Fractal dimension of 3-blocks in four-, five-, and six-dimensional percolation systems

Gerald Paul* and H. Eugene Stanley
Center for Polymer Studies and Department of Physics, Boston University, Boston, Massachusetts 02215
(Received 15 October 2002; published 5 February 2003)

Using Monte Carlo simulations, we study the distributions of the 3-block mass N_3 in four-, five-, and six-dimensional percolation systems. Because the probability of creating large 3-blocks in these dimensions is very small, we use a ''go with the winners'' method of statistical enhancement to simulate configurations having probability as small as 10^{-30}. In earlier work, the fractal dimensions of 3-blocks, d_3, in 2D (two dimensional) and 3D were found to be 1.20 ± 0.1 and 1.15 ± 0.1, respectively, consistent with the possibility that the fractal dimension might be the same in all dimensions. We find that the fractal dimension of 3-blocks decreases rapidly in higher dimensions, and estimate $d_3=0.7\pm0.2$ (4D) and 0.5 ± 0.2 (5D). At the upper critical dimension of percolation, $d_c=6$, our simulations are consistent with $d_3=0$ with logarithmic corrections to power-law scaling.

DOI: 10.1103/PhysRevE.67.026103 PACS number(s): 64.60.Fr, 05.45.Df, 64.60.Ak

I. INTRODUCTION

Percolation is a classic model for disorder [1–3]. It continues to be of interest both because of the application of the model to various physical phenomena from flow in porous media [4] to the behavior of forest fires [5] and because as a simple geometrical model of a system with a phase transition it provides an ideal environment for studying the properties of critical systems [6].

A number of years ago, it was realized that for bond percolation the incipient infinite cluster can be decomposed into simply connected ''links'' and multiply connected ''blobs'' [7]. Recently, it has been recognized for bond percolation that clusters and blobs are the $k=1$ and $k=2$ cases of k-connected graphs (k-blocks), graphs in which all vertices are connected to every other vertex in the k-block by at least k independent paths [8–10]. The values of the fractal dimension d_3 of 3-blocks in two- and three-dimensional(2D and 3D) percolation systems at the percolation threshold were found to be 1.20 ± 0.1 and 1.15 ± 0.1, respectively [9].

The fact that the fractal dimensions of 3-blocks are identical within error bars is consistent with the possibility that d_3 might be independent of dimension. This independence on dimension would be surprising because all other nontrivial exponents depend on dimension below the upper critical dimension $d_c=6$. To investigate whether d_3 is, in fact, independent of dimension, we focus in this paper on determining d_3 for $d=4$, 5, and 6 using Monte Carlo simulations.

In the following section, we study d_3 for percolation on the Cayley tree in order to gain insight into the behavior of 3-blocks in very high dimension. In Sec. III, we discuss the methods we use to generate large 3-blocks in 4D, 5D, and 6D. In Sec. IV, we discuss our results.

II. CAYLEY TREE RESULTS

Percolation on the Cayley tree has been used as a model for percolation for $d\geqslant6$, the upper critical dimension of percolation. The cluster fractal dimension and blob fractal dimension, as well as a number of other critical exponents on the Cayley tree, are identical to those of percolation for $d\geqslant6$ [1–3]. Below, we argue that for percolation on the Cayley tree $d_k=0$ for $k\geqslant3$ suggesting that while d_3 may change little between $d=2$ and $d=3$, eventually d_3 decreases more rapidly approaching zero for $d=6$. To show that $d_k=0$ for $k\geqslant3$ for percolation on the Cayley tree, we make use of the concept of k-bone. Reference [9] generalizes the concept of backbone by defining a k-bone as the set of all sites connected to k disjoint sets of points by k independent paths. Thus, clusters and backbones are k-bones with $k=1$ and 2, respectively. For a given k, the fractal dimension of k-bones and k-blocks are equal [9].

To see that for percolation on the Cayley tree the fractal dimension of a 3-bone is zero, we choose any three points on the boundary (Fig. 1) and observe that there is only one site which is connected to these points by independent paths. This result is independent of the size of the tree, even if the tree is fully populated. Hence, the fractal dimension is zero. Clearly this argument holds for larger k and is meaningful as

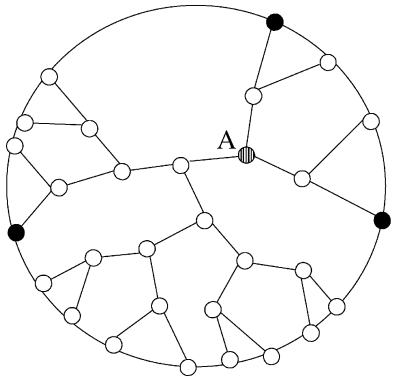

FIG. 1. Cayley tree in which each site except those on boundary has three neighbors. We note that even when all bonds are fully occupied, there can be only one site which is connected to three sites on the boundary. In this example the three sites on the boundary are the filled circles and the one site connected to them by independent paths is the striped circle denoted by *A*.

*Email address: gerryp@bu.edu

long as the branching factor in the Cayley tree is greater than or equal to k, and holds independent of size.

III. SIMULATION METHOD

A. Statistical enhancement method

Randomly generated realizations in which large 3-blocks are present become more and more rare as the system dimension increases. In fact, if traditional techniques are used to generate realizations, for d as low as 4 the range of the values of the masses of 3-blocks created are so small that one cannot determine d_3 either by finding the best collapse of plots of the distribution of masses or by inferring d_3 from the slope of the power-law regime of the distributions.

To overcome this problem, we use a ''go with the winners'' method of statistical enhancement described in Ref. [11]. The basic idea of this approach in the context of a percolation cluster growth algorithm is as follows:

(i) Before we start growing a cluster, assign a value of one to the weight W of the cluster.

(ii) We use the Leath method to grow clusters [12]. While the cluster is growing, we calculate certain properties of the state of the cluster after every interval of n chemical shells of growth.

(iii) If certain criteria on the properties of the state of the cluster that are described below are met, we ''clone'' the state so we have m copies (including the original) of the state, adjust W accordingly to W/m and continue growing each of these m clones. If these criteria are not met, simply continue growing the noncloned cluster.

Cloning can take place multiple times during the growth of a cluster; the result is a tree structure of realizations where the leaves of the tree represent the completion of cluster growth. At each of these completions of cluster growth, we calculate the quantities being studied, in this case 3-block masses, and with weight W update a histogram over all clusters. By adding them with weight W, statistical averages are not biased even though the ensemble is biased.

Here, m and n are parameters which can be tuned to achieve the desired level of ''rareness'' which can be reached. If n is large and m is small, there will be little cloning and we will generate clusters with weights only moderately smaller than without enhancement. If n is small and m is large, there will be much cloning and we will generate clusters with weights very much smaller than without enhancement. However, if n is sufficiently small and/or m is sufficiently large, cluster growth will effectively never end naturally, and we will not be able to extract useful information from the simulation.

From an implementation standpoint, it is not necessary to actually create copies of the state of the system in computer memory to create the clones; as noted in Ref. [11] we can effectively walk the clone tree in a ''depth-first'' manner, completely treating a given clone before we begin treating the next clone. What is required is that we save the state of the system before we begin growth based on a clone so that we can return to that state when we begin growth on the next clone. This saving of state is accomplished naturally with a ''last-in-first-out'' stack in which we maintain information about sites in the cluster.

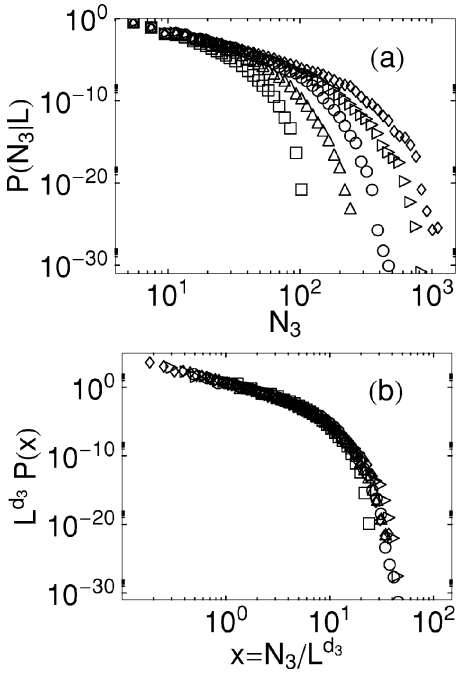

FIG. 2. $P(N_3|L)$, the distribution of 3-block mass N_3 for (from left to right) $L=8$, 16, 32, 64, and 128 for the case of four dimensions. (a) uncollapsed, (b) collapsed using a value of $d_3=0.7$.

We first attempted to create realizations with large 3-blocks by creating very dense clusters. We set as our criteria for cloning the condition that the number of occupied bonds actually created during the n shell interval be larger than the number of times we determined whether a bond should be occupied times the bond occupation probability. While this algorithm is very effective in creating dense clusters, it did not result in large 3-blocks within the clusters. We were, however, successful in creating clusters with large 3-blocks by using a criterion which results in the creation of large blobs: clone if the most massive blob found in the cluster at the end of the interval is more massive than the largest blob created before growth in the interval is begun. That is, either an existing blob grows, one or more blobs merge or a new blob is created which is more massive than any existing previous to growth in the interval.

B. Incremental cluster decomposition

The decision whether to clone depends on a knowledge of the mass of the largest blob in the cluster. It would be unacceptably inefficient to decompose the entire cluster into blobs starting from scratch each time we must make a cloning decision. Instead, we use an algorithm for cluster decomposition which allows us to incrementally decompose the cluster into blobs. At the end of an interval of n chemical shells of growth, we need only to consider the effect on the cluster decomposition of the sites and bonds we have added to the cluster during the interval. The algorithm, based on the algorithm of Ref. [13] for determining the cluster backbone, works as follows:

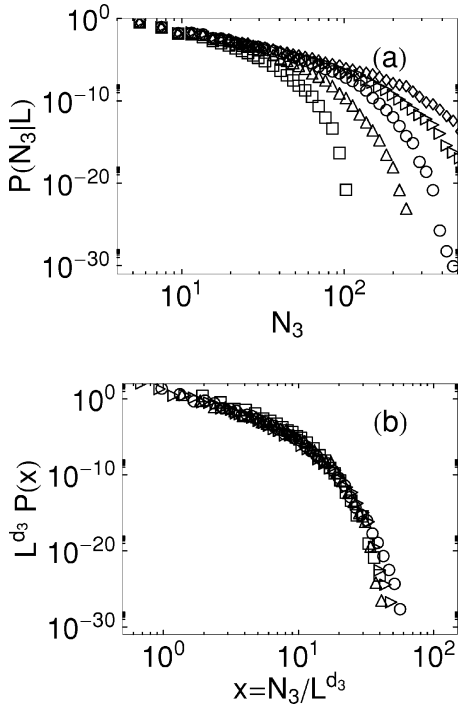

FIG. 3. $P(N_3|L)$, the distribution of 3-block mass N_3 for (from left to right) $L=8$, 16, 32, and 64 for the case of five dimensions. (a) uncollapsed, (b) collapsed using a value of $d_3=0.5$.

(i) During the growth of the cluster we identify ''loop sites.'' Loop sites are sites which are reached from two or more different growth sites simultaneously [7,13].

(ii) At the end of an interval of growth, we use the burning algorithm [7,13] to walk back from each loop site toward the origin of the cluster. When we reach a state during the walk when only one site is burning, all sites traversed so far compose a blob. If during the walk we hit an existing blob, that blob is incorporated into the blob associated with the loop site from which the walk started.

(iii) When we have exhausted all clones created at the end of an interval, we must restore the system to its state at the beginning of the interval. That is, we must (a) destroy all blobs created, (b) separate any blobs which were merged, and (c) reduce any blobs which grew during the interval back to their size at the beginning of the interval.

This is all accomplished by carefully maintaining the appropriate state information during the growth and cluster decomposition processes.

IV. RESULTS AND DISCUSSION

Using the methods described in the preceding section, we generate percolation clusters on hypercubic lattices for 2D, 3D, 4D, 5D, and 6D at their respective percolation thresholds [14,15]. To validate our use of the go with the winners approach and our incremental cluster decomposition technique, we compared our results in 2D and 3D with previous results [9,10] and found them to be consistent.

In Fig. 2(a), we plot $P(N_3|L)$, the distribution of 3-block mass N_3 in a system of size L for various L for $d=4$. In Fig. 2(b), we plot the same distributions collapsed using the estimated value $d_3=0.7$ which, visually, yields the best collapse. We show analogous plots for $d=5$ and $d=6$ in Figs. 3 and 4. Based on the value of d_3 which yields the best collapse, we estimate

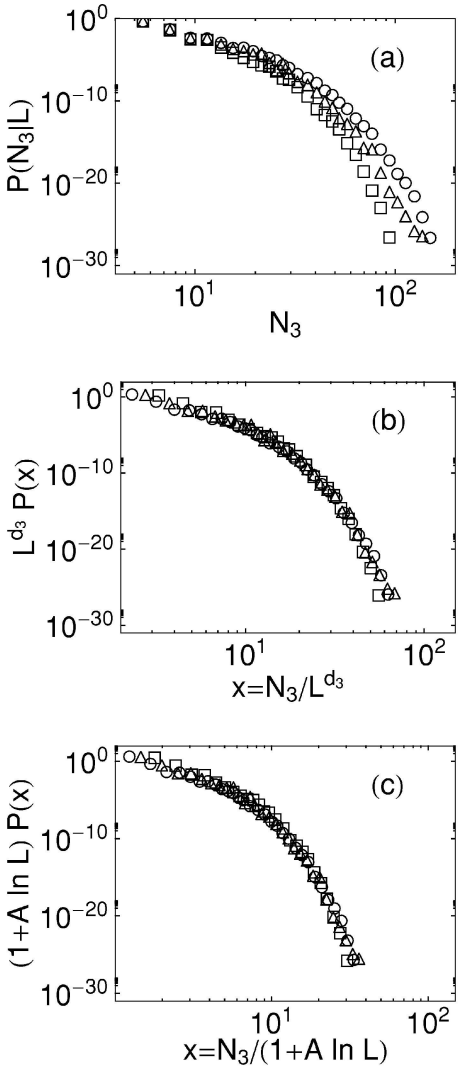

FIG. 4. $P(N_3|L)$, the distribution of 3-block mass N_3 for (from left to right) $L=8$, 16, and 32 for the case of six dimensions. (a) uncollapsed, (b) collapsed using a value of $d_3=0.25$, (c) collapsed assuming $N_3 \sim 1+A\ln L$ with $A=1.0$.

$$d_3=\begin{cases}0.7\pm0.2 & \text{(4D)}\\ 0.5\pm0.2 & \text{(5D).}\end{cases} \tag{1}$$

If we fit our results for 6D with a power law, then we find the best collapse is obtained for $d_3=0.25\pm0.2$. However, it it difficult to numerically distinguish between power-law scaling with a small exponent and logarithmic scaling. Hence, in Fig. 4(c), we also collapse the distributions for 6D assuming $d_3=0$ with logarithmic corrections to scaling

$$N_3 \sim 1+A\ln L \quad \text{(6D)}, \tag{2}$$

with $A=1.0$. The quality of the collapses for power-law scaling and logarithmic scaling seem to be comparable; however,

the facts that $d_3=0$ for the Cayley tree and that logarithmic corrections to scaling are common at the upper critical dimension favor the conclusion that $d_3=0$ for $d=6$.

Our results, despite their limited precision, indicate that d_3 is not independent of dimension below the upper critical dimension. The possibility of such independence in 2D and 3D is only a manifestation of the relatively low precision of the results and the relative closeness of the actual values of d_3 for $d=2$ and 3.

Finally, we make two observations:

(i) We note that the behavior of d_3 with dimension is qualitatively the opposite of the behavior of d_2, the blob fractal dimension, in the following sense: d_2 increases significantly between $d=2$ and $d=3$ but increases very slowly between $d=3$ and $d=6$ [2,16–19], while d_3 is slowly decreasing between $d=2$ and $d=3$ but then decreases significantly between $d=3$ and $d=6$.

(ii) Since $k=0$ corresponds to the entire system, which scales as L^d, we note that for $k=0$, 1, 2, and 3, the fractal dimensions d_k for 6D are 6, 4, 2, and 0, respectively, that is, a series of decreasing consecutive even integers.

ACKNOWLEDGMENTS

We thank Don Baker, Sergey Buldyrev, Shlomo Havlin, Greg Huber, and Sameet Sreenivasan for helpful discussions and NSF, BP, and Intevep for support.

[1] D. Ben-Avraham and S. Havlin, *Diffusion and Reactions in Fractals and Disordered Systems* (Cambridge University Press, Cambridge, 2000).
[2] D. Stauffer and A. Aharony, *Introduction to Percolation Theory* (Taylor & Francis, Philadelphia, 1994).
[3] A. Bunde and S. Havlin, in *Fractals and Disordered Systems*, edited by A. Bunde and S. Havlin (Springer-Verlag, New York, 1996).
[4] M. Sahimi, *Applications of Percolation Theory* (Taylor & Francis, London, 1994).
[5] Guido Caldarelli, Raffaella Frondoni, Andrea Gabrielli, Marco Montuori, Rebecca Retzlaff, and Carlo Ricotta, Europhys. Lett. **56**, 510 (2001).
[6] John Cardy, e-print cond-mat/0209638.
[7] H.J. Herrmann and H.E. Stanley, Phys. Rev. Lett. **53**, 1121 (1984); H.J. Herrmann, D.C. Hong, and H.E. Stanley, J. Phys. A **17**, L261 (1984).
[8] W.T. Tutte, *Graph Theory* (Cambridge University Press, Cambridge, 1984).
[9] G. Paul and H.E. Stanley, Phys. Rev. E **65**, 056126 (2002).
[10] J.L. Jacobsen and P. Zinn-Justin, e-print cond-mat/0207063.
[11] P. Grassberger, Comput. Phys. Commun. **147**, 64 (2002).
[12] P.L. Leath, Phys. Rev. B **14**, 5046 (1976).
[13] M. Porto, A. Bunde, S. Havlin, and H.E. Roman, Phys. Rev. E **56**, 1667 (1997).
[14] G. Paul, R.M. Ziff, and H.E. Stanley, Phys. Rev. E **64**, 026115 (2001).
[15] P. Grassberger, e-print cond-mat/0202144.
[16] J.L. Jacobsen and P. Zinn-Justin, J. Phys. A **35**, 2131 (2002).
[17] P. Grassberger, Physica A **262**, 251 (1999).
[18] C. Moukarzel, Int. J. Mod. Phys. C **8**, 887 (1998).
[19] H.K. Janssen and O. Stenull, Phys. Rev. E **61**, 4821 (2000).