Cluster diversity and entropy on the percolation model: The lattice animal identification algorithm

I. J. Tsang, I. R. Tsang, and D. Van Dyck
VisionLab–Department of Physics, University of Antwerp-RUCA, Groenenborgerlaan 171, Antwerp B-2020, Belgium
(Received 4 May 2000)

We present an algorithm to identify and count different lattice animals (LA's) in the site-percolation model. This algorithm allows a definition of clusters based on the distinction of cluster shapes, in contrast with the well-known Hoshen-Kopelman algorithm, in which the clusters are differentiated by their sizes. It consists in coding each unit cell of a cluster according to the nearest neighbors (NN) and ordering the codes in a proper sequence. In this manner, a LA is represented by a specific code sequence. In addition, with some modification the algorithm is capable of differentiating between fixed and free LA's. The enhanced Hoshen-Kopelman algorithm [J. Hoshen, M. W. Berry, and K. S. Minser, Phys. Rev. E **56**, 1455 (1997)] is used to compose the set of NN code sequences of each cluster. Using Monte Carlo simulations on planar square lattices up to 2000×2000, we apply this algorithm to the percolation model. We calculate the cluster diversity and cluster entropy of the system, which leads to the determination of probabilities associated with the maximum of these functions. We show that these critical probabilities are associated with the percolation transition and with the complexity of the system.

PACS number(s): 02.70.Lq, 05.70.Jk, 64.60.−i

I. INTRODUCTION

The statistics of cluster size has been widely studied in various problems in statistical physics, such as percolation [1], fragmentation processes [2], cellular automata [3,4], and complex systems [5–7]. An important breakthrough in the computational analysis for cluster size statistics occurred with the introduction of the Hoshen-Kopelman (HK) algorithm [8]. This algorithm made possible the analysis of systems with very large lattice size, due to its linear time and memory space requirements as a function of lattice size. Its applications encompass diverse fields from basic science to technology [9]. However, the measurement studied with the HK algorithm is the cluster size, which does not convey information on the shape structure of the clusters.

Several problems in physics require a proper definition and recognition of cluster or in a more general sense pattern, which can be distinguished by some physical properties. We introduce an algorithm to identify and count clusters with different shape structures, defined as lattice animals. Lattice animals (LA's) are clusters of connected sites, distinguished from each other not only by their sizes but also by their shapes. The statistics and enumeration of n-cell LA's have been of much interest and various papers have been written on LA's in connection with percolation [1,10–12], branched polymer problems [13], the renormalization group [14], and self-organized criticality [15]. They are also called polyominoes [16,17]. It is usual to make a distinction between fixed and free LA's [11,17]. A free LA is considered similar to another if it can be derived by a symmetry operation, while in fixed LA's they are regarded as different. Figure 1 shows some examples of LA's. By the definition of a fixed LA there are eight different animals, while as free LA's they are all considered the same. The exact enumeration of LA's is a problem still not solved and much work has been done concerning this subject [17–20]. Algorithms that take up this challenge have been proposed [17,21] and succeeded in counting the number of LA's up to $n=25$ [12,22].

Recently, the enhanced Hoshen-Kopelman (EHK) algorithm [9] has been proposed. It is a natural extension of the original HK algorithm and can determine information not only on the cluster size but also on the structure of the clusters, such as, for example, the internal perimeter, radius of gyration, or spatial moments. Even though these parameters yield information on the shape structure of the clusters and

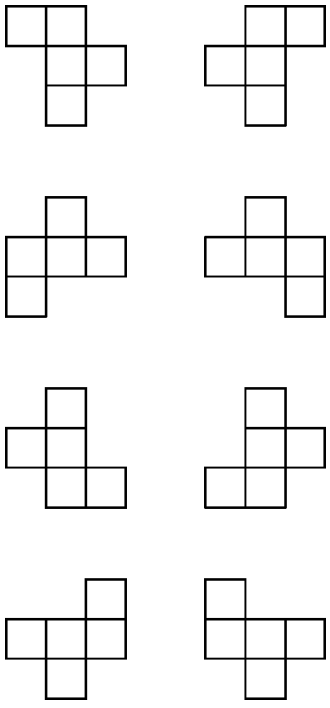

FIG. 1. Some configurations of five-cell lattice animals. From the definition of a fixed LA, there are eight different clusters, whereas, if defined as free LA's they are all considered the same.

are very important measurements in statistical physics, they do not differentiate between the LA's. The present algorithm deals with the problem of differentiating and counting the huge number of possible LA's on a lattice, so that a more refined definition and recognition of clusters (patterns) can be obtained.

Using the enhanced Hoshen-Kopelman algorithm the clusters are discriminated and each of the cells is coded according to the nearest-neighbor (NN) sites. A cluster is represented by a sequence of NN code, which is unique for each LA configuration. As a result, the identification of a LA is performed by comparison of each code in the proper sequence.

The structure of this paper is as follows. In the next section we analyze the LA structure and apply the EHK algorithm to compose the NN codes and to determine the proper sequence. The third section describes how the algorithm proceeds to identify the fixed or free LA's. In the fourth section we discuss cluster diversity and cluster entropy measurements. The fifth section presents some results of numerical simulations on a planar square lattice, where the algorithm is used to measure both the diversity and the entropy of the system. Also, we discuss the complexity of the algorithm in both computational time and memory requirement. Furthermore, in the sixth section we obtain the probabilities at which the maxima of these variables occur, taking into consideration the finite size effect. We relate these results to the percolation transition and complexity of the system. Finally, the last section discusses the limitations and further applications of the algorithm.

II. STRUCTURAL CHARACTERIZATION OF LATTICE ANIMALS

Each unit cell of a cluster is coded according to its nearest neighbors. A vector V is created where each component indicates a neighbor in one of the four directions. Therefore, $V=(n,e,s,w)$ where each letter, respectively, represents the presence or absence of a neighbor cell in the directions north, east, south, and west; hence we call V the NN vector. An order sequence for each cell is associated with the NN vectors. This order sequence must be generic and a natural way to implement it is to follow the order in which the cells of the cluster are scanned. In this way, the structural information of the clusters is represented by a set of NN vectors, which forms a distinct LA. For the identification of free LA's, it is necessary to take into account symmetry operations. This means an extra allocation of memory space to store the code sequences generated by each symmetry operation. Consequently, a higher efficiency in run time and memory requirement for the identification of fixed LA's is achieved in comparison to the identification of free LA's. Figure 2(a) shows an example of the order sequence and Fig. 2(b) the NN vectors associated with each cell of a cluster.

A. Nearest-neighbor code

It is important to verify if the proposed method is sufficient to distinguish uniquely all the possible LA's. Hence, to demonstrate that no two LA's will have the same set of NN vectors in the same sequence, let us try to perform a structural change on a LA without altering the order sequence and

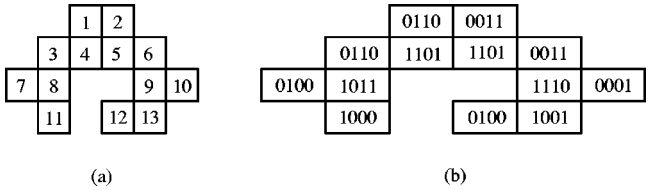

FIG. 2. (a) The order sequence for each cell of a cluster. (b) The NN vector (n,e,s,w) for each cell.

content of the NN vectors. Here, two LA's will be comparable only if they have the same number of cells. A change in the position of any cell $\mathcal{A}$ will imply a change in its ordering label and/or its NN vector content. Moreover, it also causes a change in at least one other cell $\mathcal{B}$ neighboring cell $\mathcal{A}$, since all the cells must be connected, causing a change in at least one other NN vector and/or its ordering label. Consequently, a change in any cluster cell will cause a change in at least one NN vector and/or its ordering label. This shows that any possible LA will have a distinct code sequence, and also that this method is not invariant to symmetry operations.

In practice, the NN vector can be coded as an integer where the first four bits are used to assign a possible nearest-neighbor configuration. As a result, there are 16 possible NN codes (see Table I). Now, consider an n-cell LA and let $s(n)$ be the total number of fixed LA's with size n. Not taking into account the code 0, because it represents the uninteresting case of a cluster with just one cell, an upper bound for possible LA configurations is $s(n)<15^n$, since using the above coding and ordering scheme there are 15 possible codes in n possible positions. This upper bound is in agreement with the upper bound $s(n)<(27/4)^n$ derived by Klarner [18], that is, $s(n)<(27/4)^n<15^n$. Clearly, our derivation resulting in a higher upper bound value is due to the fact that not all configurations generated by the combinations of 15 codes on an n-word represent a possible LA. Moreover, it serves to demonstrate that the present coding and ordering scheme is sufficient to characterize the LA's and also that a better coding scheme is still possible. Figure 3 shows two similar clusters and the NN code sequence generated by each one; note that even though the shapes of the clusters are similar the code sequences correctly differentiate them.

B. Algorithm implementation

For implementation of the algorithm to identify LA's, we used the EHK algorithm to discriminate the clusters and extract the NN vectors. The use of this algorithm is ideal since it is a generalization of the HK algorithm and thus a very efficient cluster identification algorithm. In addition, the HK

TABLE I. All possible NN codes. For each letter, 1 represents the presence while 0 represents the absence of a neighbor cell in the direction north, east, south, and west, respectively.

$(nesw)$	Code	$(nesw)$	Code	$(nesw)$	Code	$(nesw)$	Code
0000	0	0100	4	1000	8	1100	c
0001	1	0101	5	1001	9	1101	d
0010	2	0110	6	1010	a	1110	e
0011	3	0111	7	1011	b	1111	f

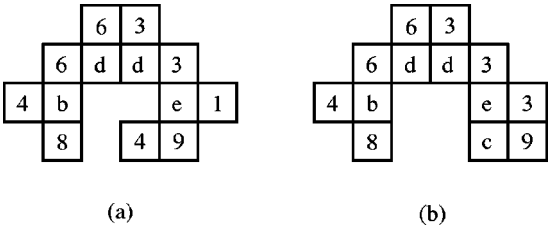

FIG. 3. Two similar cluster with distinct code sequences. (a) 636*dd*34*be***1**8**4**9; (b) 636*dd*34*be***3**8**c**9.

algorithm is a well-known and widely used algorithm in statistical physics (we assume that the reader has some knowledge of both the HK and EHK algorithms in what follows).

In the original HK algorithm a binary lattice is assumed to be occupied by two types (A and B) of molecule, which are randomly distributed with probability p and $1-p$. The HK algorithm proceeds by assigning to each site occupied by a molecule with concentration p a cluster label m_t^α, where α denotes a specific cluster. In this way multiple labels, represented by the set $\{m_1^\alpha, m_2^\alpha, \ldots, m_s^\alpha, \ldots, m_t^\alpha, \ldots\}$, can be assigned to a single cluster. However, only one of this set is the *proper* cluster label, which is represented by m_s^α. From the *pseudo* labels m_t^α, the proper cluster label can be determined by the set of integers $\{N(m_1^\alpha), N(m_2^\alpha), \ldots, N(m_s^\alpha), \ldots, N(m_t^\alpha), \ldots\}$. In this expression, $N(m_s^\alpha)$ denotes the size of the cluster, while the other labels are negative integers linking the labels m_t^α with the proper label m_s^α. The relation between these labels can be found through the following set of equations:

$$m_r^\alpha = -N(m_t^\alpha), \quad m_q^\alpha = -N(m_r^\alpha), \ldots, \quad m_s^\alpha = -N(m_p^\alpha), \tag{1}$$

where the solution follows from left to right.

In addition, the EHK algorithm extends the HK algorithm since one can calculate general cluster properties $F^{(1)}(m_s^\alpha), F^{(2)}(m_s^\alpha), \ldots$ defined by quantities $f^{(1)}(i), f^{(2)}(i), \ldots$, respectively, where $f^{(n)}(i)$ represents some properties of the ith lattice site. In the present case, $f(i)$ represents the NN code or the nonscalar quantity $f(i) = (n,e,s,w)$. We drop the index (n) since we are interested in just one cluster site property, i.e., the NN codes. In both algorithms a very important routine CLASSIFY (see Ref. [8], [9] for details of this routine) determines the proper cluster labels and coalesces different sites or clusters when they are found to form a single larger cluster. As a result only a single pass over the lattice is required to determine the clusters, the NN codes, and the order sequence of these codes.

In a two-dimensional lattice the HK algorithm requires just a single line to store the labels m_t^α or m_s^α assigned to each occupied site. However, the information of these labels, denoted by $m^\alpha(i)$, and the cluster properties $f(i)$, both corresponding to the ith site, are necessary for the identification of lattice animals. Thus, $m^\alpha(i)$ and $f(i)$ need to be stored in a structure representing the lattice topology. This can be done either using matrices, which will require memory proportional to the lattice size, or by any other clever data structure, which may be proportional to the number of occupied sites.

1	1	1		2		3	
1		1			4	3	3
	5			6	3		
7	5	5					8
	5		9	9		10	8
11		12		9		8	
11			13				14
11		15	13		16	16	14

FIG. 4. An 8×8 lattice, where the occupied sites are represented by labels m_t^α. These label numbers, associated with the cluster cells, are due to the multiple labeling technique used in the HK algorithm.

The next step is to arrange this information so that comparison of the NN code sequence of each cluster can be done efficiently. The labels $m^\alpha(i)$ indicate to which cluster a specific cell belongs and they are used to identify each cell of a cluster (see Fig. 4), while the property $f(i)$ contains the NN code of each cell and is used to compose the NN code sequence that uniquely characterizes each LA (see Fig. 5). Consequently, a list of NN code sequences, each one representing a cluster, is produced and they can be compared with each other. The next section shows the details of how the algorithm proceeds from here to the final identification of fixed or free LA's.

III. IDENTIFICATION OF LATTICE ANIMALS

Since there are no known simple measurements to define the LA's, they were represented by a series of NN codes in a proper order sequence. In this way, comparison of the code sequences of each LA will determine if two LA's are distinct or not. The order sequence of the codes must obey a general rule and an efficient way to implement it is to follow the scanning sequence used by the HK algorithm, i.e., top to bottom and left to right. From the definition of LA's, it is only necessary to compare clusters of the same size. Nevertheless, in order to identify free LA's, we need not only a proper sequence but also a transformation of the codes with respect to every symmetry operation.

6	5	3		0		2	
8		8			6	d	1
	2			4	9		
4	f	1					2
	8		4	3		6	9
2		0		8		8	
a			2				2
8		4	9		4	5	9

FIG. 5. The NN codes for each cluster cell. The codes are derived from the NN vectors, which indicate the presence or absence of sites in the directions north, east, south, and west. See Table I.

$$
\begin{array}{l}
\quad 0\ 1\ 2\ 3\ \ 4\ \ 5\ \ 6\ \ \ 7\ \ 8\ 9\ 10\ 11\ 12\ 13\ 14\ \ 15\ \ \ 16 \\
\quad \updownarrow\updownarrow\updownarrow\updownarrow\ \updownarrow\ \updownarrow\ \updownarrow\ \ \updownarrow\ \updownarrow\updownarrow\ \updownarrow\ \ \updownarrow\ \updownarrow\ \updownarrow\ \updownarrow\ \ \updownarrow\ \ \ \updownarrow \\
N(m_t^\alpha) = [0, 5, 1, 6, -3, 5, -3, -5, 4, 3, -8,\ 3\ ,\ 1\ ,\ 3\ ,\ 4\ , -13, -14, \ldots]
\end{array}
$$

FIG. 6. Due to the multiple labeling technique, $N(m_t^\alpha)$ contains either the size of the cluster or a negative value pointing to the proper label. The index above each value represents the label m_t^α.

A. Fixed lattice animals

The multiple labeling technique relies on assigning different labels m_t^α to the cells of a cluster. While a negative index in $N(m_t^\alpha)$ indicates a pointer to the proper cluster label, a positive index determines the proper label m_s^α and the size of the cluster. Note that the labels m_t^α are not stored on the computer; they represent the position on array $N(m_t^\alpha)$ where the negative pointer or the cluster size is stored (see Fig. 6).

After applying the EHK algorithm the next step is to compose the NN code sequences representing each cluster, so that they can be systematically analyzed. Thus, an index indicating the labels m_t^α is coupled to each $N(m_t^\alpha)$. The set containing $N(m_t^\alpha)$ is sorted; nevertheless the index m_t^α still preserves the correct link between the pseudo labels m_t^α and the proper label m_s^α.

Once the set $N(m_t^\alpha)$ is sorted, the classification of clusters with desired properties, such as clusters with a specific size, or the counting of cluster diversity can be done without difficulty. As our objective here is to determine the diversity of LA's on a lattice, it is not necessary to keep the clusters of size 1 and the clusters with a size that appears just once, since they will contribute a unit increase in the diversity measure (see Fig. 7). Consequently, a *new proper* cluster label r^α is defined and is related to the labels m_t^α by the following set:

$$\{K(m_1^\alpha), K(m_2^\alpha), \ldots, K(m_s^\alpha), \ldots, K(m_t^\alpha), \ldots\}. \tag{2}$$

To assign the relationship between $K(m_t^\alpha)$ and the new proper label r^α, we follow the sorted set of $N(m_t^\alpha)$. For the positive integers (proper label) that do not have size 1 or that appear more than once, set $K(m_t^\alpha) = r^\alpha$, while for the negative integers $K(m_t^\alpha) = K(-N(m_t^\alpha))$. Hence, the clusters are relabeled according to a specific criterion. It is necessary to follow the positive $N(m_t^\alpha)$ first, since they represent a proper label, so that the pseudo labels are correctly linked to the new proper label r^α. If the previous sort is performed in such a fashion that the rearrangement of its elements is in descending numerical order, verifying the positive integers in $N(m_t^\alpha)$ before the negatives becomes an easy task. This will also cause the clusters to be classified in descending order according to their size, which is ideal for comparison of the

$$
\begin{array}{l}
\quad 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 0\ 0\ 0 \quad\quad 0\ \ 0\ \ 2\ \ 3\ \ 7\ \ 4 \\
\quad \uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow \quad\quad \uparrow\ \ \uparrow\ \ \uparrow\ \ \uparrow\ \ \uparrow\ \ \uparrow \\
\quad 3\ 1\ 5\ 8\ 14\ 9\ 11\ 13\ 2\ 12\ 0 \quad\quad 4\ \ 6\ \ 7\ \ 10\ \ 15\ \ 16 \\
\quad \updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow \quad\quad \updownarrow\ \ \updownarrow\ \ \updownarrow\ \ \updownarrow\ \ \updownarrow\ \ \updownarrow \\
N(m_t^\alpha) = [6, 5, 5, 4,\ 4\ , 3,\ 3\ ,\ 3\ , 1,\ 1\ , 0, \ldots, -3, -3, -5, -8, -13, -14]
\end{array}
$$

FIG. 7. After sorting $N(m_t^\alpha)$ in descending numerical order, the coupled index m_t^α indicates the correct link between the pseudo labels and the proper label. From the bottom up, the third row indicates the new proper label after ignoring the clusters of size 1 and the clusters whose size appears just once.

$$
\begin{array}{l}
\quad 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13\ 14\ 15\ 16 \\
\quad \updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\ \updownarrow\ \ \updownarrow\ \ \updownarrow\ \ \updownarrow\ \ \updownarrow\ \ \updownarrow \\
K(m_t^\alpha) = [0, 1, 0, 0, 0, 2, 0, 2, 3, 5,\ 3\ ,\ 6\ ,\ 0\ ,\ 7\ ,\ 4\ ,\ 7\ ,\ 4\ ]
\end{array}
$$

FIG. 8. $K(m_t^\alpha)$ contains the new proper label r^α. The index above each value represents the label m_t^α.

NN code sequence of each cluster. Figure 8 shows the values of $K(m_t^\alpha)$ assuming the example given from the previous figures. For comparison of the codes that compose the LA's, each NN code sequence is arranged according to its cluster size (see Fig. 9). The following sets represent the NN code sequences:

$$\{C_1^\alpha, C_2^\alpha, \ldots, C_n^\alpha\},$$

$$\{C_1^\beta, C_2^\beta, \ldots, C_n^\beta\},$$

$$\{C_1^\gamma, C_2^\gamma, \ldots, C_n^\gamma\},$$

$$\vdots$$

where α, β, and γ represent different clusters and C_n is the NN code of the nth cell.

A second scan on the lattice is necessary to extract the set of NN code sequences. As the array containing the information of the labels (Fig. 4) is scanned, an occupied site with label l is assigned to the cluster α by the relation $\alpha = K(l)$. Let $n(\alpha)$ be the site number of cluster α, attributed as they are scanned. The NN code sequences of a lattice animal are arranged by assigning the code $f(i)$ to the position $n(\alpha)$ of C_n^α. The NN code sequences are placed in descending order according to cluster size. Hence, it is a straightforward task to compare the NN code sequences.

B. Free lattice animals

In the case of free LA identification one must take into account the symmetry operations. The NN code sequence equivalent to each symmetry requires scanning the image according to the rotation and reflection operations. Furthermore, the NN codes must be transformed into an equivalent code with respect to these operations.

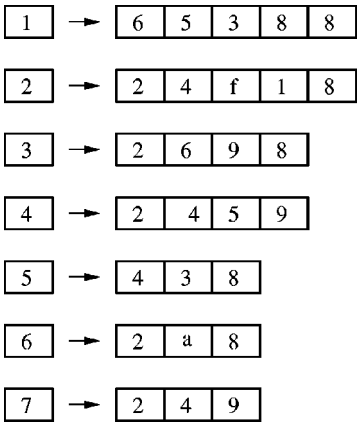

FIG. 9. The NN code sequence for each cluster according to the cluster size. Note that clusters with size 1 and cluster sizes that appear just once are not present.

TABLE II. Changes of coordinate due to symmetry operators. The angles represent counterclockwise rotations. The row at 0° represents the original coordinate. For each angle, the first row represents the rotational operator and the second row a reflection operator.

0°	north	east	south	west
	north	west	south	east
90°	east	south	west	north
	east	north	west	south
180°	south	west	north	east
	south	east	north	west
270°	west	north	east	south
	west	south	east	north

The scanning sequence establishes the orientation in which the lattice is observed. At first, a 0° orientation is associated with scanning the lattice from left to right and top to bottom, while the NN vector coordinates (north, east, south, and west) are determined for each cell of a cluster. Assuming that a LA is presented with a 90° counterclockwise rotation, to find an equivalent code in 0° one must scan this rotated LA from bottom to top and left to right, and the vector coordinate must be changed according to Table II. If we fix on a specific pixel and rotate the lattice, it is quite intuitive that a rotation will represent an exchange of coordinates on the NN vectors. In the same way, reflection operations represent an exchange between the elements north and south or between east and west.

Consequently, for identification of free LA's we rescan the label lattice (Fig. 4) in the directions equivalent to the symmetry operations and transform the NN codes according to the transformation matrix A_{ls} (see Fig. 10). In matrix A_{ls} the rows l represent the labels in comparison to the 0° orientation, while the columns s represent the symmetry operations. Hence, the second column represents a counterclockwise rotation of 90°, followed by 180° and 270° rotations and the equivalent reflection operations. In practice when comparing two LA's, $\mathcal{X}$ and $\mathcal{Y}$, if $\mathcal{Y}$ is assumed to be a 90° clockwise rotation of $\mathcal{X}$, one must apply an inverse operation (counterclockwise rotation) on the NN codes of $\mathcal{Y}$, so that they will correctly match.

Finally, the set containing the LA codes will also contain the equivalent codes in the order sequence according to every symmetry operation:

$$\{C_1^{\alpha}(s),C_2^{\alpha}(s),\ldots,C_n^{\alpha}(s)\},$$

$$\{C_1^{\beta}(s),C_2^{\beta}(s),\ldots,C_n^{\beta}(s)\},$$

$$\{C_1^{\gamma}(s),C_2^{\gamma}(s),\ldots,C_n^{\gamma}(s)\},$$

$$\vdots.$$

In the above set of integers, s represents the different symmetry operations. The identification of free LA's follows by comparing a sequence of NN codes $\{C_1^{\alpha}(1),C_2^{\alpha}(1),\ldots,C_n^{\alpha}(1)\}$, where $s=1$ is equivalent to the 0° orientation, with all the other sequences $\{C_1^{\beta}(s),C_2^{\beta}(s),\ldots,C_n^{\beta}(s)\}$, which represent all symmetries.

$$A_{ls} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 4 & 8 & 4 & 2 & 1 & 8 \\
2 & 4 & 8 & 1 & 2 & 1 & 8 & 4 \\
3 & 6 & c & 9 & 6 & 3 & 9 & c \\
4 & 8 & 1 & 2 & 1 & 8 & 4 & 2 \\
5 & a & 5 & a & 5 & a & 5 & a \\
6 & c & 9 & 3 & 3 & 9 & c & 6 \\
7 & e & d & b & 7 & b & d & e \\
8 & 1 & 2 & 4 & 8 & 4 & 2 & 1 \\
9 & 3 & 6 & c & c & 6 & 3 & 9 \\
a & 5 & a & 5 & a & 5 & a & 5 \\
b & 7 & e & d & e & 7 & b & d \\
c & 9 & 3 & 6 & 9 & c & 6 & 3 \\
d & b & 7 & e & d & e & 7 & b \\
e & d & b & 7 & b & d & e & 7 \\
f & f & f & f & f & f & f & f
\end{bmatrix}$$

$$= \begin{bmatrix}
0000 & 0000 & 0000 & 0000 & 0000 & 0000 & 0000 & 0000 \\
0001 & 0010 & 0100 & 1000 & 0100 & 0010 & 0001 & 1000 \\
0010 & 0100 & 1000 & 0001 & 0010 & 0001 & 1000 & 0100 \\
0011 & 0110 & 1100 & 1001 & 0110 & 0011 & 1001 & 1100 \\
0100 & 1000 & 0001 & 0010 & 0001 & 1000 & 0100 & 0010 \\
0101 & 1010 & 0101 & 1010 & 0101 & 1010 & 0101 & 1010 \\
0110 & 1100 & 1001 & 0011 & 0011 & 1001 & 1100 & 0110 \\
0111 & 1110 & 1101 & 1011 & 0111 & 1011 & 1101 & 1110 \\
1000 & 0001 & 0010 & 0100 & 1000 & 0100 & 0010 & 0001 \\
1001 & 0011 & 0110 & 1100 & 1100 & 0110 & 0011 & 1001 \\
1010 & 0101 & 1010 & 0101 & 1010 & 0101 & 1010 & 0101 \\
1011 & 0111 & 1110 & 1101 & 1110 & 0111 & 1011 & 1101 \\
1100 & 1001 & 0011 & 0110 & 1001 & 1100 & 0110 & 0011 \\
1101 & 1011 & 0111 & 1110 & 1101 & 1110 & 0111 & 1011 \\
1110 & 1101 & 1011 & 0111 & 1011 & 1101 & 1110 & 0111 \\
1111 & 1111 & 1111 & 1111 & 1111 & 1111 & 1111 & 1111
\end{bmatrix}$$

FIG. 10. Transformation matrix for the NN codes due to symmetry operations. The second matrix shows the equivalent binary form. The transformation can be seen as a change in the coordinates of vector $V=(n,e,s,w)$ following the scheme in Table II.

In this case, α and β represent clusters of the same size, since from the definition of a LA it is not necessary to compare clusters of different sizes.

IV. DIVERSITY AND ENTROPY

The macroscopic properties of a system are mostly determined by its microstructure. In percolation the microstructure, i.e., the clusters formed by this process, have been thoroughly studied and analyzed [1]. In addition, macroscopic properties associated with cluster size, such as diversity and entropy, have recently been investigated [7]. By applying the LA identification algorithm described in the last section, we were able to enrich the analysis of cluster diversity and cluster entropy in the percolation model.

In the percolation model a site or a bond is chosen at random and occupied with a probability p. For the site-percolation model, two occupied sites having one side in common are called nearest-neighbor sites, and the group of sites connected by nearest neighbors is defined as a cluster. A particular and important phenomenon in this model occurs when a cluster extends from one edge of the lattice to the opposite edge. This cluster is called the spanning cluster, since it percolates or spans the system. At this point the system is said to pass through a geometrical phase transition where the order parameter is the probability that an occupied site belongs to the spanning cluster. In this paper, we limit our investigation to the site-percolation model on a two-dimensional Euclidean lattice. Further study on bond perco-

lation and different lattice structures will be presented in future work.

Diversity is an important characteristic of nature and has been used to describe the complexity of different systems [4,5,7]. Here, the term ''complexity'' has been associated with the diversity in the length scales that the clusters can assume in the model. Moreover, diversity can refer to different properties of the system, such as the size or configuration assumed by the clusters. Thus, ''cluster diversity'' is defined as the differentiation of clusters with respect to their size or LA's. The mathematical definitions of the cluster size and fixed and free LA diversity are given by

$$D_s(p)=\sum_s \Theta[N(s,p)], \tag{3}$$

$$D_{fx}(p)=\sum_{fx} \Theta[N(fx,p)], \tag{4}$$

$$D_{fr}(p)=\sum_{fr} \Theta[N(fr,p)], \tag{5}$$

where p is the probability of occupying a site, and $\Theta(x)$ is the Heaviside function defined as $\Theta(x)=1$ if $x>0$ and $\Theta(x)=0$ otherwise. Also, $N(s,p)$ represents the number of clusters of size s and $N(fx,p)$ and $N(fr,p)$ the number of clusters with fixed and free LA configuration, respectively.

Entropy is a fundamental concept in physics. It is related to the information content and order/disorder of a system. In the present paper, we consider ''cluster entropy,'' which is defined using the probability that an occupied site belongs to a cluster of size s or is part of a specific free or fixed LA. This definition can be associated with the information entropy and the configurational or local porosity entropy defined in [23]. However, these entropies are based on the information content of a sliding $m\times m$ square region, thus being basically a local or short-range measurement. Conversely, our definition of entropy depends on the structure of the clusters, which are not limited to a local region and are capable of spanning the whole lattice. The mathematical definitions of cluster entropy are given by

$$H_s(p)=-\sum_s w_s(p)\ln w_s(p), \tag{6}$$

$$H_{fx}(p)=-\sum_{fx} w_{fx}(p)\ln w_{fx}(p), \tag{7}$$

$$H_{fr}(p)=-\sum_{fr} w_{fr}(p)\ln w_{fr}(p), \tag{8}$$

where p is the probability of occupation of the lattice, and $w_s(p)$ represents the probability that an arbitrary occupied site belongs to a cluster containing s sites. Similarly, $w_{fx}(p)$ and $w_{fr}(p)$ represent the probabilities that an arbitrary occupied site belongs to a fixed and a free LA configuration, respectively. Cluster entropy is related to the possible length scale or shape configuration that the clusters can assume in the system. Thus, its physical significance can be analyzed by observing its behavior with respect to the probability of occupation p. Furthermore, this measurement is intrinsically related to the notion of diversity, since the probability that an occupied site belongs to a certain cluster depends on the appearance of distinct clusters.

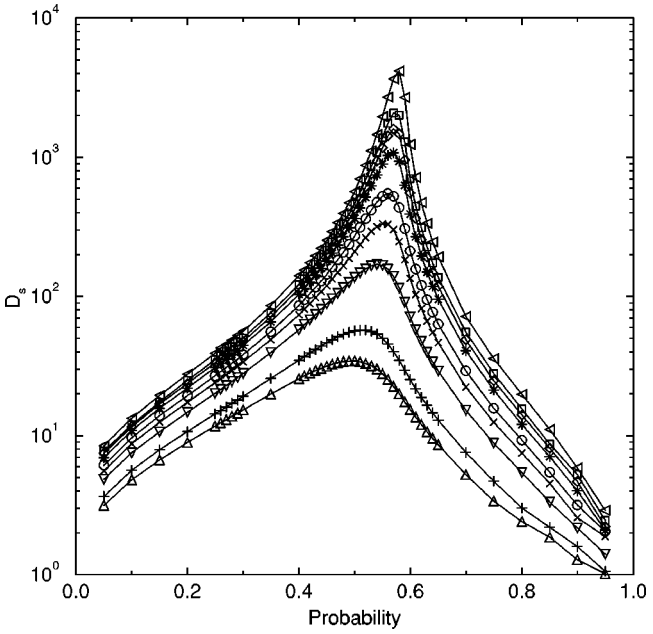

FIG. 11. Cluster size diversity versus the probability of occupation p. The curves represent $L=60$ (△), 100 (+), 300 (▽), 600 (×), 1000 (○), 2000 (★), 3000 (◇), 4000 (□), and 8000 (◁).

V. NUMERICAL SIMULATIONS

We performed Monte Carlo simulations on square lattices with sizes $L=60$, 100, 300, 600, 1000, and 2000, and averages taken on 6000, 5000, 2000, 500, 300, and 200 experiments, respectively. The lattices were randomly occupied with probabilities p ranging from 0.05 to 0.95 with steps of 0.01 between $p=0.25$ and 0.30 and $p=0.40$ and 0.65, and steps of 0.05 in the other regions. In the case of free LA's the simulations were performed until $L=1000$. However, for the cluster size case we extended the simulations up to $L=3000$, 4000, and 8000, with averages over 100, 50, and 20 experiments, respectively. The variables D_s, D_{fx}, and D_{fr} were measured as functions of both L and p.

A. Simulation results

Figure 11 shows the behavior of cluster size diversity as a function of the probability of occupation for different values of L. A tuning effect of the cluster size diversity by the parameters L and p can be observed. Figures 12 and 13 show the fixed and free LA diversity, as a function of the probability of occupation. These two plots are similar to each other and follow a bell shaped curve centered at the probability of maximum LA diversity. In addition, Fig. 14 shows the behavior of the cluster size entropy. One can see that as the probability of occupation increases the entropy increases up to the percolation threshold, where the dominance of the spanning cluster causes a sharp decrease in the entropy of the system. Figures 15 and 16 show H_{fx} and H_{fr} as functions of p. Even though these entropies have a slightly different

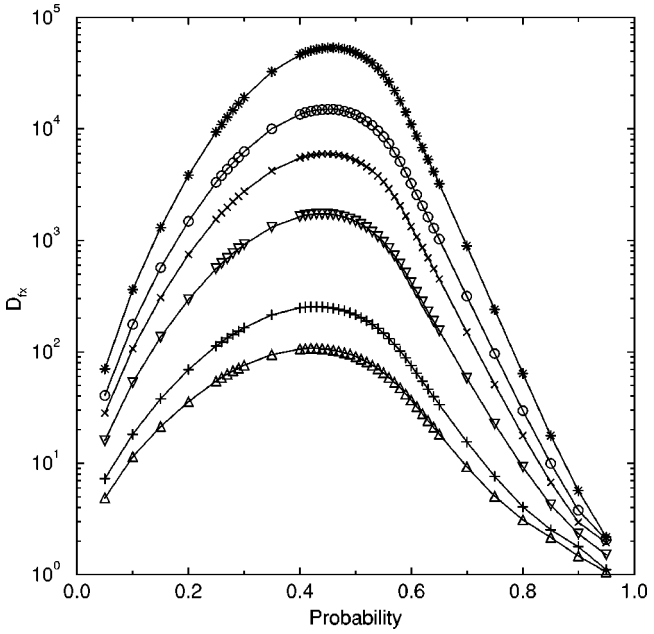

FIG. 12. A log-linear plot of D_{fx} as a function of p. The curves represent $L=60$ ($\triangle$), 100 ($+$), 300(∇), 600 ($\times$), 1000 ($\bigcirc$), and 2000 ($\bigstar$).

shape in comparison to the H_s curves, they also present a sharp drop at the point where a single cluster dominates the lattice.

In respect to both cluster diversity and cluster entropy the behavior of fixed and free LA's is similar. In these simulations this is to be expected, since the difference between their definitions is just in the symmetry aspect of the cluster configurations. Conversely, the behavior of these measurements when comparing the cluster size with LA diversity is qualitatively different. The cluster size diversity curves present a peak at the maximum of the functions, while in the LA diversity these curves are smoother. Analyzing the skewness of these curves, we find that $D_s(p)$ presents a leptokurtic distribution, in contrast with $D_{fx}(p)$ and $D_{fr}(p)$, which have mesokurtic distributions. This distinction is caused by the enormous increase in configurational possibilities of LA's with increasing cluster size. The characteristics of the diversity curves are reflected in the entropy functions, since by definition cluster diversity and cluster entropy are intrinsically related, as mentioned in the last section.

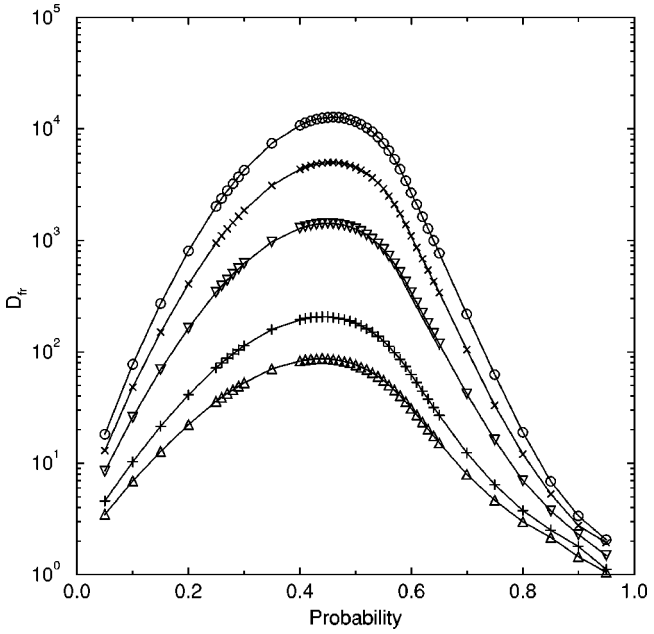

FIG. 13. Similar to Fig. 12, this figure shows D_{fr} versus p. The curves represent $L=60<(\triangle)$, 100 ($+$), 300 (∇), 600 ($\times$), and 1000 ($\bigcirc$).

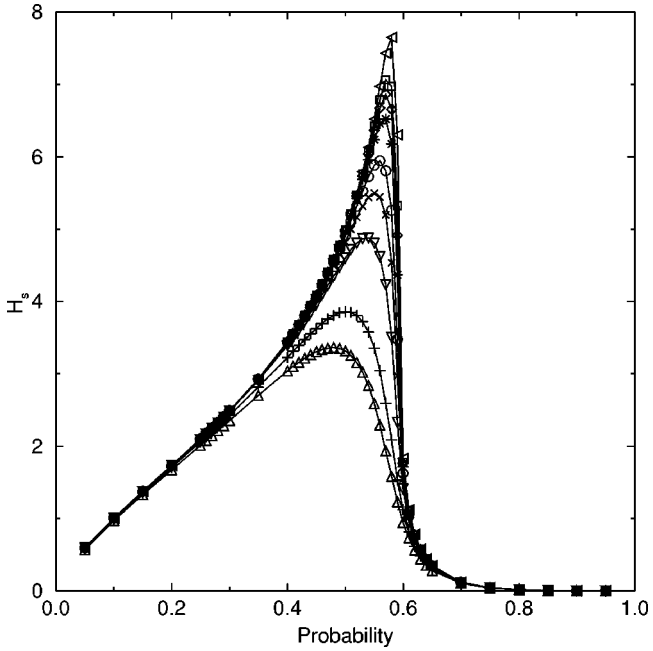

FIG. 14. H_s versus the probability of occupation p. The curves represent the same lattice sizes as in Fig. 11.

B. Computational complexity

Using a Pentium 133 computer running the LINUX operating system, we measured the average computer central pro-

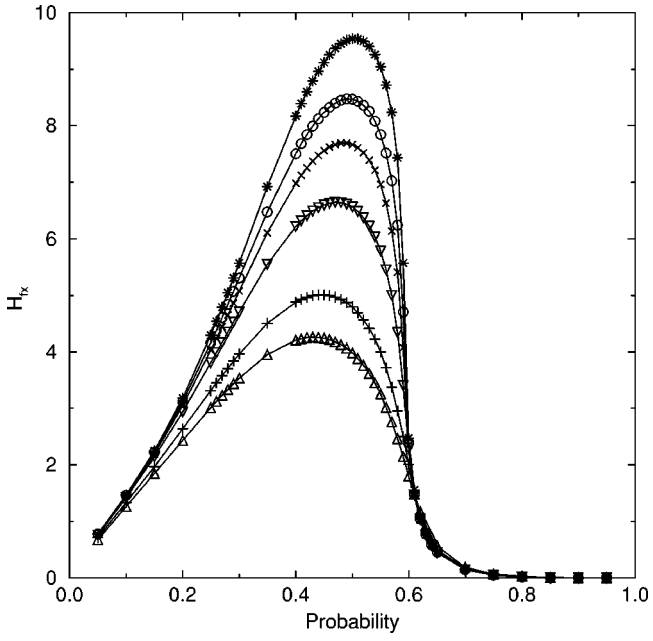

FIG. 15. H_{fx} versus p. The simulations were done using the same lattice sizes as in Fig. 12.

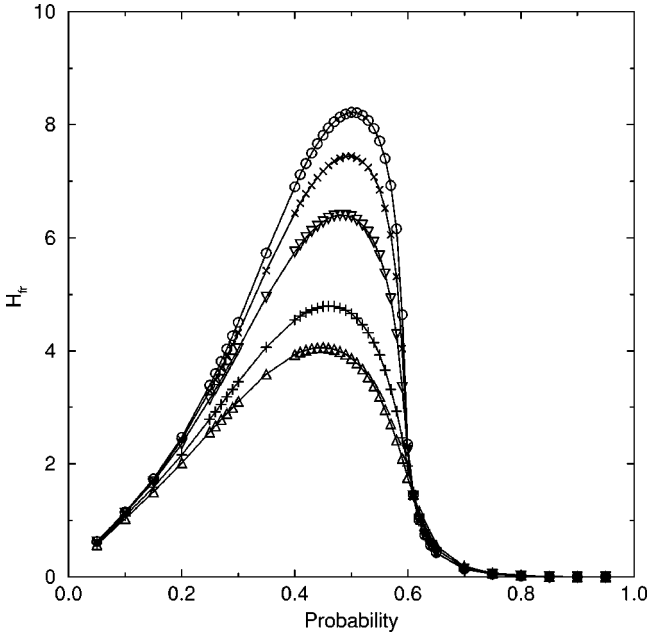

FIG. 16. H_{fr} as a function of p, using similar L as for D_{fr} in Fig. 13.

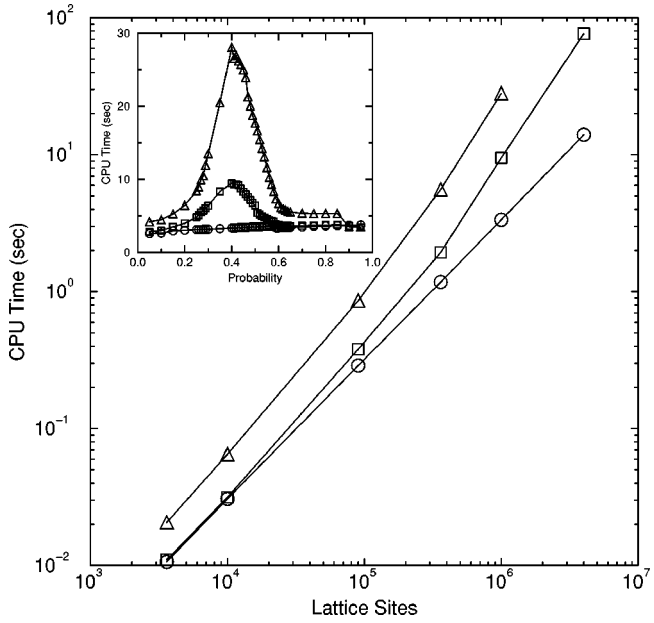

FIG. 17. Plot of the computer CPU time as a function of number of lattice sites. The inset shows the CPU time versus the probability of occupation for $L=1000$. The curves represent the EHK algorithm (○) and the fixed (□) and free (△) LA identification algorithm for different experiments.

cessing unit (CPU) time for several experiments. The time and space complexity of this algorithm depend on its implementation. The implementation of the EHK algorithm follows Ref. [9], where it is thoroughly analyzed. For creation and comparison of the tables containing the NN code sequence of the LA's, we dynamically allocate a vector of pointers with v elements, where v is equal to the number of clusters, excluding the clusters of size 1 and the clusters with size that appears just once. Each of these pointers indicates a vector of integers, and these have the sizes of the LA's, as shown in Fig. 9. The elements of these vectors are filled with the NN codes, as described in Sec. III, and they can easily be accessed by array reference. As a result, the memory space required is proportional to the number and size of LA's to be distinguished. Once this data structure is created the algorithm compares the element of the jth position of a NN code sequence with another LA's sequence of the same size. If a lexicographic sort is applied to the NN code sequences of LA's with the same size, a more efficient comparison algorithm is achieved, since fewer steps will be needed to compare all the sequences. Alternatively, hashing techniques can be used and may improve both the space and time complexity of the algorithm.

Figure 17 shows the CPU time as a function of the number of sites for the square lattice at $p=0.40$. The computational time complexity for this algorithm is not linear as in the case of the EHK algorithm. The inset in this figure shows the behavior of the CPU time as a function of the probability of occupation for $L=1000$. Clearly, there is a maximum computational time complexity, at $p=0.40$, which means a critical slowing down associated with the probability of maximum LA diversity. This maximum was found to be at $p\approx 0.45$ for $L\to\infty$ (see the next section). The difference from $p=0.40$, obtained in Fig. 17, is due to the finite size of the lattice used in this simulation. The slowing down of the algorithm happens because as the LA diversity increases the number of comparisons for different LA's also increases. The algorithm requires a larger amount of memory space in comparison to the EHK algorithm, since the sets of NN code representing the LA's need to be stored. The computational complexity for the identification of free LA's in both time and space is higher than for fixed LA's due to the symmetry operations. However, considering the proposed task the algorithm is quite efficient and yields useful measurements, especially when the morphology of the system studied is of concern.

VI. CRITICAL PROBABILITY

In the model studied both cluster diversity and cluster entropy present a define maximum. In order to estimate the probabilities at which these maxima occur at the thermodynamic limit, $L\to\infty$, finite size effects have to be taken into consideration. In Ref. [7], it was found that $p_c(D_{s\,\max})$ has statistically the same value as the percolation threshold p_c. From percolation theory we have that $L\sim|p-p_c|^{-\nu}$, where $\nu=\frac{4}{3}$ for two-dimensional site percolation. Now, let $p(L)$ be the transition probability or in this case the probability where the maxima of the functions occur, for a system of linear size L. One can estimate p_c from the scaling relation $p(L)-p_c \sim L^{-1/\nu}$. Thus, fits of $p(L)$ against $L^{-1/\nu}$ produce straight lines where the intercept at the y axis give us reasonable estimates of the value p_c (see Figs. 18 and 19).

In the case of the cluster size diversity and cluster size entropy, the present simulations yield $p_c(D_{s\,\max}) = p_c(H_{s\,\max}) = 0.58 \pm 0.02$, which is a value closer to the percolation threshold ($p_c = 0.592\,746$) than the value previously reported in [7]. The probabilities associated with $D_{fx\,\max}$ and $D_{fr\,\max}$ at the thermodynamic limit are 0.45 ± 0.02 and 0.46 ± 0.02, respectively. For cluster entropy $p(H_{fx\,\max}) = 0.49 \pm 0.02$ and $p(H_{fr\,\max}) = 0.50 \pm 0.02$. It is important to note that these probabilities are below the percolation threshold.

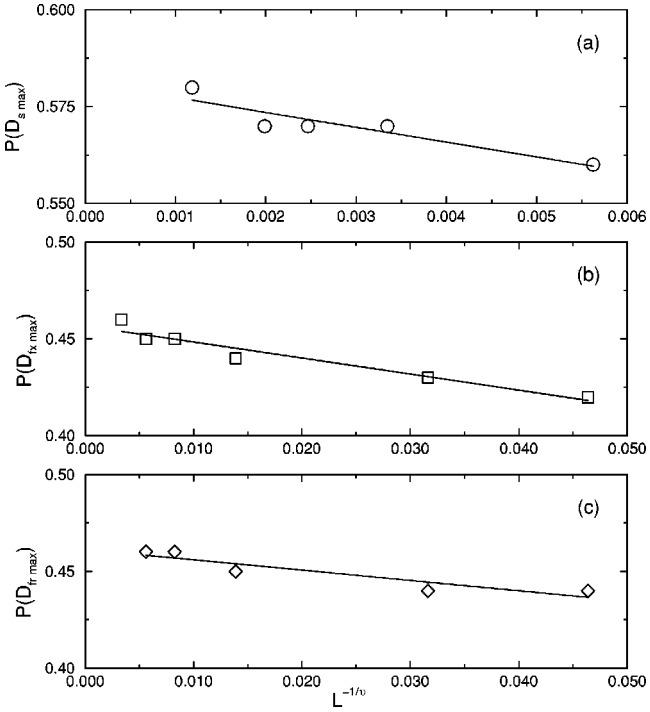

FIG. 18. Plot of the finite size scaling fits. (a) $p_c(D_{s\,\max})$ for $L=1000-8000$, (b) $p(D_{fx\,\max})$ for $L=60-2000$, and (c) $p(D_{fr\,\max})$ for $L=60-1000$, versus $L^{-1/\nu}$.

However, the finite size scaling correction was done in relation to the percolation scaling, i.e., $L^{-1/\nu}$. The use of the fit L^{-1} does not significantly change the estimates of these probabilities at $L\to\infty$.

Cluster entropy appears to be a better candidate for a possible complexity measurement for processes that present a distribution of clusters, such as fragmentation, aggregation, and percolation models. It captures the notion of having low values in both the ordered and disordered states, while presenting a maximum value in between. Moreover, it is related to the information content of the system, as derived by information theory. The system studied attains a maximum information content at a specific point between a low probability of occupation, where small scattered clusters are present (disorder), and a high p where a single cluster dominates the system (order). For the cluster size entropy we found that this maximum occurs at the same point as the percolation transition. For a complexity measurement such a fact is very important, since for this model it describes the point of phase transition as the most complex state. Other researchers have also pointed to this fact; however, they arrived at this conclusion through a different type of measurement. The measurement proposed was based on a phase space approach, derived from a scale dependent entropy or information content. As a result, a coarse-graining procedure measuring the entropy of various scales was obtained (for further details see Ref. [24]).

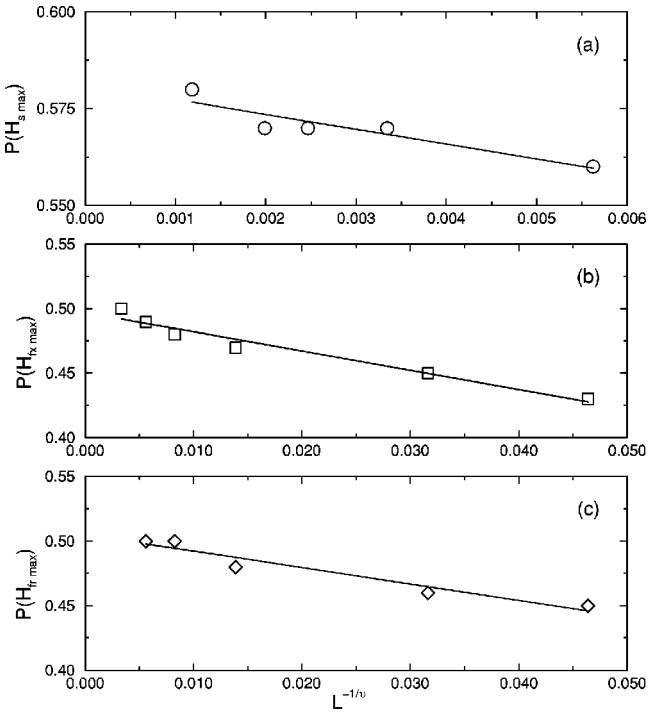

FIG. 19. Same as Fig. 18, but for cluster entropy. (a) $p_c(H_{s\,\max})$, (b) $p(H_{fx\,\max})$, and (c) $p(H_{fr\,\max})$ as functions of $L^{-1/\nu}$.

VII. DISCUSSION

Single measurements such as the radius of gyration or spatial moments are not sufficient to completely characterize different cluster shapes. Therefore, we introduce a method to code and differentiate the fixed and free LA's. It relies on coding each unit cell of a cluster using NN codes in a proper order sequence. Consequently, the memory requirement of this algorithm depends on the number of clusters present in the system, which imposes a restriction on the algorithm for very large lattice sizes. However, as computer memory becomes more available this does not constitute a serious limitation.

Another possible way to define LA's is through the external perimeter of the clusters. In the percolation problem, LA's appear in connection with an exact solution for the cluster number, which is given by

$$n_s=\sum_t g_{st}p^s(1-p)^t, \tag{9}$$

where g_{st} denotes the number of cluster configurations with size s and perimeter t [1]. Fixed LA's are defined by g_s, which encompasses the entire possible cluster configuration, irrespective of their perimeters. The definition of free LA's also does not take into account the cluster perimeter. Consequently, an alternative way to differentiate clusters is by g_{st}, such that the external perimeter becomes a parameter to distinguish the clusters. In the present simulations, the diversity of g_{st}, $D_{g_{st}}$, would give a different value if compared to fixed and free LA diversity. However, we do not address this measurement. Even though the EHK algorithm calculates the *internal* perimeter, further modification in the algorithm would be necessary to address the *external* perimeter. In addition, it is interesting to note that, due to the different definition of diversity, the following relation will hold: $D_{fx}\geqslant D_{fr}\geqslant D_{g_{st}}\geqslant D_s$.

Despite its complexity, the proposed algorithm enhances the analysis of cluster morphologies and can be used in sta-

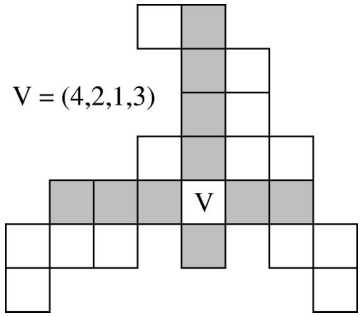

FIG. 20. Neighborhood vector $V=(n,e,s,w)$. Each element of the vector represents the numbers of neighbors in the directions north, east, south, and west.

tistical physics as a fine-grained way of characterizing clusters. Thus, it can be applied whenever the analysis of clusters is appropriate, such as in the percolation, fragmentation, and branched polymer problems. Extension of the algorithm to higher dimensions or different lattice types, such as triangular or hexagonal lattices, is possible. Furthermore, another possible variation of the algorithm is to identify part of a cluster by matching part of the NN code sequence of the clusters. As a result, degrees of similarity between clusters can be assigned.

The HK algorithm has also been applied in the fields of technology and applied science as pointed out in [9]. Likewise, by extending the concept of coding a cluster cell (pixel) according to its nearest neighbors the present algorithm can be used in the fields of image processing and pattern recognition. In this case, the pixels are coded according to the numbers of neighbors in the four directions (neighborhood code). In other words, if a pixel is in the inner part of the image, the number of neighbors in one of the directions is the number of pixels separating the aforementioned pixel from the border of the image (see Fig. 20). Each pixel is represented by a vector $V=(n,e,s,w)$, which is transformed into code $C=\mathcal{F}(n,e,s,w)$ by the function

$$C=\alpha\frac{\mathcal{L}^2+\mathcal{L}}{2}+\beta, \tag{10}$$

where

$$\alpha=n\mathcal{L}+s-\frac{n(n-1)}{2}, \tag{11}$$

$$\beta=e\mathcal{L}+w-\frac{e(e-1)}{2}, \tag{12}$$

and the image is assumed to have a size of $\mathcal{L}\times\mathcal{L}$. These equations take into consideration the fact that the greatest value that n, e, s, or w can assume is $\mathcal{L}-1$, and the boundary conditions

$$n+s<\mathcal{L}, \tag{13}$$

$$e+w<\mathcal{L}. \tag{14}$$

The possible number of codes can be enormous; thus by using a logarithmic quantization the codes are merged or reduced to a smaller set. The probability distribution of the reduced set of codes is then used as a characteristic feature of the object. We successfully applied this algorithm for the handwritten character recognition problem; for further details of this procedure see Ref. [25]. This shows that the ideas of the proposed code scheme have a broad range of applications.

ACKNOWLEDGMENT

This work was supported by CAPES (Brazilian Government Agency).

[1] D. Stauffer and A. Aharony, *Introduction to Percolation Theory*, 2nd ed. (Taylor and Francis, London, 1994).
[2] K. Coutinho, M. A. F. Gomes, and A. K. Adhikari, Europhys. Lett. **18**, 119 (1992).
[3] S. Wolfram, *Cellular Automata and Complexity: Collected Papers* (Addison-Wesley, Reading, MA, 1994).
[4] J. B. C. Garcia, M. A. F. Gomes, T. I. Jyh, T. I. Ren, and T. R. M. Sales, Phys. Rev. E **48**, 3345 (1993).
[5] M. A. F. Gomes, F. A. O Souza, and S. K. Adhikari, J. Phys. A **28**, L613 (1995).
[6] I. R. Tsang and I. J. Tsang, J. Phys. A **30**, L239 (1997).
[7] I. R. Tsang and I. J. Tsang, Phys. Rev. E **60**, 2684 (1999).
[8] J. Hoshen and R. Kopelman, Phys. Rev. B **14**, 3438 (1976).
[9] J. Hoshen, M. W. Berry, and K. S. Minser, Phys. Rev. E **56**, 1455 (1997).
[10] M. F. Sykes and M. Glen, J. Phys. A **9**, 87 (1976); M. F. Sykes, D. D. Gaunt, and M. Glen, *ibid.* **9**, 97 (1976); D. D. Gaunt, M. F. Sykes, and H. Ruskin, *ibid.* **9**, 1899 (1976).
[11] C. Domb, J. Phys. A **9**, L141 (1976).
[12] A. R. Conway and A. J. Guttmann, J. Phys. A **28**, 891 (1995).
[13] S. Alexander, G. S. Grest, H. Nakanishi, and T. A. Witten, Jr., J. Phys. A **17**, L185 (1984); M. Sahimi and G. R. Jerauld, *ibid.* **17**, L165 (1984).
[14] F. Family, J. Phys. A **13**, L325 (1980); *ibid.* **16**, L97 (1983).
[15] A. Y. Shahverdian, Fractals **5**, 199 (1997).
[16] S. W. Golomb, *Polyominoes: Puzzles, Patterns, Problems, and Packings* (Princeton University Press, Princeton, NJ, 1994).
[17] D. H. Redelmeier, Discrete Math. **36**, 191 (1981).
[18] D. H. Klarner, Can. J. Math. **19**, 851 (1967).
[19] R. C. Read, Can. J. Math. **14**, 1 (1962); D. H. Klarner and R. L. Rivest, *ibid.* **3**, 585 (1973); D. H. Klarner, Fibonacci Q. **3**, 9 (1965).
[20] D. Dhar, M. K. Phani, and M. Barma, J. Phys. A **15**, L279 (1982); A. J. Guttmann, *ibid.* **15**, 1987 (1982); A. Conway, *ibid.* **28**, L125 (1995).
[21] S. Redner, J. Stat. Phys. **29**, 309 (1982); S. Mertens, *ibid.* **58**, 1095 (1990); S. Mertens and M. E. Lautenbacher, *ibid.* **66**, 669 (1992).
[22] T. Oliveira e Silva claims to have reached $n=28$ using the algorithm described by S. Mertens in Ref. [21]; see http://www.inesca.pt/~tos/animals/44.html

[23] C. D. Van Siclen, Phys. Rev. E **56**, 5211 (1997); F. Borger, J. Feder, T. Jøssang, and R. Hilfer, Physica A **187**, 55 (1992); C. Andraud, A. Beghdadi, and J. Lafait, *ibid.* **207**, 208 (1994); C. Andraud, A. Beghdadi, E. Haslund, R. Hilfer, J. Lafait, and B. Virgin *ibid.* **235**, 307 (1997).

[24] Y. C. Zhang, J. Phys. I **1**, 971 (1991); H. C. Fogedby, J. Stat. Phys. **69**, 411 (1992).

[25] I. J. Tsang, I. R. Tsang, and D. Van Dyck, Pattern Recogn. Lett. **20**, 1279 (1999); I. R. Tsang, I. J. Tsang, P. Scheunders, and D. Van Dyck, in *Proceedings of the Fourth Joint Conference on Information Sciences*, edited by Heng D Cheng (AIM, Durham, NC, 1998), Vol. 4, p. 250.