

Estudio de la percolación de sitio en una malla bidimensional

C. D. Herrera Medina,^{*} A. Y. Bautista Sierra,^{**} A. S. Gómez Fonseca^{***}

Departamento de Física, Facultad de Ciencias, Universidad Nacional de Colombia

Mayo 19 de 2020

Introducción

La teoría de la percolación surgió en el año 1957 cuando se trataba de estudiar el comportamiento de un fluido a través de un cuerpo poroso [2]. Allí, el cuerpo poroso representa una estructura desordenada y su morfología juega un papel importante en la dinámica del fluido, puesto que efectos debidos a la topología (interconexión entre los elementos microscópicos dentro de la estructura) y la geometría (forma y tamaño de los elementos microscópicos), van a afectar el movimiento del fluido en el cuerpo desde un punto a otro [4].

El movimiento del fluido dentro del medio involucra elementos aleatorios que pueden ser atribuidos a dos mecanismos. Por un lado se considera que las partículas del fluido son las que deciden cuál camino tomar dentro del medio, esto corresponde al proceso de difusión. Por otro lado, se considera que la aleatoriedad viene del medio, el desorden en su morfología especifica cuál camino toman las partículas. Esta última situación fue estudiada por Broadbent y Hammersley en 1957 y fue llamada *proceso de percolación*. [4]

En dos dimensiones su modelo consiste en una red de puntos cuadrada y un número p que satisface $0 \leq p \leq 1$. Se examina cada uno de los bordes entre dos puntos contiguos y se establece, que el borde estará abierto con probabilidad p y cerrado con probabilidad $(1 - p)$. Los bordes representan los ca-

minos hacia el interior del cuerpo y la probabilidad p la proporción de pasajes que serán lo suficientemente anchos como para dejar pasar el fluido a través de ellos. El cuerpo es una gran parte finita de esta red cuadrada. En la figura 1 se puede ver una representación de lo descrito previamente, además se observa que el punto x estará mojado solamente si existe un camino desde este punto hasta un punto en la frontera del cuerpo conformado por bordes abiertos entre ellos. [3]

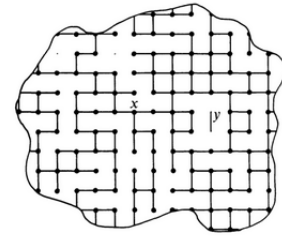


Figura 1: Estructura del cuerpo poroso bidimensional. En este diagrama se han omitido los bordes cerrados y los abiertos se muestran con líneas. Imagen tomada de [3]

Ahora bien, la teoría de la percolación clásica se centra en dos problemas principalmente: *bond percolation* y *site percolation*. El primer problema fue el descrito previamente, en el que los bordes entre dos puntos estarán abiertos al paso del fluido aleatoriamente e independientemente de sus vecinos con probabilidad p , o de otro modo estarán cerrados con probabilidad $(1 - p)$.

De forma similar, en el problema de percolación de sitio, se consideran los vértices de la red como

^{*}cdherreram@unal.edu.co

^{**}aybautistas@unal.edu.co

^{***}asgomezf@unal.edu.co

las entidades relevantes. Ciertos sitios de la malla están ocupados con probabilidad p y desocupados con probabilidad $(1 - p)$. Dos vecinos cercanos estarán conectados si ambos están ocupados, y así, un conjunto de sitios ocupados conectados conformará un *clúster*. Si la red es muy grande y la probabilidad p es pequeña, el tamaño de cualquier clúster será pequeño. Sin embargo, si la probabilidad p es cercana a 1, la red debería estar enteramente conectada, a excepción de pocos sitios. [4] Los clústers percolantes serán entonces aquellos que estén conectados de una frontera a otra a través de un camino de sitios ocupados entre ellos. En la figura 2 se muestra la diferencia entre los dos problemas de percolación.

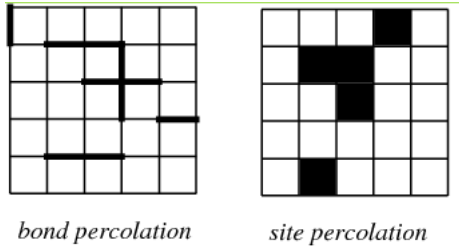


Figura 2: Diferencia entre la percolación de sitio y percolación de borde. Imagen tomada de math-world.wolfram.com/PercolationTheory.html

Debe existir entonces una probabilidad crítica p_c sobre la cual un clúster infinito ocupará toda la red. Desde que la probabilidad P de que un clúster infinito exista es 0 ó 1, según Kolmogorov, esta probabilidad es una función creciente que depende de la probabilidad de llenado de la red p . Por lo tanto, la probabilidad crítica p_c define el punto por debajo del cual la probabilidad siempre es cero o encima del cual la probabilidad siempre es uno [1]. En el caso de una red finita la probabilidad de que se forme un clúster percolante P en función de la probabilidad de llenado p , tenderá a una función paso escalón cuando el tamaño del sistema tiende a infinito.

El cálculo de la probabilidad crítica para sistemas tridimensionales se ha realizado con métodos de MonteCarlo. Ha sido posible mostrar que la probabilidad crítica p_c para ciertas redes, como la de Brethe, puede expresarse como:

$$p_c = \frac{1}{Z - 1}$$

donde Z es el número de coordinación, i.e, el número de vecinos más cercanos a un punto dado [4].

En el presente trabajo se pretende estudiar la percolación de sitio simulando el llenado aleatorio de una malla bidimensional. Para ello, se busca estudiar el comportamiento de la probabilidad $P(p, L)$ de que exista un clúster percolante, en función de la probabilidad de llenado p y del tamaño de la malla L . Además, se determina el tamaño promedio del clúster percolante más grande en función de p y de L .

Metodología

Con el fin de realizar la simulación del llenado aleatorio de una malla bidimensional, se construye varias funciones con objetivos específicos. En primer lugar, una función (*generacionMallaAleatoria*) que genera una matriz con valores aleatorios en función de la probabilidad p . Esto se realizó con las librerías *Eigen* y *random*, con el fin de asignarles valores aleatorios a las casillas con distribución binomial $Bin(1, p)$.

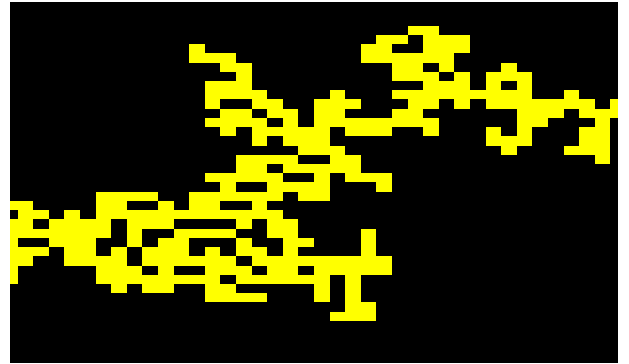


Figura 3: Malla de 40×40 sitios llenada con probabilidad $p = 0.55$

Posteriormente, teniendo en cuenta solamente los vecinos inmediatos (arriba-abajo-izquierda-derecha), la función *clusters* permite identificar clústers de vecinos con tamaños específicos que dependerán de la probabilidad p . La función identifica los elementos que son distintos de cero y les asigna un número mayor a 1 de forma recursiva hasta que se encuentra con un cero. Identificando de esta forma cada clúster con un número mayor a 1. En

las figuras 3 y 4 se muestra una visualización de un clúster que percola luego del proceso recursivo que realiza la función *clusters*.

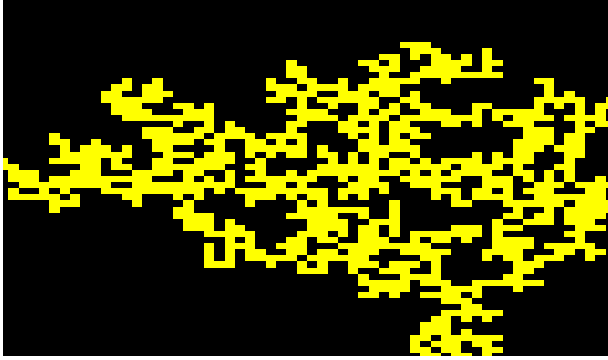


Figura 4: Malla de 60×60 sitios llenada con probabilidad $p = 0.55$

Para poder analizar el comportamiento de la probabilidad $P(p, L)$ y el tamaño del clúster más grande $s(p, L)$, la función *percolacion* determina qué mallas generaron clústers percolantes por medio de variables tipo *bool* que determinan si los elementos en la frontera de la matriz son diferentes de cero teniendo en cuenta a la función *clusters* en su ejecución. Adicionalmente, la función calcula el tamaño del clúster percolante más grande (y el identificador de dicho clúster, aunque no lo reporta).

Las simulaciones se desarrollaron para 30 valores de probabilidad $p \in (0.4, 0.7]$ y cinco tamaños de matrices diferentes $L = \{32, 64, 128, 256, 512\}$. Cada una de estas se realizó para 1600 semillas diferentes (tomando cada 20 semillas un valor promedio de tamaño de clúster percolante más grande y el valor de P) con el fin de reportar valores precisos de los promedios P y s , y sus respectivas desviaciones estándar.

Resultados y Análisis

A partir de los resultados obtenidos de las simulaciones se realiza la figura 5 donde se representa el comportamiento de la probabilidad P , que da cuenta de la existencia de un clúster percolante, en función de la probabilidad p y tamaño L .

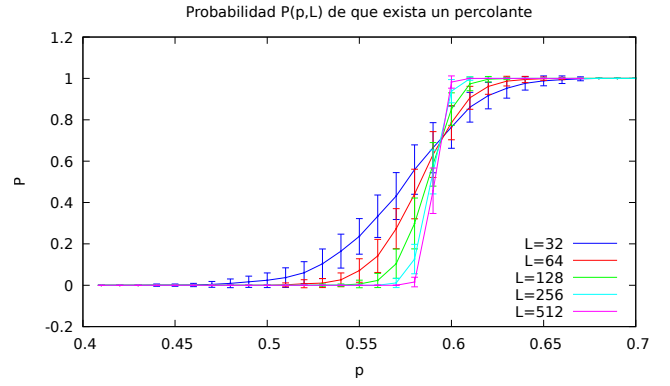


Figura 5: Dependencia de la probabilidad de existencia de un clúster P en función de la probabilidad de llenado p y tamaño de matriz $L \times L$

En la gráfica 5 se evidencia que entre más pequeño sea el tamaño de la matriz, los clústers percolantes empiezan a aparecer para valores más pequeños de p . Sin embargo, la probabilidad de existencia de percolación aumenta con menor rapidez en función de la probabilidad de llenado p . Note que el punto de inflexión de las curvas en la gráfica 5 aparece alrededor de la probabilidad crítica $p \approx 0.59271$, lo que coincide con lo esperado para una malla bidimensional.

También fue posible realizar una gráfica que representa el comportamiento del tamaño máximo promedio del clúster percolante en función de las variables p y L . Este fue normalizado al tamaño $L \times L$ de la matriz. Ver Figura 6.

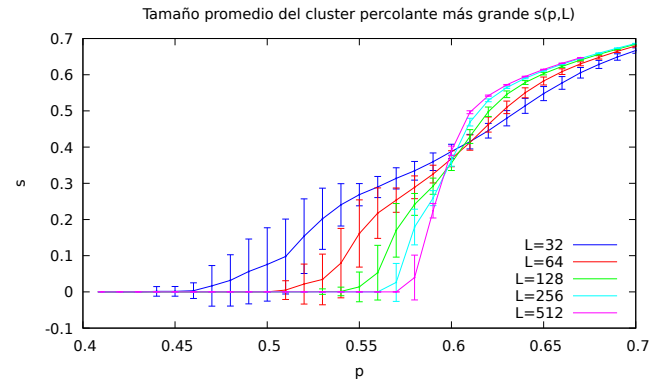


Figura 6: Tamaño promedio del clúster percolante más grande s en función de la probabilidad de llenado p y tamaño de matriz $L \times L$. Los tamaños se encuentran normalizados.

Allí se refleja que el tamaño máximo promedio

aumenta con mayor rapidez entre más grande sea la matriz. Además, en ambas gráficas se evidencia mayor dispersión en los datos entre más pequeña es la matriz.

Por otro lado, con el fin de evidenciar la influencia de dos niveles diferentes de optimización del compilador, se estudió el tiempo de cómputo para matrices de diferentes tamaños ($L \times L$) con las banderas de optimización -O2 y -O3. A partir de la medida del tiempo se realizó la gráfica que se muestra en la figura 7 que ilustra el tiempo de cómputo en función del tamaño del sistema (L).

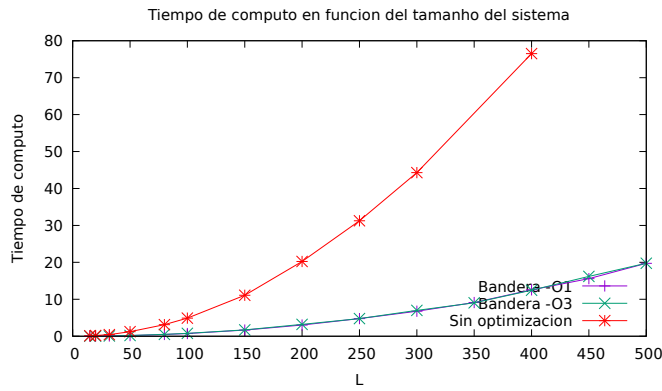


Figura 7: Tiempo de cómputo en función del tamaño del sistema para dos banderas de optimización

Se puede distinguir que el tiempo de cómputo es similar con ambas banderas de optimización, probablemente porque ambas desarrollan las mismas tareas de optimización para esta simulación. Además, es de resaltar, que el utilizar funciones recursivas para determinar los clústers e identificar aquellos que son percolantes hizo que la ejecución del programa fuera rápida para la cantidad de instrucciones que el programa debía realizar. Igualmente, el emplear la librería de C++ Eigen en la solución del problema, hace que no existan problemas de corrupción de memoria al no tener que utilizar arreglos bidimensionales.

La curva que representa el tiempo de cómputo sin ninguna bandera de optimización crece mucho más rápido que las dos curvas que representan el tiempo empleando optimización, se puede notar entonces que, a pesar de emplear librerías que ya están optimizadas y funciones recursivas, las banderas de optimización reducen considerablemente el tiempo

de ejecución y deben ser empleadas después de realizar debugging y profiling del código.

En cuanto a la comparación hecha por dos niveles de optimización en el caso de la probabilidad crítica ($p_c \simeq 0.59271$) se observa en el flat profile (reportado en el archivo *profiling-report.txt*) que la mayoría del tiempo (entre el 9 y el 10%) la función *clusters* consume gran parte del tiempo del programa, lo cual puede ocurrir debido a la recursión usada para la revisión de los clústers en el interior de la malla. La segunda función del programa escrito que consume mayor tiempo es *percolacion*, debido probablemente a que es la función que suele llamar a *clusters* y en donde se debe reportar un mayor tiempo a medida que va creciendo el tamaño de la matriz. El resto del tiempo se distribuye en las llamadas a las funciones del paquete Eigen, con el cual está generada la malla. Además, se observa que hay una reducción con la bandera -O3 en el porcentaje de tiempo usado por la función *clusters*. Otras funciones que no aparecen descritas en el flat profile, como *imprimirDatos*, son funciones que probablemente consumen muy poco tiempo en comparación con la función *clusters*.

Conclusiones

- Se estudió la percolación de sitio por medio de la simulación del llenado de sitio para un malla bidimensional, se determinó la probabilidad de percolación $P(p, L)$ en función de la probabilidad de llenado p . Se encontró que a partir de una probabilidad crítica promedio $p_c = 0,59$ de llenado de los sitios de la matriz, se generaba clúster percolante independientemente del tamaño. En la gráfica que muestra el comportamiento de la probabilidad $P(p, L)$ se evidencia que para tamaños cada vez más grandes del sistema, este comportamiento tiende a una función de paso escalón, lo que concuerda con la teoría para una malla infinita.
- Se reporta *violación de segmento* en el tamaño $L = 512$ desde la probabilidad $p = 0.68$. Esto puede ocurrir debido a que la función que construye los clústers realiza una cantidad de iteraciones que superan la capacidad de la memoria stack al recorrer sus vecinos. Este problema es difícil de resolver para tamaños grandes con va-

lores de p cercanos a 0.70, dado el recorrido necesario para la revisión de los clústers; como una posible solución (no implementada, dada la necesidad del replanteamiento de la función principal del programa escrito) se sugiere un cambio en la revisión de los clústers percolantes, y buscar una manera más restrictiva de calcular el tamaño (donde no requiera el recorrido de toda la matriz).

Referencias

- [1] Béla Bollobás y Oliver Riordan. “Sharp thresholds and percolation in the plane”. En: *Random Structures & Algorithms* 29.4 (2006), págs. 524-548.
- [2] Béla Bollobás y col. *Percolation*. Cambridge University Press, 2006.
- [3] Geoffrey Grimmett. “Percolation and disordered systems”. En: *Lectures on probability theory and statistics*. Springer, 1997, págs. 153-300.
- [4] Muhammad Sahimi. *Applications of percolation theory*. CRC Press, 1994.