Transició de fase i components connexes en grafs aleatoris

José Camilo Romero Limones, Luis Oriol Soler Cruz,
Pau Escofet Majoral, Roger González Herrera

FIB • UPC

Quatrimestre Tardor 2019/2020

Índex

1. Introducció

Entre la teoria de grafs i la teoria de probabilitats es troba el concepte de teoria de grafs aleatoris, terme que fa referència a les distribucions de probabilitat sobre grafs que son generats mitjançant un procés aleatori.

El concepte de grafs aleatoris estudia la forma de com enllaçar de forma aleatoria un conjunt de nodes mitjançant arestes. Cada procés aleatori forma un model de graf aleatori, que s'utilitza per estudiar el comportament de propietats típiques de grafs com poden ser la propietat de connectivitat o la propietat de components connexes d'un graf.

Entre els models de grafs aleatoris existents tenim el model de graf binomial aleatori i el model de graf geomètric aleatori, desenvolupats per Edgar Gilbert.

En aquest document estudiarem les transicions de fase de certes propietats de grafs en aquests dos models de graf aleatori esmentats anteriorment.

2. Descripció de les propietats estudiades

2.1 Connectivitat

Un graf G = (V, E), on V es el conjunt de vèrtexs i E es la colecció d'arestes, està conectat si, i només si, té almenys un vèrtex i existeix un camí en G desde 'u' fins a 'v' per cada parell de vèrtexs 'u,v' de V.

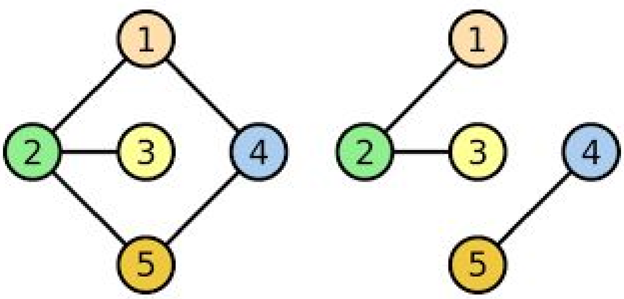


Figura 1: Graf connex (esquerra) i no connex (dreta)

Prenent d'exemple els dos grafs de la *Figura 1*, tenim el de l'esquerra que es connex i està format pels vèrtexs etiquetat com a 1,2,3,4 i 5. El graf de la dreta, en canvi, no es connex i està format per dues components connexes (concepte definit en el següent apartat): per un costat el component 1-2-3 i per l'altre el component 4-5.

2.2 Nombre de components connexes

Una component connexa d'un graf G és un subgraf G', tal que G' ⊆ G, en que qualsevol parell de vèrtexs estan connectats entre ells per un camí, el qual no està connectat a cap vèrtex addicional en el supergraf.

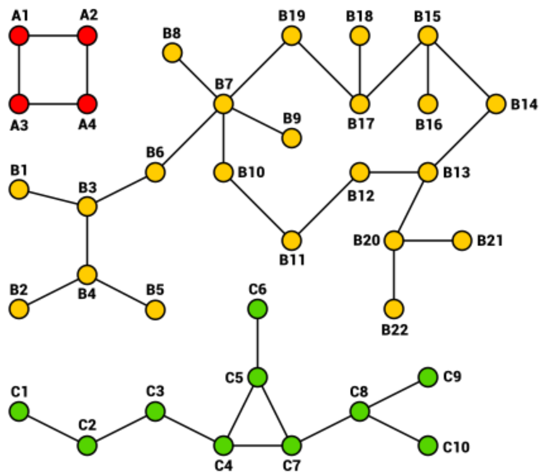


Figura 2: Graf no connex format per tres components connexes

Un exemple de graf format per diverses components connexes es el graf de la *Figura 2*. Es tracta d'un graf no connex format per tres components connexes acolorides de forma diferent.

Si ajuntem la definició de graf connex i component connexa podem extreure que un graf que es conex té exactament una component connexa, que consisteix en el graf complet. Tenim com exemple el graf de l'esquerra de la *Figura 1*.

3. Descripció dels models de grafs aleatoris estudiats

Un graf és una representació d'objectes (nodes) que es conecten a través d'ells mitjançant arestes. Aquests grafs poden ser dirigits o no dirigits, depenent de si les arestes tenen un sentit o no tenen sentit (ambdues direccions).

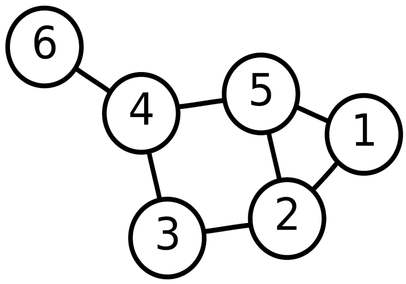


Figura 3: graf

Un clar exemple de graf és un mapa del metro, on les estacions son els nodes i les vies de metro son les arestes, es tracta d'un graf no dirigit ja que les línies de metro funcionen en els dos sentits.

3.1 Model de graf aleatori binomial

El model G(n, p) va ser introduït per primer cop en un article d'Edgar Gilbert de l'any 1959 sobre el llindar de connectivitat, el qual es basa en la generació de grafs aleatoris, es basa en un node, que s' enllaça amb igual probabilitat amb qualsevol altre del graf, per tant cada node té independencia estadística amb la resta de nodes del graf. El model binomial genera un graf aleatori mitjançant dos paràmetres, el nombre de nodes del graf, el qual anomenarem 'n', i la probabilitat de que dos nodes qualsevols estiguin connectats amb una aresta, que anomenarem 'p'. D'aquesta manera 'n' ha de ser un enter i 'p' un valor entre 0 i 1, ambdós inclosos. En cas de que 'p' sigui igual a 0 ens trobarem amb un graf amb zero arestes, és a dir, tindrà 'n' components connexes, d'altra banda, si 'p' és igual a 1 en el graf hi haurà el màxim d'arestes possibles, serà un graf complet de grau 'n'.

La generació de grafs aleatoris sota aquest model segueix una distribució de Poisson, i no és correcte pensar que són semblants a les xarxes reals, ja que poques es comporten d'una manera tan homogènia com els grafs generats. A causa d'això les aplicacions que tenen els grafs binomials son reduïdes.

3.2 Model de graf aleatori geomètric

En la teoria de grafs, un graf geomètric es defineix com el graf amb pes, on els vèrtexs es corresponen amb punts de l'espai Euclidià i els pesos de les arestes són les distàncies mètriques entre els punts que limiten aquestes arestes.

Un graf geomètric aleatori és la xarxa espacial matemàticament més simple, és a dir, un graf no dirigit construït a col·locar, de forma aleatòria, **N** nodes en algun espai mètric (per aquest treball s'utilitza l'espai Euclidià) i connectar dos nodes per una aresta si, i només si, la seva distància està en un rang prèviament definit. Es denota de la forma **G = (N, r)**, on la **|N|** és el nombre de nodes que formen el graf i **r** correspon al radi per a tot node donat.

4. Documentació dels experiments

4.1 Algorisme de connectivitat

Aquest algorisme consisteix en un recorregut en profunditat (DFS) per un graf G=(V,E), el que significa que recorre tots els nodes en una mateixa component connexa, induint un arbre d’expansió d’aquesta component.

A partir d’un node inicial ‘v’ qualsevol, que perteneix a V, anem recorrent, recursivament, cada node no visitat ‘w’, que es un node adjacent o un node successor de ‘v’, marcant-lo com a node visitat fins no trobar més nodes no visitats. Al final del recorregut, si existeix algun node del graf G que no ha estat visitat, significa que el graf es no connex, ja que existeix com a mínim dues components connexes, si no, el graf es connex.

En cas pitjor, que és quan un graf es connex, s'aplica la crida recursiva DFS per cada vèrtex 'v' de V amb un cost O(V), i com al assolir un node explorem els seus nodes adjacents amb un cost O(E) (explorem les arestes connectades al node actual), obtenim un cost temporal total O(V+E) en cas d'implementar l'algorisme amb una llista d'adjacències.

En el cas d'implementar l'algorisme amb una matriu d'adjacències, a més del cost O(V) de recorrer recursivament cada node, hauriem d'explorar una fila sencera de longitud V de la matriu d'adjacència, amb cost O(V), pel que obtindríem un cost temporal total O(V^2).

Aquesta diferència de complexitat temporal es el motiu pel qual en el nostre algorisme hem pres la decisió d'utilizar una llista d'adjacències per recórrer el graf.

4.2 Algorisme del càlcul de les components connexes

Aquest algorisme és molt similar al de connectivitat, ja que empra un recorregut en profunditat (DFS) per un graf G=(V, E). La seva principal diferència es que en lloc de recorrer una única component connexa, recorrerà cada component connexa que contingui el graf. Al final del recorregut, en haver visitat tots els nodes del graf G, s'haurà emmagatzemat cada component connexa que compon el graf.

Respecte a la complexitat temporal, és semblant a la de l'algorisme de connectivitat. La crida recursiva DFS s'aplicarà per cada vèrtex 'v' de V amb un cost O(V) i, en arribar a cada node, explorarà els seus nodes adjacents amb un cost O(E), pel que en total el cost temporal es O(V+E) en cas d'implementar una llista d'adjacències, i de O(V^2) en cas d'utilizar una matriu d'adjacències.

Pel mateix motiu que en l'algorisme de connectivitat, en el nostre disseny hem pres la decisió de representar les adjacències amb una llista d'adjacències. També, ja que l'algorisme és pràcticament igual, aprofitem aquest algorisme del càlcul de components connexes per obtenir la informació necessària per decidir si el graf G és connex o no.

4.3 Algorisme de generació de grafs aleatoris

4.3.1 Generació en model binomial

S’ha de generar un nombre aleatori entre 0 i 1 per a cada parell de nodes del graf. Si aquest nombre es menor o igual que el valor de ‘p’, hi haurà una aresta entre els dos nodes que estiguem tractant en aquell moment.

Així doncs farem (n * n-i) iteracions entre els dos bucles. Aquí hi ha una mostra del codi:

```
for (int i = 0; i < numOfNodes; ++i) {
  for (int j = i + 1; j < numOfNodes; ++j) {
    if ( generateRandom() <= probOfEdge ) {
      adjacentList[i].push_back(j);
      adjacentList[j].push_back(i);
      numOfEdges ++;
    }
  }
}
```

Figura 4: Fragment del codi generador de grafs binomials aleatoris

4.3.2 Generació model geomètric

Una aproximació és calcular la distància d'un vèrtex a cada un dels vèrtexs restants, de manera que tenim n (n-1) / 2 possibles connexions de comprovar. El resultat en termes de cost temporal és O (n ^ 2). Les mostres són generades utilitzant un generador de nombres aleatoris pertanyent al rang [0,1].

/ * L'algorisme queda de la forma:

Per a tot p, q nodes pertanyents a N, si la distància de p a q és menor a la r donada, es crea una aresta de p a q. * /

Pseudocodi:

```
V = generateSamples(n) // Generar n muestras dentro del cubo unitario
for each p of V
    for each q of V\{p}
        if  distance(p,q) <= r
            addConnection(p,q) // Añadir arista (p,q) a la estructura que guarda las aristas
        end
    end
end
```

Figura 5: Pseudocodi generador de grafs geomètrics aleatoris

És important recalcar que aquest algoritme no és escalable, ja que cada vèrtex necessita informació de tots els vèrtexs restants. D'altra banda l'algoritme és correcte perquè es recorren tots els seus components i les seves respectives distàncies es comproven amb una distància donada r.

4.4 Execució dels experiments

Hem fet un algoritme que fa experiments per veure com evolucionen els resultats canviant el nombre de nodes i la probabilitat o la distancia segons el model que escollim.

El que fa l'algorisme es simplement generar molts grafs aleatoris i en calcula les seves components connexes. Això ho fa aprofitant els algorismes esmentats anteriorment. Utilitza un bucle que, va augmentant la 'r' o la 'p' i, per cada una d'elles, en treu la mitjana de les components connexes de tots els grafs que ha generat, en l'algorisme hem utilitzat 10.000 grafs generats per cada 'n' i 'p' o 'r', i així tenir més precisió.

4.4.1 Valor esperat del nombre de components connexes

Esperem tant per el model Binomial com per el model geomètric que, a mesura que la probabilitat i la distància es facin més grans, el valor del nombre de components connexes d'un graf tendís a 1, per tant a ser connex.

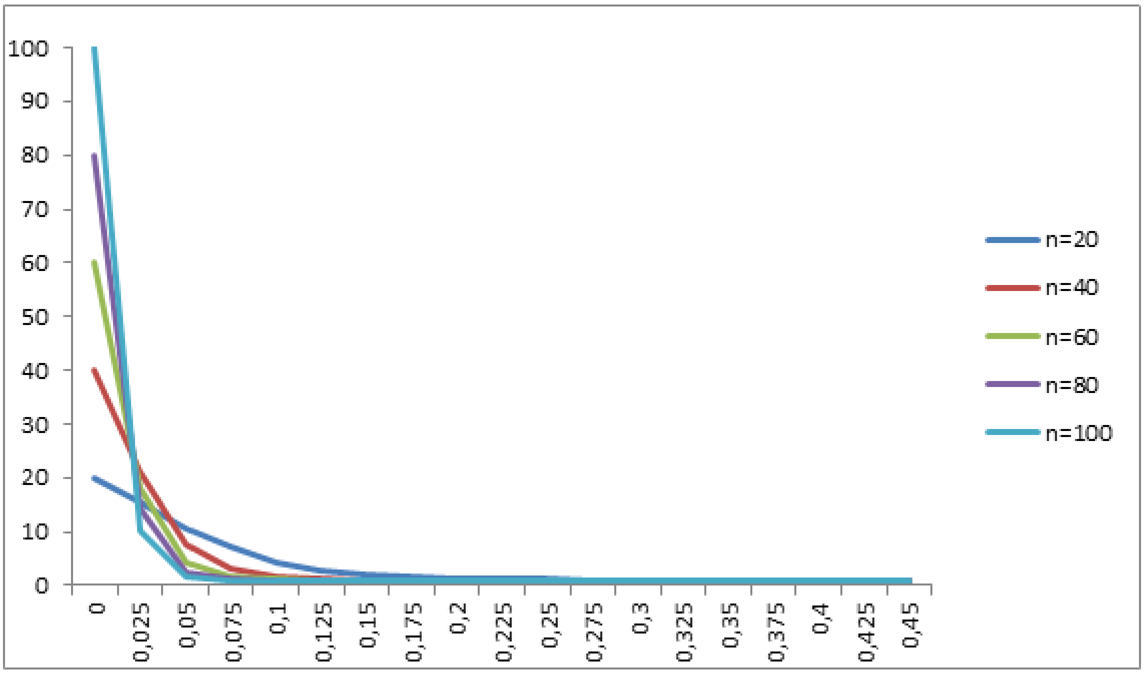
Esperem també que el resultat obtingut depengui també del número de nodes, i que actuï de forma que com més nodes hi ha més possibilitats de ser connex té.

5. Anàlisi dels resultats

Gràcies als generadors de grafs de model binomial i geomètric, hem pogut fer experiments i generar grafs aleatoris que ens han permès estudiar el nombre de components connexes.

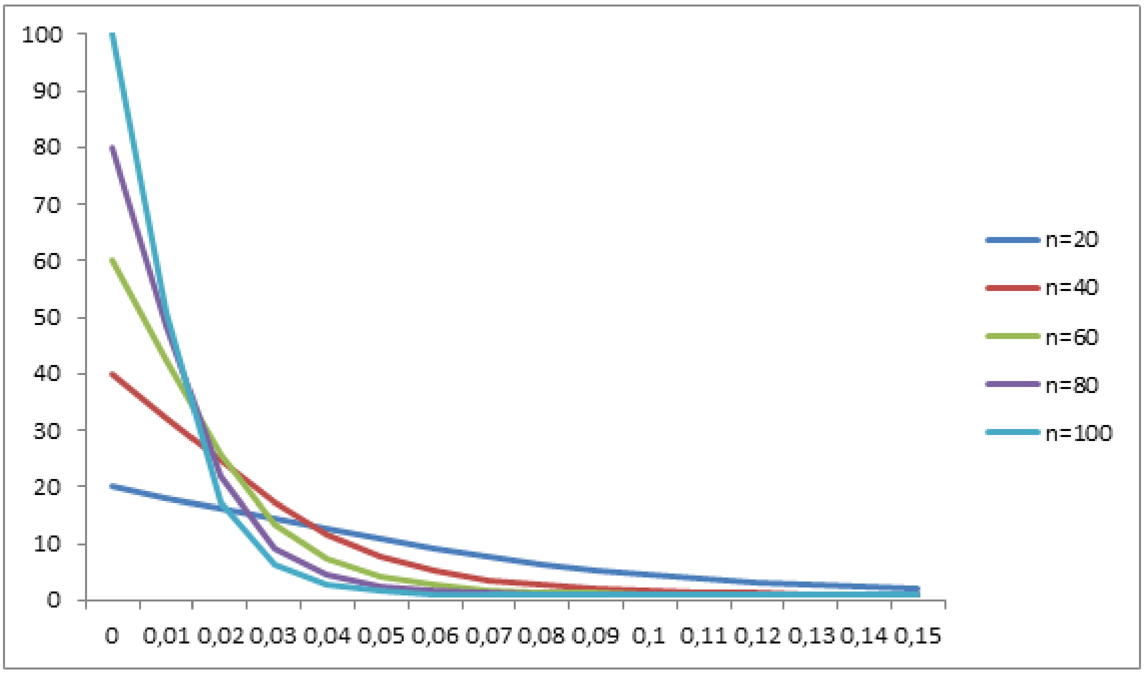
Al model Binomial veiem que, com més alta la probabilitat ‘p’, més tendeix a ser connex, és a dir, a tenir només una component connexa. Però no només depèn de la ‘p’ sino que també depèn del número de nodes, en aquest cas ‘n’. Com més nodes hi ha, és a dir, la ‘n’ més gran, cal una probabilitat més petita per que el graf tingui només una component connexa.

A la següent gràfica veiem el comportament dels grafs que hem generat a partir del model Binomial:



A l’eix vertical hi ha el número de components connexes que s’espera per cada valor ‘p’, que és l’eix horitzontal.

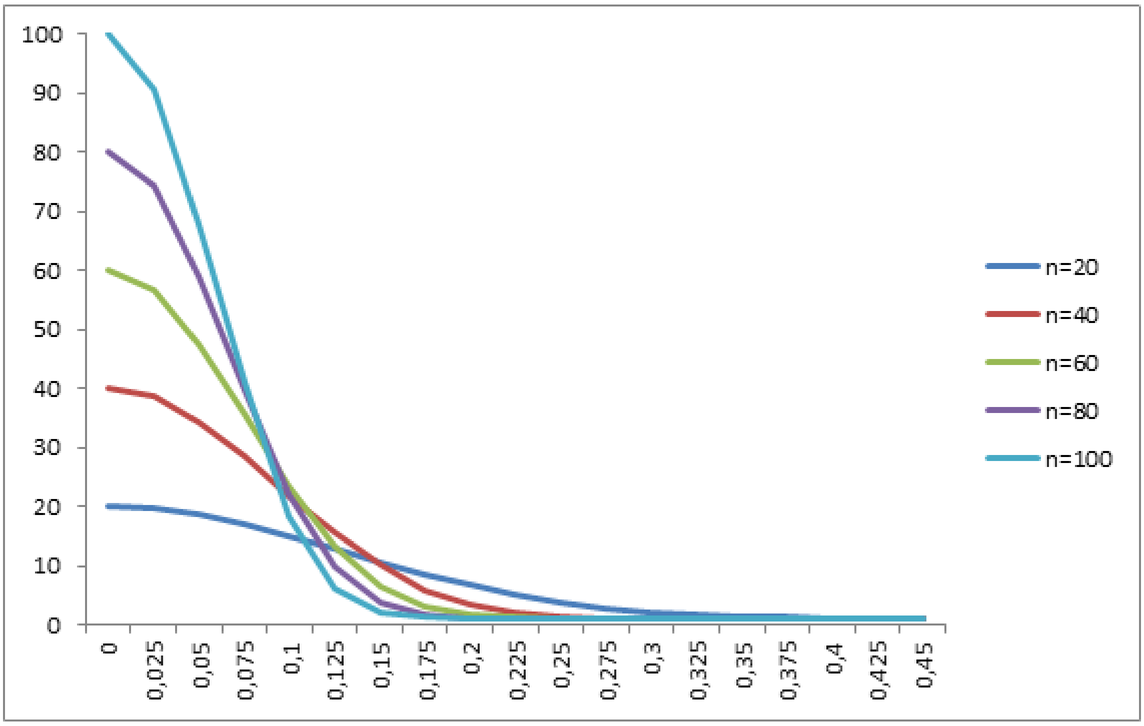
Com que visualment és bastant difícil apreciar el moment més crític de la transició de fase, hem fet un gràfic més detallat:



Podem observar que, com més gran el número de nodes més ràpid es fa connex, sent així per n=20 el model menys pronunciat però que amb poca més probabilitat també es fa connex. Tot al contrari passa amb n=100 que es el que es fa connex amb menys probabilitat.

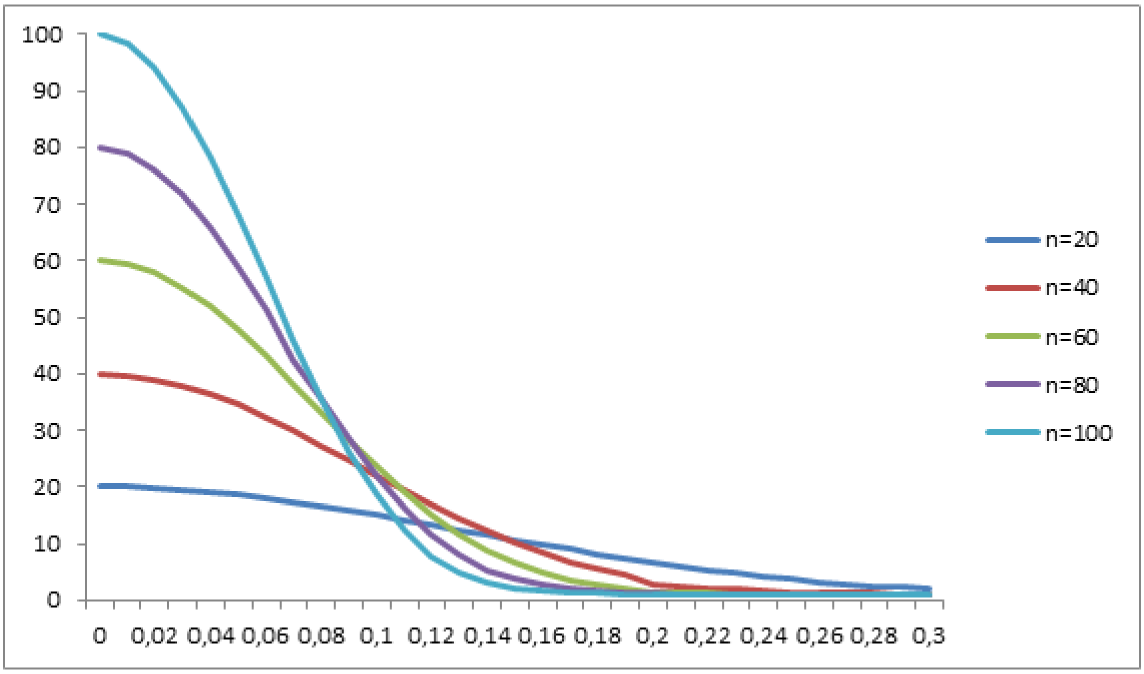
Al model Geomètric hem vist, com passava amb el Binomial, que com més gran la distancia ‘r’ més ràpid es fa connex. Com també passava amb el Binomial, el nombre de nodes es molt important a l’hora d’estudiar la funció, ja que com més gran la ‘n’ més ràpid es converteix en un graf d’una component connexa.

A la següent gràfica veiem el comportament dels grafs que hem generat a partir del model Geomètric:



A l'eix vertical hi ha el número de components connexos que s'espera per cada distancia 'r', que es l'eix horitzontal.

Com passava amb l'altre model, hem volgut estudiar més detalladament la transició de fase de la gràfica:



Veiem doncs que quan el nombre de nodes és més gran, més ràpidament es fa connexa amb una ‘r’ més petita, i com més petita, necessitem una ‘r’ més gran perquè el graf sigui connex.

6. Conclusions

A partir dels dos models que fan grafs aleatoris programats, i un cop comprovada que la seva funcionalitat fos correcta, hem pogut fer experiments que han sortit exitosos.

Mitjançant l’algoritme per fer experiments hem pogut comprovar que els resultats son molt semblants, o gairebé iguals, que els resultats que s’obtenen amb la llibreria networkx.

Això ens ha fet poder veure el comportament de cada un dels algoritmes; el Binomial i el Geomètric.

Ens esperàvem que passat un cert punt amb ‘r’ (entre 0,2-0,3) en el Geomètric i ‘p’(entre 0,1 -0,2) en el Binomial, els grafs fossin connexos. Això ho hem pogut comprovar gràcies a l’experimentació, i la nostra suposició ha estat confirmada. A més, el factor de nombre de nodes ens ha fet veure que com més gran es la ‘n’, més ràpidament es tornaria connex, tal i com esperàvem.

Cal destacar que, en comparació, hi ha un model on els grafs es fan més ràpidament connexos que l’altre. Es el cas del Binomial, que es fa connex més ràpidament que el Geomètric.

7. Referències

[1] Wikipedia: Random geometric graph [en línia]. [Última modificació 9 Agost 2019]. [Consulta 25 Setembre 2019]. Disponible en:
https://en.wikipedia.org/wiki/Random_geometric_graph

[2] Wikipedia: Graph theory [en línia] [Última modificació 20 Setembre 2019] [Consulta 25 Setembre 2019] Disponible en:

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

[3] Wikipedia: Graph [en línia] [Última modificació 19 Setembre 2019] [Consulta 25 Setembre 2019] Disponible en:
https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)

[4] Wikipedia: Connectivity [en línia] [Última modificació 17 Setembre 2019] [Consulta 23 Setembre 2019] Disponible en:
https://en.wikipedia.org/wiki/Connectivity_(graph_theory)

[5] Wikipedia: Grafo conexo [en línia] [Última modificació 1 Juliol 2019] [Consulta 24 Setembre 2019] Disponible en:
https://es.wikipedia.org/wiki/Grafo_conexo

[6] R. Sedgewick and K. Wayne. Algorithms. Addison-Wesley, 4th edition, 2011.
https://algs4.cs.princeton.edu/

[7] M.E.J. Newman. Networks. An Introduction. Oxford University Press, 2010.

[8] Viquipedia: Graf (matemàtiques) [en línia] [Última modificació 19 Agost 2019] [Consulta 25 Setembre 2019] Disponible en:
https://ca.wikipedia.org/wiki/Graf_(matemàtiques)

[9] Viquipedia: Model d'Erdős-Rényi [en línia] [Última modificació 10 Juliol 2019] [Consulta 25 Setembre 2019] Disponible en:
https://ca.wikipedia.org/wiki/Model_d%27Erdős-Rényi