Available online at www.sciencedirect.com

SciVerse ScienceDirect

Physics Procedia 34 (2012) 39 – 43

Physics Procedia

Exact enumeration of self-avoiding walks on percolation clusters

Niklas Fricke and Wolfhard Janke

Institut für Theoretische Physik and Centre for Theoretical Sciences (NTZ), Universität Leipzig, Postfach 100 920, 04009 Leipzig, Germany

Abstract

We study the scaling behavior of self-avoiding walks on critically dilute lattices. To this aim, we have developed a new enumeration technique, which is highly efficient for this particular problem. It makes use of the low connectivity and the self-similar nature of the critical percolation cluster. The problem can thus be factorized, and the exponential complexity that usually afflicts exact enumeration can be avoided. This allowed us to enumerate all conformations of walks of 1000 steps for a large random sample of percolation clusters in two dimensions. The scaling exponents could thus be determined with very high precision.

Keywords: exact enumeration, self-avoiding walks, percolation

1. INTRODUCTION

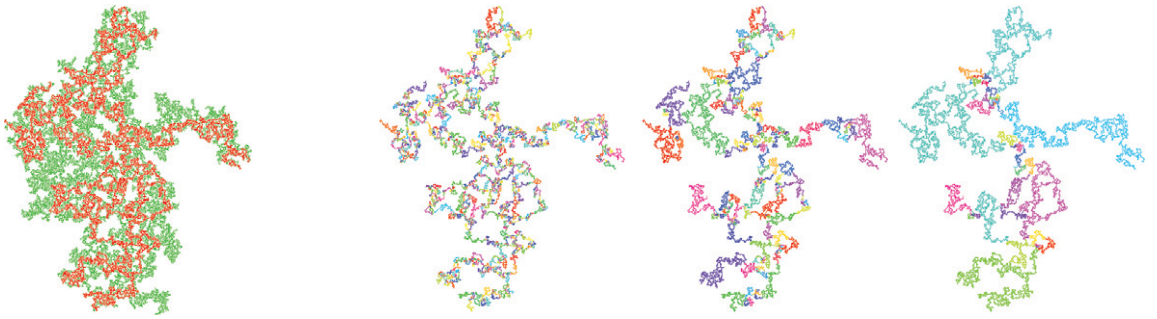
The self-avoiding walk (SAW) is among the most fundamental systems in statistical physics and a basic model for a polymer (see [1]). Although its formulation is very simple, its mathematical treatment is difficult and it has many interesting properties. Its asymptotic scaling behavior is described by non-trivial exponents, which are related to the critical exponents of the $O(n)$ symmetric spin system in the limit $n \to 0$. As these, they are universal, i.e., independent of system details. It is therefore not presumptuous to assume that the rather simplistic model of a SAW on a randomly diluted square lattice can capture some features of real polymers in disordered environments. Of particular interest is the case when the dilution is as strong as possible but an infinite (system spanning) cluster of connected sites can still exist – the percolation threshold. At that critical concentration, the fractal dimension of the incipient cluster is smaller than the embedding Euclidean dimension [2], and disorder rules over all length scales. It is assumed that the scaling laws in this case are similar to those for regular systems but have different exponents. One usually considers the scaling of the average number of SAW conformations Z and the mean squared end-to-end distance R^2 with the number of steps N:

$$[Z] \sim \mu^N N^{\gamma-1}, \quad \left[\left\langle R^2 \right\rangle\right] \sim N^{2\nu}. \tag{1}$$

Here the square brackets denote (quenched) averages over all disorder realizations, and the angle brackets denote the average over all SAW conformations. Unlike the exponents ν and γ, the connectivity constant μ is non-universal.

Starting in the late 1970s, this model has been studied using a variety of methods including real-space renormalization [3, 4], field theory [5, 6], Monte Carlo sampling [7, 8, 9], and exact enumeration [10, 11, 12]. However, each method had some serious draw-back. The analytical estimates were imprecise or uncertain, while for Monte Carlo sampling and especially for exact enumeration the manageable chain lengths were quite small. Although much insight has been gained through those studies, questions and doubts have remained. Since each method appeared to be at its limit, the interest in the problem eventually subsided.

doi:10.1016/j.phpro.2012.05.006



(a) (b)

Figure 1: Decomposition of the backbone of a critical percolation cluster. The first image (a) shows the cluster with its backbone marked in red. Sites that cannot be reached within a maximal number of steps (here 500) are removed. The following images (b) show different stages of the partitioning process, i.e., different levels of the hierarchy of nested blobs.

2. METHOD

The number of possible SAW conformations increases exponentially with the number of steps [see Eq. (1)] and so does the computational effort usually required to enumerate them. Fortunately, however, the very low connectivity and the self-similarity of a critical percolation cluster allows us to factorize the problem: Consider the simple case where two parts, A and B, of a cluster are only connected through one single site l. If we want to count the number of SAWs of length N, starting at some site a in part A and ending somewhere in B, $Z_{a\to B}[N]$, we can count the segments in A and B separately and then connect them:

$$Z_{a\to B}[N] = \sum_{n=0}^{N} Z_{a\to l}[n] \cdot Z_{l\to B}[N-n]. \tag{2}$$

By generalizing this idea and applying it recursively on all length scales, we can circumvent the exponential increase in computation time. The fragile nature of the critical cluster is best appreciated by looking at its backbone [Fig. 1 (a)], the fraction of sites that could support a lattice-spanning, self-avoiding loop. Factoring out parts with but one connection (i.e., parts that do not belong to the backbone) is trivial; therefore we can focus on the backbone only, whose connectedness is easier to visualize. Note, however, that the method works just as well on the complete cluster.

We start by partitioning the backbone into nested "blobs" that are only loosely connected to the rest. We thus establish a tree hierarchy, where each node corresponds to a blob. The idea is to enumerate the walk segments within a small blob at the bottom of the hierarchy and to then use this information when enumerating the walks through the larger blob containing it. This is repeated recursively until we have enumerated the walks through the root blob, which is simply the whole cluster.

To partition the cluster we use an amalgamation technique, which shall not be explained in detail here. It builds up the hierarchy from the bottom and proceeds by merging neighboring smaller blobs into a larger blob, which is then defined as their parent. This has to be done in such a way that the number of links of each blob will be as small as possible. Fig. 1 (b) shows snapshots taken during this process, corresponding to layers of the hierarchy.

Now we assume that we have a suitable decomposition of our critical cluster and want to determine the number of SAWs of a given length N starting from some fixed position. We begin by separating a blob that has no children (does not contain smaller blobs) with L links to the rest of the cluster. We add to it an extra site (the "parent site"), to which we connect all L links. Using the standard recursive backtracking routine (see [10]), we then generate all possible SAWs of length $n = 1, \ldots, N$ starting from the parent site. There is no self-avoidance on the parent site, so that the

walker may exit and re-enter the blob multiple times. When it does that, however, the ensemble of SAW segments outside of the blob to which the path could be connected (to produce a valid SAW conformation) changes. All paths that would fit to the same external segments are "topologically" equivalent; they connect the links in the same way. We call such an equivalence class a "state" of the blob. For each state s_i, we need a separate counter, $Z_{s_i}[n]$, storing the number of walks of each length n. If, for example, the blob has two links, l_1 and l_2, there may be four different states: an empty state (s_0) with no walk, one where the walk enters at l_1 (s_1) or at l_2 (s_2) and terminates inside, and one where the two links are connected by the walk (s_3). The number of possible states increases drastically with the number of links, which must therefore not be too large (not much larger than 10 in 2D). A blob that contains "children" can be dealt with in a similar manner: Again, we separate it from the rest and connect the links to a parent site. In addition, each area belonging to a child is replaced by a single "child site". These sites are also exempt from self-avoidance, but they change their state whenever the walker visits or leaves them. For example: If a child site with two links resides in state s_0 and is accessed via its link l_2, it will transit to s_2. When the walker leaves it via l_1 it will switch to s_3. (Note that accessing and leaving the child site are counted as only one step each, no matter the actual size of the child.) Again, we need separate containers to store the counted paths, but this time we need one for each distinct combination of states of both blob and children. Once the counting is done, we determine the number of walks of each length $n \le N$ for each state s_i of the blob by fitting the right pieces together: Consider, for example, a blob B with only one child C. Assume that we had previously determined the walks $Z^C_{s_j}[n]$ through C for each state s_j of C. For each combination of blob state s_i and child state s_j we now have counters $Z^{B,C}_{s_i,s_j}[n]$. The number of walks of length n for state s_i of the blob is then obtained as:

$$Z^B_{s_i}[n] = \sum_j \sum_{n'=0}^{n} Z^{B,C}_{s_i,s_j}[n'] \cdot Z^C_{s_j}[n-n']. \tag{3}$$

The generalization to more than one children is straightforward. When the paths for each state of the blob have been determined, all information concerning the children can be deleted. This procedure is repeated until the root blob is reached, whose only link is the SAW's spawning location. Determining the number of SAW conformations is the main part of the problem; the additional measurement of the mean end-to-end distance or other quantities that depend on the SAW's coordinates can be included (more or less) easily.

3. RESULTS

Using this new method, we successfully enumerated all conformations of SAWs of $N \le 10^3$ steps on 2×10^5 randomly chosen percolation clusters and on 8×10^4 cluster backbones. The exponent ν was then estimated using a linear fit of $\ln[\langle R^2 \rangle]$ vs. $\ln N$. To reduce finite-size effects, only values where $N > 500$ were included. We thus obtained

$$\nu = 0.7754(15) \tag{4}$$

on the full cluster and

$$\nu = 0.781(2) \tag{5}$$

on the backbone. The errors were estimated using a bootstrap resampling method (see [13]). Our results are in good agreement with those obtained by other authors, but more precise. In previous studies, it was often argued that the exponents for the backbone and the full cluster should be identical. We believe that the measured difference is significant, but this issue still needs more investigation.

Estimating the exponent γ for the scaling of $[Z]$ turned out to be more difficult. The reason are large deviations in the distribution of Z, which is in fact very close to a lognormal. The measured values for $Z(1000)$ range over almost 40 orders of magnitude. A small number of configurations therefore dominates the disorder statistics, and the mean value is probably underestimated. This explains the bumps and the sudden steepening of the slope at about $N = 70$ in Fig. 2 (a). We used a resampling technique to determine how far the values could be trusted: From our 2×10^5 configurations we picked 10^3 random subsamples S_i, each with 10^5 configurations. We averaged the estimators $\ln[Z]_{S_i,\mathrm{av}}$ over those subsamples and compared the result, $\overline{\ln[Z]_{S_i,\mathrm{av}}}$, with the estimator from the original sample, $\ln[Z]_{\mathrm{av}}$. As can be seen in Fig. 2 (b), the values start to deviate at around $N = 60$. Using only the data for

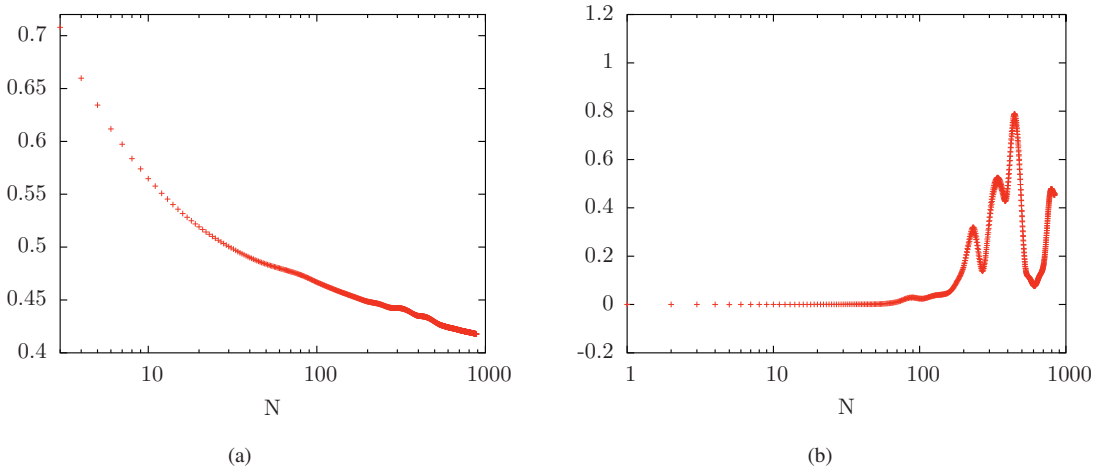


Figure 2: (a) Plot of $\ln [Z]_{\mathrm{av}}/N$ vs. N from which the exponent γ and the connectivity constant μ are estimated using a non-linear fit. (b) Bias estimator $\ln [Z]_{\mathrm{av}}/N - \overline{\ln [Z]_{S_i,\mathrm{av}}}/N$. $\ln [Z]_{S_i,\mathrm{av}}$ denotes an average over smaller random subsamples.

$10 < N \leq 60$, we obtained $\gamma = 1.35(3)$ and $\mu = 1.567(3)$. However, finite-size effects must be expected to be quite strong in this regime. It is unfortunate that the results for larger N could not be used here, but one could probably obtain more reliable estimates for γ by analyzing the distribution of Z.

The decomposition and enumeration for one cluster took on average roughly eight minutes on a 2 GHz processor. (Even the best MC method would need weeks to produce an unbiased estimator while the standard enumeration method would take over 10^{150} years). The maximum required RAM was about 3 GB, although 1 GB was sufficient for most clusters. Both, the required memory and the computation time depend on the fabric of the cluster and on the quality of the decomposition. Obviously, the method will fail (or take as long as the standard method) in some unfortunate cases, for example if all lattice sites happen to be occupied. However, these cases are extremely rare (we met none so far), so that the error introduced by resorting to stochastic methods to treat them will be negligible. The distribution of the computation time did not seem to have a fat tail, so we think that the average case complexity is polynomial even with those rare events included. We have successfully tried increasing the length to up to $N = 3000$ for a couple of configurations. While the numbers become evermore unwieldy, there does not seem to be a principle limitation so far. Our current implementation is only for two-dimensional lattices, but adaptation to three or more dimensions is in progress.

The method can also be used slightly above the percolation threshold ($p_c \approx 0.593$ in 2D), but only up to some finite length. Our preliminary trials suggest that for $p = 0.61$ we may still go up to $N = 500$ and up to $N = 180$ for $p = 0.63$. The method will also work for (finite) clusters below p_c, where the connectivity is lower still.

4. CONCLUSIONS

We have presented a novel method for the exact enumeration of self-avoiding walks on critically dilute lattices, which makes use of the structure of the percolation clusters. We could demonstrate that it is highly efficient and outperforms any other numerical method for this kind of problem. We have employed it to determine the scaling exponents of the end-to-end distance as $\nu = 0.7754(15)$ on the full cluster and as $\nu = 0.781(2)$ on the backbone. Due to large deviations in the distribution of Z, the exponent for the number of conformations could only be determined with lesser accuracy.

While our implementation is so far only for two dimensions, the method is in principle independent of the dimension. It could furthermore be adapted to similar problems. It would obviously work for other lattice types or other

types of percolation, such as bond percolation, invasion percolation or directed percolation. Other thin fractals might also be considered. In addition, the method could be extended to other polymer models that are based on the SAW, e.g., by adding bending stiffness (i.e., semi-flexible polymers) or nearest-neighbor interactions (i.e., θ-polymers).

ACKNOWLEDGEMENTS

We wish to thank the Deutsche Forschungsgemeinschaft (DFG) via FOR 877, the Graduate School BuildMoNa and the Deutsch-Französische Hochschule (DFH) for supporting this work.

[1] P.-G. de Gennes, *Scaling Concepts in Polymer Physics* (Cornell University Press, Ithaca, 1976).
[2] D. Stauffer and A. Aharony, *Introduction to Percolation Theory* (Taylor and Francis, London, 1992).
[3] Y. Meir and A. B. Harris, Phys. Rev. Lett. **63**, 2819 (1989).
[4] P. Le Doussal and J. Machta, J. Stat. Phys. **64**, 541 (1991).
[5] C. von Ferber, V. Blavatska, R. Folk, and Y. Holovatch, Phys. Rev. E **70**, 035104(R) (2004).
[6] H.K. Janssen and O. Stenull, Phys. Rev. E **75**, 020801(R) (2007).
[7] S. B. Lee, H. Nakanishi, and Y. Kim, Phys. Rev. B **39**, 9561 (1989).
[8] P. Grassberger, J. Phys. A 26, 1023 (1993).
[9] V. Blavatska and W. Janke, Europhys. Lett. **82**, 66006 (2008); Phys. Rev. Lett. **101**, 125701 (2008); J. Phys. A **42**, 015001 (2009).
[10] P. M. Lam, J. Phys. A **23**, L831 (1990).
[11] A. Ordemann, M. Porto, H. E. Roman, S. Havlin, and A. Bunde, Phys. Rev. E **61**, 6858 (2000).
[12] A. R. Singh, D. Giri, and S. Kumar, J. Chem. Phys. **131**, 065103 (2009).
[13] B. Efron, Ann. Statist. **7**, 1 (1979).