Annales UMCS Informatica AI 5 (2006) 87-91

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Reducing algorithm for percolation cluster analysis

Norbert Sendra*, Tomasz Gwizdałła, Jerzy Czerbniak

Department of Solid State Physics, University of Łódź, Pomorska 149/153, 90-236 Łódź, Poland

Abstract

The determination of percolation threshold is the substantial question for a lot of problems which may be modeled by the formalism of cellular automata. There is a set of well known algorithms which deal with this topic. All of them have some advantages and drawbacks connected to calculational or memory complexity. In our work we are going to present a new approach which we call "reducing algorithm". In our procedure we avoid the large memory occupancy which is usually connected to the algorithms aiming not only at confirming the existence of percolation cluster. Our approach makes it also possible to reduce time complexity by only single scan through the analyzed space. In the paper we present some basics of algorithm and the comparison of its effectiveness to other, mentioned earlier, ones.

1. Introduction

Very often physicists are unable to resolve real physical problems due to the complexity of real systems. Then they are trying to introduce simple models which simulate the behavior of real system and at the same time can be simply simulated by computer. The ideas of percolation and cellular automata belongs to such models.

The percolation concept in science has appeared for the first time in the early forties by Flory and Stockmayer [1,2]. Since then the model has been applied in a lot of disciplines, starting with geology (problem of water-flow in rock) and finishing with chemistry (problem of gelation).

We would like to present simple implementation of the method which has been applied in order to analyze effectively the percolation model.

2. Percolation

Let us consider through this paper the most widely known model of simple regular lattice. The classic model could be presented in two versions which may, in general, analyze within the frame of the same formalism:

*Corresponding author: *e-mail address*: nsendra@uni.lodz.pl

- Bond percolation – there exists n-dimensional regular grid of sites. Two adjacent sites may be connected with probability **p** (which we call the occupied bond) randomly and independently. Certainly, every bond can be unoccupied (there is no connection between sites) with probability **1-p**. If we are talking about vast systems, it corresponds to the case when **1-p** sites of the system have been dropped. We say that two sites are connected, if there is at least one path between these two sites consisting of only occupied bonds. The structure connected by bonds and limited by unoccupied bonds we call cluster. In big systems, if **p** is satisfactorily small we observe only clusters with a small number of bonds. In the vicinity of **p=1** almost the whole system will consist of occupied bonds, except for some unoccupied 'holes'. For proper **p** in the system there will appear a transition in the topological structure of random lattice from macroscopically unconnected structure into connected one. The value **p** for which such a connection appears we call percolation threshold $\mathbf{p_{tr}}$.
- Site percolation – similar to the bond percolation, sites of the system are occupied with probability **p** or unoccupied with probability **1-p**. The nearest neighbors are connected only if both sites are occupied. Cluster and percolation threshold concepts are defined in the system similarly.

3. Classic algorithms

Classic methods of percolation systems analysis consist of two stages. The first one, responsible for identifying elements (sites, bonds) of concrete clusters and the second one, responsible for collecting cluster parameters.

Existing algorithms, devoted to the first stage processing, may be classified as belonging to some general classes:

- scattering algorithms – by so called "system scattering", such algorithms are based on the idea of enlarging the cluster from arbitrary, initially found occupied site/bond [3],
- recurrent algorithms – process of discovering is based on recurrent function, which makes it possible to consider smaller structures, see e.g. [4],
- straying algorithms – the same idea as in the recurrent algorithm but recurrence function has been replaced by linear function [5],
- Newman-Ziff algorithm – very fast algorithm, where we avoid generation of a lot of random numbers [6].

Implementation of the second stage usually relies on routing through analyzed by the first stage system and collecting appropriate information.

4. Reducing algorithm

Reducing algorithm is built on the basis of cluster counting algorithm by Hoshen and Kopelman. The native HK algorithm is dedicated to recognize membership of each site (bond) in the percolation system to appropriate cluster [7].

We introduce modification of Hoshen-Kopelman algorithm by the location of clusters details on a doubly linked list and additionally by introduction of a singly linked list to keep information about merged clusters. The classic HK algorithm does not keep detailed information about cluster and the only information which exists in the model is the array with cluster numbers (there is also information about merged clusters).

The example of single passage through the system is presented in Fig. 1 for the site percolation analysis.

We start with the system where sites are occupied with probability **p** (upper left diagram). We will analyze the system from the extreme left-bottom site to the extreme right-top one. If we encounter a new occupied site which does not neighbor on the left and bottom with another occupied site we will create a new cluster by creating a new element of the doubly linked list. Each new cluster is identified by a unique number-cluster number. After analyzing each site is filled in by the appropriate cluster number.

If we encounter a site which is adjacent to the only one site already belonging to an existing cluster we add the considered site to this cluster.

If the investigated site has two neighbors and each of them is the element of a different cluster then we have to merge both clusters by updating the details of one of them and dropping the other one. Additionally in this case we have to create a new element in a simply linked list which includes information which cluster has been dropped.

The difference between the proposed algorithm and the Hoshen-Kopelman one lies in the fact that the goal of native HK algorithm is to discover membership of each site (bond) to appropriate cluster but the reducing algorithm during the analyzing process additionally collects cluster parameters. By such implementation we join both stages of the classic method and obviously improve the efficiency of algorithm.

Which way is the mentioned collecting realized in the system in? Simply, during every operation performed on the cluster (creating, merging and dropping) we are updating its parameters in the appropriate element of a double linked list.

The example of element of a doubly linked list is presented below:

```
struct element
   {
      int cluster_number;
```

```
    int cluster_elements;
    int cluster_energy;
    …
    element* previous;
    element* next;
};
```

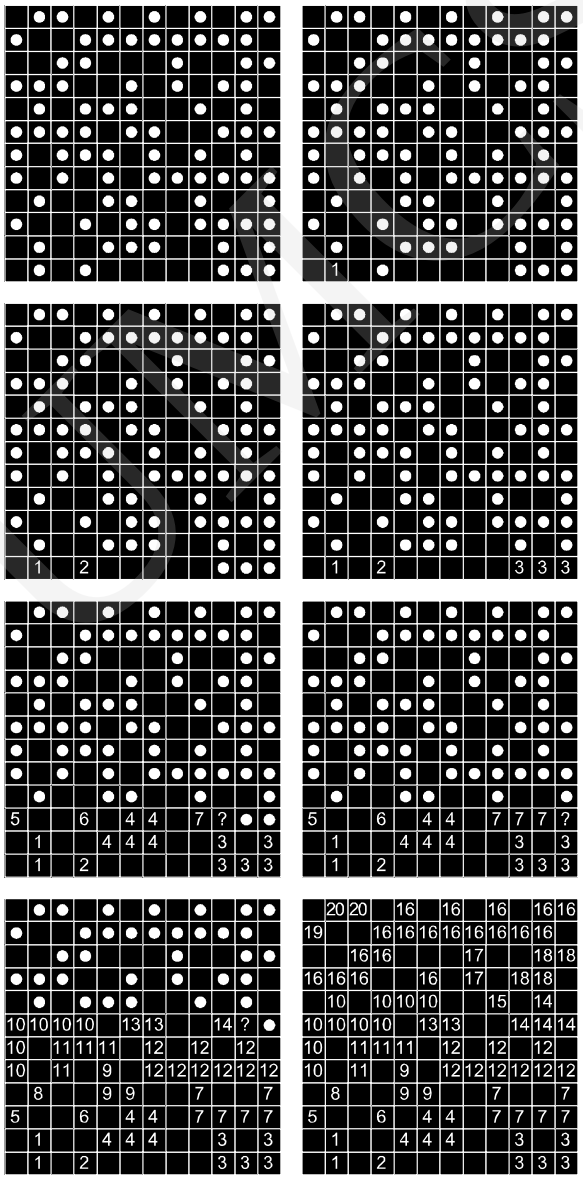

Fig. 1 Example of the single route through percolation system has been analyzed by reducing algorithm

Conclusions

The presented 'reducing algorithm' is more efficient than the classic ones (e.g. spreading, recurrent and straying) because it joins both stages of percolation system analysis.

The studied procedure displays some advantages in comparison to the Newman-Ziff algorithm as well. Firstly, the algorithm can penetrate only interesting regions for appropriate value of probability **p**. If we use the Newman-Ziff algorithm we have to start analysis with **p=0**. Reducing algorithm is more universal and can be used to analyze of dynamical systems i.e. systems where we are interested in studying of current state of system.

The presented approach may be very efficient from the point of view of parallel computing. The only additional information which has to be kept in the object containing cluster data are the positions of both boundary sites (bonds). The subsystems of the whole system are, in that approach, analyzed by individual units and then the results of such computation are sent to the central unit in the form of a doubly linked list where after global analysis of individual computation the final result has been determined.

In the presented paper we use the concept of 'reducing algorithm' as a modified Hoshen-Kopelman algorithm. The cause is connected to the fact that because we collecting all the parameters for all clusters existing in the sample during single passage through the system, we do not need to keep the states of all bonds in computer memory. All what we need is the state of two lines: those currently investigated and the former ones. Such property reduces the investigation of the d-dimensional system to the (d-1)-dimensional one.

References

[1] Flory D.J., J. Chem. Soc., 63 (1941) 3083.
[2] Stockmayer W.H., J. Am. Phys. 11 (1943) 45.
[3] Stauffer D., Aharony A., *Introduction to Percolation Theory*, Taylor & Francis, 2nd ed. London, (1992).
[4] Ballesteros H.G., Fernandez L.A., Martın-Mayor V., Munoz Sudupe V., Parisi G., Ruiz-Lorenzo J.J., Phys. Lett. B, 346 (1997).
[5] Sahimi M., *Applications of Percolation Theory*, Taylor & Francis, London, (1994).
[6] Newman M.E.J., Ziff R.M., Phys. Rev. E64 (2001) 016706.
[7] Hoshen J., Kopelman R., Phys. Rev. B, 14 (1976) 3438.