

Algorísmia

Estudi Experimental de Connectivitat i Percolació de Grafs

Pau Belda, Guillem Cabré, Marc Peñalver, Prisca Oleari

Curs 2024-25, Quadrimestre de tardor

Índex

1	Introducció	2
2	Definicions	3
2.1	Percolació	3
2.2	Transició de Fase	3
2.3	Objectius de la Experimentació	3
3	Grafs Seleccionats	5
3.1	Graella Quadrada	5
3.2	Graella Triangular	6
3.3	Graf Geomètric Aleatori	6
3.4	Graf de Barabási-Albert	7
4	Algoritmes	9
4.1	Percolació per Nodes	9
4.2	Percolació per Arestes	10
4.3	Càlcul de Components Connexes	10
5	Experimentació	11
5.1	Metodologia	11
5.1.1	Programa Main	12
5.2	Lectura de taules i gràfics	12
5.3	Graella quadrada	13
5.3.1	Percolació per arestes	13
5.3.2	Percolació per nodes	14
5.4	Graella triangular	15
5.4.1	Percolació per aresta	16
5.4.2	Percolació per node	17
5.5	Graf geomètric aleatori	18
5.5.1	Percolació per arestes	19
5.6	Graf de Barabási-Albert	20
5.6.1	Percolació per nodes	21
6	Conclusions	23
6.1	Coneixements Adquirits	23
7	Bibliografia	24
8	Annex	25

1 Introducció

L'estudi de la connectivitat en grafs sota processos de percolació és un camp d'interès clau en diverses disciplines, incloent la teoria de grafs, la física estadística i l'anàlisi de xarxes complexes. La percolació modela situacions en què nodes o arestes d'un graf poden fallar de manera aleatòria, i aquestes fallades poden tenir un impacte significatiu en la connectivitat global del sistema. Un aspecte crucial en aquests estudis és la transició de fase, un fenomen en què una propietat estructural del graf, com ara la connexió entre els seus components, pateix un canvi sobtat a mesura que es varia un paràmetre, com la probabilitat de fallida.

En aquest estudi, ens centrarem en l'efecte de la percolació sobre diferents tipus de grafs, tant deterministes com aleatoris. Concretament, ens proposem investigar com la percolació de nodes i d'arestes afecta la connectivitat del graf i, més específicament, com es produeix la transició de fase que determina el pas d'un graf connex a un graf fragmentat en múltiples components. Les nostres hipòtesis inicials estan basades en la variació d'aquests fenòmens en funció del tipus de graf i del procés de percolació.

Aquestes hipòtesis ens permetran explorar com diferents configuracions estructurals i processos de fallida afecten la transició de fase i la connectivitat dels grafs.

2 Definicions

2.1 Percolació

La percolació en un graf G consisteix en eliminar o desactivar nodes o arestes, i posteriorment es mesura com això afecta una certa propietat global del graf. Quan desactivem una aresta o un node, direm que ha tingut una fallida.

En termes generals, l'objectiu és estudiar com el graf passa d'estar completament connectat a parcialment o totalment desconnectat a mesura que es treuen alguns dels seus components. Quan parlem de percolació, considerem una probabilitat p que determina si una component del graf (node o aresta) es desactiva aleatòriament. Aquest mecanisme és especialment rellevant per a l'estudi de xarxes complexes, ja que ens ajuda a comprendre com de robust o vulnerable és el sistema que volem analitzar.

- **Percolació per nodes:** Cada node té una probabilitat p de ser desactivat. Un cop fet això, s'analitza com ha canviat la connectivitat del graf.
- **Percolació per arestes:** En aquest cas, les arestes es desactiven en lloc dels nodes. Això també afecta la connectivitat, ja que les connexions directes entre nodes es perden.

Després d'aplicar el procés de percolació (sobre nodes o arestes), s'obté un graf percolat, que és la versió modificada del graf original, amb una connectivitat reduïda i, possiblement, components desconnectades. Aquest graf l'anomenarem G_p .

2.2 Transició de Fase

Una transició de fase d'un graf per a una propietat concreta Π fa referència a un resultat satisfactori d'un procés de percolació aplicat al graf. En el nostre cas aquesta propietat Π serà la connectivitat del graf. O dit d'una altra manera, que el graf només tingui una component connexa.

Definim un resultat com a satisfactori si, donat que es troba una probabilitat de valor q tal que es compleix la propietat Π al graf G_q (definim aquesta probabilitat com q_Π), per als grafs $G_{q'}$ on $q' > q_\Pi$, aquests verifiquen la propietat Π , i als grafs $G_{q'}$ on $q' < q_\Pi$, no la verifiquen (ambdues afirmacions són vàlides si es compleixen amb una probabilitat prou alta).

Quan s'ha obtingut aquest resultat, diem que la propietat Π presenta una transició de fase al voltant de q_Π .

2.3 Objectius de la Experimentació

En aquest projecte, realitzem un estudi experimental sobre la transició de fase en grafs sotmesos a un procés de percolació, modelat mitjançant un paràmetre $p \in [0, 1]$ que representa la probabilitat que un node o aresta falli. L'objectiu principal és analitzar com varia el nombre de components connexes d'un graf durant el procés de percolació, avaluant l'existència d'un valor crític, conegut com a *threshold* o llindar de transició de fase, al voltant del qual el graf experimenta canvis significatius en la seva connectivitat.

Per a aquest anàlisi, considerem diversos models de grafs, incloent xarxes quadrades, grafs geomètrics aleatoris i altres models paramètrics. Els experiments es realitzaran per grafs de diferents mides, enfocant-nos en el comportament asimptòtic a mesura que el nombre de nodes creix. L'estudi es complementarà amb la implementació d'algorismes per a generar aquests grafs, aplicar percolació i calcular el nombre de components connexes.

Objectius específics:

- Estudiar la possible transició de fase en graells quadrades $n = m \times m$, sent n el nombre de nodes, sota un procés de percolació per nodes i arestes amb probabilitat p de que aquestes fallin. La hipòtesi plantejada és que la percolació de nodes generarà una transició de fase més ràpida que la percolació d'arestes.

La raó d'aquesta hipòtesi és que, en eliminar nodes, també s'eliminen totes les arestes associades a aquests nodes, fet que altera la connectivitat de manera més dràstica. Per tant, s'espera que la connectivitat del graf es vegi més afectada quan fallin nodes en comparació amb quan només fallin les arestes.

- Estudiar la possible transició de fase sota un procés de percolació d'arestes en graelles triangulars $n = \frac{rows \times (rows+1)}{2}$, sent n el nombre de nodes.

La nostra hipòtesi és que, com més arestes connectin un node, més resistent serà el graf davant la percolació d'arestes, ja que la seva connectivitat global es veurà menys alterada. Això ens permetrà analitzar com el nombre d'arestes per node influeix en la capacitat del graf per mantenir la seva integritat davant fallades. Per validar aquesta hipòtesi, compararem la percolació per arestes en dos models: el graf de graella quadrada, on cada node està connectat a quatre veïns, i el graf de graella triangular, on cada node està connectat a sis veïns. Aquest contrast ens permetrà estudiar com la diferència en el nombre de connexions per node afecta la robustesa del graf.

- Estudiar la possible transició de fase en grafs geomètrics aleatoris connexos (*Random geometric graphs*), sota un procés de percolació d'arestes.

La nostra hipòtesi és que, a mesura que augmenta el radi de connexió, el graf esdevé més dens, fet que dificulta la fragmentació del graf quan es produeixen fallades. Per tant, preveiem que, a mesura que augmenta la densitat del graf, serà necessària una probabilitat de fallida més alta per provocar una transició de fase.

- Estudiar la possible transició de fase sota un procés de percolació de nodes en un graf de tipus *Hub Graph* o graf Barabási-Albert.

El graf de Barabási-Albert és un model que genera grafs amb una estructura de xarxa d'escala lliure, on alguns nodes (hubs) concentren un nombre desproporcionat d'arestes. La hipòtesi que proposem és que la tolerància d'aquesta xarxa a les fallades es veurà afectada principalment per la percolació de nodes. En concret, si els hubs comencen a fallar, la xarxa perdrà la seva connectivitat molt més ràpidament que en un graf més uniforme, ja que els hubs són crítics per mantenir el graf connectat.

Els resultats obtinguts permetran caracteritzar el comportament de diferents models de grafs sota condicions de percolació i explorar la robustesa d'aquestes xarxes davant de fallades aleatòries.

3 Grafs Seleccionats

En aquesta secció s'exposen els grafs seleccionats per a l'estudi experimental. Concretament, explicarem les peculiaritats de cada graf. Així mateix, es detallaran els algorismes utilitzats per a la generació de cada tipus de graf.

3.1 Graella Quadrada

Algorisme 1 Generació de Graf de Graella Quadrada $G(m \times m)$

Entrada: m (dimensió de la graella)

Sortida: Graf g

```
1: Inicialitzar  $n = m \times m$  (nombre de nodes del graf)
2: Crear un graf buit  $g$  amb  $n$  nodes
3: for  $i = 0$  fins a  $m - 1$  do
4:   for  $j = 0$  fins a  $m - 1$  do
5:      $nodeActual = i \times m + j$ 
6:     if  $i < m - 1$  then
7:        $nodeFilaInferior = (i + 1) \times m + j$ 
8:       Afegir una aresta entre  $nodeActual$  i  $nodeFilaInferior$ 
9:     end if
10:    if  $j < m - 1$  then
11:       $nodeColumnaDreta = i \times m + (j + 1)$ 
12:      Afegir una aresta entre  $nodeActual$  i  $nodeColumnaDreta$ 
13:    end if
14:  end for
15: end for
16: Retornar el graf  $g$ 
```

El generador de grafs de graella quadrada crea un graf amb una estructura regular en què cada node és adjacent als nodes de la fila superior, inferior, esquerra i dreta, si aquests existeixen. Això s'aconsegueix comprovant per cada node si existeixen nodes a la fila inferior i a la columna dreta, i si és així, s'afegeixen les arestes corresponents.

3.2 Graella Triangular

Algorisme 2 Generació de Graf de Graella Triangular $G(rows)$

Entrada: $rows$ (nombre de files)

Sortida: Graf g

```
1: Inicialitzar  $n = \frac{rows \times (rows + 1)}{2}$  (nombre de nodes del graf)
2: Crear un graf buit  $g$  amb  $n$  nodes
3:  $nodeActual = 0$ 
4: for  $i = 0$  fins a  $rows - 1$  do
5:   for  $j = 0$  fins a  $i$  do
6:     if  $j < i$  then
7:        $nodeDreta = nodeActual + 1$ 
8:       Afegir una aresta entre  $nodeActual$  i  $nodeDreta$ 
9:     end if
10:   if  $i < rows - 1$  then
11:      $nodeInferiorEsquerra = nodeActual + i + 1$ 
12:     Afegir una aresta entre  $nodeActual$  i  $nodeInferiorEsquerra$ 
13:      $nodeInferiorDret = nodeActual + i + 2$ 
14:     Afegir una aresta entre  $nodeActual$  i  $nodeInferiorDret$ 
15:   end if
16:    $nodeActual = nodeActual + 1$ 
17: end for
18: end for
19: Retornar el graf  $g$ 
```

El generador de grafs de graella triangular crea un graf amb n nodes, on $n = 1 + 2 + 3 + \dots + rows = \frac{rows \times (rows + 1)}{2}$ per tal de aconseguir una estructura triangular. Aquests nodes tenen una estructura en què cada node està connectat als veïns de la dreta i l'esquerra, així com als nodes superiors i inferiors, tant a l'esquerra com a la dreta, sempre que aquests existeixin. Això s'aconsegueix comprovant per a cada node si hi ha nodes a la fila inferior i a la columna dreta, i si és així, s'afegeixen les arestes corresponents.

3.3 Graf Geomètric Aleatori

Algorisme 3 Generació de Graf Geomètric Aleatori $G(n, r)$

Entrada: n (nombre de nodes), r (radi de connexió)

Sortida: Graf g

```
1: Crear un graf buit  $g$  amb  $n$  nodes
2: Inicialitzar un vector de coordenades  $coords$  de longitud  $n$ 
3: for  $i = 0$  fins a  $n - 1$  do
4:    $coords[i].x = rand01()$ 
5:    $coords[i].y = rand01()$ 
6: end for
7: for  $i = 0$  fins a  $n - 1$  do
8:   for  $j = i + 1$  fins a  $n - 1$  do
9:     if  $distanciaEuclidiana(coords[i], coords[j]) < r$  then
10:      Afegir una aresta entre  $i$  i  $j$  al graf  $g$ 
11:    end if
12:   end for
13: end for
14: Retornar el graf  $g$ 
```

El generador de grafs geomètrics aleatoris crea un graf amb n nodes, on cada node es col·loca aleatòriament en un espai bidimensional unitari. Dos nodes estan connectats per una aresta si la distància entre ells és

menor que un radi r especificat. Això es determina calculant la distància euclidiana entre tots els parells de nodes i afegint arestes quan la distància és inferior a r .

3.4 Graf de Barabási-Albert

Algorisme 4 Generació de Graf de Barabási-Albert $G(n, m_0, m)$

Entrada: n (nombre de nodes), m_0 (nombre de nodes inicials), m (grau de connexió per nou node)
Sortida: Graf g

```

1: Crear un graf buit  $g$  amb  $n$  nodes
2: Inicialitzar un vector connection_degree de longitud  $n$ 
3: for  $i = 0$  fins a  $m_0 - 1$  do
4:   for  $j = i + 1$  fins a  $m_0 - 1$  do
5:     Afegir una aresta entre  $i$  i  $j$  al graf  $g$ 
6:   end for
7:   connection_degree[ $i$ ] =  $m_0 - 1$ 
8: end for
9: for  $i = m_0$  fins a  $n - 1$  do
10:  Inicialitzar un vector buit candidates
11:  while la longitud de candidates és menor que  $m$  do
12:    selectedNode = preferentialAttachment(connection_degree)
13:    if selectedNode no està en candidates then
14:      Afegir selectedNode a candidates
15:    end if
16:  end while
17:  for cada  $j$  en candidates do
18:    Afegir una aresta entre  $i$  i  $j$  al graf  $g$ 
19:    connection_degree[ $j$ ] += 1
20:  end for
21:  connection_degree[ $i$ ] =  $m$ 
22: end for
23: Retornar el graf  $g$ 

```

El generador de grafs de Barabási-Albert crea un graf que segueix el model de creixement de xarxes, on s'afegeixen nodes nous que es connecten a nodes existents en funció del seu grau de connexió. Comença amb un conjunt inicial de m_0 nodes completament connectats, i cada nou node que s'afegeix selecciona m nodes existents per connectar-se, amb una probabilitat proporcional al seu grau de connexió.

Aquesta selecció es realitza mitjançant la funció `preferentialAttachment`, que s'encarrega de seleccionar un node existent basant-se en el seu grau de connexió. La funció funciona de la següent manera:

Algorisme 5 Preferential Attachment

Entrada: *connection_degree* (vector de graus de connexió)

Sortida: *chosen* (node seleccionat)

```
1: Inicialitzar degree_sum = 0, temp_sum = 0
2: for cada i en connection_degree do
3:   degree_sum += i
4: end for
5: random_num = rand() mod degree_sum
6: for i = 0 fins a length(connection_degree) - 1 do
7:   temp_sum += connection_degree[i]
8:   if random_num ≤ temp_sum then
9:     chosen ← i
10:
11:   end if
12: end for
13: Retornar chosen
```

Aquesta funció calcula la suma total dels graus de connexió de tots els nodes existents i selecciona aleatòriament un node, on la probabilitat de seleccionar cada node és proporcional al seu grau de connexió. Així, els nodes amb més connexions tenen una major probabilitat de ser seleccionats, promovent el creixement de xarxes amb característiques d'escala.

4 Algoritmes

En aquesta secció es presenten els principals algorismes emprats en l'estudi de la transició de fase. A continuació, detallarem cada algorisme i donarem una breu explicació d'aquests. Els algorismes abordats són els següents:

4.1 Percolació per Nodes

Algorisme 6 Percolació de Nodes en un Graf

Entrada: Graf $G = (V, E)$ amb n nodes, probabilitat p

Sortida: Graf percolat $G_p = (V', E')$ on $V' \subseteq V$ i $E' \subseteq E$

```
1: Inicialitzar un vector posicioNodes de longitud  $n$ 
2: Assignar  $nbNodesVius = n$ 
3: for cada node  $u$  en  $V$  do
4:   Generar un valor aleatori  $r = \text{rand01}()$ 
5:   if  $r < p$  then
6:     Marcar el node  $u$  com a fallat
7:     Decrementar  $nbNodesVius$ 
8:   else
9:     Actualitzar la posició del node viu  $posicioNodes[u] = u - n + nbNodesVius$ 
10:  end if
11: end for

12: Crear un graf buit  $G_p$  amb  $nbNodesVius$  nodes
13: for cada node  $u$  en  $V$  do
14:   if  $u$  no ha fallat then
15:     for cada node  $v$  en la llista d'adjacència de  $u$  en  $G$  do
16:       if  $v$  no ha fallat  $\wedge u < v$  then
17:         Afegir l'aresta  $(posicioNodes[u], posicioNodes[v])$  a  $G_p$ 
18:       end if
19:     end for
20:   end if
21: end for
22: Retornar el graf  $G_p$ 
```

Aquest algorisme retorna el graf percolat a partir del graf original eliminant nodes amb una probabilitat p . Per a cada node, es genera un valor aleatori entre 0 i 1 mitjançant la funció `rand01()`. Si aquest valor és inferior a p , el node es considera eliminat (fallat) i no es conservarà en el graf resultant.

Els nodes que sobreviuen són reindexats per assegurar que el nou graf té una numeració consecutiva de nodes. Les arestes només es conserven si ambdós nodes que connecten han sobreviscut, i es manté la condició $u < v$ per evitar afegir la mateixa aresta dues vegades, ja que en els grafs no dirigits una aresta (u, v) és equivalent a (v, u) . Així, el resultat és un graf amb una mida reduïda en funció de la probabilitat p , mantenint només els nodes i arestes que han "sobreviscut" al procés de percolació.

4.2 Percolació per Arestes

Algorisme 7 Percolació d'Arestes en un Graf

Entrada: Graf $G = (V, E)$ amb n nodes, probabilitat p

Sortida: Graf percolat $G_p = (V, E')$ on $E' \subseteq E$

- 1: Crear un nou graf buit G_p amb n nodes (còpia profunda de G)
 - 2: **for** cada node u en V **do**
 - 3: **for** cada node v en la llista d'adjacència d' u d'el graf G **do**
 - 4: **if** $u < v \wedge \text{rand01}() > p$ **then**
 - 5: Afegir l'aresta (u, v) al graf G_p
 - 6: **end if**
 - 7: **end for**
 - 8: **end for**
 - 9: Retornar el graf G_p
-

Aquest algorisme retorna el graf percolat a partir del graf original. Això s'aconsegueix eliminant les arestes amb una probabilitat p . Per evitar processar una mateixa aresta més d'una vegada, s'utilitza la condició $u < v$, ja que en grafs no dirigits una aresta (u, v) és equivalent a (v, u) . Tal com s'ha mencionat prèviament, la funció `rand01()` genera un nombre aleatori entre 0 i 1, que es compara amb la probabilitat p per decidir si es manté o s'elimina una aresta.

4.3 Càlcul de Components Connexes

Algorisme 8 Càlcul del Nombre de Components Connexes

Entrada: Graf $G = (V, E)$ amb n nodes

Sortida: Nombre de components connexos *componentCount*

- 1: Inicialitzar un vector *visited* de longitud n amb valors *false*
 - 2: Inicialitzar *componentCount* = 0
 - 3: **for** cada node $i = 0$ fins a $n - 1$ **do**
 - 4: **if** el node i no ha estat visitat **then**
 - 5: Incrementar *componentCount*
 - 6: Realitzar una DFS a partir del node i , marcant els nodes visitats
 - 7: **end if**
 - 8: **end for**
 - 9: Retornar *componentCount*
-

L'algorisme utilitza una cerca en profunditat (DFS) per explorar cada component connexa del graf. Cada vegada que es troba un node no visitat, es crida la funció `dfs` per explorar recursivament tots els nodes connectats a aquest node. Aquesta crida recursiva assegura que tots els nodes del mateix component quedïn marcats com a visitats, evitant comptar-los més d'una vegada.

5 Experimentació

5.1 Metodologia

Per dur a terme l'experimentació del projecte, hem utilitzat diferents eines. Hem programat dos programes en C++, un llenguatge que ens ofereix molta eficàcia temporal i espacial. Aquests programes són el **main** i el **runner**. També hem dissenyat un fitxer de classe **graph** amb tots els atributs i funcions necessàries per operar amb els grafs. Aquesta classe representa els grafs com a llistes d'adjacència.

Per compilar aquests programes, hem fet ús del programari lliure **make**, que automatitza i paral·lelitzla la compilació i l'enllaç.

A més, hem dissenyat scripts per a l'interpret **R**, que és un programari de tractament de dades que ens analitzarà i generarà gràfics dels resultats dels estudis, que estaran en format **.csv**.

Més informació del procés d'experimentació es pot trobar en el GitHub del projecte, premeu [aquí](#) per accedir-hi. Allà, a part del codi, també podreu consultar més informació sobre la generació de grafs, les dependències del programa per compilar-lo i executar-lo, com inserir els paràmetres pels programes i més.

El programa **main**, mitjançant la classe **graph**, ens ha permès analitzar les propietats del canvi de fase a partir dels paràmetres inicials. Aquests paràmetres són els següents:

- **RandomSeed**: La llavor per al generador aleatori.
- **NúmeroMínimNodes**: El nombre mínim de nodes del graf. En el cas de les graelles, el nombre mínim de files del graf.
- **NúmeroMàximNodes**: El nombre màxim de nodes del graf. En el cas de les graelles, el nombre màxim de files del graf.
- **NúmeroNodesStep**: Increment dels nodes en cada iteració.
- **IteracionsPerObtenirResultat**: El nombre de vegades que es provarà la configuració per probabilitat p de percolació i per nombre de vèrtex n .
- **ModePercolació**: Tipus de percolació per nodes o per arestes.
- **PathResultat**: Fitxer on es guardaran els resultats.
- **AlgorismeGeneradorGraf**: Algoritme utilitzat per generar el graf (per exemple, Erdős-Rényi, Square-Grid, etc.).
- **ParàmetresAlgorisme**: Paràmetres addicionals per al generador de graf (opcional segons l'algorisme).

A partir d'aquests paràmetres, el programa **main** escriurà un fitxer **PATH.csv** que posteriorment serà analitzat mitjançant el software de tractament de dades **R**.

Per altra banda, tenim el programa **runner**, que rebrà com a input un fitxer de text. Aquest fitxer tindrà un llistat de paràmetres per diferents experiments del programa **main**. Un exemple d'això seria:

RGN	MIN	MAX	STEP	ITs	PERC-MODE	RESULT-PATH	GEN-ALGORITM	PARAMETERS-GEN
21312	10	100	10	1000	NODE_PERC	./data/test1.csv	Erdos-Renyi	0.1
35353	50	500	50	1000	EDGE_PERC	./data/test2.csv	Random-Geometric	0.3
72479	100	1000	100	100	EDGE_PERC	./data/test3.csv	Square-Grid	

El programa **runner**, per cada fila del fitxer que rep, inicialitzarà una instància del programa **main**, aconseguint d'aquesta manera automatitzar molt més els tests, podent córrer diferents programes **main** simultàniament.

5.1.1 Programa Main

Per entendre els resultats també s'ha d'entendre les decisions que s'han pres per la recollida de dades. Analitzarem el programa `main` mitjançant un pseudocodi per no entrar en conceptes avançats de C++. A continuació vegeu una mostra del pseudocodi:

Algorisme 9 Descripció de l'experiment

```
1: Seleccionar opcions de configuració
2: Inicialitzar el generador de nombres aleatoris
3: Obrir l'arxiu CSV i escriure la capçalera
4: for  $n$  in range(MIN_NB_NODES, MAX_NB_NODES + 1, NB_NODES_STEP) do
5:   for  $p$  des de 0 fins a 1 amb pas 0.01 do
6:     Inicialitzar el comptador de grafs connexos
7:     for  $i = 0$  fins a TRIES_PER_P do
8:       repeat
9:         Generar el graf seleccionat( $p, n, params$ )
10:      until el graf és connex
11:      Aplicar percolació (per nodes o arestes) al graf
12:      if el graf percolat és connex then
13:        Incrementar el comptador de grafs connexos
14:      end if
15:    end for
16:    Escriure entrada resultant al CSV
17:  end for
18: end for
```

Ara, analitzarem el codi. Per començar, el programa preguntarà per totes les opcions necessàries. D'aquesta manera, ens podem permetre tenir un sol programa que pugui fer tot el que necessitem i que sigui altament modular. S'utilitzarà la llavor per generar nombres aleatoris, i així l'experiment podrà ser repetit amb els mateixos resultats. Després, crearà el fitxer `PATH.csv`, al qual s'hi inseriran entrades que posteriorment s'analitzaran.

Ara analitzarem l'algorisme encarregat de generar els resultats. Vegeu com primerament iterarem sobre n tantes vegades com s'hagi indicat en la entrada. Alhora, també iterarem per cada n sobre una probabilitat de fallida de percolació. Aquest bucle tindrà 100 iteracions, $p \in \{0.00, 0.01, \dots, 1.00\}$. A més d'aquests dos bucles, iterarem una altra vegada sobre p i n tantes vegades com l'usuari hagi indicat en l'apartat *IteracionsPerObtenirResultat*. Així, es farà una mitjana amb més o menys mostres.

S'iniciarà un comptador a 0 que representarà el nombre de grafs percolats connexos. S'utilitzarà el generador de grafs seleccionat a les opcions per generar el graf que posteriorment serà percolat. Vegeu que aquí generarem grafs fins a aconseguir un graf connex. Aquesta decisió la vam prendre per tenir una representació més acurada de la transició de fase. Més endavant, es tornarà a considerar aquesta decisió, ja que hi ha grafs, com ara el *Random Geometric Graph*, que requereixen un paràmetre r , el qual, amb valors petits de r , acostuma a generar grafs no connexos.

Per acabar, percolarem el graf $G(V, E)$ de la manera que s'hagi especificat a l'input, ja sigui per nodes o per arestes, obtenint G_p . A G_p se li aplicarà un algorisme que determinarà si el graf és connex. Si ho és, s'incrementarà el comptador. Quan les *IteracionsPerObtenirResultat* s'hagin completat, s'escriurà l'entrada resultant al fitxer `PATH.csv`.

5.2 Lectura de taules i gràfics

Abans de procedir amb els experiments, es presenten les claus per interpretar correctament les taules i gràfics que acompanyen els resultats.

Gràfics: Els gràfics mostren l'evolució dels experiments en funció del paràmetre p (probabilitat de fallida) a l'eix de les X, mentre que l'eix Y mostra la probabilitat que el graf percolat G_p sigui connex. A més, en cada gràfic col·locarem més d'un experiment, donats diferents valors de n (nombre de nodes),

pintarem de colors diferents cada experiment. Aquest últim paràmetre a vegades canviarà, ja que en grafs, com per exemple en el quadrat, mesurarem amb el paràmetre m , que és el nombre de files. També, al peu de cada graf, hi haurà la seqüència de inputs amb el que s'ha generat aquest, l'ordre i significat de cada valor el podreu trobar a la subsecció 5.1.

Taules: Les taules contenen les mètriques que són més difícils de veure amb els gràfics. Cada fila representa una instància específica de l'experiment, mentre que les columnes mostren:

- **N:** nombre de nodes de l'experiment (pot variar dependent del graf).
- **Threshold:** punt d'inflexió en què apareix la transició de fase.
- **% Connex:** la probabilitat, en tant per u, de que el graf percolat sigui connex en el punt p .
- **Pendent:** pendent del segment en el punt *threshold*.

Creiem convenient explicar com calculem el punt d'inflexió o punt *threshold*. El trobarem com el segment amb el pendent més pronunciat.

5.3 Graella quadrada

Per estudiar la possible transició de fase a graelles quadrades, hem decidit fer proves amb grafs de mides diferents. Més concretament, hem agafat dues mostres: una que contempla grafs d'entre $n = 2 \times 2$ fins a $n = 20 \times 20$, i l'altra mostra serà de $n = 20 \times 20$ fins a $n = 200 \times 200$. Això ens servirà per estudiar com afecta la percolació a la connectivitat, tant percolació per nodes com per arestes. Hem triat aquestes mides de grafs perquè creiem que ens permetran observar com evoluciona la transició de fase fins que aquesta arribi a un llindar, ja que per grafs amb $|V|$ majors el canvi de transició de fase serà cada cop més similar, arribant fins a un llindar.

5.3.1 Percolació per arestes

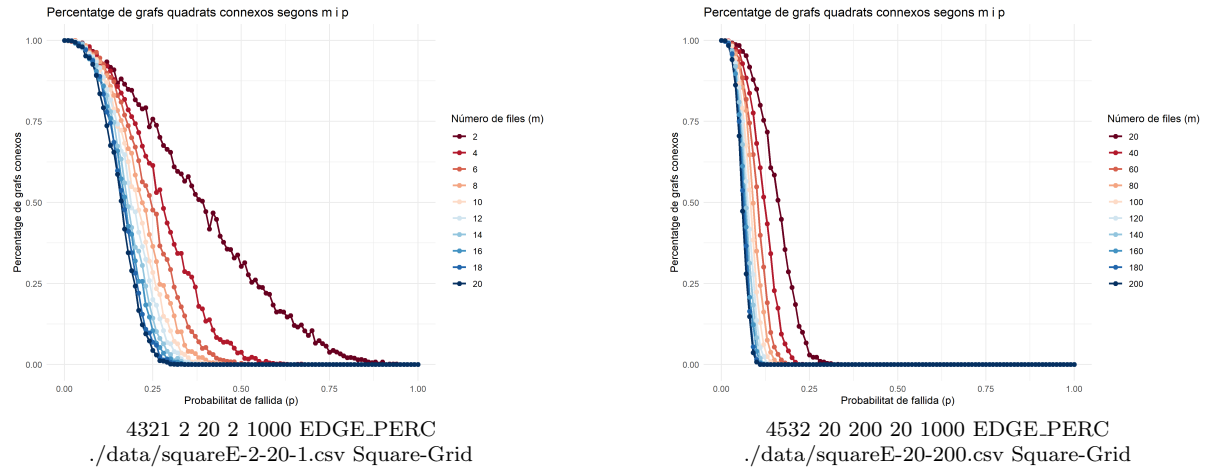


Figura 1: Comparativa de percolació per arestes del grafs quadrats amb mides diferents

La *Figura 1* mostra una comparació de la percolació per arestes en grafs quadrats de mides diferents. Ambdós gràfics representen la relació entre la probabilitat de fallada p (eix x) i el percentatge de grafs connexos (eix y). A mesura que p augmenta, s'eliminen més arestes i, per tant, hi ha menys connectivitat en els grafs resultants.

En els dos gràfics, podem veure com a mesura que creix el nombre de nodes dels grafs, la transició de fase s'intensifica i apareix en valors de p més petits.

Amb aquests gràfics, podem observar que, tot i que el punt inicial on comença la transició de fase és força similar per a totes les mides de grafs observats, l'abruptesa del canvi de fase s'intensifica a mesura

que $|V|$ augmenta progressivament. A continuació, analitzarem els valors referents al punt *threshold* de cada mida de graf.

M	Threshold	% Connex	Pendent	M	Threshold	% Connex	Pendent
2	0.24	0.73	-5.90	20	0.17	0.42	-8.70
4	0.26	0.53	-8.30	40	0.15	0.23	-11.40
6	0.27	0.37	-9.80	60	0.11	0.40	-14.90
8	0.24	0.40	-7.50	80	0.09	0.50	-14.90
10	0.19	0.55	-7.70	100	0.10	0.22	-16.00
12	0.18	0.56	-8.70	120	0.07	0.56	-18.50
14	0.17	0.56	-7.40	140	0.07	0.48	-19.60
16	0.18	0.43	-9.00	160	0.08	0.21	-21.60
18	0.15	0.60	-8.80	180	0.06	0.54	-21.30
20	0.17	0.42	-8.70	200	0.06	0.47	-23.20

Taula 1: Dades sobre la transició de fase de grafs quadrats amb percolació per arestes

Com hem mencionat anteriorment, la *Taula 1*, ens mostra el *threshold* on apareix la transició de fase i la probabilitat que el graf sigui connex en aquest valor de p , per cada nombre de files del graf. En les graelles de poques files podem veure un *threshold* que ronda el 0.25, i a mesura que augmenta la mida del graf aquest valor va disminuint. En la taula observem com varia la velocitat amb la que disminueix el *threshold*, aquesta cada cop és menor. Això ens indica que, per molt que anem augmentant la mida de la graella, el punt on aparegui la transició de fase serà cada vegada en valors de p menors, fins arribar a un llindar.

A més, amb la taula podem veure numèricament les observacions que fèiem amb els gràfics. El valor de p on hi ha la fallida total disminueix a mesura que la m augmenta. I que la transició de fase és més abrupta, ja que observant el pendent que ens indica la taula, podem veure que el pendent dels segments on està el *threshold* creix a mesura que m també creix.

5.3.2 Percolació per nodes

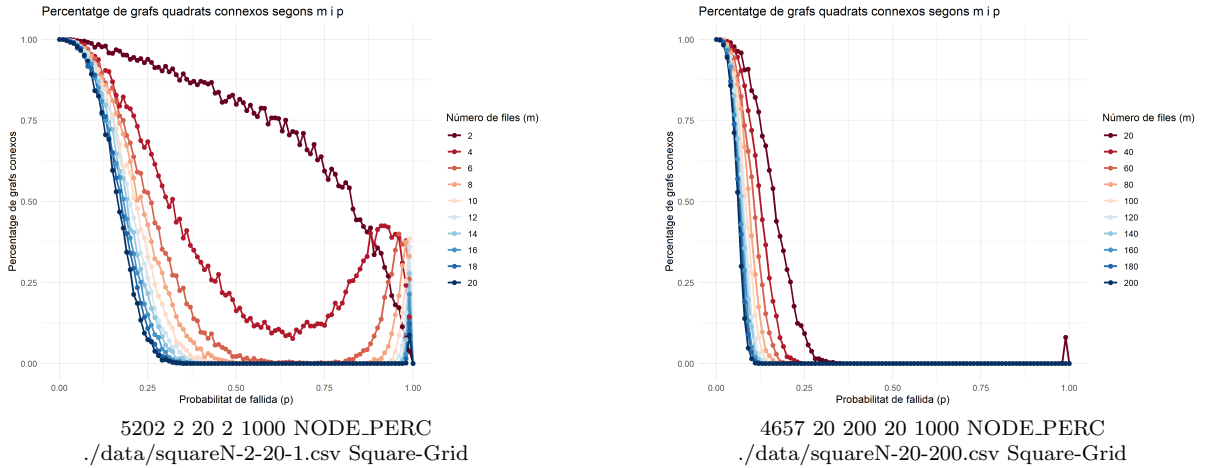


Figura 2: Comparativa de percolació per nodes del grafs quadrats amb mides diferents

La *Figura 2* mostra la comparació de la percolació per nodes en grafs quadrats de mides diferents. Com en la percolació per arestes, el gràfic de l'esquerra mostra els resultats per grafs amb un nombre de vèrtex menor que els del gràfic de la dreta. Hem utilitzat les mateixes mides que en la percolació per aresta per poder comparar els resultats i observar diferències en el mètode que utilitzem per percolar.

Els resultats representats en el gràfic ens indiquen que graelles quadrades amb molt poques files, menys de 8, continuen mantenint un percentatge alt de connectivitat en qualsevol valor de p . A mesura que creix la graella això es perd i apareix una transició de fase. Igual que en la percolació per arestes, la

mida del graf influeix en el pendent de la transició de fase, quan més gran és el graf més ràpida i brusca és la pèrdua de connectivitat.

En graelles petites la transició és més suau i, fins i tot, observem una pujada després de la baixada inicial en valors de p per sobre del 0.75, que indica un increment en el percentatge de grafs connexos quan hi ha una probabilitat de fallida alta. Aquest fenomen apareix quan la percolació genera un sol *cluster*. Això significa que al eliminar la gran majoria de nodes, es queda una petita component connexa, anomenada *cluster*. A mesura que els grafs augmenten en nombre de vèrtex, la probabilitat de que aparegui un *cluster* és més petita, ja que la distància entre *clusters* generats serà major, i amb que n'hi hagi dos separats el graf resultant ja no serà connex.

Podem calcular la probabilitat de que aparegui un *cluster*, per simplicitat calcularem la probabilitat de que aquest tingui un sol node. Sigui m el nombre de files, $n = m \times m$ nombre de nodes, p probabilitat de fallida de node i $q = 1 - p$ probabilitat de que aquest no falli. Tindrem que $t = n \times q \times p^{n-1}$, on t serà la probabilitat de que el graf percolat tingui un sol *cluster* amb un node. Vegem un exemple que verifiqui aquest càlcul: donats $m = 6 \Rightarrow n = 36$, $p = 0.95 \Rightarrow q = 0.05$, ara calculem $t = 36 \times 0.05 \times 0.95^{36-1} \approx 0.299$. Veiem com en el gràfic (*Figura 2*), quan m val 6, i la probabilitat p s'aproxima a 0.95, tenim una connectivitat propera al 29.9% que acabem de calcular.

Ara ens interessa fer una comparació de resultats amb les diferents percolacions. Com hem vist que els grafs tendeixen a un *threshold* cada vegada més baix quan el graf incrementa el número de nodes, farem la comparació amb els *threshold* calculats per graelles de mides a partir de $m = 20$. A més, la diferència en els grafs de mida inferior és més fàcil de veure gràficament, amb la pujada final en la percolació per node sent el principal canvi. En graelles grans és més difícil veure alguna diferència, així que numèricament serà més clar.

M	Percolació per arestes			Percolació per nodes		
	Threshold	% Connex	Pendent	Threshold	% Connex	Pendent
20	0.17	0.42	-8.70	0.17	0.44	-10.10
40	0.15	0.23	-11.40	0.13	0.44	-10.50
60	0.11	0.40	-14.90	0.11	0.45	-12.60
80	0.09	0.50	-14.90	0.10	0.39	-14.50
100	0.10	0.22	-16.00	0.08	0.52	-15.60
120	0.07	0.56	-18.50	0.07	0.58	-16.80
140	0.07	0.48	-19.60	0.07	0.52	-17.70
160	0.08	0.21	-21.60	0.07	0.43	-20.10
180	0.06	0.54	-21.30	0.08	0.19	-20.80
200	0.06	0.47	-23.20	0.07	0.30	-22.90

Taula 2: Comparació de la transició de fase de graelles quadrades grans amb diferents percolacions

La *Taula 2* conté la comparació de dades de transició de fase per graelles quadrades grans al fer la percolació per aresta o per node. Si ens fixem en els valors del *threshold* en els dos casos són valors molt semblants, així com tampoc podem veure un gran canvi en el percentatge de grafs connexos que tenim en aquest valor de p . Amb aquestes dades podem concloure que no hi ha molta diferència entre fer la percolació per nodes o per arestes en les graelles quadrades a partir de que tinguin un mida que deixi de generar connectivitat per *clusters*.

5.4 Graella triangular

En el cas de les graelles triangulars, hem estudiat la percolació per arestes seguint el mateix enfocament utilitzat per a les graelles quadrades. Els grafs considerats van des de 5 fins a 55 files, amb increments de 5 files, i des de 50 fins a 150 files, amb increments de 10 files.

Hem optat per estudiar la percolació per aresta amb l'objectiu d'analitzar si el fet de tenir una graella similar a la quadrada, però amb els vèrtexs interiors (aquells que no es troben a la vora del graf i, per tant, tenen totes les arestes possibles) de grau superior, afecta la probabilitat de fallida en què es produeix la

transició de fase. El grau dels vèrtex interiors de les graelles quadrades era de 4, el de les triangulars de 6. Considerem que, en aquest cas, la probabilitat de fallida associada a la transició de fase serà superior a la de les graelles quadrades, ja que el grau dels vèrtexs és més alt, i per tant, hi ha més arestes que han de fallar per desconectar els nodes. Posteriorment, hem decidit estendre l'estudi a la percolació per node per tal de comparar els resultats amb els obtinguts en graelles quadrades.

5.4.1 Percolació per aresta

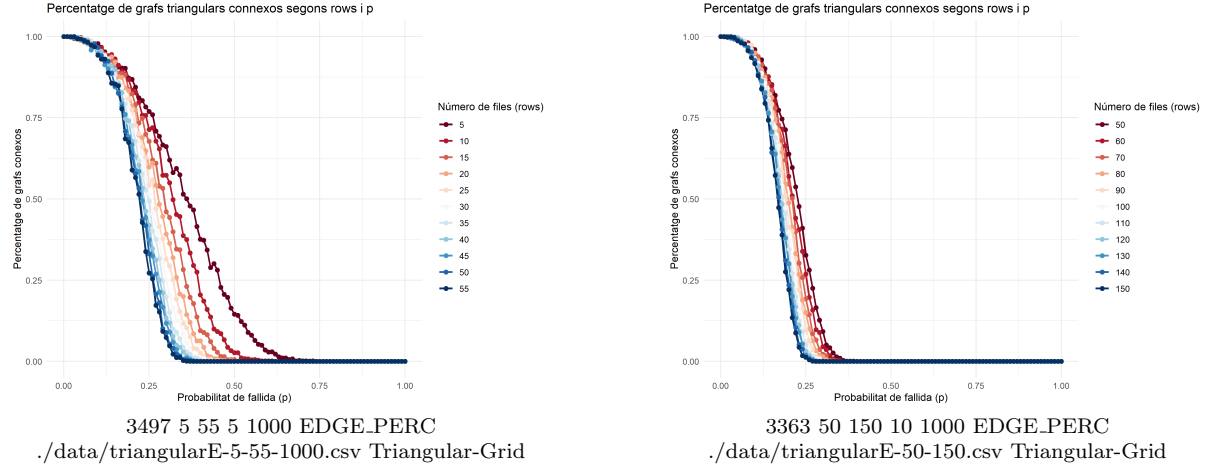


Figura 3: Comparativa de percolació per arestes de graelles triangulars amb mides diferents

La *Figura 3* mostra la comparació de la percolació per arestes en graelles triangulars amb els resultats obtinguts en dos grafs de mostres diferents. El gràfic de l'esquerra presenta un conjunt de nodes $|V|$ inferior al de la dreta. La informació proporcionada en ambdós gràfics és la mateixa que en l'experimentació amb graelles quadrades.

En ambdós casos es pot observar una transició de fase, evidenciada per una caiguda brusca en un valor determinat de p . Tanmateix, en el cas dels grafs amb menys files, aquesta caiguda és més gradual i la transició de fase és menys pronunciada. En canvi, en grafs amb més files, la transició es produeix de manera més ràpida. A més, si s'analitza el valor del punt *threshold*, es a major nombre de files menor és aquest punt en p .

Per tant, de manera similar a com ocorre en els grafs quadrats, la mida del graf influeix en el valor de la probabilitat de fallida en què es produeix la transició de fase, la qual disminueix a mesura que el graf augmenta de mida.

M	Threshold	% Connex	Pendent	M	Threshold	% Connex	Pendent
5	0.35	0.52	-5.90	50	0.25	0.33	-8.30
10	0.29	0.57	-8.10	60	0.20	0.57	-9.40
15	0.30	0.46	-7.20	70	0.22	0.40	-10.20
20	0.22	0.70	-7.60	80	0.19	0.55	-9.80
25	0.27	0.47	-8.80	90	0.20	0.41	-9.80
30	0.24	0.56	-7.70	100	0.18	0.54	-12.20
35	0.26	0.40	-9.20	110	0.20	0.34	-11.00
40	0.18	0.75	-8.00	120	0.19	0.38	-10.70
45	0.23	0.49	-9.10	130	0.17	0.54	-10.10
50	0.23	0.44	-9.70	140	0.15	0.64	-9.90
55	0.18	0.68	-9.20	150	0.19	0.28	-11.30

Taula 3: Dades sobre la transició de fase de grafs triangulars amb percolació per arestes

Amb la *Taula 3* podem observar numèricament el que hem descrit en els gràfics: el valor on es produeix la transició de fase disminueix a mesura que M augmenta. Com en les graelles quadrades, el valor

del *threshold* disminueix lentament amb cada increment de mida, però, en canvi, el pendent en el punt de transició augmenta. Això indica que la pèrdua de connectivitat és més abrupta en grafs més grans i que la transició de fase està influenciada per la mida del graf.

Pel que fa a la comparació amb les graelles quadrades, tot i que ja es pot apreciar als gràfics que la p on es produeix la desconexió total és més petita en les graelles quadrades, en aquesta taula es poden veure els resultats de grafs amb el mateix nombre de files, la qual cosa permet una comparació més detallada.

# files	Threshold square	Threshold triangular
10	0.19	0.29
20	0.17	0.22
40	0.15	0.18
60	0.11	0.20
80	0.09	0.19
100	0.10	0.18
120	0.07	0.19
140	0.07	0.15

Taula 4: Comparació del punt *threshold* entre els grafs quadrats i triangulars amb percolació per arestes

La *Taula 4* compara el punt de *threshold* en grafs amb el mateix nombre de files però de diferent tipus: graelles quadrades i triangulars. Els resultats mostren que el punt de *threshold* en les graelles quadrades és menor que en les triangulars. Això ens permet confirmar la hipòtesi formulada durant l'experimentació, segons la qual un grau superior dels nodes influeix en el valor de p on es produeix la transició de fase.

5.4.2 Percolació per node

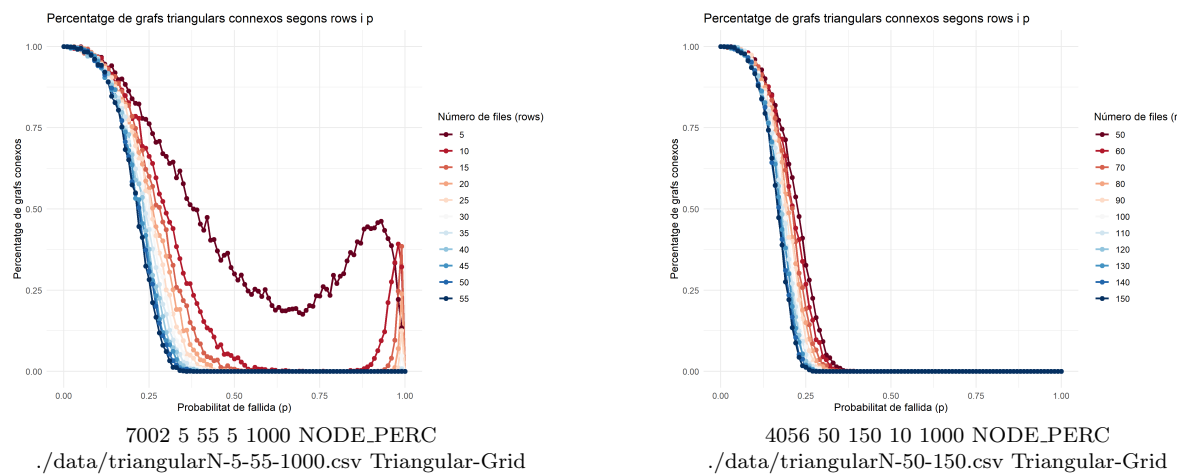


Figura 4: Comparativa de percolació per nodes de graelles triangulars amb mides diferents

Després de fer la comparació amb la percolació per arestes entre graelles triangulars i quadrades hem decidit fer-ho també per percolació per nodes, la *Figura 4* ens mostra els resultats obtinguts de la percolació per nodes en graelles triangulars.

En aquest cas, podem observar resultats similars a la resta obtinguts de graelles, on podem veure una transició de fase que a mesura que la mida del graf augmenta és més brusca i en un valor de p menor. En els grafs amb menys de 25 podem observar el mateix fenomen que en les graelles quadrades, on hi ha una caiguda de la connectivitat quan p ronda el 0.3, però en valors de p molt alts el graf recupera una mica el percentatge de connectivitat.

De forma similar a les graelles quadrades, aquesta anormalitat és deguda a que la percolació de nodes dona peu a que pugui ocórrer la següent casuística: donat un graf de n nodes fallin $n - 1$ nodes, aleshores el graf resultant, d'un sol node serà connex. La probabilitat de que això passi la podem calcular de la

següent manera. Sigui p , la probabilitat de fallida, $q = 1 - p$ probabilitat de que un node no falli, sigui m nombre de files i $n = \frac{m \times (m+1)}{2}$ el nombre de nodes. Sigui t la probabilitat de que el graf sigui connex amb un sol node, $t = n \times q \times p^{n-1}$. Provarem el següent cas, $m = 5 \Rightarrow n = 15, p = 0.95 \Rightarrow q = 0.05$, aleshores veiem que $t = 15 \times 0.05 \times 0.95^{15-1} \approx 0.366$. Veiem com a partir d'aquest càlcul podem explicar el fenomen que apareix en probabilitats de fallida majors. Ja que per $m = 5$ observem com la probabilitat de grafs connexos en $p = 0.95$, s'aproxima al 36.6% que acabem de calcular.

A continuació vegeu una taula que ens explica numèricament els valors referents al punt *threshold*. Hem obviat els grafs de mida inferior a $m = 50$ ja que tots ells presenten una transició difusa ja sigui perquè és inexistent o per les anormalitats que apareixen quan el valor p s'aproxima 1.0 per lo explicat en el paràgraf anterior.

M	Percolació per arestes			Percolació per nodes		
	Threshold	% Connex	Pendent	Threshold	% Connex	Pendent
50	0.25	0.33	-8.30	0.23	0.42	-8.90
60	0.20	0.57	-9.40	0.21	0.50	-8.30
70	0.22	0.40	-10.20	0.24	0.22	-11.40
80	0.19	0.55	-9.80	0.23	0.24	-11.00
90	0.20	0.41	-9.80	0.21	0.31	-11.40
100	0.18	0.54	-12.20	0.21	0.29	-10.50
110	0.20	0.34	-11.00	0.21	0.26	-10.60
120	0.19	0.38	-10.70	0.20	0.29	-9.20
130	0.17	0.54	-10.10	0.18	0.43	-12.00
140	0.15	0.64	-9.90	0.20	0.22	-10.80
150	0.19	0.28	-11.30	0.18	0.38	-9.40

Taula 5: Comparació de la transició de fase de grafs triangulars grans amb diferents percolacions

Com podem veure en la *Taula 5* els valors que obtenim al fer la percolació per aresta o per node en graelles triangulars és molt semblant. En general, els valors de *threshold* obtinguts en la percolació per node són una mica més alts, però per contra el percentatge de grafs connexos en aquell punts és una mica més alt en la percolació per arestes.

Per tant si volem grafs amb més connectivitat seria millor fer la percolació per arestes, però la diferència és petita així que s'obtingran resultats molt semblants si es fa la percolació per nodes. Els pendents són molt similars en els dos casos així que la rapidesa en la que els grafs perden la connectivitat, com hem pogut veure en els gràfics, és casi igual.

5.5 Graf geomètric aleatori

L'estudi de transició sobre grafs geomètrics aleatoris depen, a més de la probabilitat de percolació p , dels dos paràmetres que s'utilitzen per a la creació del graf: el nombre de nodes n i el radi de connectivitat r . En aquesta experimentació, s'han fet 6 casos de prova, on es compara l'impacte de tots dos paràmetres en diferents nivells. En el cas del nombre de nodes n , els valors que s'han utilitzar són de 10 a 100 i de 100 a 1000, donant-nos resultats per a grafs amb pocs nodes fins a grafs amb una quantitat relativament gran de nodes. Per part del radi de connectivitat r , els valors que s'han escollit són de 0.2, 0.5, 0.8, que ens permeten observar grafs amb una connectivitat bastant feble en el cas del primer valor, grafs amb una connectivitat molt alta amb $r = 0.8$ i un mig entre tots dos valors.

5.5.1 Percolació per arestes

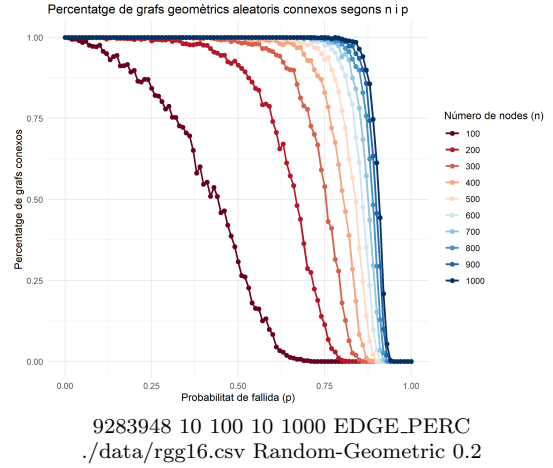
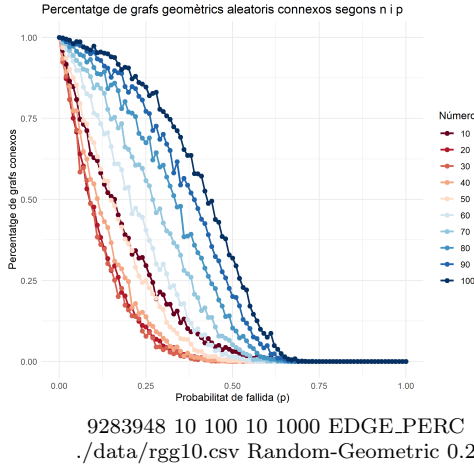


Figura 5: Comparativa de percolació per arestes del grafs geomètrics aleatoris amb $n = 10..100$ (1) i $n = 100..1000$ (2), $r = 0.2$

En aquesta primera comparativa, *Figura 5*, veiem com grafs amb una connectivitat feble amb pocs nodes deixen de generar grafs connexos quan hi ha una probabilitat de percolació relativament baixa. No obstant, a mesura que augmenten els nodes, la quantitat de grafs connexos en funció de la probabilitat de percolació comença a créixer de manera notable (imatge 2). Això ens mostra que, tot i ser grafs amb una connectivitat teòrica feble, amb una quantitat de nodes suficientment gran es pot aconseguir una transició de fase tot i que la probabilitat de percolació sigui alta.

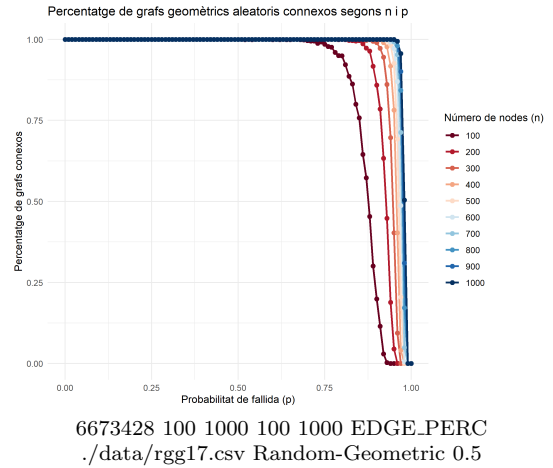
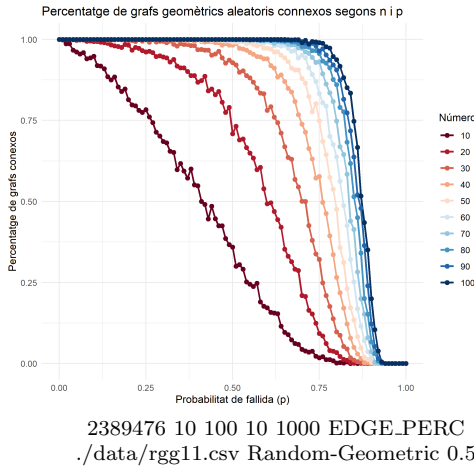
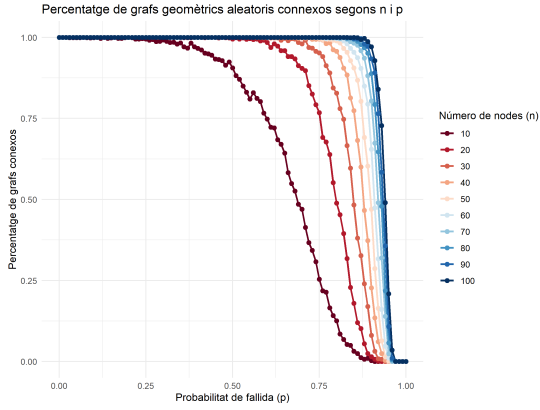
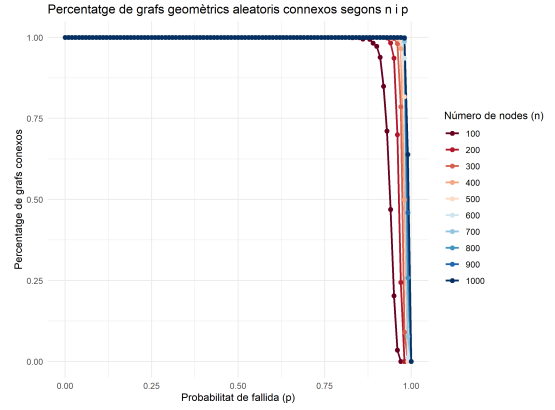


Figura 6: Comparativa de percolació per arestes del grafs geomètrics aleatoris amb $n = 10..100$ (1) i $n = 100..1000$ (2), $r = 0.5$

Per altra banda, la *Figura 6* mostra els resultats dels grafs amb una connectivitat teòrica major als anteriors. L'impacte d'aquesta connectivitat major es veu reflectit clarament en tots dos experiments. En els casos de grafs amb una quantitat de nodes no gaire elevada, el percentatge de grafs connexos en funció de la probabilitat de percolació és bastant elevat fins i tot per al cas de grafs d'únicament 10 nodes (imatge esquerra). Més enllà d'aquesta quantitat, el percentatge incrementa de forma notòria fins a arribar un punt en el que la diferència entre les probabilitats de percolació per a cada quantitat de nodes és irrisòria.



8273563 10 100 10 1000 EDGE_PERC
./data/rgg12.csv Random-Geometric 0.8



9293745 100 1000 100 1000 EDGE_PERC
./data/rgg18.csv Random-Geometric 0.8

Figura 7: Comparativa de percolació per arestes del grafs geomètrics aleatoris amb $n = 10..100$ (1) i $n = 100..1000$ (2), $r = 0.8$

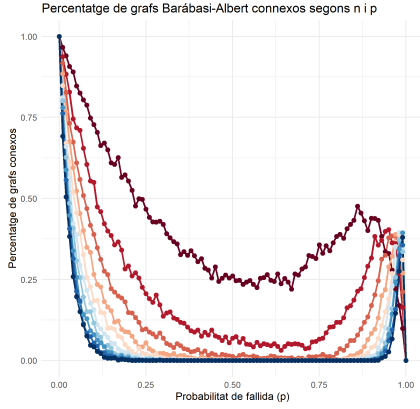
Per últim, a *Figura 7* s'utilitzen grafs geomètrics aleatoris amb un radi de connectivitat molt elevat, donant-lis una forma semblant a un graf complet. Per aquest motiu, fins que la probabilitat de percolació no és molt alta, els grafs generats segueixen sent connexos una vegada aplicada la transició de fase. Evidentment, l'augment de nodes provoca que el percentatge de grafs connexos creixi ja que els nodes tenen més probabilitats d'estar units a altres nodes (imatge 2), fet que ha succeït a totes les comparatives.

5.6 Graf de Barabási-Albert

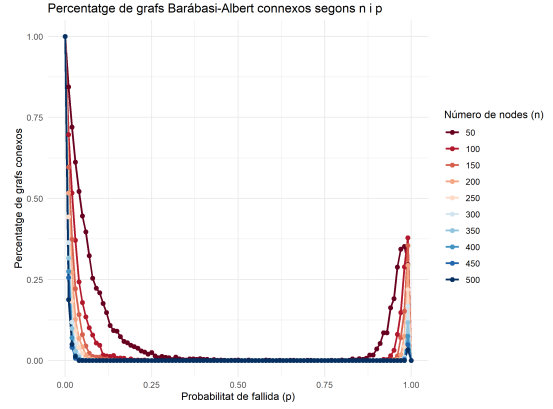
L'estudi que s'ha realitzat sobre els grafs generats amb el model de Barabási-Albert té com a objectiu estudiar la transició de fase sota percolació per nodes. Volem veure l'impacte de la percolació a un graf que conté molts vèrtexs units entre sí per un o més *hubs* centrals, l'impacte que tindria la fallida d'algun d'aquests nodes o, per altra banda, el no impacte que pot tenir sobre nodes que tenen una estructura de fulla, és a dir, nodes que estan connectats únicament a un altre node i per tant la seva fallida segueix permetent un graf connex.

Els experiments realitzats sobre aquests grafs depenen de tres paràmetres: nombre de nodes n , nombre de nodes inicials o *hubs* m_0 , i el nombre d'arestes que han de tenir com a mínim cada node que no és *hub*. Per als nostres experiments, hem decidit donar els següents valors: per al nombre de nodes $n = 10..100$, $n = 50..500$, per al nombre inicial de nodes o *hubs* $m_0 = 2$, $m_0 = 5$, $m_0 = 10$, i per el nombre d'arestes incidents a la resta de nodes $m = 1$, $m = 2$, $m = 5$. Abans de veure els resultats, hem cregut adient una explicació del perquè s'han agafat aquests valors i no d'altres. Els grafs construïts amb el model de Barabási-Albert són típicament utilitzats com a xarxes on els nodes s'uneixen mitjançant *hubs*. Per aquest motiu, hem decidit utilitzar aquests valors ja que representen d'una manera força realista una xarxa generada per aquest model d'una forma computacionalment viable.

5.6.1 Percolació per nodes



4575745 10 100 10 1000 NODE_PERC ./data/ba9.csv
Barabasi-Albert 2 1

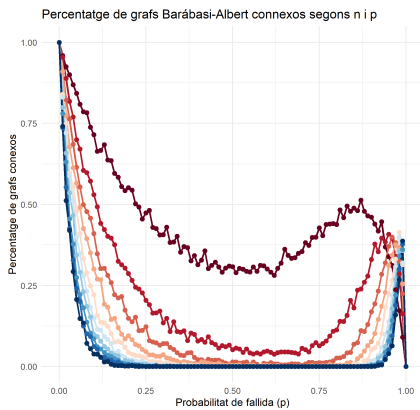


7673658 50 500 50 1000 NODE_PERC ./data/ba13.csv
Barabasi-Albert 2 1

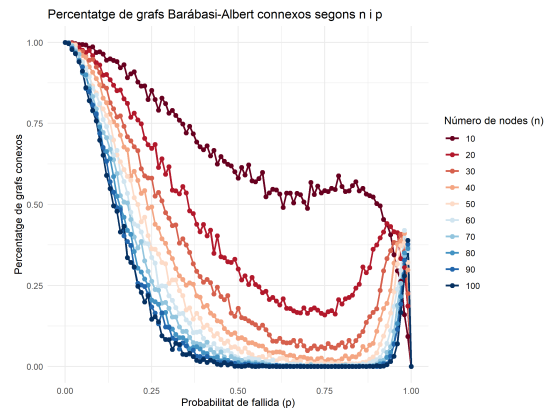
Figura 8: Comparativa de percolació per arestes del grafs Barabási-Albert amb $n = 10..100$ (1) i $n = 50..500$ (2), $m_0 = 2$, $m = 1$

En aquesta primera comparativa (*Figura 8*) entre grafs amb nodes que varien des de $n = 10..100$ i $n = 50..500$, podem veure una tendència similar a tots dos experiments, però molt més pronunciada en el segon cas. Aquesta tendència mostra que a partir d'una probabilitat de percolació relativament petita, el percentatge de grafs conexas disminueix de manera abrupta. Això es pot explicar degut a que, com que el paràmetre m , que representa el nombre d'arestes a les que es connecten els nodes inicialment, és igual a 1, molts nodes estan connectats únicament a un altre node que fa de *hub*. Si aquest últim falla, el graf deixa de ser connex.

Es pot observar també com, quan la probabilitat de percolació s'aproxima a 1, el percentatge de connectivitat augmenta sobretot en grafs amb pocs nodes. Aquest comportament ja s'ha observat en els experiments de grafs quadrats i la seva explicació és la mateixa, a mesura que la probabilitat de fallida s'aproxima a 1, la probabilitat que fallin tots els nodes excepte un (que equival a un graf connex) augmenta. Aquest fenomen es dona quan menys nodes té un graf, ja que serà més probable que només quedi un sol node després de la percolació.



9937843 10 100 10 1000 NODE_PERC ./data/ba10.csv
Barabasi-Albert 5 1



8800547 10 100 10 1000 NODE_PERC ./data/ba11.csv
Barabasi-Albert 5 2

Figura 9: Comparativa de percolació per arestes del grafs Barabási-Albert amb $n = 10..100$, $m_0 = 5$, $m = 1$ (1) i $m = 2$ (2)

En aquest experiment s'han utilitzat com a variables $m_0 = 5$, $m = 1/2$ i un nombre de nodes $n = 10..100$. Amb aquesta comparativa podem veure l'impacte d'augmentar el nombre d'arestes inicials que tenen els nodes que no són *hubs* inicialment. Tot i que l'augment és simplement d'una aresta més per node, la probabilitat de percolació en la qual apareix la transició de fase és significativament superior

a la de l'experiment anterior. A més, podem observar a través del primer gràfic de la *Figura 9*, com l'augment d'únicament el nombre de *hubs* inicials m_0 , no és gaire significatiu en els resultats de connectivitat, donant uns gràfics molt similars als del primer experiment de Barabási-Albert.

Una altra vegada, podem observar com quan la probabilitat de percolació s'aproxima a 1.0, la connectivitat dels grafs augmenta pel mateix motiu que feia anteriorment en la percolació per nodes.

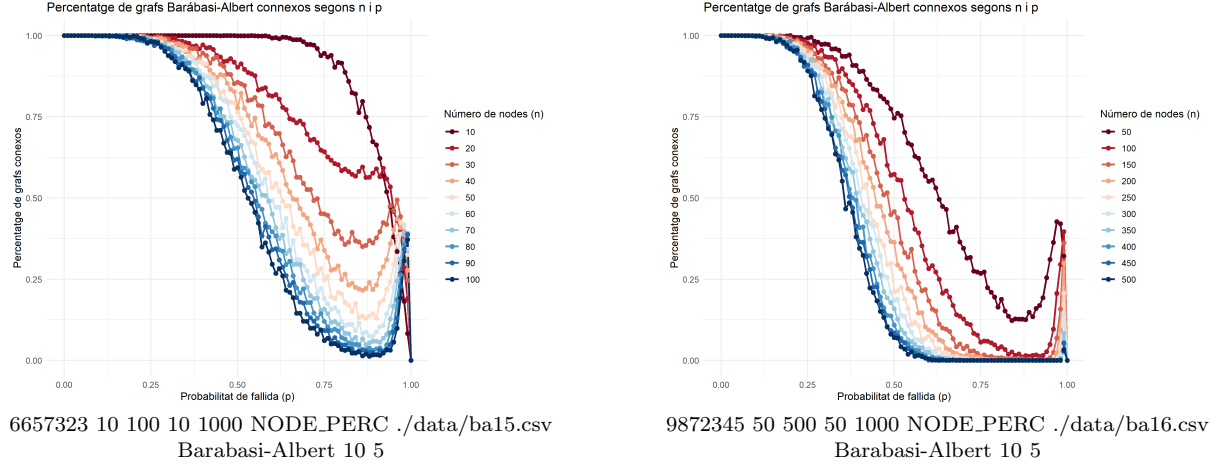


Figura 10: Comparativa de percolació per arestes del grafs Barabási-Albert amb $n = 10..100$ (1) i $n = 50..500$ (2), $m_0 = 10$ $m = 5$

Per últim, en aquest experiment hem utilitzat els paràmetres $m_0 = 10$, $m = 5$, i un nombre de nodes $n = 10..100$ i $n = 50..500$, amb la fi de comprovar l'impacte d'augmentar tots els paràmetres sense deixar que els grafs generats tinguin una estructura de xarxa més que d'un graf complet. Tot i que no semblin similars, totes dues imatges mostren el mateix comportament que han tingut els altres dos experiments, amb una baixada abrupte de la connectivitat més pronunciada com més nodes hi han i un augment d'aquesta quan la probabilitat de percolació s'aproxima a 1.

No obstant, la principal diferència és que, amb tots els paràmetres augmentats, la connectivitat és manté alta en totes dues imatges fins que la probabilitat de percolació no arriba fins a un valor molt superior als altres dos experiments. Això és degut a que els grafs creats en aquest experiment, tot i seguir el model de Barabási-Albert, són els grafs més semblants al seu equivalent complet dels tres experiments, augmentant la seva connectivitat per defecte i sota una percolació de nodes.

6 Conclusions

En aquest estudi, hem explorat els efectes de la percolació en diversos models de grafs, analitzant com la probabilitat de fallada i el nombre de nodes afecten la seva connectivitat i com es produeix la transició de fase. Els models seleccionats inclouen la graella quadrada, la graella triangular, els grafs geomètrics aleatoris i el model de Barabási-Albert. Hem aplicat tant percolació per arestes com per nodes, observant com varien els resultats en funció de la mida i estructura del graf.

Els resultats obtinguts mostren patrons clars: a mesura que augmenta el nombre de nodes, la transició de fase esdevé més abrupta i ocorre amb valors més petits de probabilitat de fallada. Això és especialment evident en grafs més densos, com les graells triangulars i els grafs geomètrics amb un radi de connexió gran, on la desconexió completa es retarda. En canvi, en el model de Barabási-Albert, la percolació afecta més ràpidament la connectivitat quan fallen els hubs, mostrant la fragilitat de les xarxes amb concentració de connexions.

En futurs treballs, seria interessant ampliar l'estudi amb altres tipus de grafs, com els grafs Erdős-Rényi o els grafs dirigits, així com investigar a part de la connectivitat com a propietat booleana, el nombre de components connexes després del procés de percolació. A més, es podria explorar l'impacte de percolacions selectives o correlacionades, on la fallada d'un node o aresta depengui de la fallada dels seus veïns, per entendre millor les propietats de robustesa i vulnerabilitat de les xarxes complexes.

6.1 Coneixements Adquirits

Al llarg d'aquest projecte, hem adquirit una sèrie de coneixements tècnics i habilitats que han estat essencials per a la seva execució i anàlisi. Entre els principals aspectes apresos, es poden destacar:

- **Latex i Markdown:** Hem millorat considerablement les nostres habilitats en l'ús de \LaTeX i Markdown per a la redacció de documents tècnics i científics, facilitant la presentació clara i estructurada dels resultats.
- **GitHub:** Hem après a gestionar el codi del projecte de manera col·laborativa i eficient utilitzant GitHub, permetent un control de versions robust i la integració contínua dels diferents components desenvolupats.
- **C++ Modern:** S'ha perfeccionat l'ús de característiques avançades de C++ modern, així com la paral·lelització eficient de codi utilitzant *threads*, accelerant els processos computacionals.
- **Exportació de dades:** Hem après a exportar els resultats dels experiments en format *.csv* mitjançant C++, permetent una anàlisi més detallada i estructurada.
- **Gràfics amb R:** S'ha utilitzat R per a la creació de gràfics que mostren clarament les transicions de fase i els resultats de les simulacions de percolació.
- **Teoria de Percolació:** Hem aprofundit en el procés de percolació i les seves implicacions en la transició de fase, especialment en diferents models de grafs.
- **Generadors de Grafs:** Hem aplicat diferents tipus de generadors de grafs, incloent els models d'Erdős-Rényi (que finalment no el vam incloure a la documentació), grafs geomètrics aleatoris (RGGs) i el model de Barabási-Albert, observant els patrons característics de percolació en cadascun.

7 Bibliografia

- [1] E. Weisstein, "Site Percolation", *MathWorld – A Wolfram Web Resource*. Disponibile: https://es.wikipedia.org/wiki/Modelo_Barab%C3%A1si%E2%80%9393Albert. [Accedit: 17-oct-2024].
- [2] "Lattice graph", *Wikipedia*. Disponibile: https://en.wikipedia.org/wiki/Lattice_graph. [Accedit: 2-oct-2024].
- [3] E. Weisstein, "Triangular Grid Graph", *MathWorld – A Wolfram Web Resource*. Disponibile: <https://mathworld.wolfram.com/TriangularGridGraph.html>. [Accedit: 2-oct-2024].
- [4] R. Sedgewick i K. Wayne, "Union-Find Algorithms", *Princeton University*. Disponibile: <https://algs4.cs.princeton.edu/lectures/keynote/15UnionFind-2x2.pdf>. [Accedit: 2-oct-2024].
- [5] "Percolation Theory", *Wikipedia*. Disponibile: https://en.wikipedia.org/wiki/Percolation_theory. [Accedit: 21-oct-2024].
- [6] E. Weisstein, "Bond Percolation", *MathWorld – A Wolfram Web Resource*. Disponibile: <https://mathworld.wolfram.com/BondPercolation.html>. [Accedit: 17-oct-2024].
- [7] G. R. Grimmett i A. M. Stacey, "CRITICAL PROBABILITIES FOR SITE AND BOND PERCOLATION MODELS", Disponibile: <http://www.statslab.cam.ac.uk/~grg1000/papers/UScrit6.pdf>. [Accedit: 17-oct-2024].
- [8] "Random Geometric Graph", *Wikipedia*. Disponibile: https://en.wikipedia.org/wiki/Random_geometric_graph. [Accedit: 19-oct-2024].
- [9] "Barabasi-Albert model", *Wikipedia*. Disponibile: https://en.wikipedia.org/wiki/Barab%C3%A1si%E2%80%9393Albert_model. [Accedit: 19-oct-2024].

8 Annex

Enllaç al Codi

El codi font i documentació addicional del projecte estan disponibles al repositori de GitHub. Podeu accedir-hi mitjançant el següent enllaç:

[GitHub - Estudi de Percolació i Connectivitat](#)

En aquest repositori, a més del codi, es proporciona informació sobre la generació de grafs, les dependències del programa, instruccions de compilació i execució, així com exemples de com configurar els paràmetres per realitzar els experiments.