

ALGORITHMICS

FALL 2020-2021

G12

CONECTIVIDAD Y PERCOLACIÓN: ESTUDIO DEL NÚMERO DE COMPONENTES CONEXAS EN UN GRAFO BAJO UN PROCESO DE PERCOLACIÓN

IVÁN LÓPEZ RODRÍGUEZ
DANIEL SANTIAGO CORONA



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



Tabla de Contenidos

1. Problema de estudio.....	3
1.1 Conceptos.....	3
2. Implementación.....	6
2.1 Grafo.....	6
2.2 Generadores de grafos.....	6
2.3 Estructura de datos Union-Find.....	7
2.4 Algoritmos de los apartados A y B.....	8
3. Experimentación.....	9
3.1 Estudio de transición de fase en grafos <i>grid</i> cuadrados.....	9
3.1.1 Procesos de percolación de nodos.....	9
3.1.2 Procesos de percolación de aristas.....	14
3.2 Estudio de transición de fase en grafos geométricos aleatorios conexos.....	17
3.2.1 Determinación de distancia de conectividad.....	17
3.2.2 Procesos de percolación de aristas.....	19
4. Conclusiones.....	21
5. Bibliografía.....	22

1. Problema de estudio

El objetivo de este proyecto es el estudio de la existencia de transiciones de fase para la propiedad del número de componentes conexas en grafos sometidos a procesos de percolación.

Comprender los elementos que formaban parte de este problema fue el primer paso para poder estudiar el proyecto propuesto, y a continuación nos proponemos sintetizar los conceptos necesarios para comprender la implementación y experimentación que siguen.

1.1 Conceptos

Una percolación, en el contexto de las matemáticas, es un proceso sobre un grafo que consiste en la eliminación de nodos o aristas. El estudio de los efectos de las percolaciones en los grafos se conoce como teoría de percolaciones y, de entre sus numerosas aplicaciones, destacamos el estudio de la conectividad de redes o la modelización de procesos en las ciencias de materiales.

El estudio del efecto de las percolaciones sobre un grafo para una determinada propiedad de éste, como la conectividad (propiedad de ser conexo), el número de componentes conexas o la existencia de un camino entre nodos destacados; está fuertemente relacionado con el concepto del cambio de fase.

Dada una probabilidad q , definimos $P(q)$ como el proceso de aplicar percolaciones en un grafo con una probabilidad q independiente para cada nodo o arista. Consideraremos la existencia de un cambio de fase o valor crítico la aplicación de $P(q)$ sobre el grafo G cuando:

1. La probabilidad de que se dé la propiedad π en G es muy baja en $P(q')$ para cualquier $q' < q$.
2. La probabilidad de que se dé la propiedad π en G es muy alta en $P(q')$ para cualquier $q' > q$.

En la teoría de percolaciones se definen las regiones:

- Subcrítica para la región de valores de $q' < q$.
- Supercrítica para la región de valores de $q' > q$.
- Crítica para el valor o valores q donde se encuentra una singularidad, el efecto de $P(q)$ sobre la propiedad π no está determinado y además es el límite entre una región subcrítica y otra región supercrítica.

El estudio de los valores de probabilidad q que determinan estas fases es frecuentemente un estudio experimental, este estudio se realiza de manera iterativa sobre números suficientemente grandes de muestras para llegar a conclusiones significativas del comportamiento de una propiedad en G bajo los efectos de fallos en sus nodos o aristas. Para poder disponer de un número significativo de muestras para la experimentación, además de procurar disminuir la probabilidad de *bias* en las muestras, los grafos aleatorios son una herramienta muy útil.

Los grafos aleatorios son grafos que se han generado por la aplicación de un proceso aleatorio. Con esta definición observamos que los grafos que han sufrido un proceso de percolación son en sí mismos grafos aleatorios, ya que son el resultado de la aplicación del proceso $P(q)$ que elimina nodos o aristas de G según una distribución de probabilidad que utiliza la probabilidad q .

Los modelos de generación de grafos aleatorios son procesos de generación de grafos utilizando procesos aleatorios. Para nuestro estudio nos centraremos en los modelos de generación de *Erdős–Rényi*, modelos de generación de grafos aleatorios muy relacionados:

- El modelo $G(n,p)$ construye un grafo con n nodos, y cada una de las aristas del grafo completo K_n existe independientemente con probabilidad p . Es interesante ver que la probabilidad de generar un determinado grafo $G=(V,E)$ con $|V| = n$ y $|E| = m$ es la composición de las m probabilidades q de que exista una arista existente por la composición de las probabilidades de que no existan las M ($|V_{K_n}| = M$) menos m aristas que no existen.

$$p^M (1 - p)^{\binom{n}{2} - M}.$$

Figure 1: Probabilidad de generar un determinado grafo con n nodos y m aristas.

- El modelo $G(n,m)$ construye un grafo aleatoriamente de entre la colección de todos los grafos posibles con n nodos y m aristas, con la misma probabilidad entre todos ellos de ser escogido.

En concreto será el primer modelo mencionado el que utilizaremos para la experimentación, al resultar mucho más coherente con las generaciones de grafos que realizaremos, como veremos más adelante.

También es interesante apuntar que un grafo generado por un modelo de *Erdős–Rényi* es una percolación sobre el grafo completo con tantos nodos como el grafo generado.

Con los conceptos previamente explicados, podemos proceder a explicar como hemos abordado el estudio de los efectos de la percolación de nodos y aristas en grafos sobre la propiedad de su número de componentes conexas. Concretamente estudiaremos sus efectos sobre dos tipos de grafos:

- Grafos *grid* o grafos de cuadrícula: Una instancia de los *lattice graph*, los pertenecientes a \mathbb{R}^2 .
- Grafos geométricos conexos generados de manera aleatoria, encuadrados en el cuadrado unitario de \mathbb{R}^2 .

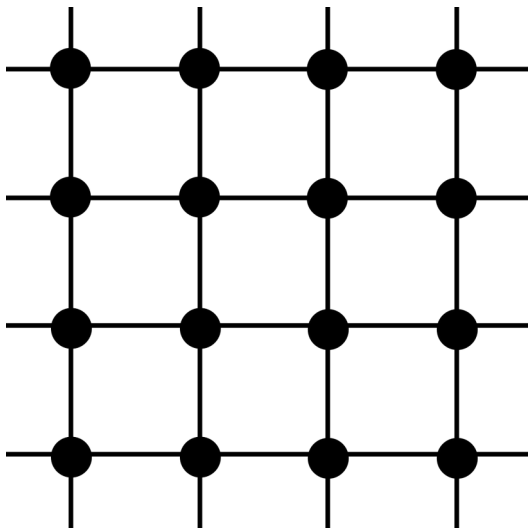


Figure 2: Representación de un grafo grid.

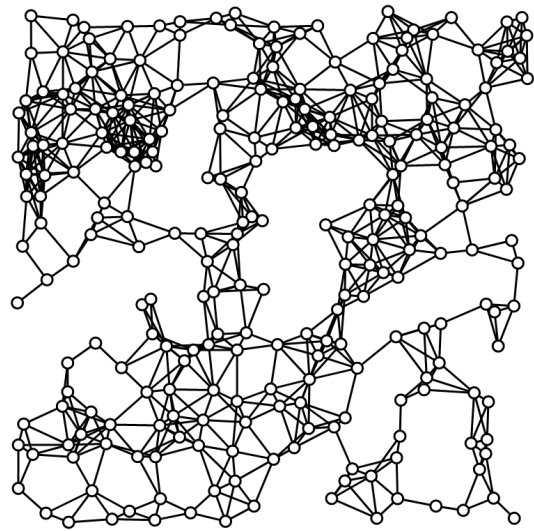


Figure 3: Representación de un grafo geométrico aleatorio en \mathbb{R}^2

2. Implementación

Nuestra aproximación a modelizar las percolaciones en grafos desde un principio fue trabajar con una clase `Graph` que modelizará un grafo de manera genérica, que pudieramos utilizar para representar cualquier tipo de grafo de una manera cómoda para realizar las percolaciones.

2.1 Grafo

Inicialmente consideramos una clase de grafo como las que habíamos utilizado previamente, representando una instancia de grafo por su lista de adyacencias o por una matriz de adyacencia, y si bien estas implementaciones hubieran sido válidas, tuvimos la idea de representar al grafo de una manera más afín a su definición matemática: como un conjunto de nodos y un conjunto de aristas.

Hay un motivo claro por el que esta no es su implementación habitual al trabajar con grafos para tareas computacionales, esta estructura es poco eficiente para trabajar con algoritmos de grafos. Aún así creemos que en este caso es especialmente adecuada (y hasta más elegante) para los dos algoritmos sobre grafos para los que los utilizaremos: **la percolación sobre nodos o aristas** y **la creación de una estructura de datos para conjuntos disjuntos** (*union-find* a partir de ahora).

2.2 Generadores de grafos

Además de los principales algoritmos de esta clase `Graph`, implementamos dos algoritmos necesarios para construir los dos tipos de grafos sobre los que experimentaremos:

- Generador de grafos grid: Este generador toma como parámetro un entero n que corresponde al número de elementos de cada costado del grafo y construye un grafo con $(n * n)$ nodos y la configuración de aristas propia de un grafo grid. En el momento de su construcción este grafo no presenta ningún elemento de aleatoriedad, la construcción de un grafo de tipo grid dado su tamaño es única.
- Generador de grafos geométricos aleatorios: Este generador toma como parámetros un entero n que corresponde al número total de nodos del grafo a construir y un número decimal r que corresponde al umbral de distancia para la creación aristas. El algoritmo genera un grafo con n nodos encuadrados dentro del cuadrado unitario de \mathbb{R}^2 , dos nodos de ese conjunto están unidos cuando su distancia euclídea es igual o menor que r .

Por las características de los nodos de los dos tipos de grafos con los que teníamos que trabajar: posiciones en una matriz y posiciones en un espacio bidimensional consideramos adecuado trabajar

con nodos cuya información fuera un par de valores, así podemos trabajar de manera cómoda con estos dos tipos de grafos.

Pese a esta última implementación de los nodos pensada para encajar con nuestros dos tipos de grafos, pensamos que nuestra clase Graph es muy polivalente y para trabajar con diferentes tipos de grafos solo requeriría implementar el constructor correspondiente al nuevo tipo de grafo, y con cambios mínimos en la implementación sería capaz de tratar con grafos grid 3 o n dimensionales.

2.3 Estructura de datos Union-Find

Tras descubrir la estructura de datos union-find, se vio reforzada la coherencia del uso de nuestra implementación de grafo como un conjunto de nodos y un conjunto de aristas, ya que esta resultó muy adecuada en la construcción de las estructuras union-find.

Después de investigar distintas implementaciones de esta estructura de datos en la literatura científica, encontramos muy interesante la idea vista en la referencia 2, implementar la estructura union-find como un bosque. Esta estructura es especialmente adecuada además de natural, todos sus elementos serán árboles. Además favorece la creación de los elementos disjuntos al poder leer el conjunto de nodos y aplicar iterativamente la operación make-set para cada uno de los elementos del conjunto y la unión de estos conjuntos por la lectura del conjunto de aristas, que debe añadir solo aquellas que no rompen la condición de que los elementos dentro del bosque son árboles. En la referencia mencionada también encontramos interesante la implementación de la *path compression* para la función find-set, cuyo coste queda amortizado por las sucesivas llamadas a *union*.

La implementación de estos tres elementos ya nos permitió tener un marco sobre el que poder trabajar y empezar nuestra experimentación. Consideraremos que pese a que no era el objetivo principal de este proyecto, ha sido muy positivo realizar el trabajo de programar los anteriores componentes. Si bien habíamos programado clases de grafos antes (aunque no con la implementación de conjuntos), programar la estructura de los union-find nos ha permitido profundizar mucho más en ella.

El proceso de programarla ha sido altamente iterativo y hemos probado bastantes alternativas, desde un uso intensivo de punteros hasta el uso de vectores para guardar los representantes de cada conjunto, para encontrar la que finalmente nos ha convencido. Aún así, debido a limitaciones en el tiempo disponible para realizar nuestro proyecto, nos hemos quedado con ideas pendientes para la implementación; creemos que con más tiempo hubiéramos podido probar nuevas funcionalidades para nuestras clases o quizás explorar implementaciones más eficientes.

Aún así creemos que el código generado puede ser útil para futuros proyectos y no descartamos utilizarlo para futuros trabajos.

2.4 Algoritmos de los apartados A y B

Nos gustaría concretar que los algoritmos solicitados por el enunciado corresponden a:

- a) Los métodos *node_percolation(double q)* y *edge_percolation(double q)* de la clase Graph, que se llaman sobre un elemento G de esta clase y realizan la percolación de nodos o aristas respectivamente con la probabilidad q.
- b) El método *print_connected_components()* de la clase Graph se llama sobre un elemento G de esta clase y imprime por pantalla el número de componentes conexas de G utilizando la clase de Union-Find.

3. Experimentación

Para poder obtener resultados experimentales significativos se han tomado las siguientes medidas:

- Un tamaño de grafos suficientemente grande, concretamente de 32×32 (1024) nodos para los grafos grid, y de 500 nodos para los grafos geométricos.
- La aleatoriedad en los procesos probabilísticos se ha implementado utilizando los métodos de la librería *random* de C++.
- Todos los experimentos se realizan sobre 10 instancias de grafos distintas, generados respectivamente por las semillas enumeradas del 1 al 10 en la librería *random*.
- Los experimentos son reproducibles utilizando los ejecutables generables por el *Makefile* en la carpeta adjunta *build*. En esta carpeta también se encuentra un fichero README con las instrucciones correspondientes. Así, el archivo main no implementa ningún experimento y está disponible para ejecutar pruebas.
- En la carpeta *doc* se adjunta los datos obtenidos en los experimentos y sus representaciones gráficas.

3.1 Estudio de transición de fase en grafos *grid* cuadrados

3.1.1 Procesos de percolación de nodos

El experimento consiste en 10 iteraciones, cada una de ellas asociada a una semilla de aleatoriedad. Por cada iteración se iterará por un rango de valores de probabilidad, el parámetro q . Por cada valor del parámetro q , se crea una instancia de un grafo grid y se le aplica un proceso de percolación de nodos con probabilidad q . Al realizar la percolación se obtiene como efecto secundario de la llamada su número de componentes conexas después del proceso.

El rango de valores de probabilidad es $[0,1]$, pero el número de valores que puede tomar q varía en función de la versión del experimento escogida: la implementación *grid_node_percolation* varía q en incrementos de 0.1 mientras que *detail_node_percolation* varía en incrementos de 0.05.

Por lo general hemos encontrado más interesantes los resultados de las implementaciones con mayor número de valores de probabilidad, aunque sucesivas pruebas nos hicieron darnos cuenta que incrementos de q demasiado pequeños añadían mucho ruido en las muestras, quizá sería subsanable con iteraciones sobre un mayor número de semillas, por ejemplo 100 en vez de 10.

Los resultados obtenidos en la experimentación (detallada) pueden ser vistos en la figura 4 y la media, útil para reducir el ruido presente en las muestras, en la figura 5.

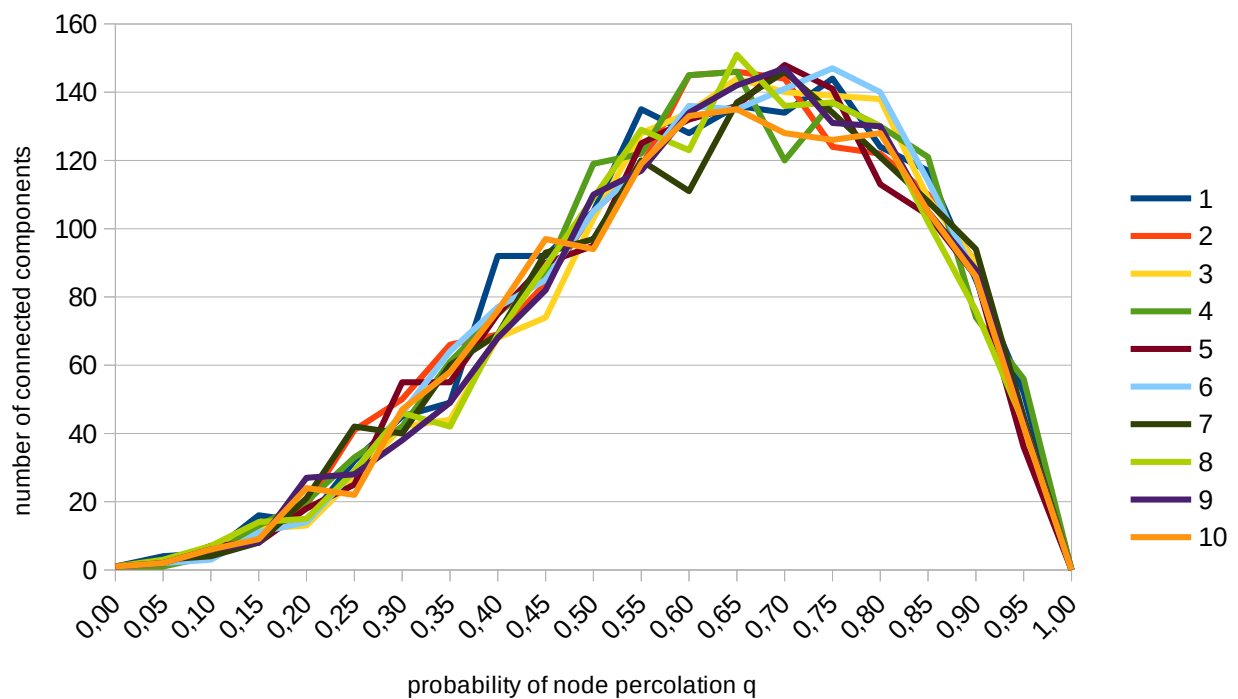


Figure 4: Resultados de la experimentación de la percolación de nodos sobre un grafo grid

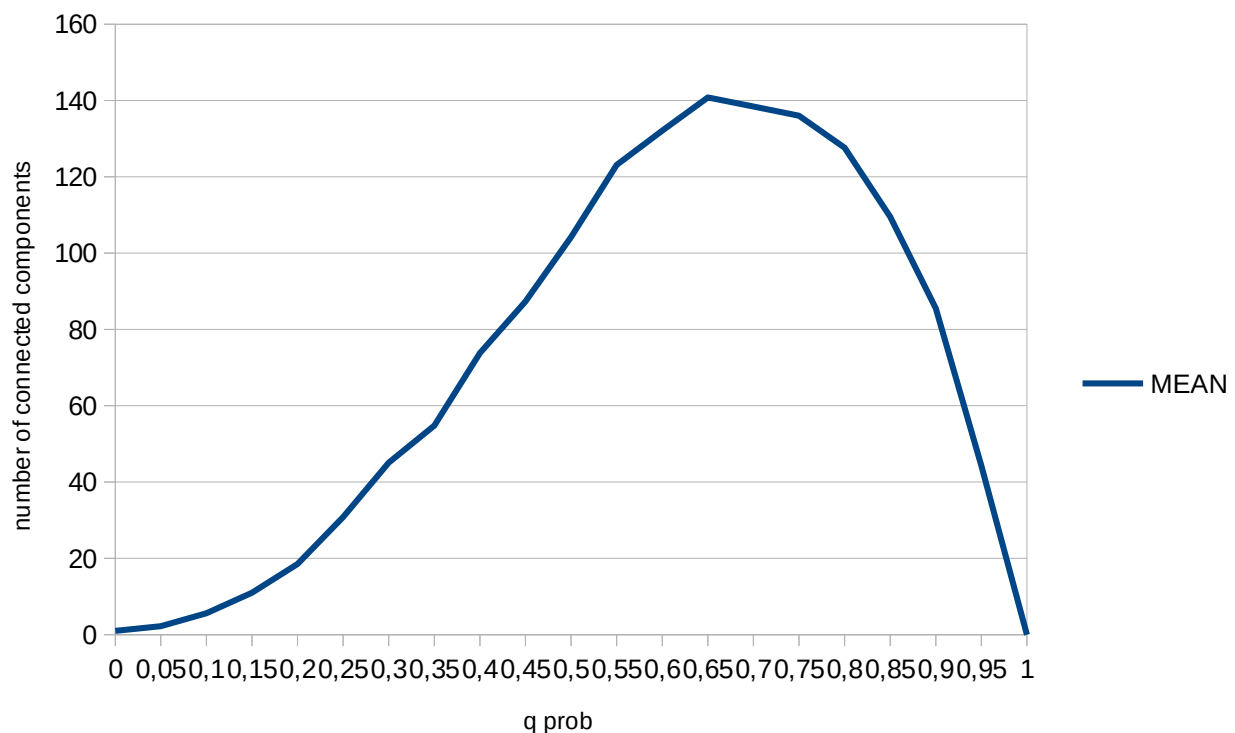


Figure 5: Media de los resultados representados en la figura 4.

Los resultados obtenidos son coherentes con la intuición, con una probabilidad nula de realizarse las percolaciones ningún nodo es eliminado (y por lo tanto ninguna arista) así que el número de componentes conexas del grafo es 1, ya que el grafo grid cuadrado es conexo por definición.

Conforme aumenta la probabilidad de que desaparezcan nodos (desapareciendo sus aristas asociadas), es natural pensar que algún nodo o conjunto de nodos puede quedar aislado del resto del conjunto y así empieza a crecer el número de componentes conexas del grafo. Cuanto mayor es la probabilidad de percolación, mayor es la probabilidad de que este fenómeno ocurra, y así observamos crecer el número de componentes conexas junto a la probabilidad de percolación q .

Es interesante ver que el numero de componentes crece con la probabilidad de percolación pero el número de componentes conexas siempre se encuentra lejos del máximo posible inicialmente (1024 componentes conexas correspondientes a 1024 nodos aislados), esto es debido a que el proceso de percolación de nodos que está provocando la aparición de componentes conexas a la vez destruye los propios nodos candidatos a convertirse en la unidad mínima de componente conexas. De hecho el número máximo de componentes conexas observado en nuestro experimento no alcanza 160.

Es este mismo fenómeno el que provoca que en el entorno del valor $q = 0.65 (\pm 0.05)$ se encuentre una región en la que cambia la tendencia del número de componentes conexas respecto a q , si para valores previos a esta región el número de componentes conexas aumentaba con q a partir de esta región de valores el número de componentes conexas decae con el crecimiento de q .

Este caída en el número de componentes conexas se mantiene hasta el límite $q = 1$, cuando la percolación de nodos es segura por cada nodo, es natural que la cantidad de componentes conexas en el grafo es 0, ya que no quedan componentes en absoluto.

Si bien podemos encontrar una región de cambio de fase, tal y como mencionamos anteriormente, alrededor de $q = 0.65$, este cambio de fase es respecto a la tendencia del número de componentes conexas y no respecto al número de ellas en si mismo.

En los datos obtenidos no observamos ninguna región con un crecimiento notable o repentino del número de componentes conexas entorno a un valor de q , tal y como habíamos observado en otros estudios de cambios de fase en la literatura científica. El crecimiento de la función puede parecer en un principio exponencial para más tarde recordar más a una función parabólica debido al efecto de la desaparición de nodos, esperábamos encontrar resultados con regiones de verdadero cambio en el número de componentes conexas.

Así nos animamos a estudiar concretamente la región de los valores de q muy pequeños: con $q = 0.0$ el número de componentes conexas es 1, como mencionamos, y eventualmente al crecer este valor nos empezaremos a encontrar más de ellas. ¿Podemos acotar una región de cambio de fase para este proceso? Los datos de el experimento `beg_zoom_grid_node` se muestran en la figura 6:

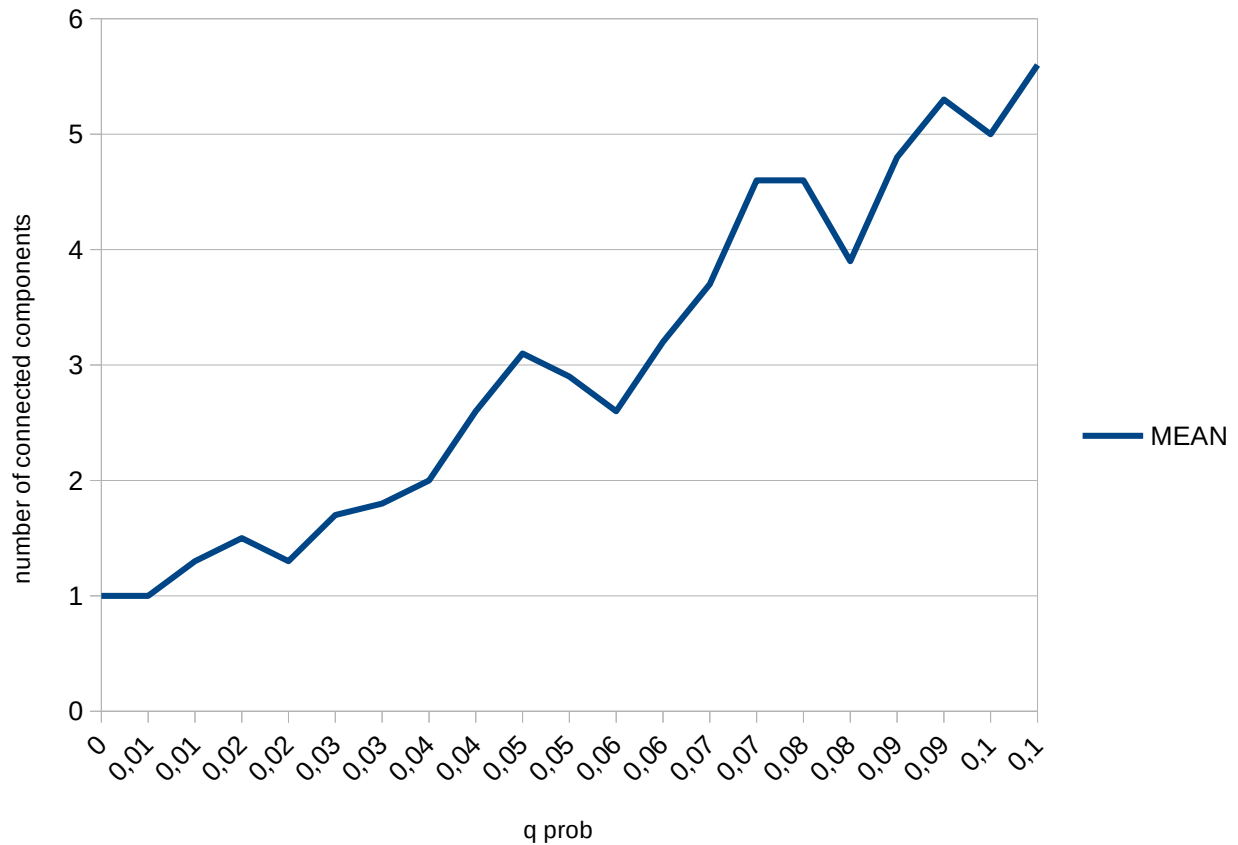


Figure 6: Media de los resultados obtenidos en *beg_zoom_grid_node*.

En la representación de la media de los datos obtenidos podemos observar que para $q = 0.01$ podemos observar un cambio de fase, ya que para valores menores es muy improbable encontrar más de una componente conexa mientras que para valores de q mayores es muy improbable encontrar una sola componente conexa.

De hecho este resultado es el mismo que surge de buscar el cambio de fase para la conectividad del grafo en si mismo, pero éste es el cambio de fase más constatable que encontramos.

3.1.2 Procesos de percolación de aristas

El experimento es muy similar al anterior pero para percolaciones de aristas en vez de nodos.

El experimento consiste en 10 iteraciones, cada una de ellas asociada a una semilla de aleatoriedad. Por cada iteración se iterará por un rango de valores de probabilidad, el parámetro q . Por cada valor del parámetro q , se crea una instancia de un grafo grid y se le aplica un proceso de percolación de aristas con probabilidad q . Al realizar la percolación se obtiene como efecto secundario de la llamada su número de componentes conexas después del proceso.

Otra vez los resultados más representativos se obtuvieron con `detail_grid_edge_percolation` así que este será el experimento usado para nuestras explicaciones.

Los resultados obtenidos en la experimentación pueden ser vistos en la figura 7 y la media, útil para reducir el ruido presente en las muestras, en la figura 8.

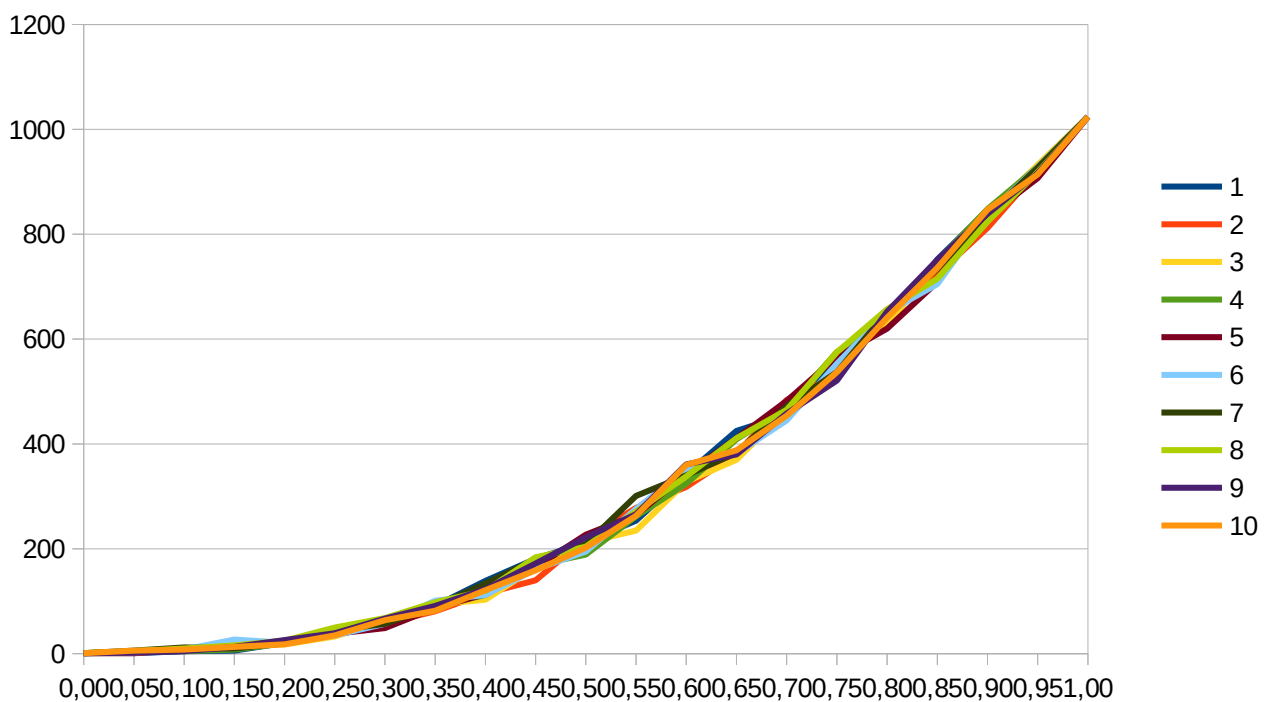


Figure 7: Resultados de la experimentación de la percolación de aristas sobre un grafo grid

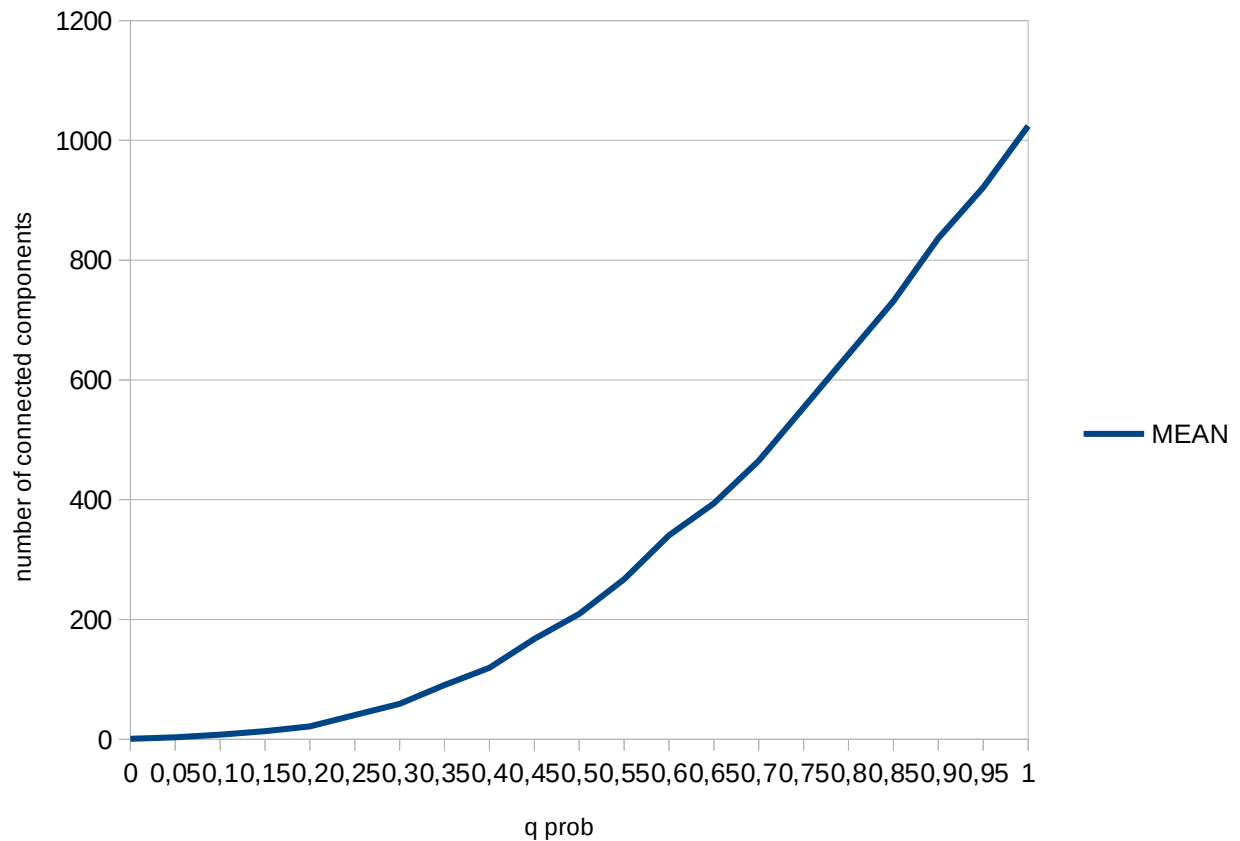


Figure 8: Media de los resultados representados en la figura 7.

De manera similar al experimento anterior, los resultados son coherentes con la intuición que podamos tener sobre la percolación de aristas, cuando la probabilidad de percolación de arista es nula el grafo grid permanece conexo por construcción. Conforme aumenta la probabilidad de percolación de arista aumenta el número de componentes conexas que nos podemos encontrar, al empezar a quedar aislados nodos del grafo y convirtiendose en componentes conexas. A diferencia del caos anterior, no se eliminan nodos así que el límite máximo inicial de componentes conexas (1024, tantas como nodos) si que se alcanza cuando el valor de q es 1, al desaparecer todas las aristas, todos los nodos son componentes conexas.

Esta vez el número de componentes conexas presenta un crecimiento claramente exponencial, pero que tampoco da a pie a observar una región de cambio de fase en su número de componentes conexas. Otra vez recurrimos a analizar su punto de creación de componentes conexas frente a la situación inicial de una única componente conexas en el experimento `zoom_grid_edge_percolation`.

probability	0,000	0,010	0,020	0,030	0,040	0,050
MEAN	1	1,2	1,8	1,7	2	3,4

Figure 9: Datos obtenidos en el experimento *zoom_grid_edge_percolation*

En esta ocasión, los datos obtenidos (representados en la figura 9) son más representativos debido al ruido que se percibe en las gráficas asociadas. Podemos observar como en la región $q = [0.01, 0.03]$ podemos observar un cambio de fase, para valores anteriores a esta región la probabilidad de tener más de una componente conexas es muy baja, mientras que para valores mayores es muy improbable encontrar solo una componente conexas.

Ha sido objeto de debate si deberíamos centrarnos en encontrar cambios de fase en el resto del rango de valores de q , por ejemplo entorno a $q = 0.15 \pm 0.05$ (figura 8) parece ser un punto con un incipiente crecimiento notable del número de componentes conexas, pero no encontramos indicadores lo suficientemente buenos, como en el caso de la aparición de más de una componente conexas arriba discutido, que puedan justificarlo. De hecho los datos reflejan que en realidad, en ningún rango de los valores de q el crecimiento relativo del número de componentes conexas respecto a los valores de su entorno (la constante que multiplica su crecimiento) es tan grande como en la zona inicial, donde nosotros hemos indicado el cambio de fase de la conectividad del grafo.

3.2 Estudio de transición de fase en grafos geométricos aleatorios conexos

3.2.1 Determinación de distancia de conectividad

El siguiente experimento difiere del resto de experimentos realizados. Recordamos que la distancia r en un grafo geométrico aleatorio es la distancia de creación de una arista entre dos nodos. El objetivo es determinar de manera experimental el valor de r a partir del cual podemos tener una confianza grande en que el grafo resultante será conexo.

La experimentación consiste en 10 iteraciones, cada una de ellas asociada a una semilla de aleatoriedad entre el 1 y el 10 respectivamente. Por cada una de estas iteraciones, se iterará por un rango de posibles valores de r , desde 0.0 para el caso donde nunca generaremos una arista entre dos nodos (ya que el generador se ocupa de que dos nodos no ocupen la misma posición), hasta la distancia máxima posible en el cuadrado unitario: $2^{1/2}$.

Por cada uno de los posibles valores de r , con la ayuda del generador de números aleatorios asociado a la semilla, se generan 500 nodos en posiciones aleatorias encuadradas en el cuadrado unitario de \mathbb{R}^2 . Una vez generados los nodos procedemos a añadir una arista entre todas las parejas de nodos cuya distancia sea menor o igual a r . Finalmente se informa si el grafo generado por este proceso es conexo.

Tal y como comentamos, este experimento difiere en que no se basa en la realización de percolaciones a un grafo y el posterior estudio de su efecto. En vez de percolar nodos o aristas con una determinada probabilidad sobre la que iteraremos, en nuestro caso el parámetro sobre el que iteramos está relacionado con la probabilidad de crear una arista. Sin embargo nuestro objetivo final también consiste en encontrar una región de cambio de fase (concretamente sabemos que el para $r = 0.0$ el grafo nunca será conexo ya que no hay aristas posibles, y para $r = 2^{1/2}$ el grafo siempre será conexo ya que coincidirá con K_n), la región de cambio de fase donde el grafo pasará de ser muy probablemente no conexo a ser muy probablemente conexo.

Los primeros resultados de la experimentación nos descubrieron que este cambio de fase ocurría para valores mucho más pequeños de r de lo que imaginábamos, este hecho lo hemos contrastado y es coherente con la literatura científica. Así para obtener datos más útiles para encontrar la región de interés acotamos r a un rango más pequeño, concretamente a $[0.0, 0.2]$.

Los resultados corresponden al experimento `zoom_geo_radius` y se muestran a continuación.

r	0,00000	0,00500	0,01000	0,01500	0,02000	0,02500	0,03000	0,03500	0,04000
MEAN	0	0	0	0,5	1	1	1	1	1

Figure 10: Datos asociados al experimento zoom_geo_radius.

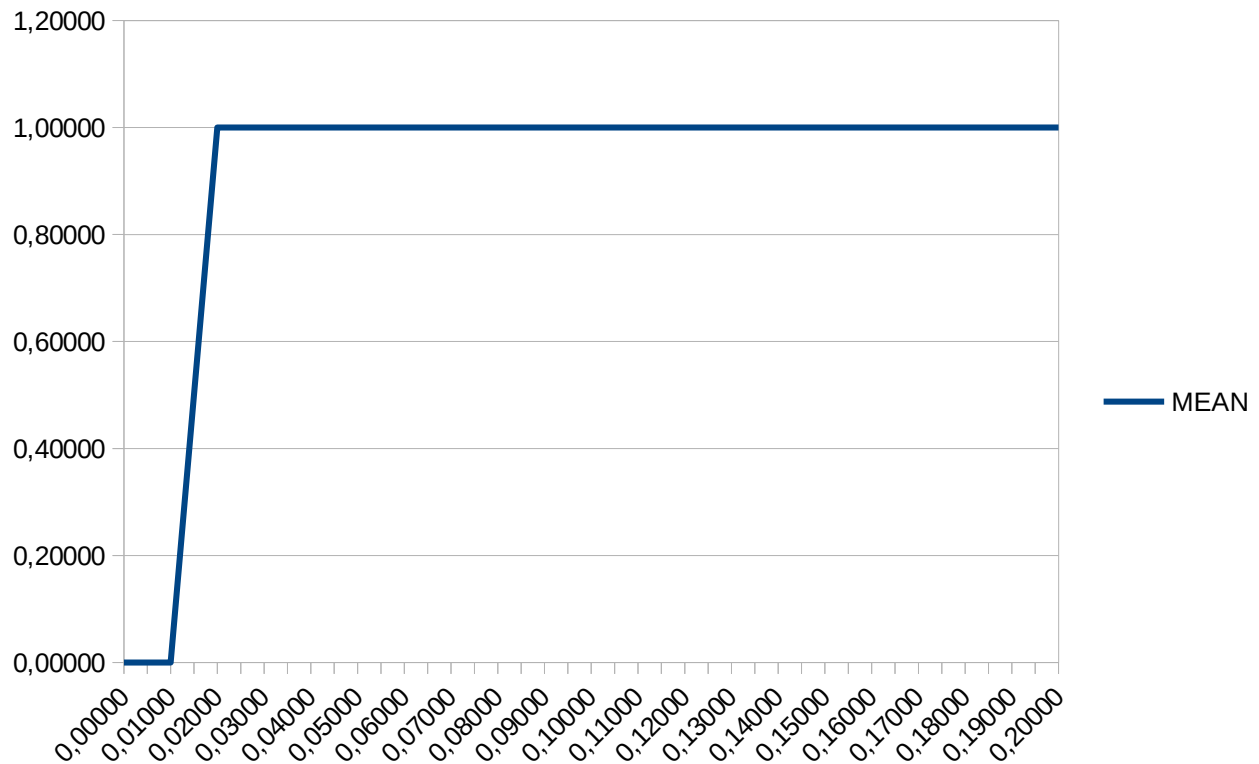


Figure 11: Representación gráfica de los datos asociados al experimento zoom_geo_radius.

Esta vez observamos de manera clara la región crítica centrada en el valor $r = 0.015$, con probabilidad muy baja de encontrar grafos conexos para distancias más pequeñas y probabilidad muy grande de encontrarlo para valores mayores.

3.2.2 Procesos de percolación de aristas

El experimento aplicado es el mismo que hicimos para la percolación de aristas, pero ahora aplicado en grafos geometricos aleatorios.

Dicho experimento consiste en 10 iteraciones, cada una de ellas asociada a una semilla aleatoria. Por cada iteración se iterará por un rango de valores de probabilidad, el parámetro q . Por cada valor del parámetro q , se crea una instancia de un grafo geometrico aleatorio y se le aplica un proceso de percolación de aristas con probabilidad q . Al realizar la percolación se obtiene como efecto secundario de la llamada su número de componentes conexas después del proceso.

El valor de r , parámetro definido como la distancia máxima para que entre dos nodos tengan una arista asociada entre ellos, le hemos dado un valor igual a 0.2, ya que como bien hemos visto en el experimento anterior, es un valor suficiente para que con 500 nodos, muy probablemente nos cree un grafo conexo.

En este caso hemos tomado los valores con `geo_edge_percolation`, donde la probabilidad va aumentando de 0.1 en 0.1 por cada iteración. Vistos estos resultados (figura 12), como el número de componentes conexas no variaba de 1 hasta llegar a una probabilidad de aproximadamente 0.8, decidimos hacer un nuevo experimento donde la probabilidad de percolar aristas estuviera en el rango $[0.7,1]$, incrementando q en 0.05 por iteración.

GRAPH SEED \ q	0.00	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00
1	1	1	1	1	1	1	1	1	1	1	500
2	1	1	1	1	1	1	1	1	1	2	500
3	1	1	1	1	1	1	1	1	2	2	500
4	1	1	1	1	1	1	1	1	1	2	500
5	1	1	1	1	1	1	1	1	1	1	500
6	1	1	1	1	1	1	1	1	1	4	500
7	1	1	1	1	1	1	1	1	1	1	500
8	1	1	1	1	1	1	1	1	1	1	500
9	1	1	1	1	1	1	1	1	1	1	500
10	1	1	1	1	1	1	1	1	1	2	500
MEAN	1	1	1	1	1	1	1	1	1.1	1.7	500

Figure 12: Tabla obtenida mediante `geo_edge_percolation`, donde podemos ver el número de componentes conexas dada una semilla y una probabilidad de percolación de aristas.

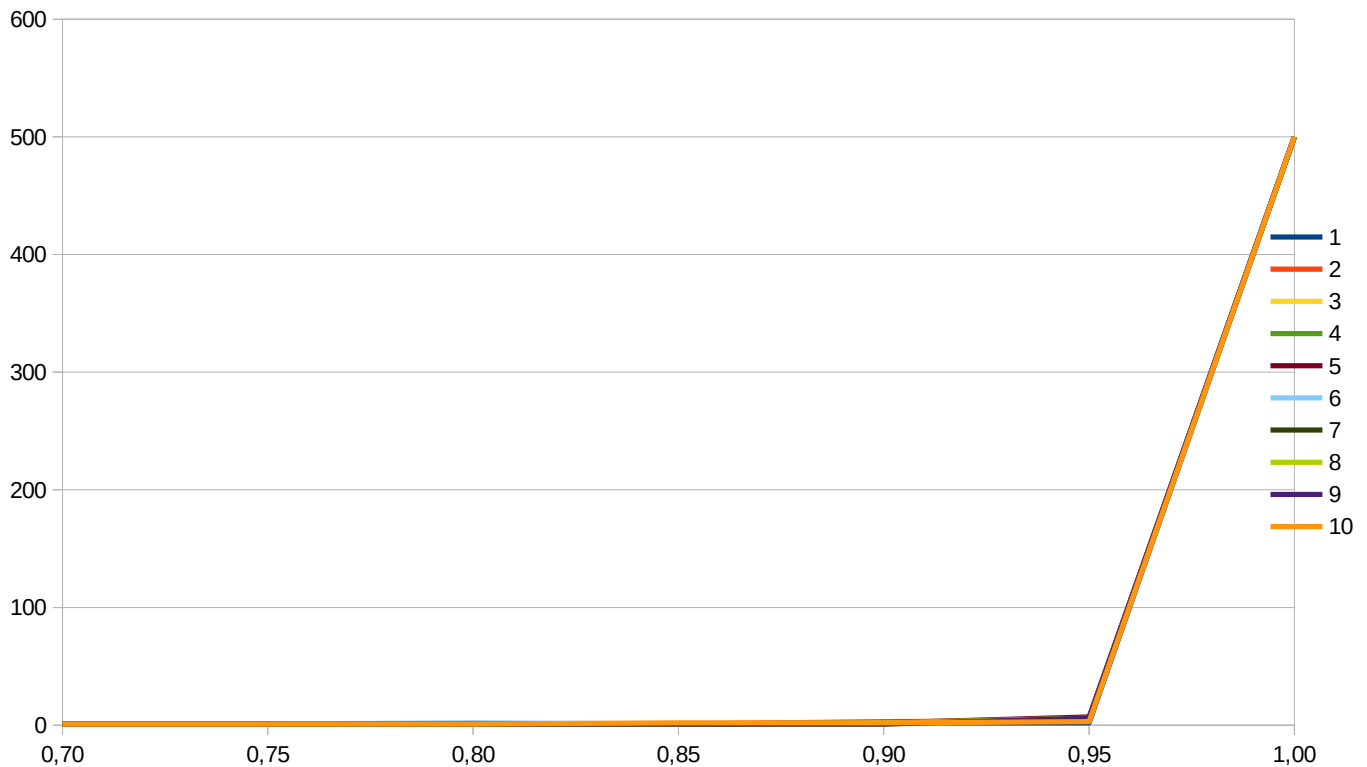


Figure 13: Gráfica obtenida mediante `mid_zoom_edge_percolation`, donde vemos el número de componentes conexas por cada semilla y probabilidad de percolar una arista.

Como hemos enunciado antes, decidimos hacer un segundo experimento para tratar de encontrar una probabilidad más precisa donde encontrar el cambio de fase. Se puede observar en la figura 13, de manera clara como a partir de $q = 0.95$, el número obtenido de componentes conexas empieza a crecer de manera drástica, por tanto, podemos concluir que nuestro cambio de fase se encuentra entorno a ese valor. El número de componentes conexas se mantiene de manera estable en valores bajos para valores de q menores a 0.95, mientras que la cantidad de componentes conexas que podemos encontrar para valores mayores es, con una gran probabilidad, mucho mayor que los encontrados en la anterior región.

En este caso, a diferencia de los anteriores, hemos encontrado de manera clara un cambio de fase para el número de componentes conexas (y no la conectividad) del grafo. Creemos que sería muy útil encontrar un indicador matemático para ayudarnos a encontrar estas regiones críticas en los datos de nuestras experimentaciones. Intentamos utilizar algunos, como comparar los crecimientos relativos de una iteración a la siguiente para cada seed, pero ninguno dio resultados consistentes y finalmente recurrimos a la observación manual de los resultados para encontrar los puntos críticos.

4. Conclusiones

Los resultados de la experimentación han sido interesantes, a nuestro parecer. Hemos podido observar como fenómenos como la ruptura de una componente conexa relativamente grande se da para probabilidades relativamente pequeñas de percolaciones, o como un grafo se vuelve conexo con probabilidades relativamente pequeñas de generar aristas. Creemos que hubiera sido aún más interesante poder profundizar en los resultados experimentales probando a iterar sobre grafos aún más grandes y, especialmente, iterar sobre más instancias de grafo distintas, es decir utilizar más semillas. Realizar los experimentos con estas magnitudes podría habernos ayudado a intentar esclarecer si los resultados obtenidos son o no extrapolables más allá del tamaño de grafo actual, si los resultados obtenidos son generales para los tipos de grafos con los que hemos trabajado.

Asimismo el estudio de los efectos de las percolaciones en grafos y de los cambios de fase de las propiedades nos ha parecido muy interesante. De haber contado con más tiempo tanto para formarnos, implementar como para documentar nos hubiera gustado adaptar el estudio de la aparición de la conectividad en el grafo geométrico conexo al estudio de la evolución de una pandemia, por las similitudes que encontramos en los modelos.

El conjunto de nodos iniciales en el grafo geométrico podrían inicializarse como individuos sanos, mientras que uno o unos pocos de esos individuos aparecerán como infectados. La distancia r de creación de aristas modeliza la capacidad de la enfermedad para propagarse. En caso de encontrarse a una distancia de contagio r o menor, podría producirse un contagio (arista) cuando uno de los individuos esté enfermo y con una probabilidad determinada, que modeliza la agresividad del virus y las medidas de prevención de la población. Este modelo permitiría realizar estudios estáticos de una posible evolución de una pandemia dadas unas condiciones iniciales y una modelización de las características de la enfermedad y la población. Podría ser interesante añadir un movimiento estocástico de los individuos en función del tiempo para romper la estaticidad y generar un modelo más dinámico.

Además como comentamos previamente, nos hubiera gustado poder investigar más sobre indicadores sobre los datos que nos ayuden a determinar los puntos de cambios de fase.

En general creemos que tanto la teoría de percolaciones como el estudio de los cambios de fase tienen multitud de posibles utilidades en una gran cantidad de disciplinas dado el potencial que tienen para modelar procesos naturales.

5. Bibliografía

1. Blum A., Hopcroft J., Kannan R. (2018). Foundations of Data Science, Random Graphs (pp, 245-301):

<https://www.cs.cornell.edu/jeh/book.pdf>

2. Cormen T., Leiserson C., Rivest R., Stein C. (2009). Introduction to Algorithms, Data Structures for Disjoint Sets (pp 561-573). Massachusetts Institute of Technology:

https://edutechlearners.com/download/Introduction_to_algorithms-3rd%20Edition.pdf

3. Gao J., (2006). Percolation Theory and Network Connectivity [Presentación Power Point]. Computer Science Department, Stony Brook University:

<https://www3.cs.stonybrook.edu/~jgao/CSE590-fall06/Slides/lecture13.pdf>

4. Phase Transition (s.f). En Wikipedia. Recuperado de:

https://en.wikipedia.org/wiki/Phase_transition

5. Percolation Theory (s.f). En Wikipedia. Recuperado de:

https://en.wikipedia.org/wiki/Percolation_theory

6. Percolation Threshold (s.f). En Wikiperdia. Recuperado de:

https://en.wikipedia.org/wiki/Percolation_threshold#Site-bond_percolation_in_2D

7. Random Graph. (s.f). En Wikiperdia. Recuperado de:

https://en.wikipedia.org/wiki/Random_graph

8. Erdős-Rényi Model. (s.f). En Wikipedia. Recuperado de:

https://en.wikipedia.org/wiki/Erd%C5%91s%E2%80%93R%C3%A9nyi_model