Minimal spanning tree and percolation on mosaics: graph theory and percolation

1999 J. Phys. A: Math. Gen. 32 2611

(http://iopscience.iop.org/0305-4470/32/14/002)

J. Phys. A: Math. Gen. **32** (1999) 2611–2622. Printed in the UK PII: S0305-4470(99)98872-8

Minimal spanning tree and percolation on mosaics: graph theory and percolation

C d'Iribarne, M Rasigni and G Rasigni

Département de Physique des Interactions Photons-Matière, Case EC1, Faculté des Sciences et Techniques de Saint Jérôme, 13397 Marseille Cedex 20, France

Received 3 November 1998, in final form 18 January 1999

Abstract. Graph theory, through the minimal spanning tree (MST), is used to determine site percolation thresholds p_c related to 2D regular lattices. It is shown that there is a direct relation between p_c and a geometrical parameter characterizing the lattices. Moreover, the methodology developed in this paper makes it possible to study long-range percolation in an efficient way.

1. Introduction

Concepts of percolation theory have played an important role in our current understanding of disordered systems and their properties [1, 2]. Recall that in the site percolation problem, sites of a network are occupied with probability p and vacant with probability of $1 - p$. Two nearest-neighbour sites are called connected if they are both occupied, and connected clusters on the network are again defined in the obvious way. There exists a site percolation threshold p_c above which an infinite cluster of occupied sites spans the whole network [3].

Besides exact analysis, a variety of numerical methods have been used for the solution of percolation problems, namely Monte Carlo and series expansion methods [4, 5]. Although these methods turned out to be successful in solving some problems, they often require a high computation time—except for a few approaches like that of Hoshen and Kopelman [6], for instance. Moreover, the conventional methods to exactly compute percolation thresholds are restricted so far to two dimensions [7, 8]. For that case p_c values are only known for triangular [9, 10], square [11, 12], honeycomb [10–13] and Kagomé [4] lattices. But there are also 2D regular structures that are of interest, mainly all arrangements called mosaics which are composed of an infinite set of regular polygons [14, 15]. The knowledge of p_c for those mosaics may provide arguments for learning more about the percolation phenomenon.

Given that the essence of percolation theory is to determine how a given set of sites regularly or randomly positioned in some space is interconnected, it seems quite realistic to try using graph theory for tackling percolation problems. For this purpose Babalievski [16] has recently discussed two basic approaches to the cluster counting task in the percolation and related models, namely the Hoshen–Kopelman multiple labelling technique [6] for cluster statistics and the graph-theoretical basis for the spanning tree approaches.

This paper is an enlargement of a previous approach [17] using graph theory. It is aimed at showing that the minimal spanning tree (MST)—a special graph [18–20]—computed by means of the Rohlf's algorithm [21] is particularly well suited for determining the p_c values related to the above-mentioned mosaics. Moreover, such a new methodology makes it possible

0305-4470/99/142611+12$19.50

to study the long-range percolation problem [2, 22–24], namely the percolation related to the second, third ... nearest neighbour.

The paper is organized as follows. In section 2 we provide basic definitions of the graph theory and we describe the algorithms that may be used to construct a MST. We give arguments why Rohlf's algorithm is best suited for the problem at hand. Section 3 deals with numerical experimentation. Regular and semi-regular mosaics allowed in the plane are described and details about the generation of point distributions are given. Section 4 is aimed at computing the values of critical probability thresholds p_c for the various mosaics. It is shown that there is a direct relation between p_c and the geometrical parameter characterizing the mosaics. Computations are also extended to the study of long-range percolation. In conclusion (section 5) we discuss our results in the context of the results of other studies.

2. Minimal spanning tree analysis

2.1. Definitions

Basic definitions of the graph theory are given in numerous papers and may be found for instance in [18, 19]. Let us mention the fundamental definitions. An edge-weighted linear graph $G = (X, E)$ is composed of a set of points $X = \{x_1, x_2, \ldots\}$ called nodes and a set of node pairs $E = \{(x_i, x_j)\}$ called edges, with a number called weight (in this study the Euclidean distance) assigned to each edge. A tree is a connected graph without closed loops. A MST is a tree which contains all the nodes and where the sum of the edge weights is minimal. It has been shown in previous papers [25, 26] that the MST is a powerful tool for studying order and disorder in various systems. Indeed, depending on the starting point, there may be more than one MST for a given set of points, but all the MSTs have the same edge-length histogram. It follows that the statistical information deduced from the histogram, such as the average edge-length m, the standard deviation σ or the higher moments (skewness and kurtosis), may be used as characteristics for the corresponding set of points.

2.2. Algorithms

Numerous algorithms are available that enable us to compute the MST constructed from a set of points. Four of them are particularly used, namely Prim's [27], Kruskal's [28], Dijkstra's [29] and Rohlf's [21].

2.2.1. Prim's algorithm. In Prim's algorithm we start with a fixed node and, one by one, add nodes along the least-cost edges. In other words, we add the nearest neighbour to the subgraph that we have built so far, taken as a whole. We must avoid adding an edge that completes a cycle, since a tree cannot have any cycle. The process is stopped when there are no new nodes to add.

2.2.2. Kruskal's algorithm. The methodology that leads to Kruskal's algorithm is quite different. The algorithm begins by preprocessing the edges of the complete graph, sorting them by weight from cheapest to mostly costly. The algorithm then loops exactly $(n - 1)$ times, if n points are considered, adding each time a cheapest edge that will not create a cycle when added to the edges already included. Of course, as it adds an edge, it includes in the graph the nodes incident to the edge.

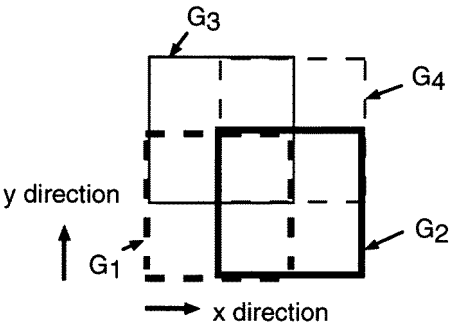

Figure 1. Partition of the space by means of a system of four partially overlapping grids. G_1: thick dashed line; G_2: thick full line; G_3: thin full line; G_4: thin dashed line.

2.2.3. Dijkstra's algorithm. Dijkstra's algorithm consists in retrieving a MST from a supergraph which has to be determined prior to the construction of the MST. The principle is as follows. It may be shown that the longest edge of a closed loop never belongs to a MST. Thus a MST can be obtained from the supergraph by removing the longest edge of each cycle, the order of removing being unimportant.

2.2.4. Rohlf's algorithm. We use Rohlf's algorithm which makes it possible to determine the MST constructed on n points in an efficient way, actually in time $n \ln n$. Moreover, this algorithm is particularly well suited for determining and studying clusters in the percolation phenomenon. We will describe it in some detail. This algorithm proceeds in stages which correspond to the selection of successively larger threshold distances δ_i between two nearest points. In the following we are only interested in 2D sets of points. Initially each point is considered as a component of the MST or equivalently as a cluster of unit size.

At the first stage the δ_1-value is determined by finding the minimal distance (in this paper the Euclidean distance) for a random sample of n pairs of points. Then a point is chosen at random, and a possible nearest neighbour at a distance $d \leqslant \delta_1$ is investigated. To do that, the space is partitioned by using a system of four partially overlapping grids G_k ($k = 1, 2 \ldots 4$) consisting of $2\delta_1$-sized square cells [30–32]. The grid G_1 is superimposed to the sampling window while G_2 and G_3 correspond to G_1 translated to δ_1 in the x and y directions respectively; G_4 corresponds to G_1 translated of δ_1 both in the x and y directions (figure 1). These grids have the property that each pair of points such as $d \leqslant \delta_1$ must be found together inside at least one cell of at least one grid. Thus, in order to link the randomly chosen point to its nearest neighbour such as $d \leqslant \delta_1$, one only needs to compute distances between the given point and the other points found in the same cells of the grid. At this stage, two cases may occur.

(i) There are no other points. Then the chosen point remains a cluster of unit size. Another point is chosen at random among the $(n-1)$ remaining, and a possible nearest neighbour at a distance $d \leqslant \delta_1$ is again investigated.

(ii) There are effectively other points. The chosen point is linked by an edge to the nearest point and thus a cluster of size two is obtained. Then the nearest neighbour related to the two points belonging to this cluster is tested. Other edges with $d \leqslant \delta_1$ are added to this component of the MST currently under consideration wherever possible. When no more additions can be made, another point is considered, chosen at random among those which have not already been taken into account, and so on, until all the points of the starting set have been considered. At the end of this stage one obtains a non-connected graph which consists of various-sized clusters union. The size of each cluster is known, and in particular the size of the largest one.

At the second stage the δ_2-value is determined by finding the minimum distance for a random sample of n pairs of points from the different components of the MST constructed at the first stage. One operates as previously. Now the partially overlapping grids consist of $2\delta_2$-sized square cells. Components of the MST are enlarged as possible by adding the edges such as $d \leqslant \delta_2$. For this purpose one only needs to compute distances between points in the given component and the other points which do not belong to the given component but are found in the same cells of the grid. Further stages are carried out until the single final MST is obtained.

The good efficiency of the algorithm originates in the use of systems of partially overlapping grids to partition the space. It must be emphasized that the first stage provides an attractive tool for studying percolation on lattice networks. Indeed it is enough to take the lattice parameter value as δ_1-value, to detect the clusters and determine their size easily. Thus one can evaluate the probability for a point to belong to the largest cluster and study this probability as a function of the probability p of site occupation. Moreover, it is possible to consider also the second, third ... nearest-neighbour, i.e. to study the so-called long-range percolation without additional difficulty.

2.2.5. Comparison between algorithms. All the above MST algorithms need a similar computation time if n is smaller than 500. The first three are easy to implement. Rohlf's algorithm becomes much more efficient when n is large because its complexity is $O(n \ln n)$ instead of $O(n^2)$ for the others. Despite the fact that it needs a large memory, Rohlf's algorithm is particularly well suited for simultaneously studying the first as well as the kth nearest-neighbour connections.

3. Numerical experimentation

3.1. Material

Mosaics in the plane are arrangements which are composed of an infinite set of adjacent regular polygons, i.e. of polygons with the same edge length. Figure 2 shows the 11 possibilities permitting such arrangements. Mosaics are of two types: (i) regular ones, i.e. constructed from only one regular polygon (triangle, square, hexagon); (ii) semi-regular ones, i.e. constituted of two or more regular polygons. Conventionally, mosaics are characterized by standard notation [33] which is reported in figure 2.

To compare the different mosaics or the corresponding lattices, one may characterize each by a normalized parameter, for instance the normalized edge length λ_0. For this purpose we impose the density of vertices to be $\rho = 1$ for each mosaic. If $\Delta X\lambda_0$ and $\Delta Y\lambda_0$ stand for the size of an elementary cell $\mathcal{M}$ in the x and y directions respectively, and $n_{\mathcal{M}}$ the number of sites in $\mathcal{M}$ (figure 3), then the normalized edge length λ_0 is given by

$$\lambda_0 = (n_{\mathcal{M}}/\Delta X\Delta Y)^{1/2}.$$

The corresponding values of λ_0 for the 11 lattices are reported in table 1.

3.2. Generation of point distributions

Points are independently uniformly distributed over the sites of mosaics as follows. Let N be the total number of considered sites for a given mosaic (in this paper $N = 600\,000$). Sites are numbered 1 to N line by line. A sequence $(v_k)_{1\leqslant k\leqslant N}$ of N numbers distributed randomly between zero and unity is generated from a pseudo random number generating program [34].

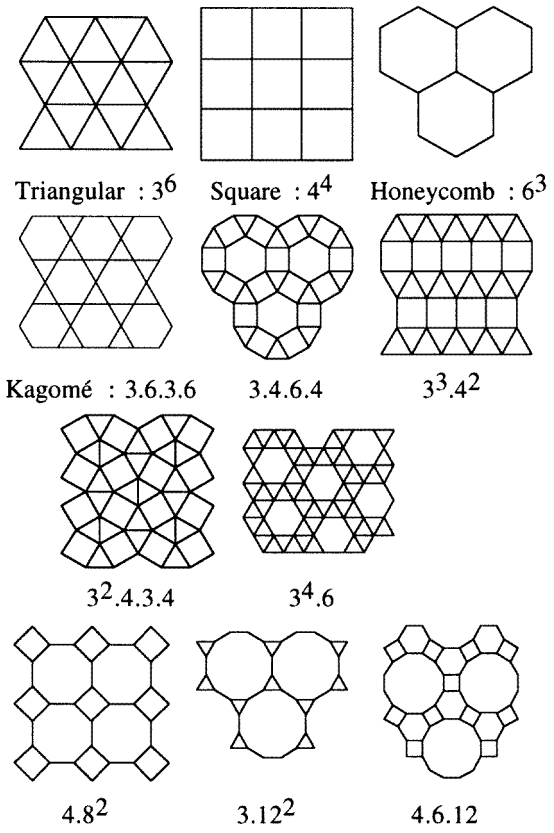

Figure 2. Eleven possible arrangements of a set of points on a plane along regular or semi-regular mosaics. Each mosaic is characterized by standard notation (for instance 3^6 for the triangular lattice) according to [33].

To each site S_k ($k = 1, 2, \ldots N$) is assigned the kth random number ν_k. Then we fix a probability of occupation $\tilde{p}$ at each site. A site S_k is considered as occupied by a point if $\nu_k \leqslant \tilde{p}$. The process begins with $\tilde{p} = \tilde{p}_0 = 0$: no site is occupied. The second stage corresponds to a probability of occupation $\tilde{p}_1$ defined as $\tilde{p}_1 = \tilde{p}_0 + \Delta p$ where Δp is a uniform step of probability ($\Delta p = 0.01$ in this study)—and so on and so forth until the ith stage where $\tilde{p}_i$ is given by $\tilde{p}_i = \tilde{p}_{i-1} + \Delta p$. The process is carried out until $\tilde{p} = 1$. Actually, the real probability of occupation at each site at the ith step is $p_i = N_i/N$ where N_i is the number of effectively occupied sites. The probability p_i is a random variable such as $\tilde{p}_i = E[p_i]$, where E stands for the expected value. Thus one obtains a set $\mathcal{E}$ of point distributions, each one generated by a random process and corresponding to a given probability of occupation at each site. The number of distributions depends on the chosen value Δp. To avoid effects due to the use of finite lattices, mosaics are placed on a torus. Moreover, with a view to reducing statistical fluctuations, each mosaic is studied from nine sets $\mathcal{E}_\ell (\ell = 1, 2, \ldots 9)$ of point distributions, each set being characterized by a sequence $(\nu_k)_{1 \leqslant k \leqslant N}$. Results are given as averages over these nine sets to obtain data with sufficiently small sampling errors.

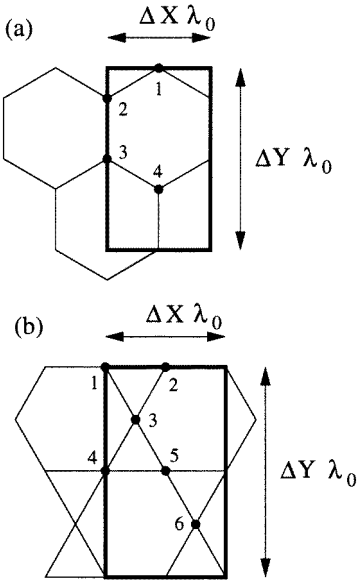

Figure 3. Possible elementary cell for the two lattices: (*a*) Triangular lattice with $\Delta X = \sqrt{3}$, $\Delta Y = 3$, $n_{\mathcal{M}} = 4$; (*b*) Kagomé lattice with $\Delta X = 2$, $\Delta Y = 2\sqrt{3}$, $n_{\mathcal{M}} = 6$.

Table 1. Values of p_c, normalized lattice parameter λ_0, site degree D_g, normalized MST average edge length m_c and $c = m_c/\lambda_0$ for the mosaics represented in figure 2.

	p_c					
Mosaics		Literature values	λ_0	D_g	m_c	$c = m_c/\lambda_0$
3^6	0.500	0.5000 [9, 10]	1.075	6	0.779	0.725
Triangular		0.500 [22, 23]				
$3^3.4^2$	0.549		1.035	5	0.781	0.755
$3^2.4.3.4$	0.550		1.035	5	0.782	0.756
$3^4.6$	0.580		0.995	5	0.776	0.780
4^4	0.590	0.5927 [11, 12]	1	4	0.784	0.784
Square		0.590 [22, 23]				
3.4.6.4	0.621		0.963	4	0.773	0.803
3.6.3.6	0.650	0.652 [4]	0.931	4	0.770	0.837
Kagomé						
6^3	0.698	0.6962 [10, 13]	0.877	3	0.759	0.865
Honeycomb		0.700 [22, 23]				
4.8^2	0.729		0.828	3	0.726	0.877
4.6.12	0.746		0.787	3	0.693	0.882
9.12^2	0.807		0.705	3	0.650	0.922

3.3. MST computation time

Computations have been performed on a Digital Alpha Station 255/233 MHz. As a guide, the computation time for constructing a MST from 5×10^5 occupied sites over 10^6, was about 30s cpu for the honeycomb lattice.

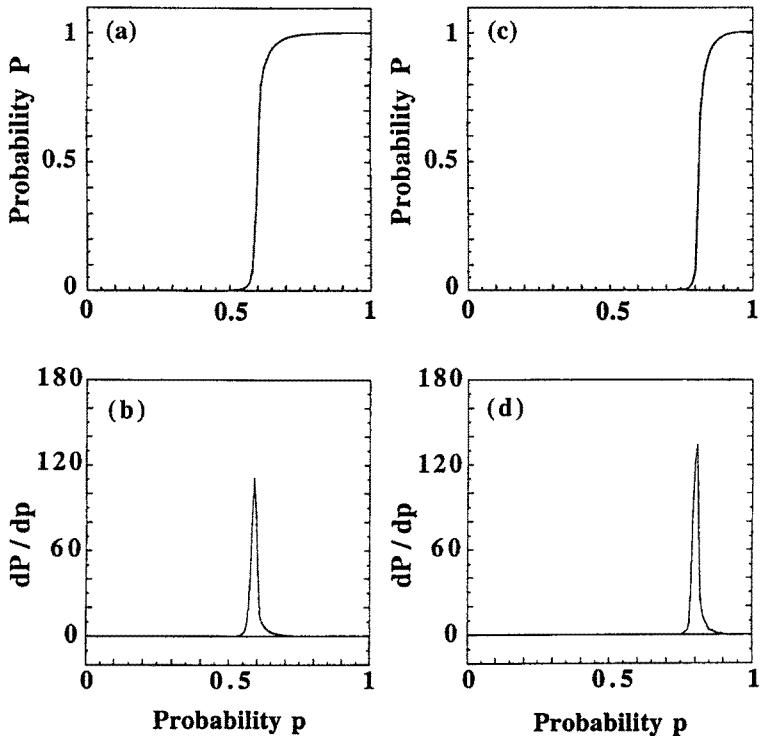

Figure 4. (*a*) Plot of the probability P that an occupied site belongs to the largest cluster versus the probability p of occupancy of the sites for the triangular lattice; (*b*) plot of the derivative of P versus p; (*c*) and (*d*) are the same as (*a*) and (*b*) for the (3.12^2) mosaic.

4. Results

4.1. First nearest-neighbour connections (classical percolation)

For a probability $p_i = N_i/N$ of site occupation, the first stage of Rohlf's algorithm enables us to know the size n_i of the largest cluster easily and so to compute the probability $P_i = n_i/N_i$ that a point belongs to it. To do that it is enough to take the lattice parameter value as the δ_1-value. By considering various p_i values it is then possible to study the variations of P_i versus p_i. Depending on the behaviour of clusters when percolation occurs, one expects to obtain a particular behaviour for P_i. Figure 4(*a*) shows the graph $P = f(p)$ for the triangular lattice (3^6). The curve actually exhibits a pronounced inflexion point for a critical probability p_c, which corresponds to a jump in P values. Quite similar graphs are obtained for each mosaic, the critical probability p_c differing from one to the other, as can be seen, for instance, in figure 4(*c*) related to the (3.12^2) mosaic. This p_c value is determined from the p value for which the derivative $\mathrm{d}P/\mathrm{d}p$ exhibits a maximum (figures 4(*b*) and 4(*d*)).

Results obtained for the 11 mosaics are reported and classified in ascending order in table 1. It can be noted that the p_c values related to triangular (3^6), square (4^4), honeycomb (6^3) and Kagomé (3.6.3.6) lattices are in good agreement with percolation threshold values found either in a theoretical way [9] or by other methods [10–13, 22, 23]. With a view to clarify the link between the computed critical probabilities p_c and the geometrical parameters characterizing the mosaics, the normalized lattice parameter values λ_0 as well as the site degree D_g have also

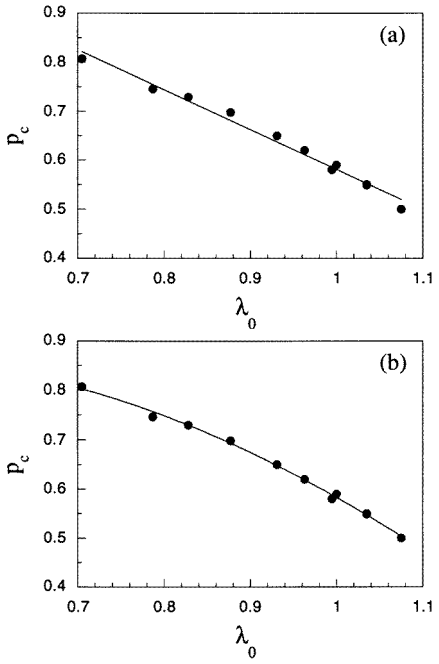

Figure 5. Plots of the site percolation thresholds for the 11 mosaics of figure 2 versus the normalized lattice parameter λ_0 characterizing the mosaics. The solid curve is a least-squares fit to the data: (*a*) linear approach, (*b*) polynomial approach.

been reported in table 1 for each mosaic. It may be seen that p_c increases as λ_0 decreases. Moreover, for the two mosaics ($3^3.4^2$) and ($3^2.4.3.4$) which have the same λ_0 value, the p_c values are found to be equal within the accuracy range of their measurements. Figure 5(*a*) shows the variation of p_c versus λ_0. There is roughly a linear dependence between p_c and λ_0. The solid curve is a least-square fit to the data and has the equation:

$$p_c = 1.397 - 0.816\lambda_0. \tag{1}$$

A more accurate fit to the data may be obtained by means of a polynomial approach, namely:

$$p_c = 0.715 + 0.735\lambda_0 - 0.866\lambda_0^2 \tag{2}$$

as is shown in figure 5(*b*). So, there is clearly a relation between the percolation threshold and the geometric parameter λ_0 which characterizes the mosaic. This relation may be made conclusive if a new dimensionless parameter is introduced. Indeed, let us consider the values of m_c, namely the normalized values of the MST average edge length for $p = p_c$, which are reported in table 1. The MST edge length values have been normalized according to the process described by Hofmann and Jain [35]. Given that the expected length of a randomly chosen edge of a MST constructed from N_i uniformly distributed points in a sampling window V, is asymptotically proportional [36] to

$$(N_i V)^{1/2}/(N_i - 1) \tag{3}$$

the normalized values are obtained by dividing the original ones by expression (3). For N_i large enough ($N_i \sim N_i - 1$) and a density of vertices in the lattice $\rho = 1$, it amounts to multiplying the original edge length values by $(p_i)^{1/2}$. The variation of p_c versus m_c (figure 6) does not exhibit an interesting behaviour. On the other hand, if we consider the dimensionless variable $c = m_c/\lambda_0$ whose values are reported in table 1, it clearly appears in figure 7 that the p_c values vary linearly with c. The solid line which is a least-square fit to the data has the equation:

$$p_c = -0.598 + 1.514c. \tag{4}$$

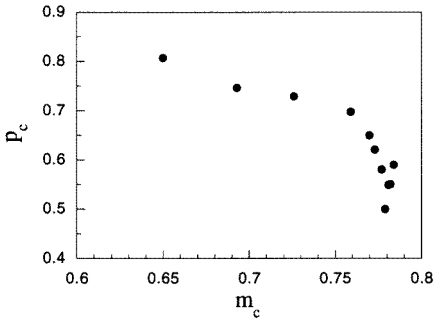

Figure 6. Plot of the site percolation thresholds for the 11 mosaics of figure 2 versus the normalized MST average edge length m_c.

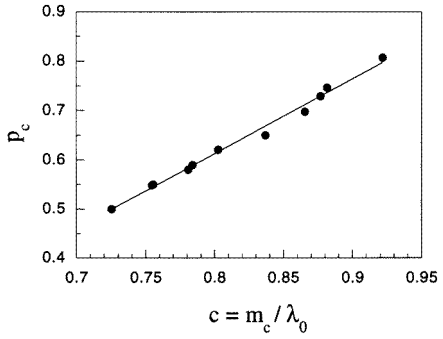

Figure 7. Plot of the site percolation thresholds for the mosaics of figure 2 versus $c = m_c/\lambda_0$.

That shows there is a link between p_c, the MST and the geometry of the system. Links between percolation and the MST have recently been shown theoretically by Bezuidenhout *et al* [37]. The new above-defined parameter c should be of interest in studies on conductivity, for instance. One also notes that as p_c increases the site degree D_g of the corresponding mosaics decreases, but not as a monotone function. So an initial analysis shows that there is no simple dependence between p_c and the site degree D_g of the mosaics.

4.2. kth nearest-neighbour connections ($k \geqslant 2$) *(long-range percolation)*

If we do not stop it at the first stage, Rohlf's algorithm enables us to determine, in only one run, the size of the largest cluster relative to first, second, third ... nearest-neighbour connections successively. This is possible by taking the increasing normalized inter-site distances $d_j = \alpha_j \lambda_0 (j = 1, 2, 3 \ldots)$ as δ_j-values.

With a view to testing the method, second and third-neighbour connections for the square lattice (4^4) were first considered. For this mosaic $\alpha_2 = \sqrt{2}$ and $\alpha_3 = 2$. The values determined for p_c, namely 0.404 and 0.290 respectively, compare favourably on the one hand with 0.401 obtained by Dean [38] and 0.410 obtained by Dalton *et al* [22, 23], and on the other hand with 0.292 given by Dalton *et al* [22, 23].

Then a study was carried out on the triangular (3^6), the honeycomb (6^3) and the Kagomé (3.6.3.6) lattices, respectively. These mosaics have the same α_j-values. The first five $\alpha_j (j = 1, 2, 3, 4, 5)$ are such as $\alpha_j^2 = \{1, 3, 4, 7, 9\}$. For a given probability $p_i = N_i/N$ of site occupation and for each α_j, the probability $P_{ij} = n_{ij}/N_i$ that a point belongs to the largest cluster of size n_{ij} was computed. Figure 8(*a*) shows the variation of P with p, for $\alpha = 1$, $\sqrt{3}$, 2, $\sqrt{7}$, 3 respectively, in the case of the honeycomb lattice (6^3). Curves corresponding to $\alpha = \sqrt{3}$, 2, $\sqrt{7}$, 3 are similar to those relative to $\alpha = 1$: they all present an inflection point for a critical probability p_c which is all the smaller as d is larger. The p_c values were evaluated as previously, i.e. from the p value for which the derivative $\mathrm{d}P/\mathrm{d}p$ is maximum (figure 8(*b*)).

A similar behaviour of $P = f(p)$ is obtained for the triangular (3^6) and the Kagomé (3.6.3.6) mosaics when considering successive inter-site distances d. Results for the three lattices are summarized in table 2, where the p_c values and the corresponding normalized inter-site distances $d = \alpha\lambda_0$ are also reported. Figure 9 shows the plot of $\ln(p_c)$ versus $\ln(d)$ for all the inter-site distances considered in this study. It can be seen that for higher nearest-

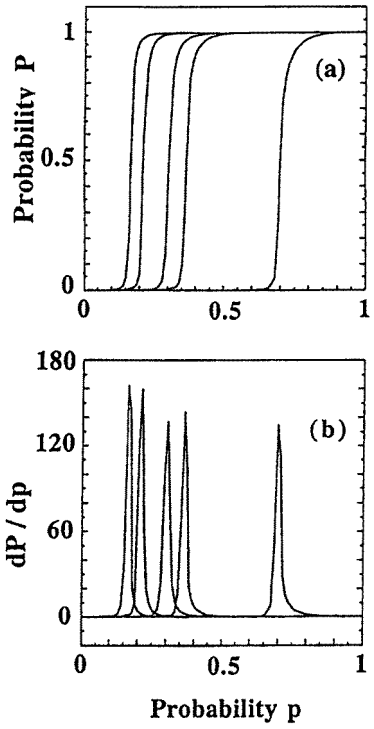

Figure 8. (*a*) Probability P for a point to belong to the largest cluster versus the probability of site occupation for the honeycomb lattice. From right to left the curves are related to the first, second, third, fourth and fifth nearest-neighbour connections respectively, i.e. to normalized inter-site distances $d = \alpha\lambda_0$ with $\alpha = 1$, $\sqrt{3}$, 2, $\sqrt{7}$ and 3; (*b*) derivative $\mathrm{d}P/\mathrm{d}p$ versus p.

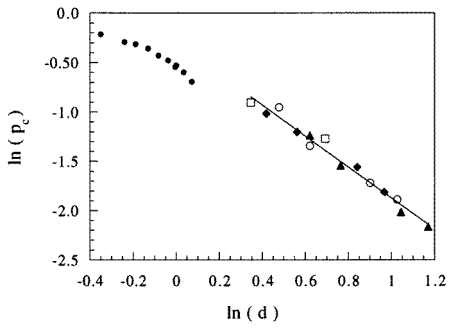

Figure 9. $\ln(p_c)$ versus $\ln(d)$ (d: normalized inter-site distance); filled circles: first nearest-neighbour connections for the mosaics represented in figure 2; square: second and third nearest-neighbour connections for the (4^4) lattice; filled triangles: second, third, fourth and fifth nearest-neighbour connections for the (3^6) lattice; filled lozenges: as previously for the (6^3) lattice; circles: as previously for the (3.6.3.6) lattice.

Table 2. Values of p_c for the second to fifth nearest-neighbour connections which are characterized by the normalized inter-site distance $d = \alpha\lambda_0$, for the three mosaics (3^6), (3.6.3.6) and (6^3). In italic, p_c literature values.

	Second nearest-neighbour connections		Third nearest-neighbour connections		Fourth nearest-neighbour connections		Fifth nearest-neighbour connections	
Lattices	$d_2 = \sqrt{3}\lambda_0$	p_c	$d_3 = 2\lambda_0$	p_c	$d_4 = \sqrt{7}\lambda_0$	p_c	$d_5 = 3\lambda_0$	p_c
3^6 Triangular	1.861	0.290 *0.295* [22, 23]	2.149	0.214 *0.225* [22, 23]	2.843	0.133	3.224	0.115
3.6.3.6 Kagomé	1.612	0.386	1.861	0.261	2.462	0.179	2.792	0.151
6^3 Honeycomb	1.520	0.361	1.755	0.300 *0.300* [22, 23]	2.321	0.210	2.632	0.163

neighbour connections data may be fitted with a good approximation by a linear relation:

$$\ln(p_c) = a\ln(d) + b \tag{5}$$

with $a = -1.572$, $b = -0.299$. This is not the case when only first nearest-neighbour connections are taken into account. Thus the dependence of p_c on d differs depending on whether we consider first nearest-neighbour connections or higher nearest-neighbour connections. Nevertheless, the above approach obtained for (3^6), (3.6.3.6), and (6^3) lattices, has to be confirmed for the other mosaics.

5. Conclusion

It is clear that the results presented in this paper emphasize the potentialities of the graph theory via the MST to study percolation. Indeed, by using this procedure it became possible to determine the percolation thresholds related to all the regular and semi-regular mosaics allowed in the plane in quite a sensible computation time. Most of these thresholds were unknown until now and were difficult to find by using conventional methods. For this purpose, Rohlf's algorithm turned out to be very efficient. Indeed, unlike other algorithms, such as Hoshen-Kopelman's [6] for instance, Rohlf's algorithm takes only the number of occupied sites into account instead of the total number of lattice network sites. This decreases the computational time for weak probabilities of site occupation. However, although the memory complexity is linear, it is larger than the one related to Hoshen–Kopelman's approach. So, according to Babalievski [16], Hoshen–Kopelman's algorithm is more suitable for very large systems whereas the spanning tree approach can easily solve more sophisticated tasks in systems of a moderate size.

Our results indicate that there is a direct relation between percolation threshold and the normalized lattice parameter characterizing the mosaics. The nature of this relation is central to learning more about the percolation phenomenon. Recently, Galam and Mauger in a series of papers [39] collected many percolation thresholds to compare them with some general fitting formula, similar in spirit to the present aims. On the other hand, we have shown there is a linear dependence between percolation threshold and a new parameter linked to the MST and the geometry of the system. Unfortunately, we are not able to interpret it yet. Finally, the flexibility of Rholf's algorithm enabled us to study long-range percolation, a phenomenon that very few people have investigated before, and the methodology we used should allow us to study lattices at dimensions greater than two and also non-regular arrangements.

References

[1] Stauffer D 1979 *Phys. Rep.* **54** 2
Stauffer D 1985 *Introduction to Percolation Theory* (London: Taylor and Francis)
[2] Grimmett G R 1989 *Percolation* (Berlin: Springer)
[3] Broadbent S R and Hammersley J M 1957 *Proc. Camb. Phil. Soc.* **53** 629
[4] Sahimi M 1994 *Applications of Percolation Theory* (London: Taylor and Francis)
[5] Isichenko B 1992 *Rev. Mod. Phys.* **64** 961
[6] Hoshen J and Kopelman R 1976 *Phys. Rev.* B **14** 3438
[7] Essam J W, Gaunt D S and Guttmann A J 1978 *J. Phys. A: Math. Gen.* **11** 1983
[8] Kesten H 1982 *Percolation Theory for Mathematicians* (Basel: Birkhäuser)
[9] Sykes M F and Essam J W 1963 *Phys. Rev. Lett.* **10** 3
[10] Dean P and Bird N F 1967 *Proc. Camb. Phil. Soc.* **63** 477
[11] Ziff R M 1986 *Phys. Rev. Lett.* **56** 545
[12] Kertesz J 1986 *J. Phys. A: Math. Gen.* **19** 599
[13] Sykes M F, Gaunt D S and Glen M 1976 *J. Phys. A: Math. Gen.* **9** 97
Sykes M F, Gaunt D S and Glen M 1976 *J. Phys. A: Math. Gen.* **9** 725
[14] Ball W W R 1959 *Mathematical Recreations and Essays* (London: MacMillan)
[15] Wells A F 1956 *The Third Dimension in Chemistry* (Oxford: Oxford University Press)

[16] Babalievski F 1998 *Int. J. Mod. Phys.* C **9** 43
[17] d'Iribarne C, Rasigni G and Rasigni M 1995 *Phys. Lett.* A **209** 95
[18] Berge C 1983 *Théorie des Graphes et ses Applications* (Paris: Gauthier-Villars)
[19] Harary F, Norman R Z and Cartwright D 1965 *Structural Models* (New York: Wiley)
[20] Zahn C T 1971 *IEEE Trans. Comput.* C **20** 68
[21] Rohlf F J 1978 *Inform. Process. Lett.* **7** 44
[22] Dalton N W, Domb C and Sykes M F 1964 *Proc. Phys. Soc.* **83** 496
[23] Domb C and Dalton N W 1966 *Proc. Phys. Soc.* **89** 856
[24] Shante V K S and Kirkpatrick S 1971 *Adv. Phys.* **20** 325
[25] Dussert C, Rasigni G, Rasigni M, Palmari J and Llebaria A 1986 *Phys. Rev.* B **34** 3528
[26] Loeffler A, Rasigni M and Raidt U 1996 *J. Phys. A: Math. Gen.* **29** 2969
[27] Prim R C 1957 *Bell Sys. Tech. J.* **36** 1389
[28] Kruskal J B 1956 *Proc. Am. Math. Soc.* **7** 48
[29] Dijkstra E W 1960 *Indag Math.* **28** 196
[30] Yuval G 1975 *Inform. Process. Lett.* **3** 113
[31] Yuval G 1976 *Inform. Process. Lett.* **5** 63
[32] Rabin M O 1976 Probabilistic algorithms *Algorithms and Complexity* vol 21, ed J Treub (New York: Academic)
[33] Grünbaum B and Shepard G C 1987 *Tilings and Patterns* (New York: Freeman)
[34] Knuth D E 1969 *The Art of Computer Programming* vol 2 (New York: Addison-Wesley)
[35] Hofmann R and Jain A K 1983 *Patt. Recog. Lett.* **1** 175
[36] Steele J M 1988 *Ann. Prob.* **16** 1767
[37] Bezuidenhout C, Grimmett G and Loeffler A 1998 *J. Stat. Phys.* **92** 1
[38] Dean P 1963 *Proc. Camb. Phil. Soc.* **59** 397
[39] Galam S and Mauger A 1998 *Eur. Phys. J.* **1** 255