Estimarea Pragului Critic de Percolație în Rețele 2D folosind Metode Monte Carlo

Student: Ignat Marius Florentin
Facultatea de Matematică și Informatică

12 Ianuarie 2025

1 Descrierea Problemei și Justificarea Alegerii

Proiectul își propune studiul fenomenului de **percolație** pe o rețea bidimensională pătratică (*square lattice*). Percolația este un model canonic în fizica statistică, esențial pentru descrierea tranzițiilor de fază în sisteme dezordonate. Aplicațiile practice variază de la curgerea fluidelor prin medii poroase până la robustețea rețelelor de comunicații în fața cedării nodurilor.

Obiectivul Principal

Determinarea experimentală a **pragului critic de percolație (p_c)**. Acesta reprezintă valoarea limită a densității pentru care sistemul suferă o tranziție de fază geometrică, trecând de la o stare neconectată la una care permite trecerea (percolează).

Justificarea Metodei

Am ales această problemă deoarece, pentru o rețea pătratică 2D, **nu există o soluție analitică exactă** pentru determinarea pragului p_c, spre deosebire de alte topologii (ex. rețeaua triunghiulară). Valoarea acceptată în literatura de specialitate, $p_c \approx 0.592746$, a fost determinată exclusiv prin metode numerice [1]. Astfel, utilizarea simulării **Monte Carlo** este indispensabilă, fiind instrumentul standard definit de Metropolis și Ulam [2] pentru abordarea problemelor deterministe prin eșantionare statistică.

2 Formularea Matematică

Considerăm o rețea $L \times L$, reprezentată ca o matrice binară M, unde starea fiecărui sit (i, j) este o variabilă aleatoare Bernoulli independentă:

$$P(M_{i,j} = 1) = p \quad \text{(sit deschis)} \tag{1}$$

$$P(M_{i,j} = 0) = 1 - p \quad \text{(sit blocat)} \tag{2}$$

Definim variabila aleatoare indicator Π_p:

$$\Pi_p = \begin{cases} 1, & \text{dacă există un cluster infinit (drum continuu sus-jos)} \\ 0, & \text{altfel} \end{cases} \tag{3}$$

Cantitatea de interes este probabilitatea de percolație $\theta(p) = \mathbb{E}[\Pi_p]$. Scopul este identificarea valorii p pentru care funcția $\theta(p)$ prezintă cea mai abruptă creștere (tranziția de la 0 la 1).

3 Algoritmul Utilizat

Pentru estimare, am implementat un algoritm Monte Carlo în următorii pași:

1. **Generare:** Crearea unei grile aleatoare $L \times L$ pentru o probabilitate p fixată.
2. **Verificare:** Verificarea percolației utilizând un algoritm de tip **Flood Fill iterativ** (echivalent cu o parcurgere Breadth-First Search pe graful implicit al rețelei). Algoritmul propagă un "fluid" virtual din toate siturile deschise de pe prima linie către vecinii adiacenți, marcând celulele accesibile.
3. **Decizie:** Dacă la finalul propagării cel puțin un sit de pe ultima linie a fost marcat, sistemul este considerat percolant ($\Pi_p = 1$).

4 Deducerea Teoretică și Analiza Erorilor

Fiind o metodă stocastică, rezultatul simulării $\hat{\theta}_N(p)$ este o aproximare a valorii reale $\theta(p)$. Pentru a garanta rigoarea matematică a rezultatelor, utilizăm **Inegalitatea lui Hoeffding** [3], care oferă o margine superioară a probabilității ca estimarea să devieze de la valoarea reală cu mai mult de o eroare ϵ:

$$P(|\hat{\theta}_N(p) - \theta(p)| \geq \epsilon) \leq 2e^{-2N\epsilon^2} \tag{4}$$

Această relație ne permite să dimensionăm corect experimentul. De exemplu, pentru a asigura o marjă de eroare $\epsilon = 0.01$ cu un nivel de încredere de 95% ($P_{eroare} \leq 0.05$), impunem:

$$0.05 \geq 2e^{-2N(0.01)^2} \tag{5}$$

Rezolvând inegalitatea (prin logaritmare), rezultă necesitatea unui număr minim de simulări $N \approx 18.445$. Acest calcul teoretic validează numărul de iterații ales în implementarea noastră software pentru a asigura precizia necesară.

5 Rezultate Experimentale și Vizualizare

În urma rulării simulării conform parametrilor deduși teoretic, am obținut curba sigmoidă caracteristică tranziției de fază (Figura 1). Se observă creșterea bruscă a probabilității de percolație în jurul valorii critice $p_c \approx 0.59$.

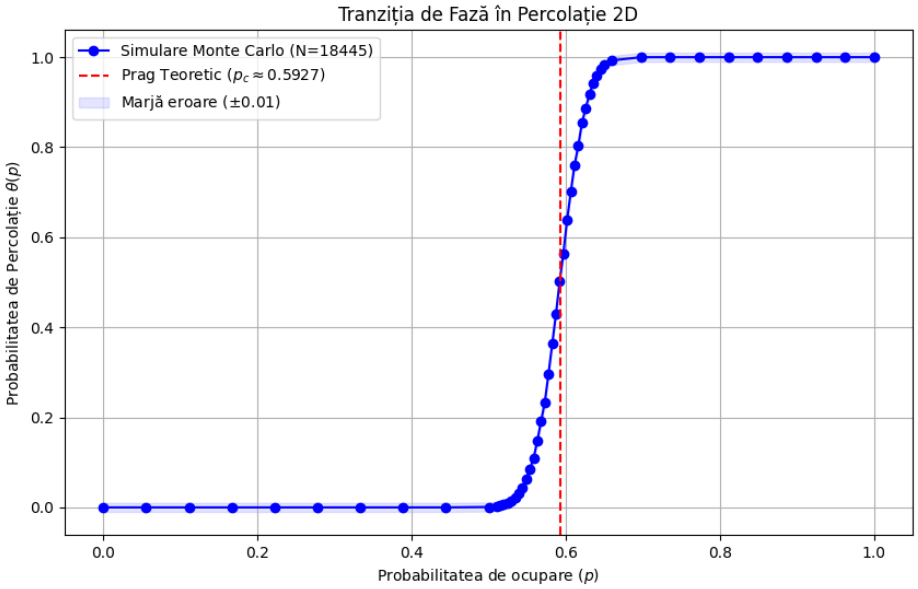

Figura 1: Estimarea Monte Carlo a probabilității de percolație $\theta(p)$. Zona albastră reprezintă marja de eroare teoretică (Hoeffding).

De asemenea, am generat vizualizări comparative ale stării grilei (Figura 2) pentru a valida comportamentul sistemului sub și peste pragul critic.

a) $p = 0.4$ (Nu percolează) b) $p = 0.6$ (Percolează)

Figura 2: Comparație vizuală între o configurație sub-critică (stânga) și una supra-critică (dreapta). Celulele albastre indică drumul fluidului.

Anexa A: Documentația Tehnică a Implementării

1. Structura Codului și Funcțiile Principale

Simularea a fost implementată în limbajul Python (v3.x), utilizând bibliotecile `numpy` pentru manipularea matricelor și `matplotlib` pentru vizualizarea datelor. Arhitectura codului este

modulară, fiind compusă din trei funcții esențiale:

- `verifica_percolatie(L, p)`: Nucleul simulării.
  - *Intrări:* Dimensiunea grilei L și probabilitatea de ocupare p.
  - *Funcționalitate:* Generează o matrice booleană aleatoare și aplică un algoritm de parcurgere **Breadth-First Search (BFS/Flood Fill)** pentru a verifica existența unui drum continuu de la prima la ultima linie.
  - *Ieșire:* Valoare booleană (`True/False`) indicând starea de percolație.
- `calculeaza_n_hoeffding(epsilon, confidence)`: Modulul de rigoare statistică.
  - *Funcționalitate:* Implementează formula derivată din inegalitatea Hoeffding pentru a returna numărul minim de iterații necesare (N) care garantează marja de eroare ϵ cu nivelul de încredere specificat.
- `ruleaza_studiu_monte_carlo()`: Funcția principală (Driver).
  - *Funcționalitate:* Gestionează bucla de simulare, iterează prin intervalul de probabilități $p \in [0, 1]$ (cu o densitate mai mare de puncte în zona critică 0.59) și agregează rezultatele statistice.

2. Instrucțiuni de Utilizare

Pentru a replica rezultatele experimentale, urmați pașii:

1. Asigurați-vă că aveți instalat Python 3 și pachetele necesare: `pip install numpy matplotlib`.
2. Rulați fișierul sursă: `python main.py`.
3. Scriptul va calcula automat numărul necesar de simulări și va afișa progresul în consolă.
4. La final, se vor genera automat două ferestre grafice corespunzătoare rezultatelor prezentate în acest raport.

Bibliografie

[1] Stauffer, D., & Aharony, A. (1994). *Introduction to Percolation Theory* (2nd ed.). Taylor & Francis.

[2] Metropolis, N., & Ulam, S. (1949). "The Monte Carlo Method". *Journal of the American Statistical Association*, 44(247).

[3] Hoeffding, W. (1963). "Probability inequalities for sums of bounded random variables". *Journal of the American Statistical Association*, 58(301).