if } B_{N,x} \\ 0 & \text{otherwise.} \end{cases}$$

The boxes $\Lambda_{N,x}$ and $\Lambda_{N,y}$ intersect only if $x, y \in \mathbf{Z}^d$ are L_∞-neighbors, whence Y and Z are both 1-dependent percolation processes.

Proof of Proposition 2.2 (a). Assume for contradiction that $p < p_c$ and that $\mathbf{P}_p(B_N) > 1 - 8^{-9}$. For $x \in \mathbf{Z}^d$ such that $B_{N,x}$ happens, write C_x for the connected component of X whose restriction to $\Lambda_{N,x}$ connects all $2d$ sides of $\Lambda_{N,x}$. Note that if $x, y \in \mathbf{Z}^d$ are L_1-neighbors, then $\{Z(x) = 1\} \cup \{Z(y) = 1\}$ implies that C_x and C_y coincide. By iterating this argument, we get for arbitrary $x, y \in \mathbf{Z}^d$ that if $Z(x) = 1$, $Z(y) = 1$ and x and y are in the same connected component of Z, then C_x and C_y coincide. In particular, if $x \in \mathbf{Z}^d$ satisfies $Z(x) = 1$ and sits in an infinite connected component of Z, then C_x is infinite. But since

$$\begin{aligned} \mathbf{P}(Z(x) = 1) &= \mathbf{P}_p(B_{N,x}) \\ &= \mathbf{P}_p(B_N) > 1 - 8^{-9} \end{aligned}$$

for all x, Lemma 3.2 (a) tells us that Z contains a.s. some infinite cluster. Hence C_x is infinite for some $x \in \mathbf{Z}^d$, so X contains an infinite cluster, contradicting $p < p_c$. □

Proof of Proposition 2.2 (b). Assume for contradiction that $p > p_c$ and that $\mathbf{P}_n(A_N) < (2d-1)^{-3^d}$. If the origin $\mathbf{0}$ is in an infinite cluster of the bond percolation process X, then we must (by the definition of $A_{N,x}$ and Y) have $Y(\mathbf{0}) = 1$, and furthermore that $\mathbf{0}$ belongs to an infinite cluster of Y. But since

$$\begin{aligned} \mathbf{P}(Y(x) = 1) &= \mathbf{P}_p(A_{N,x}) \\ &= \mathbf{P}_p(A_N) < (2d-1)^{-3^d} \end{aligned}$$

for all x, we get by Lemma 3.2 (b) that Y contains a.s. no infinite cluster. Hence $\mathbf{0}$ is a.s. not in an infinite cluster of X, contradicting $p > p_c$. □

References

[1] Antal, P. and Pisztora, A. (1996) On the chemical distance for supercritical Bernoulli percolation, *Ann. Probab.* **24**, 1036–1048.

[2] van den Berg, J. (1982) A note on percolation, *J. Phys. A* **15**, 605–610.

[3] Blass, A. and Gurevich, Y. (2003) Algorithms: a quest for absolute definitions, *Bull. Europ. Assoc. Theoret. Comp. Sci.* **81**, 195–225.

[4] Grimmett, G.R. (1999) *Percolation* (2nd ed), Springer, New York.

[5] Grimmett, G.R. and Marstrand, J.M. (1990) The supercritical phase of percolation is well behaved, *Proc. Royal Soc. A* **430**, 439–457.

[6] Häggström, O. and Jonasson, J. (2006) Uniqueness and non-uniqueness in percolation theory, *Probab. Surveys* **3**, 289–344.

[7] Kesten, H. (1980) The critical probability of bond percolation on the square lattice equals $\frac{1}{2}$, *Comm. Math. Phys.* **74**, 41–59.

[8] Knuth, D.E. (1973) *The Art of Computer Programming, Second Edition, Volume 1/Fundamental Algorithms*, Addison-Wesley, New York.

[9] Leader, I. and Markström, K. (2006) Uncountable families of vertex-transitive finite degree graphs, *Discrete Math.* **306**, 678–679.

[10] Liggett, T.M., Schonmann, R.H. and Stacey, A.M. (1997) Domination by product measures, *Ann. Probab.* **25**, 71–95.

[11] Meester, R. and Steif, J.E. (1998) Consistent estimation of percolation quantities, *Statist. Neerlandica* **52**, 226–238.

[12] Menshikov, M.V. (1986) Coincidence of critical points in percolation problems, *Soviet Math. Dokl.* **33**, 856–859.

Dept of Mathematics
Chalmers University of Technology
412 96 Göteborg
Sweden
`olleh@math.chalmers.se`
`http://www.math.chalmers.se/~olleh/`