Estimation of bond percolation thresholds on the Archimedean lattices

2007 J. Phys. A: Math. Theor. 40 9253

(http://iopscience.iop.org/1751-8121/40/31/005)

J. Phys. A: Math. Theor. **40** (2007) 9253–9258 doi:10.1088/1751-8113/40/31/005

Estimation of bond percolation thresholds on the Archimedean lattices

Robert Parviainen

ARC Centre of Excellence for Mathematics and Statistics of Complex Systems,
139 Barry Street, The University of Melbourne, Victoria 3010, Australia

E-mail: robertp@ms.unimelb.edu.au

Received 27 April 2007, in final form 14 June 2007
Published 19 July 2007
Online at stacks.iop.org/JPhysA/40/9253

Abstract

We give accurate estimates for the bond percolation critical probabilities on seven Archimedean lattices, for which the critical probabilities are unknown, using an algorithm of Newman and Ziff.

PACS numbers: 64.60.Ak, 05.70.Jk
Mathematics Subject Classification: 60K35, 82B43

1. Introduction

Since the introduction of percolation theory, it has been an interesting and challenging problem to determine the percolation thresholds. Only a few non-trivial graphs are exactly solved, such as the bond model on the square, triangular and hexagonal lattices, and the site model on the triangular and Kagomé lattices. Recently, Scullard and Ziff have found many new thresholds for various classes of lattices [6, 9]. They also conjecture [10] the value of the threshold for one lattice considered here: the $(3, 12^2)$ lattice and also for the Kagomé lattice. Precise estimates have been calculated for example for the site model on the Archimedean lattices [7] and for the bond model on the Kagomé lattice [11].

Bounds, more or less tight, have been found by various authors for some lattices. For the bond model on the Archimedean lattices, see [4] for a review of rigorous bounds on critical probabilities. Recently, Riordan and Walters [5] have given tight rigorous confidence intervals for both site and bond percolation on all Archimedean lattices.

In this paper, we provide precise estimates for the bond percolation thresholds for the unsolved Archimedean lattices.

1.1. Archimedean lattices

The Archimedean lattices are the vertex transitive graphs that can be embedded in the plane such that every face is a regular polygon. A polygon is regular if all edges have the same

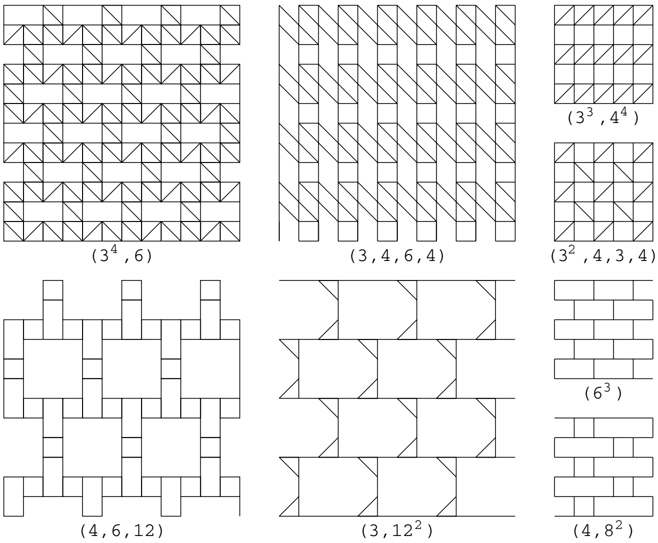

Figure 1. Finite subgraphs of square embeddings of eight Archimedean lattices.

length and all interior angles are the same. Kepler [1] showed that there exist exactly 11 such graphs.

The lattices are given names according to the sizes (number of sides of the polygon) of faces incident to a given vertex. The face sizes are listed in order, starting with a face such that the list is the smallest possible in lexicographical order. The square lattice thus gets the name (4, 4, 4, 4), abbreviated to (4^4), and the Kagomé lattice the name (3, 6, 3, 6).

Square representations of the Archimedean lattices studied here are shown in figure 1.

The bond percolation threshold is known exactly for the square, triangular and hexagonal lattices (the values are 0.5, $2\sin(\frac{\pi}{18})$ and $1-2\sin(\frac{\pi}{18})$), and the threshold for the Kagomé lattice has previously been estimated to 0.524 4053 by Ziff and Suding [11]. Ziff and Scullard's [10] conjectured value for $(3, 12^2)$ is 0.740 421 178 . . ., and it is consistent with the estimate given here: 0.740 4219(8)—the difference is less than one standard deviation (further, there is a small positive bias in the estimate).

In this work, we estimate the thresholds for all the Archimedean lattices with unknown (and in one case conjectured) threshold except the previously studied Kagomé lattice. The hexagonal lattice is used as a benchmark.

1.2. Percolation

The bond percolation process is defined as follows. For each edge of a graph G, declare the edge to be open with probability p, independently of all other edges and closed otherwise.

For the kind of graphs studied here, it is well known that there exists a critical value $p_c(G)$, called the critical probability, or threshold, such that for $p > p_c(G)$, there exists an unique infinite connected component of open edges, while for $p < p_c(G)$, only finitely large connected components of open edges exist.

1.3. Motivation

One objective of this simulation study is to get an empirical answer to the following question. Recall that the self-avoiding walk connective constant μ for a lattice is given by the limit $\mu = \lim_{n\to\infty} g_n^{1/n}$, where g_n is the number of self-avoiding walks of length n.

Question 1.1. *If G and H are two Archimedean lattices, with self-avoiding walk connective constants $\mu(G)$ and $\mu(H)$, site percolation thresholds $p_c^s(G)$ and $p_c^s(H)$, and bond percolation thresholds $p_c^b(G)$ and $p_c^b(H)$, is it true that*

$$\mu(G) \leqslant \mu(H) \Leftrightarrow p_c^s(G) \geqslant p_c^s(H) \Leftrightarrow p_c^b(G) \geqslant p_c^b(H)?$$

The answer is negative for general two-dimensional, quasi-transitive, planar graphs, as shown by Wierman [8] and Parviainen [3].

Available estimates and rigorous bounds for site percolation thresholds and connective constants suggest that these two models give the same order on the Archimedean lattices.

The estimates given in this work suggest, however, that there exist two pairs of Archimedean lattices for which the bond and site percolation thresholds are in opposite order, namely the pair $(3, 6, 3, 6)$ (Kagomé) and $(3, 4, 6, 4)$ (Ruby), and the pair $(3^3, 4^2)$ and $(3^2, 4, 3, 4)$.

2. The method of Newman and Ziff

Newman and Ziff [2] have developed a fast algorithm for estimating percolation thresholds (both bond and site); the running time is nearly *linear* in the system size (the number of vertices or edges of the subgraph), while still producing accurate estimates.

It turns out that it is profitable to modify the subgraphs used, and use torus-shaped regions. We will consider $Q(p)$, the probability that a cluster wraps around the torus in one direction, but not both.

If we generate a percolation process on a subgraph with M edges, by adding edges in random order, we get $M + 1$ different, but dependent, realizations of a percolation process. These give a rough estimate of the function $q(n)$, the probability that a cluster spans the torus in one direction but not the other, given that there are exactly n open edges.

Assume for the moment that $q(n)$ is exactly known. The law of total probability then gives us $Q(p)$:

$$Q(p) = \sum_{n=0}^{M} q(n) \binom{M}{n} p^n (1-p)^{M-n}.$$

Thus, from a single realization of the Newman–Ziff algorithm we get an estimate of $Q(p)$ for *all* values of p, from 0 to 1.

In our case, p_c can be estimated by the value of p at which $Q(p)$ is maximized, since the probability $Q(p)$ tends to zero both for p both above and below p_c, as the system size grows.

For the method to achieve its impressive running time, it is necessary to keep track of clusters efficiently. This can be achieved by a tree-based union/find algorithm. For each added edge, we *find* the clusters to which the endpoints belong. If the clusters are different, the *union* of the clusters is calculated. Both steps are rapidly done by representing the set of clusters as a directed forest. By a small modification of the union/find algorithm, detection of cluster wrapping is also easy.

We can speed up the execution by delaying the convolution (equation (2)) and maximization step, by averaging over a batch of, say m, estimates of $q(n)$ and use the average to estimate $Q(p)$ and p_c. One minor drawback by doing so is that we get fewer samples for the estimation of the statistical error. Also, the standard error[1] does not decrease as $1/\sqrt{m}$—but this is only true for small m, and a $1/\sqrt{m}$ factor dominates from $m \approx 10$

[1] The standard error is an estimate of the standard deviation of the error of the estimate—if the estimator is a mean of n values, the standard error is s.e. $= s/\sqrt{n}$, where s is the sample standard deviation.

Table 1. Simulation results. The standard error is denoted by s.e., the number of realizations by n, the system size by M and the estimated value of the exponent a is denoted by $\hat{a}$.

Lattice	$\hat{p}_c$	s.e./10^{-7}	n	M	$\hat{a}$
$(3, 12^2)$	0.740 421 95	8.0	2485	67 500	0.276
$(4, 6, 12)$	0.693 733 83	7.2	2930	73 728	0.277
$(4, 8^2)$	0.676 802 32	6.3	2983	60 000	0.284
$(3, 4, 6, 4)$	0.524 832 58	5.3	7568	64 800	0.286
$(3^4, 6)$	0.434 306 21	5.0	7397	64 000	0.274
$(3^3, 4^2)$	0.419 641 91	4.3	9822	64 000	0.274
$(3^2, 4, 3, 4)$	0.414 137 43	4.6	7504	64 000	0.269

and onwards. For the largest system sizes considered here, the standard error for large m is approximately $2/\sqrt{m}$ times that for $m = 1$. We thus lose only a factor 2 which is cheap considering that the convolution and maximization step can be of the order of 100 times more time-consuming than the generation step.

We have also run short simulations with the hull gradient method (see, for example, [7]), using different representations (of the covering graphs) of the lattices, to verify the implementations. (A reason for not using the hull gradient for longer runs was the memory requirement; the simulations were run on standard desktop computers, simultaneously used in daily work by the author's colleagues.)

3. Numerical results

The finite size error decreases very fast. For our benchmark case, the hexagonal lattice, the finite size bias was of the order 10^{-8} for systems of size 50 000.

This prompted us to concentrate the computations on only one large system per lattice. We used square-shaped subgraphs, with between 60 000 and 75 000 edges, for which we believe the finite size errors to be of the order 10^{-8}, an order of magnitude smaller than the statistical errors. The results are summarized in table 1. The standard error given is $s_m/\sqrt{m}$, where s_m is the sample standard deviation when m iterations are used for each maximization step. If we do n realizations, s_m is given by

$$s_m^2 = \frac{1}{N-1}\sum_{i=1}^{n}\left(\hat{p}_c^{(i)} - \hat{p}_c\right)^2,$$

where $\hat{p}_c^{(i)}$ is the estimate from realization i and $\hat{p}_c$ is the grand estimate, the mean of $\hat{p}_c^{(i)}$. Approximate $(1-\alpha)\%$ confidence intervals are given by

$$\hat{p}_c \pm z_{\alpha/2}\,\text{s.e.},$$

where $z_{\alpha/2}$ is the $\alpha/2$ quantile of the Gaussian distribution[2]. For all lattices, we used batches of 10^5 realizations for each maximization step.

In [2], it was observed that the standard deviation scales as

$$s \sim M^{-3/8},$$

where M is the system size (the number of edges). In our simulations, we generally observe a slightly slower convergence. Fitting $s = CM^{-a}$ we found some variations in the estimated

[2] Strictly speaking, t-distribution quantiles should be used, but for the values of n used here they agree with the Gaussian quantiles.

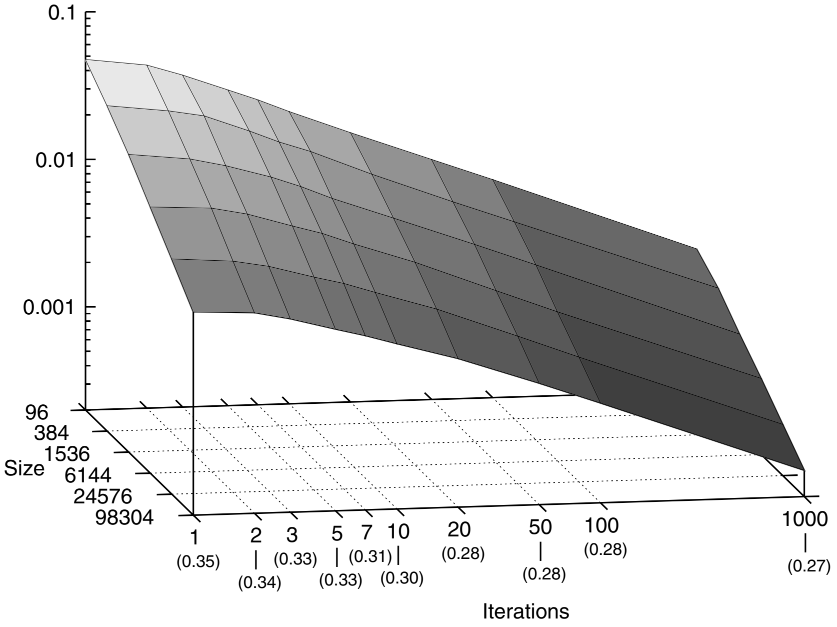

Figure 2. Sample standard deviation as a function of the system size and the number of iterations per maximization step.

values of a between lattices; we observed values in the range 0.25–0.40. The values decrease as the number of iterations m used per maximization step increases, but appear to settle down for $m \approx 20$. The between-lattice variation also decreases with m. As the purpose here is to get precise estimates for a number of lattices, we have not studied the convergence rate thoroughly for all combinations of lattices and parameters. In table 1, we report estimates of a for $m = 100$. It is hard to get a rigorous estimate of the precision of these estimates, but based on these values it is unlikely that these lattices share a common value of the exponent. For the hexagonal lattice, figure 2 shows a log–log–log plot of the sample standard deviation as a function of the system size and the number of iterations per maximization step. The numbers in parentheses give the estimated values of the exponents a for the fixed values of m.

For the hexagonal lattice, we did long simulations for six different system sizes to study the precision. We used batches of 10^4 realizations. The results are summarized in table 2. The exact value of the bond percolation threshold is $p_c = 1 - 2\sin(\pi/18) \approx 0.652\,703\,6446$.

The small finite size bias observed in [2] is confirmed by our simulations. The estimates $\hat{p}_c$ are conjectured to converge like

$$|\hat{p}_c - p_c| \sim M^{-11/8}.$$

Fitting

$$\hat{p}_c = p_c + bM^{-11/8}$$

to data, excluding the $M = 96$ data point, gives $b = 0.0017$ and a finite size error for $M = 50\,000$ of 3.4×10^{-8}.

Excluding further data points for small values of M gives slight variations, but the finite size error is always below 4×10^{8}. This supports our belief that the finite size error for our main estimates is considerably less than the statistical error.

Table 2. Simulation results for the hexagonal lattice. Here, M denotes the number of edges, n the total number of realizations used for the estimate $\hat{p}_c$, with standard error s.e. Further, s_1 is the sample standard deviation when one repetition is used for each maximization step, and s_∞ is an estimate of the limiting sample standard deviation per repetition when m repetitions are used for each maximization step.

M	n	$\hat{p}_c$	s.e.	s_1	s_∞
96	754 700	0.652 685 19	6.68×10^{-7}	0.048	0.058
384	634 300	0.652 707 81	5.38×10^{-7}	0.032	0.043
1 536	460 520	0.652 704 77	4.49×10^{-7}	0.021	0.030
6 144	444 786	0.652 703 77	3.12×10^{-7}	0.013	0.021
24 576	41 727	0.652 703 87	6.58×10^{-7}	0.0079	0.014
98 304	5 326	0.652 703 67	1.32×10^{-6}	0.0048	0.0097

It is also worthwhile noting that, except for very small sizes, the finite size bias is positive in contrast to the site percolation case studied in [2]. (The same is observed with the hull gradient method; for the bond cases studied here, the finite size bias is positive, while the site cases on the same lattices studied in [7] have negative finite size biases.)

Acknowledgments

I am grateful to Robert Ziff for many helpful answers and comments and for encouragement. I also acknowledge the support received from the Australian Research Council Centre of Excellence for Mathematics and Statistics of Complex Systems (MASCOS).

References

[1] Kepler J 1619 *Harmonices Mundi* (Lincii)

[2] Newman M E J and Ziff R M 2001 A fast Monte Carlo algorithm for site or bond percolation *Phys. Rev.* E **64** 016706

[3] Parviainen R 2004 Pairs of graphs with connective constants and critical probabilities in the same order (in connectivity properties of Archimedean and Laves lattices) *Uppsala Dissertations Math.* **34** (Paper IV) 1–5

[4] Parviainen R and Wierman J C 2005 Inclusions and non-inclusions among the Archimedean and Laves lattices, with applications to bond percolation thresholds *Cong. Numer.* **176** 89–128

[5] Riordan O and Walters M 2007 Rigorous confidence intervals for critical probabilities *Preprint* math.PR/0702232v1

[6] Scullard C R 2006 Exact site percolation thresholds using a site-to-bond transformation and the star-triangle transformation *Phys. Rev.* E **73** 016107

[7] Suding P N and Ziff R M 1999 Site percolation thresholds for Archimedean lattices *Phys. Rev.* E **60** 275–83

[8] Wierman J C 2003 Pairs of graphs with site and bond percolation critical probabilities in opposite orders *Disc. Appl. Math* **129** 545–8

[9] Ziff R M 2006 Generalized cell-dual-cell transformation and exact thresholds for peroclation *Phys. Rev.* E **73** 016134

[10] Ziff R M and Scullard C R 2006 Predictions of bind percolation thresholds for the Kagomé and Archimedean $(3, 12^2)$ lattices *Phys. Rev.* E **73** 045102

[11] Ziff R M and Suding P N 1997 Determination of the bond percolation threshold for the Kagomé lattice *J. Phys. A: Math. Gen.* **30** 5351–9