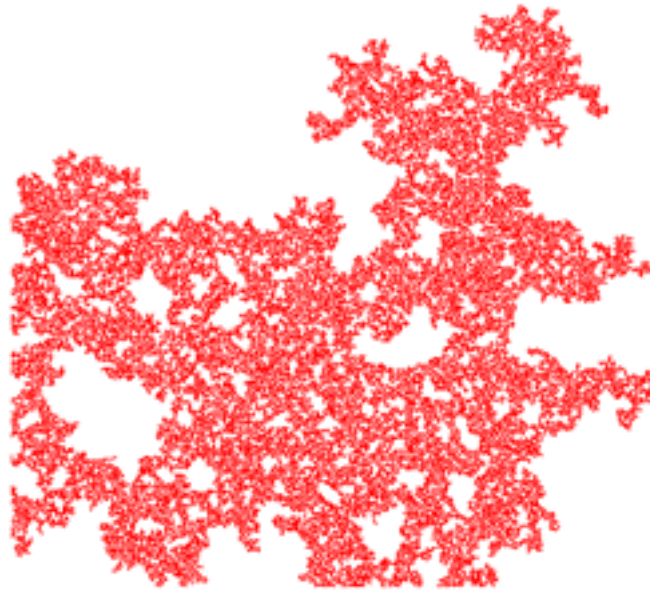




UNIVERSITÉ DE FRIBOURG SUISSE
UNIVERSITÄT FREIBURG SCHWEIZ

Une introduction à la théorie de la percolation

Pablo Crotti



Séminaire
Printemps 2009

Sous la direction du Professeur Christian Mazza

Table des matières

Introduction	4
1 Rappels de Théorie des graphes, Mesure et Probabilités	5
1.1 Théorie des graphes	5
1.2 Théorie de la mesure et des Probabilités	7
2 Percolation sur \mathbb{L}^d	9
2.1 Espace de probabilités sur \mathbb{L}^d	9
2.2 Cluster sur \mathbb{L}^d	10
2.3 Le phénomène critique	11
2.4 Théorème de Kesten et valeurs critiques	14
2.5 Approximation numérique de la probabilité critique	14
3 Loi 0-1 de Kolmogorov, Inégalité FKG, Théorème d'existence et unicité	17
3.1 Loi 0-1 de Kolmogorov	17
3.2 Inégalité FKG	18
3.3 Théorème d'existence	19
4 Applications de la percolation	21
4.1 Feux de forêt	21
4.2 Percolation orientée	21
4.3 Diffusion d'une épidémie	21
5 Simulation de feu de forêt	23
5.1 Modélisation	23
5.1.1 Modélisation du problème	23
5.1.2 Algorithme	23
5.1.3 Problèmes rencontrés	24
5.2 Résultats	24
Bibliographie	25
A Code Java	29

Introduction

La théorie de la percolation est souvent abordée sous un angle physique car c'est dans ce domaine qu'elle est la plus pertinente.

En effet, grâce à la théorie de la percolation, d'autres théories ont émergé et la théorie elle-même s'est rapidement développée.

La théorie de la percolation ne comporte pas uniquement un intérêt dans le domaine de la physique. En effet, on peut modéliser le comportement de feu de forêts ou encore le mouvement d'une épidémie dans une population grâce à la percolation.

Pour comprendre toutes ces applications, il est déjà très important de pouvoir comprendre le principe même de ce qu'on appelle des clusters, une probabilité critique et enfin de la percolation.

Ce séminaire est une introduction à la "site percolation" et à la "bond percolation".

Nous traiterons les notions de bases (qui permettent déjà de faire bon nombre d'applications) ainsi que certaines propriétés importantes et utiles.

Une simulation est donnée en annexe de manière à pouvoir visualiser et comprendre le fonctionnement de toute la théorie qui aura été étudiée.

Comme le sujet requiert certaines connaissances en théorie des graphes, en théorie de la mesure et en théorie des probabilités, le premier chapitre sera consacré à quelques rappels importants de manière à pouvoir suivre ce séminaire sans avoir de connaissances préalables.

Chapitre 1

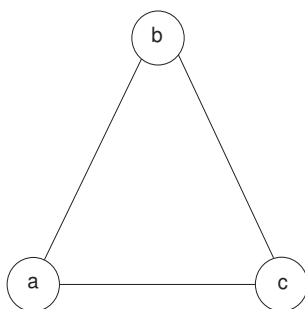
Rappels de Théorie des graphes, Mesure et Probabilités

Avant de commencer à parler des phénomènes de percolation il est important de rappeler certaines notions de théorie des graphes ainsi que celles concernant la théorie de la mesure et des probabilités.

1.1 Théorie des graphes

On rappelle qu'un *graphe* $G = (V, E)$ est une paire (V, E) où V est l'ensemble des *sommets de G* et E est l'ensemble des *arêtes de G* reliant les sommets. Chaque arête peut être représentée par deux sommets.

Pour illustrer ceci, soit $G = (E, V)$ avec $V = \{a, b, c\}$, $E = \{(a, b), (b, c), (c, a)\}$. Ce graphe est représenté de la manière suivante



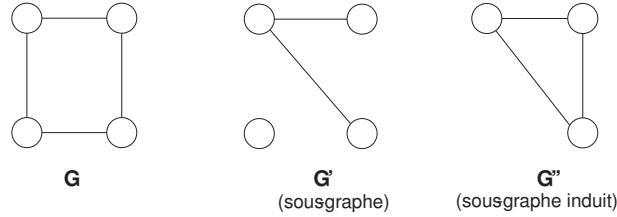
Nous devons aussi donner quelques définitions sur la manière dont sont utilisés les sommets et les arêtes.

Définition.

- Un graphe G est dit *orienté* si on tient compte de l'ordre dans lequel sont inscrits les sommets relatifs aux arêtes de E . Par exemple $(a, b) \in E$ n'est pas la même arête que (b, a) . L'une va de a à b et l'autre va de b vers a .
- Un graphe G est *non orienté* si on ne tient pas compte de l'ordre dans lequel sont inscrits les sommets relatifs aux arêtes de E (i.e. $(a, b) = (b, a) \in E$).
- Un sommet x est *adjacent* à un sommet y si les deux sont sur la même arête.

Comme les graphes sont des ensembles, la théorie nous permet de dire qu'il existe des sous ensembles de E et de V . On appelle $G' = (V, E')$, le *sous-graphe* d'un graphe $G = (V, E)$ où $E' \subseteq E$.

Le *sous-graphe induit* d'un graphe G est un graphe $G'' = (V'', E'')$ pour lequel $V'' \subseteq V$ et $E'' \subseteq E$ où E'' possède toutes les arêtes de E étant entre deux sommets de V'' .



Une fois le graphe établi, il est intéressant de pouvoir se promener dessus. Ainsi il devient naturel de définir certaines notions de chemins.

- Une *marche* est une séquence de sommets et d'arêtes $(v_1, (v_1, v_2), v_2, \dots, v_m)$ où les v_i représentent les sommets.
- Un *chemin* est une marche où tous les sommets sont distincts, i.e,
 $(v_1, \dots, v_m), v_i \neq v_j \forall 1 \leq i \neq j \leq m$.
- Un *cycle* est une marche fermée, i.e, $v_1 = v_m$.
- Un graphe est *connexe* s'il existe un chemin entre chaque sommet.

Nous allons maintenant nous intéresser à une catégorie de graphes particuliers. Ce sont les graphes créés à partir de \mathbb{Z}^d . Pour considérer notre construction nous devons d'abord établir la définition suivante.

Définition. Soit $\mathbb{Z}^d = \{x = (x_1, \dots, x_d) \mid x_i \in \mathbb{Z}\}, d \geq 1$ et soit la distance définie par

$$\delta(x, y) := \sum_{i=1}^d |x_i - y_i|, \forall x, y \in \mathbb{Z}^d.$$

Lorsqu'on introduit une arête entre chaque sommet (séparés d'une distance unitaire) on à la définition suivante.

Définition. Soit le graphe \mathbb{L}^d obtenu à partir de \mathbb{Z}^d en reliant chaque sommet distant d'une unité par une arête.

$$\mathbb{L}^d := (\mathbb{Z}^d, E^d)$$

où E^d est l'ensemble de ces arêtes.

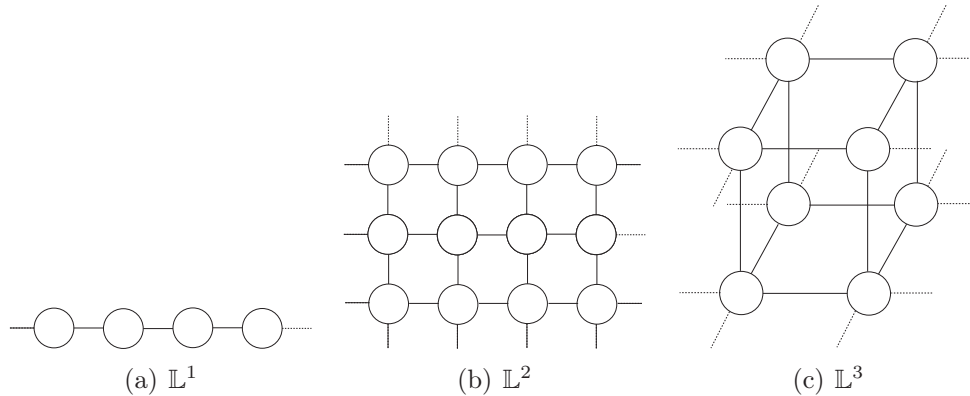


FIG. 1.1.1 – Différentes visualisations de \mathbb{L}^d

1.2 Théorie de la mesure et des Probabilités

La théorie de la percolation étant un domaine particulier des probabilités il est important d'énoncer certaines définitions et propriétés que l'on aborde souvent lorsque l'on étudie des espaces de probabilités.

On rappelle la définition d'une σ -algèbre : pour un ensemble Ω , $\mathcal{F} \subset \mathcal{P}(\Omega)$ est une σ -algèbre si :

- $\Omega \in \mathcal{F}$.
- Si $A \in \mathcal{F}$ alors $A^c \in \mathcal{F}$.
- Pour $(A_n)_{n \geq 1}$, $A_n \in \mathcal{F}$ on a $\bigcup_{n=1}^{\infty} A_n \in \mathcal{F}$

L'ensemble des parties de Ω (i.e $\mathcal{P}(\Omega)$) étant très grand nous nous intéressons uniquement à une certaine catégorie de σ -algèbre.

Définition. Soit $\mathcal{E} \subset \mathcal{P}(\Omega)$ et soit $\mathcal{G}(\mathcal{E}) \subset \mathcal{P}(\Omega)$ la famille des σ -algèbres contenant \mathcal{E} . Alors

$$\sigma(\mathcal{E}) := \bigcap_{\mathcal{F} \in \mathcal{G}} \mathcal{F}$$

est appelée σ -algèbre engendrée par \mathcal{E} et \mathcal{E} le générateur de $\sigma(\mathcal{E})$.

Lorsque notre espace (Ω, \mathcal{F}) est un espace topologique munit de sa topologie \mathcal{T} , on appelle σ -algèbre de Borel la σ -algèbre engendrée par les ouverts de \mathcal{T} (i.e $\sigma(\mathcal{T})$).

Définition. Dans le cas où $\Omega = \mathbb{R}^d$ avec la topologie usuelle, on note

$$\mathbb{B}^d := \sigma(\mathbb{R}^d)$$

Un espace munit d'une σ -algèbre est dit *mesurable*. Pour mesurer notre ensemble nous avons besoin de définir la notion de mesure.

Définition. Soit Ω un ensemble quelconque. Soit $\mathcal{F} \subset \mathcal{P}(\Omega)$ une σ -algèbre. Une fonction

$$\mu : \mathcal{F} \rightarrow [0, \infty]$$

est appelée *mesure* si :

- $\mu(\emptyset) = 0$
- Si pour tout $(A_n)_{n \geq 1}, A_n \in \mathcal{F}, A_n \cap A_m = \emptyset$ si $n \neq m$, alors

$$\mu\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} \mu(A_n)$$

De plus si $\mu(\Omega) = 1$ on appelle μ une *mesure de probabilité*.

On peut voir par exemple que si $\Omega = \mathbb{N}$ et $\mathcal{F} = \mathcal{P}(\Omega)$, l'application définie comme

$$\mu(A) := \text{Card}(A), \quad \forall A \in \mathcal{F}$$

est une mesure.

Un espace muni d'une mesure (μ par exemple) et noté $(\Omega, \mathcal{F}, \mu)$ est appelé *espace mesuré*. Dans le cas où la mesure est une mesure de probabilité on dit que c'est un *espace de probabilité*.

Considérons maintenant deux espaces mesurables et leur produit cartésien (i.e $\Omega = \Omega_1 \times \Omega_2$) ainsi que leur σ -algèbre respectives \mathcal{F}_1 et \mathcal{F}_2 .

La σ -algèbre produit est définie comme $\mathcal{F} := \mathcal{F}_1 \times \mathcal{F}_2$ et la *mesure produit* $\mu = \mu_1 \otimes \mu_2$ comme :

$$\mu(A_1 \times A_2) := \mu_1(A_1)\mu_2(A_2), \quad \forall A_1 \in \mathcal{F}_1, A_2 \in \mathcal{F}_2$$

On considère maintenant une suite de variables aléatoires $(X_i)_{i \geq 1}$. Avec ces suites il est possible d'engendrer des σ -algèbres. On a la définition suivante :

Définition. Soit $\mathbb{B}^{(n)} = \sigma(X_n, X_{n+1}, \dots)$ et $\mathbb{B}^{(n+1)} = \sigma(X_{n+1}, X_{n+2}, \dots)$. Ainsi $\mathbb{B}^{(n+1)} \subset \mathbb{B}^{(n)}$. On appelle *tribu asymptotique*

$$\mathbb{B}^{(\infty)} = \bigcap_{n \geq 1} \mathbb{B}^{(n)}$$

Dans ce séminaire nous aurons aussi besoin de la notion de classe monotone, c'est-à-dire qu'une famille \mathcal{A} de parties de Ω est fermée pour les limites monotones d'éléments de \mathcal{A} .

Chapitre 2

Percolation sur \mathbb{L}^d

Il y a deux types de percolation. La percolation par lien (bond percolation) et la percolation par site (site percolation). Considérant un graphe G quelconque, la percolation par lien est l'étude des phénomènes de percolation sur les arêtes de G . La percolation par site, quant à elle, est l'étude des phénomènes de percolation sur les sommets de G .

Ces deux formes de percolation possèdent, bien que différentes, des propriétés similaires. Nous allons étudier ces propriétés sur l'espace \mathbb{L}^d .

Mais avant toute chose, nous devons introduire un espace de probabilité sur ceux-ci.

2.1 Espace de probabilités sur \mathbb{L}^d

Pour pouvoir utiliser une mesure de probabilité il est nécessaire d'indiquer quelle sera la σ -algèbre que nous utilisons sur notre graphe. Comme chaque sommet ou arête est indépendant des autres nous utiliserons la σ -algèbre des cylindres.

Ici, nous considérerons que les arêtes ou les sommets sont indépendants des autres.

En posant $\Omega = \{0, 1\}^{E^d}$ et $\mathcal{F} \subset \mathcal{P}(\Omega)$ la σ -algèbre des cylindres, on dit qu'un élément $\omega = (\omega(e) | e \in E^d) \in \Omega$ est une *configuration pour \mathbb{L}^d* et que l'arête est ouverte si $\omega(e) = 1$ et fermée si $\omega(e) = 0$.

Définition. Soit $\mathbb{L}^d = (\mathbb{Z}^d, E^d)$. On dit qu'une arête $e \in E^d$ est *ouverte* avec probabilité p et est *fermée* avec probabilité $1 - p$ pour $0 \leq p \leq 1$.

La même terminologie est utilisée pour les sommets de \mathbb{Z}^d .

Définition. Soit $(\Omega, \mathcal{F}, \mu_e)$. La mesure μ_e est appelée *mesure de Bernoulli* si

$$\mu_e(\omega(e) = 0) = 1 - p \quad , \quad \mu_e(\omega(e) = 1) = p$$

Maintenant que l'on sait définir l'état d'une arête ou d'un sommet, regroupons les arêtes ouvertes.

Définition. Soit $(\Omega, \mathcal{F}, \mathbb{P}_p)$ notre espace de probabilités pour \mathbb{L}^d . On définit

$$K(\omega) = \{e \in E^d \mid \omega(e) = 1\}$$

l'ensemble des arêtes ouvertes.

Pour $\omega_1, \omega_2 \in \Omega$, on a l'ordre partiel suivant :

$$\omega_1 \leq \omega_2 \iff \omega_1(e) \leq \omega_2(e) \forall e \in E^d.$$

Nous pouvons maintenant formaliser notre intuition de probabilité sur tout notre espace avec la définition suivante.

Définition. Soit $(\Omega, \mathcal{F}, \mu_e)$ un espace mesuré où μ_e possède une densité p . On définit la *mesure produit* comme

$$\mathbb{P}_p := \prod_{e \in E^d} \mu_e$$

On note \mathbb{P}_p pour la probabilité et \mathbb{E}_p pour l'espérance en fonction de p .

2.2 Cluster sur \mathbb{L}^d

En considérant les sommets ou les arêtes ouverts de \mathbb{L}^d comme sous-ensemble, on obtient des sous-graphes de \mathbb{L}^d .

Dans le cadre général, pour un graphe $G = (V, E)$, un *cluster* ou un *amas* est un ensemble C de composantes connexes de G .

Il est nécessaire d'approfondir la notion de connexion :

- Pour la percolation par lien, deux arêtes sont connectées si elles ont un sommet en commun.
- Pour la percolation par site, deux sommets sont connectés s'ils sont adjacents.

En utilisant la notion d'ouvert pour un amas nous obtenons

Définition. Soit $G = (\mathbb{Z}^d, K(\omega))$ un sous-graphe de \mathbb{L}^d contenant tous ses sommets et les arêtes ouvertes. Chaque composante connexe de G est appelée *cluster ouvert*. On note

$$C(x) = \{y \in \mathbb{Z}^d \mid y \text{ est connecté à } x \text{ par un chemin ouvert}\}$$

On appelle $C(x)$ le cluster contenant x .

Lorsque $x = (0, \dots, 0)$ (i.e x est le centre de \mathbb{L}^d), on note $C(0) = C$.

S'il n'y a qu'un élément on écrit $C(x) = \{x\}$.

On peut toujours ramener x vers 0 par translation (automorphisme). Par conséquent, nous travaillerons la plupart du temps avec C .

Voici quelques figures pour illustrer tout ce qui a été défini précédemment.

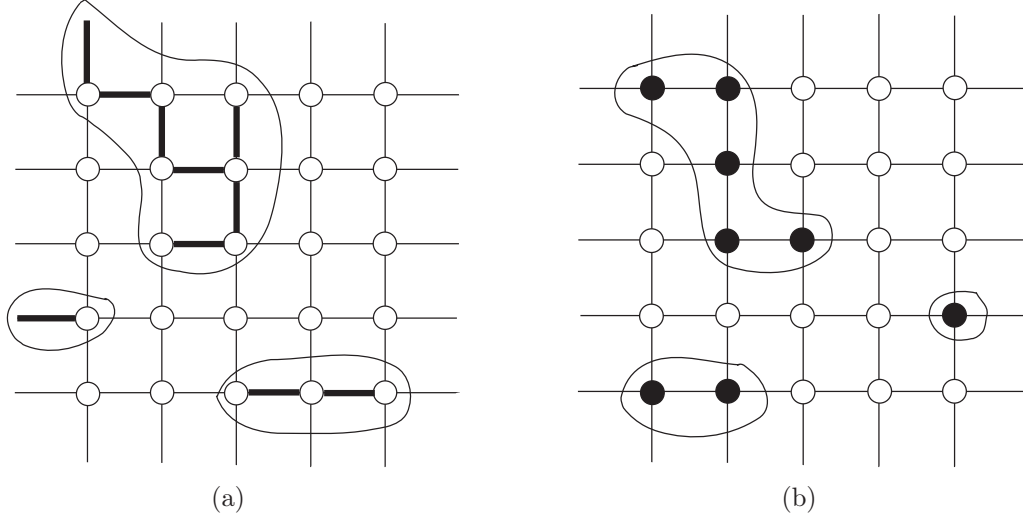


FIG. 2.2.1 – (a) Percolation par liens. On remarque qu’il y a plusieurs clusters dont un qui ne possède qu’un élément.

(b) Percolation par sites. Ici aussi, il y a plusieurs cluster dont un qui ne possède qu’un élément.

Soient $A, B \subset \mathbb{Z}^d$. On note $A \leftrightarrow B$ s’il existe un chemin ouvert joignant un sommet de A à un sommet de B . Cette notation est valable pour chaque éléments de A et B . Nous pouvons donc mieux exprimer notre cluster $C(x)$ en utilisant la notion de chemin. Au aura

$$C(x) = \{y \in \mathbb{Z}^d \mid x \leftrightarrow y\}$$

2.3 Le phénomène critique

On cherche désormais la probabilité p nous permettant d’obtenir des clusters de taille infinie. Il est aussi utile de savoir s’il y a plusieurs clusters de ce type.

On dira qu’il y a percolation si notre graphe \mathbb{L}^d possède un cluster de taille infinie(i.e. $|C| = \infty$).

On introduit la fonction $\theta_x(p) : [0, 1] \rightarrow [0, 1]$ comme *probabilité de percolation* (i.e. $\theta_x(p) = \mathbb{P}_p(|C(x)| = \infty)$).

Elle indique à partir de quelle probabilité p , le sommet x , appartient à un cluster de dimension infinie.

On omet souvent d’écrire le sommet x et on prend l’origine. On note $\theta(p) = \mathbb{P}_p(|C| = \infty)$.

Définition. La *probabilité critique* $p_c(d)$ est définie comme

$$p_c(d) := \sup\{p \mid \theta(p) = 0\}.$$

Remarque. Plus généralement pour n’importe quel graphe G on omet d’écrire la dimension d et on écrit p_c . Ainsi on peut redéfinir la probabilité de percolation comme suit :

$$\theta(p) = \begin{cases} = 0 & \text{si } p < p_c \\ > 0 & \text{si } p > p_c \end{cases}$$

Lorsque l'on parle de percolation par site on note la probabilité critique, $p_c^{\text{site}}(d)$ et pour la percolation par lien, $p_c^{\text{lien}}(d)$. Lorsqu'on ne précise pas, on a que

$$p_c(d) = p_c^{\text{site}}(d) = p_c^{\text{lien}}(d).$$

Remarque. Dans le cas ou notre graphe est \mathbb{L}^1 , il est clair que $p_c(1) = 1$, sinon il n'y a pas de chemin de $-\infty$ vers ∞ .

Pour $d \geq 2$ on remarque que \mathbb{L}^d s'injecte dans \mathbb{L}^{d+1} et donc on a $p_c(d+1) \leq p_c(d)$. Avec ceci on obtient le théorème suivant.

Théorème. *Pour $d \geq 2$ on a,*

$$0 < p_c(d) < 1$$

Démonstration.

$$\boxed{0 < p_c(d)}$$

On montre que pour un certain $p > 0$ (i.e proche de 0) on a $\mathbb{P}_p(0 \leftrightarrow \infty) = 0$.

On considère $\sigma(n)$ le nombre de chemins partant de l'origine, de longueur n ne passant qu'une fois sur ses sommets. Soit $N(n)$ le nombre de chemins ouverts de $\sigma(n)$ (ce nombre est aléatoire). On a ainsi

$$\mathbb{P}_p(0 \leftrightarrow \infty) \leq \mathbb{P}_p(N(n) \geq 1) \leq \mathbb{E}_p[N(n)] = p^n \sigma(n).$$

Cependant, $\sigma(n) \leq 2d(2d-1)^{n-1}$. En effet, il y a $2d$ possibilités de chemin pour le premier sommet et ensuite $2d-1$ pour les autres (voir avec un schéma). Donc

$$\mathbb{P}_p(0 \leftrightarrow \infty) \leq p^n (2d(2d-1)^{n-1}) = \frac{2d}{2d-1} (p(2d-1))^n.$$

Pour $0 < p < \frac{2d}{2d-1}$ on a bien que $\mathbb{P}_p(0 \leftrightarrow \infty)$ tend vers 0 quand n tend vers l'infini.

$$\boxed{p_c(d) < 1}$$

Comme énoncé plus haut, on sait que \mathbb{L}^d s'injecte dans \mathbb{L}^{d+1} et donc que $p_c(d+1) \leq p_c(d)$. On pourra ainsi ne considérer que le cas où $d = 2$. La notion de graphe dual va nous être utile dans cette démonstration. Le graphe dual de \mathbb{L}^2 est

$$\mathbb{L}_d^2 = \left(\mathbb{Z}^2 + \left(\frac{1}{2}, \frac{1}{2}\right), E^2 + \left(\frac{1}{2}, \frac{1}{2}\right) \right)$$

Une arête ouverte (resp. fermée) sur le graphe dual croise une arête fermée (resp. ouverte) de \mathbb{L}^d .

Ainsi, la probabilité qu'une arête soit ouverte sur le dual est $1 - p$.

A un graphe aléatoire $G = (V, E)$ de \mathbb{L}^2 on associe son graphe dual $G' = (\mathbb{Z}^2 + (\frac{1}{2}, \frac{1}{2}), E')$ où les arêtes ouvertes de G' (donc les éléments de E') sont celles de \mathbb{L}^2 coupant les arêtes fermées de G .

Nous utilisons un théorème (sans démonstration) qui dit qu'une composante connexe

ouverte contenant 0 dans G est finie si et seulement si il existe un circuit ouvert de G' entourant 0.

En désignant A_γ comme l'événement " γ est un circuit ouvert de \mathbb{L}_d^2 " on a

$$\begin{aligned} 1 - \mathbb{P}_p(0 \leftrightarrow \infty) &= \mathbb{P}_p(0 \leftrightarrow \infty) \\ &= \mathbb{P}_p\left(\bigcup_{\gamma \text{ circuit ouvert entourant } 0} A_\gamma\right) \\ &\leq \sum_{\gamma \text{ circuit ouvert entourant } 0} \mathbb{P}_p(A_\gamma) \end{aligned}$$

De même qu'auparavant, nous avons une arête ouverte dans le dual avec une probabilité $p - 1$, ainsi pour un circuit γ de longueur n on aura que

$$\mathbb{P}_p(A_\gamma) = (1 - p)^n.$$

Soit maintenant $\rho(n)$ le nombre de circuits de longueur n entourant 0. On remarque que $\rho(n) \leq n3^n$. En effet, il y a n possibilités de chemin et à chaque sommet il y a trois directions possibles.

En remplaçant ceci dans notre équation nous obtenons

$$1 - \mathbb{P}_p(0 \leftrightarrow \infty) \leq \sum_{n=1}^{\infty} n3^n(1 - p)^n.$$

En choisissant $p < 1$ (i.e proche de 1) de manière à ce que

$$\sum_{n=1}^{\infty} n3^n(1 - p)^n < 1$$

on obtient que $1 - \mathbb{P}_p(0 \leftrightarrow \infty) < 1$, donc que $\mathbb{P}_p(0 \leftrightarrow \infty) > 0$ □

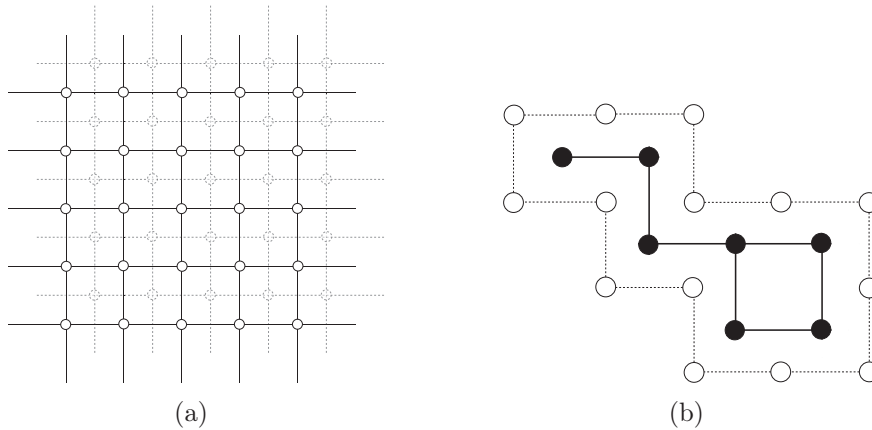


FIG. 2.3.1 – (a) Graphe dual de \mathbb{L}^2 . (b) Circuit ouvert (en blanc) entourant un cluster ouvert (en noir).

2.4 Théorème de Kesten et valeurs critiques

Ici nous allons présenter (sans démonstration) le théorème de Kesten qui donne une valeur exacte pour $p_c^{\text{lien}}(2)$ ainsi que différentes approximations de p_c pour des graphes différents.

Sur \mathbb{Z}^2 on voit qu'un sommet possède 4 arêtes. Il est naturel de se dire que pour qu'il y ait percolation il faut qu'au moins deux des quatre arêtes se touchent. En utilisant le rapport du nombre total d'arêtes sur le nombre d'arêtes utiles à la percolation, nous en déduisons qu'il faut que la probabilité soit plus grande ou égale à $\frac{1}{2}$. Ceci est une première approche pour déterminer la probabilité critique. Cependant, il nous faut le théorème de Kesten pour le constater

Théorème. (*Théorème de Kesten*)

Pour la percolation par lien et $p \geq \frac{1}{2}$, on a

$$\mathbb{P}_p (\text{il existe un cluster de taille infinie}) = 1$$

Démonstration. La démonstration demandant des outils assez techniques, le lecteur est renvoyé à la démonstration disponible dans [\[Bollobàs06\]](#)[p. 67] \square

Nous donnons ici quelques valeurs critiques (tirées de [\[Stauffer94\]](#)[p.17]) pour différents graphes.

Graphe	Site	Lien
\mathbb{L}^2	0.592746	0.5
\mathbb{L}^3	0.3116	0.2488
\mathbb{L}^4	0.197	0.1601
Triangulaire	0.5	0.34729
Diamand	0.43	0.388

2.5 Approximation numérique de la probabilité critique

Nous avons vu qu'il est possible de déterminer la valeur de la probabilité critique de manière théorique. Cela reste, cependant, délicat.

Pour palier à ce problème, nous utilisons des modèles numériques puissants donnant de bonnes approximations de p_c .

En considérant la probabilité de percolation $\theta(p)$ on sait que sous p_c la probabilité est nulle et qu'elle devient positive au dessus de p_c . Il devient alors intéressant de faire varier p de manière à faire varier $\theta(p)$ et ceci pour différentes tailles de sous-graphes de \mathbb{L}^d .

Pour chaque simulation d'un sous-graphe de taille $L \times L$ on définit

$$X_i(\omega) = \mathbb{1}_{\{|C|=\infty\}}(\omega) \quad , \quad S_n = X_1 + \dots + X_n,$$

ainsi pour un grand nombre de simulations, on obtiendra (par la loi des grands nombres) ceci

$$\frac{S_n}{n} \rightarrow \mathbb{E}_p[X_1] = \mathbb{E}_p[\mathbb{1}_{\{|C|=\infty\}}(\omega)] = \mathbb{P}_p(|C| = \infty).$$

Ainsi, on obtient plusieurs valeurs de $\theta(p)$ sur lesquelles nous pouvons effectuer une interpolation et obtenir une courbe.

Pour chaque L on aura définie une courbe donnant l'allure de $\theta(p)$.

Tout cela pris en considération, il ne nous reste plus qu'à examiner le point d'intersection de toutes nos interpolations pour avoir une approximation de p_c . La figure ci-dessous (tirée de [Wanzeller04]) nous montre une approximation de $p_c^{\text{site}(2)}$

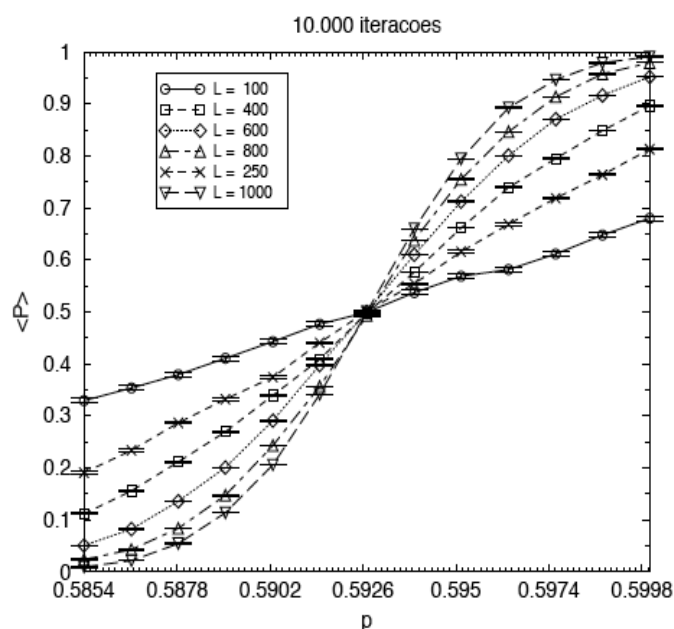


FIG. 2.5.1 – Approximation de la probabilité critique

Chapitre 3

Loi 0-1 de Kolmogorov, Inégalité FKG, Théorème d'existence et unicité

Nous avons vu au chapitre précédent que la probabilité de percolation $\theta(p)$ est une fonction croissante de p . Il est clair que plus la probabilité d'être ouvert est grande, plus il y aura de chance d'avoir un cluster de taille infinie.

Nous savons qu'il est possible d'avoir un cluster ouvert de taille infinie. Nous cherchons maintenant à connaître le nombre de clusters de ce type et leur comportement.

Nous utiliserons ici la notion d'ordre partiel évoquée précédemment pour introduire la définition d'événement croissant dans la tribu engendrée.

Définition. Un événement $A \in \mathcal{F}$ est appelé *croissant* si

$$\mathbb{1}_A(\omega_1) \leq \mathbb{1}_A(\omega_2) \quad \text{pour} \quad \omega_1 \leq \omega_2, \omega_1, \omega_2 \in \Omega.$$

A est *décroissant* si A^c est croissant.

On peut étendre cette notion aux variables aléatoires réelles de (Ω, \mathcal{F}) . Une variable aléatoire X est croissante si $X(\omega_1) \leq X(\omega_2)$ pour $\omega_1 \leq \omega_2$. X est décroissante si $-X$ est croissante.

3.1 Loi 0-1 de Kolmogorov

Avant d'introduire la loi, il nous faut voir la définition d'événement queue.

Définition. Soit $(\Omega, \mathcal{F}, \mathbb{P})$ un espace probabilisé. Soit $(X_i)_{i \in I}$ une suite de variables aléatoires i.i.d telles que $\mathcal{F} = \sigma(X_1, X_2, \dots)$. Un *événement queue* est un événement déterminé par les X_i mais qui est indépendant de toute sous-suite finie de variables X_i (i.e si l'événement est dans $\mathbb{B}^{(\infty)}$).

Par exemple lorsque l'on tire à pile ou face, le fait que l'événement $\{5 \text{ "pile" sortent à la suite}\}$ soit réalisé une infinité de fois fait que cet événement est un événement queue.

Nous pouvons maintenant introduire la loi.

Théorème. (*Loi 0 – 1 de Kolmogorov*)

Soient $(X_i)_{i \in I}$ des variables aléatoires indépendantes. Soit \mathcal{F} la tribu engendrée par les X_i . Soit $A \in \mathcal{F}$ un événement queue. Alors

$$\mathbb{P}(A) = 0 \quad \text{ou} \quad \mathbb{P}(A) = 1$$

Démonstration. Soit A un événement queue, (i.e $A \in \mathbb{B}^{(\infty)}$).

On a par définition que $\mathbb{B}^{(\infty)} \subset \mathbb{B}^{(n+1)}$ et aussi que $\mathbb{B}_n = \sigma(X_1, X_2, \dots, X_n)$.

Pour tout $n \geq 1$, A est indépendant de $\bigcup_n \mathbb{B}_n$. On montre (sans démonstration) que l'ensemble \mathcal{A} des parties indépendantes de A est une classe monotone, ainsi $\bigcup_n \mathbb{B}_n \in \mathcal{A}$.

On considère maintenant \mathbb{B} , la σ -algèbre engendrée par $\bigcup_n \mathbb{B}_n$.

Par définition, \mathbb{B} est indépendante de A mais aussi par définition, $A \in \mathbb{B}$, donc A est indépendant de lui-même. Ainsi

$$\mathbb{P}(A) = \mathbb{P}(A \cap A) = \mathbb{P}(A) \mathbb{P}(A) = \mathbb{P}(A)^2.$$

Donc $\mathbb{P}(A) = 0$ ou $\mathbb{P}(A) = 1$. □

3.2 Inégalité FGK

Si on sait qu'il existe un chemin ouvert joignant un sommet a à un sommet b , il devient plus probable qu'il existe un chemin ouvert joignant un sommet x à un sommet y . En termes d'événements croissant, on parle de corrélation positive.

Harris a été le premier à donner une formulation mais celle-ci a été largement modifiée et on ne retient maintenant que les noms de Fortuin, Kasteleyn et Ginibre.

Théorème. (*Inégalité FKG*)

Si A, B sont des événements croissants alors

$$\mathbb{P}(A \cap B) \geq \mathbb{P}(A) \mathbb{P}(B)$$

Démonstration. Voir [Grimmet99] □

Cette inégalité est particulièrement intéressante quand elle est utilisée avec la probabilité de percolation. En effet, pour deux sites x et y distincts, séparés d'une distance d , on sait que $\{x \leftrightarrow y\} \cap \{y \leftrightarrow \infty\} \subset \{x \leftrightarrow \infty\}$.

En utilisant la mesure de probabilité on a $\mathbb{P}_p(\{x \leftrightarrow \infty\}) \geq \mathbb{P}_p(\{x \leftrightarrow y\} \cap \{y \leftrightarrow \infty\})$ et donc par l'inégalité FKG

$$\mathbb{P}_p(\{x \leftrightarrow \infty\}) \geq \mathbb{P}_p(\{x \leftrightarrow y\}) \mathbb{P}_p(\{y \leftrightarrow \infty\})$$

i.e

$$\theta_x(p) \geq \mathbb{P}_p(\{x \leftrightarrow y\}) \theta_y(p)$$

ainsi $\theta_x(p) = 0$ pour tout x ou est plus grand que 0 pour tout autres sites.

3.3 Théorème d'existence

On connaît la probabilité qu'un sommet appartienne à un cluster ouvert de taille infinie. La question qui se pose est de savoir s'il existe un tel cluster.

Définition. Soit $0 < p < 1$. On définit

$$\psi(p) = \mathbb{P}_p (\text{ il existe un cluster ouvert de taille infinie})$$

Avec cette définition, nous pouvons maintenant donner le théorème d'existence d'un cluster ouvert de taille infinie.

Théorème. *La probabilité $\psi(p)$ qu'il existe un cluster ouvert de taille infinie satisfait*

$$\psi(p) = \begin{cases} 1 & \text{si } p > p_c \\ 0 & \text{si } p < p_c \end{cases}$$

Démonstration. Nous considérons que l'événement $\{\mathbb{L}^d \text{ contient une infinité de clusters ouverts}\}$. Cet événement ne dépend pas des états d'une collection finie d'arêtes.

Par la loi 0 – 1 de Kolmogorov, ψ prend comme valeur, 0 ou 1.

En effet, si $\theta(p) = 0$ alors

$$\psi(p) \leq \sum_{x \in \mathbb{Z}^d} \mathbb{P}_p (|C(x)| = \infty) = 0.$$

Par ailleurs, si $\theta(p) > 0$ alors

$$\psi(p) \geq \mathbb{P}_p (|C| = \infty) > 0$$

et par la loi 0 – 1 on a que $\psi(p) = 1$. □

La question est maintenant de savoir s'il existe un unique cluster ouvert de taille infinie. Pour répondre à cette question nous n'énoncerons que le théorème de Burton & Keane car la preuve est très technique.

Théorème. *Pour $0 < p < 1$,*

$$\mathbb{P}_p (\text{ il existe exactement un cluster ouvert de taille infinie }) = 1$$

Démonstration. Voir [Grimmet99][p.198] □

Chapitre 4

Applications de la percolation

Nous allons ici montrer un aperçu des possibilités de modélisation qu'offre la théorie de la percolation. Comme vous le verrez, il est parfois plus utile d'utiliser un modèle de percolation par lien ou par site pour étudier certains phénomènes.

4.1 Feux de forêt

On souhaite regarder le mouvement d'un feu dans une forêt. En effet, il est intéressant de voir si le feu peut brûler une grande partie de la forêt, ou tous les arbres de la forêt. Ainsi, au lieu d'effectuer une batterie de tests en grandeur nature, qui serait (comme vous l'imaginez) une catastrophe, on peut modéliser une forêt comme une partie de \mathbb{L}^2 de taille L , sur laquelle on place des arbres avec une certaine probabilité p .

On dira qu'un élément de \mathbb{L}^2 possède un arbre et on lui attribue la valeur 1 avec probabilité p et on dira qu'il est libre en lui attribuant la valeur 0 avec probabilité $1 - p$.

On regarde ensuite quel est le cluster qui part du haut vers le bas ou de gauche à droite.

4.2 Percolation orientée

Lorsqu'on regarde de près par exemple une résistance sur un circuit électrique, on remarque qu'il y a un réseau aléatoire dessus. De la même manière, si on prend une éponge, il y a un réseau aléatoire de canaux qui transporte l'eau. Dans le premier cas, le courant électrique doit passer dans le circuit dans un certain sens tout comme l'eau (par l'effet de l'apesanteur) doit circuler dans l'éponge, il y a donc une orientation dans le réseau. La percolation orientée consiste donc à l'étude des phénomènes de percolation sur des graphes aléatoires possédant une orientation, i.e, où les arêtes possèdent une direction.

4.3 Diffusion d'une épidémie

Pour prévenir la dispersion d'épidémies, il est utile de regarder le comportement asymptotique de celles-ci. La percolation est ici utilisée comme dans le cas des feux de forêt. On considère un site $x \in \mathbb{Z}^2$ comme un individu et une fonction d'état $\xi_t(x)$

donnant l'état de x au temps t .

Les états sont :

- 1 , l'individu est en bonne santé.
- i , l'individu est infecté.
- 0, l'individu est immunisé.

Lorsqu'un individu est infecté il propage des germes (suivant une loi de Poisson d'intensité α) avec une certaine probabilité à ses voisins directs (i.e. $x + (1, 0)$, $x + (0, 1)$, $x + (-1, 0)$, $x + (0, -1)$).

Si le germe va sur un site en bonne santé, celui-ci devient immédiatement infecté et commence à son tour à propager des germes.

Un individu qui est infecté, le sera durant une période de temps suivant une loi possédant une certaine distribution F . On peut imaginer que soit l'individu combat le virus (par exemple pour de petits virus comme la grippe) ou alors il décède.

On peut dès à présent étudier le comportement de chaque site x . Le temps moyen d'infection d'un site, la probabilité qu'un existe un cluster infecté après un certains temps...etc. Une bonne interprétation de ce phénomène ainsi que certains résultats sont donnés dans [\[Durrett88\]](#).

Chapitre 5

Simulation de feu de forêt

Ici nous reprenons l'exemple d'application sur les feux de forêt. En effet, il est intéressant d'étudier une application concrète de l'apport de la percolation au sujet qui nous intéresse.

5.1 Modélisation

5.1.1 Modélisation du problème

Nous modélisons le problème de la manière suivante :

- Percolation par site sur \mathbb{Z}^2 .
- Chaque arbre est un sommet du graphe.
- La taille L de la forêt peut varier de 0 à 500.

Pour simplifier les choses notre graphe sera un tableau bi-dimensionnel où chaque élément est une case pouvant contenir un arbre ou non.

5.1.2 Algorithme

L'algorithme est le suivant

1. On place les arbres de manière aléatoire sur notre graphe. On définit $X = 0$.
2. On met le feu de manière aléatoire à un arbre du graphe.
3. On va sur chaque case, dès qu'il y a un arbre dedans on observe s'il y a un arbre en feu directement à côté. Si c'est le cas l'arbre courant prend feu et on ajoute 1 à X .
4. On recommence le troisième point en parcourant à nouveau tout le tableau jusqu'à ce que X redevienne nul.

5.1.3 Problèmes rencontrés

La première difficulté rencontrée est la façon de déterminer une probabilité de Bernoulli. En effet, les ordinateurs peuvent tirer des nombres pseudo-aléatoires distribués uniformément sur $(0, 1)$ mais ils ne savent pas tirer soit 0 soit 1 avec une probabilité p et respectivement $1 - p$.

Pour contourner cette difficulté, il faut redéfinir une nouvelle variable aléatoire Y , utilisant la réalisation x de la variable aléatoire Uniforme(0, 1) X avec

$$Y = \begin{cases} 1 & \text{si } x \leq p \\ 0 & \text{si } x > p \end{cases}$$

où p est la probabilité que nous choisissons pour la simulation.

Un petit calcul montre que notre variable aléatoire Y suit bien un loi de Bernoulli.

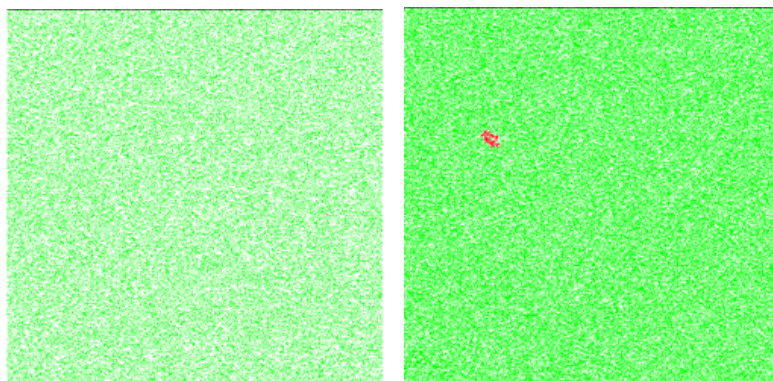
La seconde difficulté rencontrée est de déterminer si une fois les arbres placés, il y a un cluster dans notre graphe. Il n'y a malheureusement pas de subtilité informatique pour trouver de manière efficace une solution à cette difficulté.

Pour ne pas surcharger le programme en calcul, il a été décidé de placer aléatoirement un arbre en feu dans le graphe et d'observer le comportement de ses voisins après coup.

5.2 Résultats

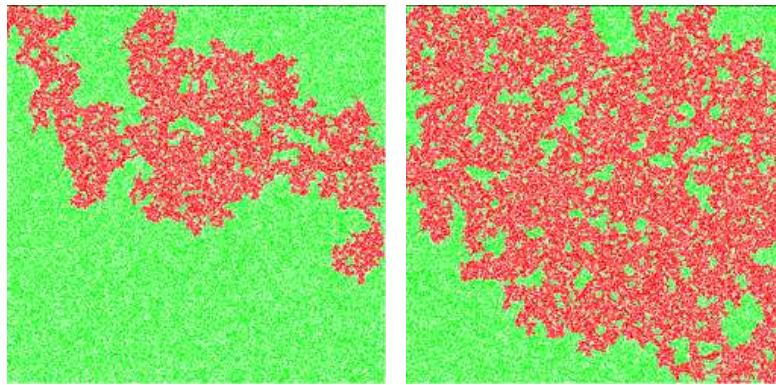
Une fois la simulation réalisée pour différentes valeurs de p_c il est très intéressant de voir des clusters apparaître quand $p > p_c$.

On remarque sur les figures ci-dessous qu'en s'approchant de la probabilité critique, la taille ainsi que la forme des clusters varient énormément.



(a) 0.3

(b) 0.55



(c) 0.59

(d) 0.6

On remarque que pour 0.59 il y'a presque un cluster, en effet le feu se propage de gauche à droite mais pas de haut en bas.

Bibliographie

- [Bollobàs06] Bollobàs B. (2006) *Percolation*, Cambridge University Press, Cambridge.
- [Diestel05] Diestel R. (2005) *Graph Theory, Electronic Edition*. Springer-Verlag, New York.
- [Durrett88] Durrett R. (1988) *Lecture notes on particle systems and percolation*, Wadsworth & Brooks
- [Durrett96] Durrett R. (1996) *Probability, Theory and Examples, Seconde Edition*, Duxbury Press
- [Gabriel08] Gabriel J-P. (2008) *Cours de théorie de la mesure*. Université de Fribourg, Fribourg.
- [Legall06] Le Gall J-F. (2006) *Intégration, Probabilités et Processus Aléatoires*. Ecole Normale Supérieure FIMFA, Paris.
- [Grimmet99] Grimmet J. (1999) *Percolation theory*. Springer-Verlag, New York.
- [Mazza08] Mazza C. (2008) *Cours de probabilités*. Université de Fribourg, Fribourg.
- [Stauffer94] Stauffer D. & Aharony A. (1994) *Introduction to percolation theory, revised second edition*. CRC Press, Florida.
- [Wanzeller04] Wanzeller W.G. , Cucchieri A. , Mendes T. , Krein G. *Percolation of Monte Carlo Clusters*, Brazilian Journal of Physics, vol 34, no. 1A, March 2004

Annexe A

Code Java

```
import java.awt.*;
import java.awt.geom.*;
import java.util.*;
import java.awt.event.*;
import javax.swing.*;
import java.awt.Color.*;
import java.lang.Math.*;
import java.lang.*;

class Cell {
public Cell(int x,int y, boolean Occupied){
    this.x = x;
    this.y = y;
    IsOccupied = Occupied;
    if(IsOccupied)
        {
        Status = 1;
        }
    else{
        Status = 0;
        }
    NumberOfCheck = 0;
    HasBeenChecked = false;
}
public void setStatus(int S){ Status = S;}
public int getStatus(){ return Status;}
public int getSpecialStatus(){
    int S=0;
    if(Status == 2) S=1;
    return S;
}
public void Check(){ HasBeenChecked = true;}

public void addNumberOfCheck(){NumberOfCheck++;}
public int getNumberOfCheck(){ return NumberOfCheck; }
public int getX(){ return x; }
public int getY(){ return y; }

private int Status;
private int NumberOfCheck;
private boolean IsOccupied;
private boolean HasBeenChecked;
private int x,y;
}

class Coord{
public Coord(int x, int y){
    this.x = x;
    this.y = y;
}

public int getX(){ return x;}
public int getY(){ return y;}

private int x,y;
}

class Lattice{
public Lattice(int L, double P, Coord Start){
```

```

    this.P = P;
    this.L = L;
    StartPoint = Start;
    NumberOfOpen = 0;
    NumberOfTouched = 0;
    addCell();
    setStart(Start);
}
public void addCell(){
    Random R;
    Date D = new Date();
    R = new Random();
    R.setSeed((D.getTime()+((long)(Math.random()*1000))));
    int i,j;
    World = new Cell[L][L];
    for (i=0;i<L;i++){
        for (j=0;j<L;j++){
            boolean O = true;
            Double T = R.nextDouble();
            if ( T > P){
                O = false;
            }
            World[i][j]= new Cell(i,j,O);
            if(O){
                NumberOfOpen++;
            }
        }
    }
}

public void ShowLattice(){
    int i,j;
    for (i=0;i<L;i++){
        for (j=0;j<L;j++){
            System.out.print(World[i][j].getStatus()+"_");
        }
        System.out.println("\n");
    }
}

public String ShowStringLattice(){
    int i,j;
    String T="";
    for (i=0;i<L;i++){
        for (j=0;j<L;j++){
            T += World[i][j].getStatus()+"_";
        }
        T += '\n';
    }
}

return T;
}

public int checkCell(int i, int j){
    int NumOfN=0;

    if (i==0 && j==0) {
        NumOfN = World[0][1].getSpecialStatus() +
                World[1][0].getSpecialStatus();
    }

    else if (i==L-1 && j==0) {
        NumOfN = World[L-1][1].getSpecialStatus() +
                World[L-2][0].getSpecialStatus();
    }

    else if (i==0 && j==L-1) {
        NumOfN = World[0][L-2].getSpecialStatus() +
                World[1][L-1].getSpecialStatus();
    }

    else if (i==L-1 && j==L-1) {
        NumOfN = World[L-2][L-1].getSpecialStatus() +
                World[L-1][L-2].getSpecialStatus();
    }

    else if (i==0 && j>0 && j <L-1){
        NumOfN = World[i][j-1].getSpecialStatus() +
                World[i][j+1].getSpecialStatus() +
                World[i+1][j].getSpecialStatus();
    }

    else if (i==L-1 && j>0 && j <L-1){
        NumOfN = World[i-1][j].getSpecialStatus() +
                World[i][j-1].getSpecialStatus() +
                World[i][j+1].getSpecialStatus();
    }

    else if (j==0 && i>0 && i <L-1){
        NumOfN = World[i-1][j].getSpecialStatus() +
                World[i][j+1].getSpecialStatus() +
                World[i+1][j].getSpecialStatus();
    }

    else if (j==L-1 && i>0 && i <L-1){
        NumOfN = World[i-1][j].getSpecialStatus() +
                World[i][j-1].getSpecialStatus() +
                World[i+1][j].getSpecialStatus();
    }
}

```

```

        }
        else{
            NumOfN = World[i-1][j].getSpecialStatus () +
                    World[i][j-1].getSpecialStatus () +
                    World[i+1][j].getSpecialStatus () +
                    World[i][j+1].getSpecialStatus ();
        }
        return NumOfN;
    }
}
public void setStart (Coord SP){
    World[SP.getX ()][SP.getY ()].setStatus (2);
}
public void RunSimulation (){
    int k;
    do{
        k=0;
        for(int i=0;i<L;i++){
            for(int j=0;j<L;j++){
                if( World[i][j].getStatus()==1 && World[i][j].getNumberOfCheck() == 0 && checkCell(i,j) >=N){
                    World[i][j].setStatus (2);
                    World[i][j].Check ();
                    World[i][j].addNumberOfCheck();
                    NumberOfTouched++;
                }
                k++;
            }
        }
    }while(k!=0);
}

public boolean checkPath(int x, int y){
    if(World[x+1][y].getStatus()==1){
        return true;
    }
    else{
        return false;
    }
}

public boolean CheckCluster(){
    boolean HasCluster, isTrue;
    HasCluster = false;
    isTrue = false;
    int i=-1;
    do{
        i = i+1;
        if(i==(L-1)) break;
        System.out.println(i);
        for(int j=0;j<L;j++){
            isTrue = checkPath(i,j);
        }
    }while(isTrue);

    if(isTrue && i==L-1) HasCluster = true;
    return HasCluster;
}

public Cell[][] getWorld(){ return World;}
public int getNumberOfOpen(){return NumberOfOpen;}
public int getNumberOfTouched(){return NumberOfTouched;}

private int NumberOfOpen,NumberOfTouched;
private double W;
private int L, N=1;
private double P;
private Coord StartPoint;
private Cell[][] World = new Cell[L][L];
}
class Statistician{
public Statistician(double proba, int N, int size){
    this.proba = proba;
    this.N = N;
    this.size = size;
}
}
public double[] doMath(){
    double Store;
    double value = proba;
    for(int i=0;i<10;i++){
        Store = 0;
        System.out.println(i);
        for(int j=0;j<10;j++){
            x = (int)(Math.random()*size);
            y = (int)(Math.random()*size);
            Coord SP = new Coord(x,y);
            L = new Lattice(500, value ,SP);
            L.RunSimulation ();
            Store += ((double)L.getNumberOfTouched()/((double)L.getNumberOfOpen ());
        }
    }
}

```

```
        System.out.println("Average_for_proba_"+value+" : "+(Store/10));
        Result[i] = Store/10;
        value += 0.001;
    }
    return Result;
}
double Result[] = {0,0,0,0,0,0,0,0,0,0};
private double proba;
private int N, size, x, y;
private Lattice L;
}
public class percolation
{
    public static void main(String[] args)
    {
        System.out.println("Site_Percolation_(c)_2008_Pablo_CROTTI");
        Coord StartPoint = new Coord(0,0);
        Lattice N = new Lattice(10,0.5,StartPoint);
        N.RunSimulation();
        N.ShowLattice();
        System.out.println("End_of_Simulation");
    }
}
```