Eina de simulació de models de difusió basant en grafs

Albert Ros Fite

Resum—L'objectiu d'aquest projecte és construir una eina que permeti simular un model de propagació epidèmica sobre una estructura de graf. Els grafs són un conjunt de nodes o vèrtexs que es comuniquen entre ells mitjançant arestes. Partim d'una base de dades de població fictícia i aquest el traduïm a graf, de manera que cada node serà una persona i aquests estan comunicats mitjançant les arestes a les persones que coneixen. Per dur a terme les relacions entre els nodes del graf m'he centrat principalment en dos factors: si dos individus pertanyen a la mateixa comunitat i si tenen la mateixa edat.

Si sabem que una persona està infectada podem establir una probabilitat de què les persones adjacents estiguin infectades i així successivament. Això ens ajudaria a predir, per exemple, si una comunitat estarà afectada i haurà de fer confinament o no. Podem modificar els paràmetres de l'algorisme perquè estableixi diferents valors als llindars de contagi o índexs de contagi entre individus.

L'eina construïda permet parametritzar un model SIR (Susceptible-Infectat-Recuperat) i executar una simulació de propagació, visualitzant l'evolució del procés epidèmic. L'eina podria utilitzar-se per fer prediccions sobre processos de contagi, i prendre decisions sobre intervencions.

Paraules clau—graf, propagació, simulació, base de dades, Python

Abstract— The aim of this project is to build a tool to simulate an epidemic spread model on a graph structure. Graphs are a set of nodes or vertices that communicate with each other through edges. We start from a fictitious population dataset and translate it into a graph, so that each node will be a person and these are communicated through the edges to the people they know. To work out the relationship between graph nodes, I focused on two factors: whether two individuals belong to the same community and whether they are the same age.

If we know that a person is infected we can establish a probability that the adjacent people are infected and so on. This would help us to predict, for example, whether a community will be affected and will have to do confinement or not. We can modify the parameters of the algorithm to set different values on the thresholds of infection or rate of infection between individuals.

The build-in tool allows you to parameterize a SIR (Susceptible-Infected-Recovered) model and run a propagation simulation, visualizing the evolution of the epidemic process. The tool could be used to make predictions about contagion processes, and to make decisions about interventions.

Index Terms— graph, simulation, propagation, dataset, Python

1 Introducció

En aquesta època de COVID hem vist moltes dades que es basen en models de propagació de processos epidèmics. Aquests models es basen en una representació d'una població en graf i a partir de determinats paràmetres com el llindar de contagi, índex de contagi i temps de recuperació preveuen com varia en el temps el nombre de contagis. Un dels models de difusió és el model SIR, on individu forma part d'un dels tres estats: susceptible, infectat, recuperat.

En teoria de grafs, un graf és una representació abstracta d'un conjunt d'objectes on alguns parells dels objectes estan connectats per enllaços. Els objectes interconnectats són representats per abstraccions matemàtiques anomenades vèrtexs i els enllaços que connecten alguns parells de vèrtexs s'anomenen arestes.

Típicament, un graf es descriu de forma esquemàtica com a conjunt de punts o vèrtexs per als vèrtexs, units per línies o corbes per les arestes.

Les arestes poden ser dirigides o no dirigides. Per exemple, un graf es pot construir escollint com a vèrtexs els primers 1000 enters positius, i definint que hi ha una aresta entre dos vèrtexs si i només si els dos enters corresponents tenen com a mínim una xifra decimal en comú.

- *E-mail de contacte: 1362389@autonoma.cat*
- *Menció realitzada: Computació*
- *Treball tutoritzat per: Josep Lladós Canet*
- *Curs 2021/22*

En altres casos la relació entre els vèrtexs no és simètrica: per exemple, un graf es pot construir escollint els vèrtexs per ser els primers 1000 enters positius, i definint que hi ha un vèrtex des de i fins a j si i és un divisor de j. Aquest tipus de graf s'anomena un graf dirigit i les arestes s'anomenen dirigides o arcs; per contrast, un graf on no es dirigeixen les arestes s'anomena no dirigit.

En el nostre cas treballarem amb grafs no dirigits, ja que dos individus han d'estar en relació mútua. No pot ser que un node A estigui en contacte amb un node B i que aquest node B no tingui contacte amb A.

Si poguéssim enregistrar els contactes entre individus, podríem fer simulacions a partir de processos de propagació en grafs. Haurem de crear relacions entre els nodes de la manera més real possible com, per exemple, per edat o comunitat. Això, simularia per exemple l'aplicació RadarCOVID que es va intentar posar en marxa per enregistrar amb els mòbils

L'objectiu d'aquest projecte és agafar una població real o simulada i mitjançant models de difusió basats en grafs poder extreure diferents característiques d'aquest. Una vegada tinguem les dades a partir de diferents models i establint un nivell de risc de contagi podem predir quina és la probabilitat de què una persona estigui afectada per la pandèmia.

En particular, el projecte consta de tres objectius:

-Importació de dades: Dissenyar un model de dades basat en grafs i les corresponents funcions d'importació a partir de dades reals sobre evolució de la pandèmia, o bé per generar dades sintètiques. Això ens permetrà tenir una base de dades amb la que podem començar a treballar. Crear base de dades amb individus amb assignació d'atributs aleatòria.

-Estructura: Implementar un model de difusió sobre el graf. Aquest model ha de permetre configurar els paràmetres de probabilitat de contagi, recuperació, etc. El model ha de permetre executar una simulació de propagació en un nombre determinat d'iteracions. El model de difusió que utilitzarem és el SIR.

-Visualització: una vegada tinguem el graf amb les diferents connexions i feta la simulació es tracta de poder visualitzar-ho d'una manera que sigui més amigable. També tinc pensat crear un vídeo amb les imatges resultants de la simulació.

La resta d'aquest informe està estructurat com segueix. A la secció 2 descrivim la planificació del projecte. A la secció 3 es detallen les eines que s'han fet servir per al desenvolupament. A la secció 4 s'explica la metodologia usada per al projecte: model de difusió SIR, model de representació en forma de graf i el front end del sistema.
La secció 5 detallem els diferents experiments que hem fet per veure els resultats del nostre model. Per últim, tenim la secció 6 de conclusions del projecte.

2 Planificació

Per assolir els objectius platejats, el projecte es va organitzar en les tasques que es descriuen a continuació.

- Creació de bases de dades d'experimentació. Ens vàrem plantejar diferents opcions, o bé trobar dades públiques amb informació d'individus i contagis, o bé treballar amb dades simulades. Finalment, l'hem creat nosaltres aleatòriament.
- Preparar l'entorn per poder fer feina amb llibreries com networkX [2] o openCV [5], entre d'altres.
- Creació dels diferents grafs segons la relació entre els individus.
- Implementar el model de propagació SIR.
- Dissenyar una UI per poder configurar els paràmetres d'execució de l'algorisme.
- Visualitzar la simulació amb un vídeo.
- Fer l'anàlisi dels resultats del procés de simulació.

A la figura (1) mostro el diagrama de Gantt del projecte:

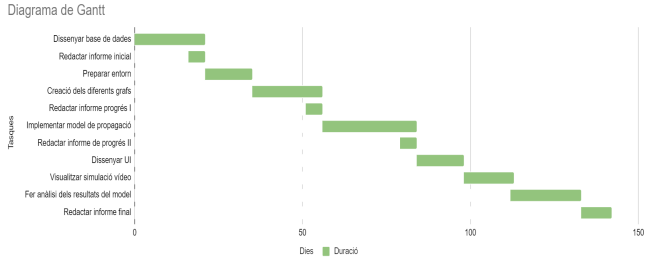

Fig 1. Diagrama Gantt projecte

El dia 0 correspon a la data inicial del projecte 19/09/2021 i com a data final tenim 8/02/2022.

2.1 Arquitectura del projecte

El projecte conte 4 fitxers principals: data_generation.py, reader.py, menu.py i video.py.

El data_generation.py conté la creació aleatòria dels individus. Crea un arxiu amb extensió .csv que serà el que després llegirà el fitxer reader.py.

El reader.py conté la part gran del projecte. Conté funcionalitats com la lectura de dades, creació del graf, inicialització de node infectat, acolorir nodes segons estat, assignació de pesos, model de propagació SIR i dibuixar la simulació.

El menu.py conté tant el disseny com la funcionalitat de la interfície d'usuari. La principal funcionalitat és agafar al paràmetres seleccionats per l'usuari i al clicar al botó de simular enviar-los a les funcionalitats del fitxer reader.py per fer la simulació.

Per últim, tenim el fitxer video.py que si l'executem ens crearà el vídeo amb les imatges de l'última simulació feta.

A la figura (2) mostro la vista estructural del projecte:

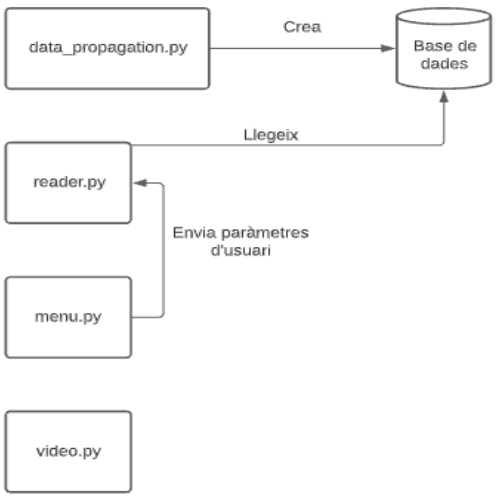

Fig 2. Vista estructural projecte

3 Eines Utilitzades

Per dur a terme aquest projecte s'han fet servir un conjunt d'eines que tot seguit llistem:

- Entorn de desenvolupament: Pycharm [3].
- Diferents llibreries de Python com networkX [2], numpy o openCV [5].
- Metodologia àgil: Scrum (Sprint cada dues setmanes).
- Software de visualització de grafs: Gephi [6]
- Eina per informes: Google Docs.

4 Mètode

4.1 Model de representació en forma de graf

Tal com hem dit a la introducció, els grafs estan formats per un conjunt de nodes i arestes. En el meu cas, els nodes són les persones i les arestes són les relacions entre els diferents nodes del graf. Per poder fer un sistema que faci la simulació de la propagació de l'epidèmia primer de tot necessitem una base de dades d'individus per representar-la en forma de graf i així poder començar a treballar.

Utilitzarem la llibreria networkX [2] de Python que ens proporciona diferents funcionalitats tant per a l'estudi de grafs com per a l'anàlisi d'aquests. Ha sigut difícil trobar una base de dades amb població real, ja que aquestes dades normalment estan protegides,així que, vaig decidir crear-ne un de fictici.

Per a cada individu vaig crear els diferents camps: identificador, comunitat, edat i estat. Per crear la base de dades simplement vaig crear un array de comunitats i un d'edats amb diferents valors. Per a cada individu, agafo una comunitat i una edat a l'atzar dins de l'array i li assigno. La funció permet indicar el nombre d'individus aleatoris que vols que et generi.

Els nodes dels grafs es caracteritzen amb els següents atributs:

- Identificador únic
- Comunitat
- Edat
- Estat (SIR)

Els atributs rellevants per les arestes que caracteritzen el model de difusió són:

- Índex de contagi alpha (comunitat)
- Índex de contagi beta (edat)
- Llindar de contagi

Una vegada tenim la base de dades creada ens disposem a crear les relacions entre els diferents nodes. Els nodes els comuniquem si pertanyen a la mateixa comunitat o si tenen la mateixa edat i aquesta és inferior o igual a 20.

La idea és que si dos individus viuen a la mateixa comunitat, podem assignar una probabilitat de contagi entre aquests, ja que segurament acabaran creuant-se al llarg del temps. En el cas de l'edat, ho relaciono de manera que si un individu té vint anys o menys llavors significa que va al col·legi, per tant, també estarà en contacte amb altres individus que vagin al col·legi.

Una vegada tenim el graf creat, ens disposem a assignar un risc de contagi per cada aresta del graf. Aquest pes dependrà de si la connexió és deguda al fet que els nodes pertanyen a la mateixa comunitat o tenen la mateixa edat. El risc de contagi si dos nodes viuen a la mateixa comunitat, l'anomenarem alpha i el de l'edat beta. Perquè

el risc de contagi tingui efecte hem d'assignar manualment l'estat infectat a algun node del graf perquè pugui infectar també als altres depenen de les diferents relacions esmentades abans.

4.2 Model de difusió: SIR

Una vegada tenim els pesos inicialitzats s'ha d'aplicar una funció de propagació al graf per veure l'evolució d'aquest amb el pas del temps. He triat el model SIR per dur a terme la propagació.

El nom del model prové de les inicials S (població susceptible), I (població infectada) i R (població recuperada). El model relaciona les variacions dels tres tipus d'individus de la població a través de la taxa d'infecció i el temps de recuperació. El model consta de tres estats: susceptible, infectat i recuperat i cada individu ha d'estar sempre dins d'un estat. No tots els individus passaran d'estat infectat a estat recuperat, també n'hi haurà alguns que poden no recuperar-se. Per això el model té una taxa de recuperació que significa el percentatge d'individus que es recuperaran de forma correcta.

A la figura (3) mostro un esquema bàsic del model de propagació SIR:

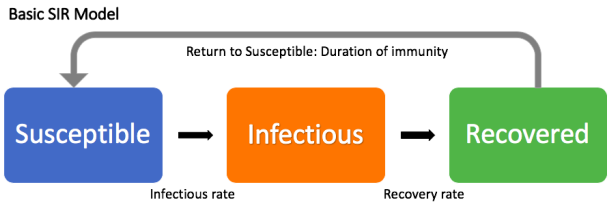

Fig 3. Model Bàsic SIR

La funció de propagació segons el model SIR rep tres paràmetres: el llindar de contagi, temps de recuperació i índex de recuperació. El llindar de contagi és el valor a partir del qual considerem que un individu passa de susceptible a infectat. Això serà degut al fet que el pes de les arestes dels nodes veïns iguala o supera el llindar establert. Després tenim el temps de recuperació, que són els dies que un individu necessita per recuperar-se i sortir de l'estat infectat. Per últim, l'índex de recuperació és el percentatge d'individus infectats que es recuperen de forma correcta. Si un individu no es recupera, eliminem el node del graf i les relacions amb els seus veïns.

Una vegada creat el graf assignem a cada aresta el pes igual a la fórmula de la figura (4):

```
for i in g.nodes():
    for j in g.neighbors(i):
        if nodes[i]['comunitat'] == nodes[j]['comunitat'] and nodes[i]['edat'] == nodes[j]['edat'] and nodes[i]['edat'] <= 20:
            g.edges[i,j]['weight'] = alpha + beta
        elif nodes[i]['edat'] == nodes[j]['edat']
            g.edges[i,j]['weight'] = beta
        else:
            g.edges[i,j]['weight'] = alpha
```

Fig 4. Assignació del pes a les arestes del graf

En cada etapa del model de difusió recalculem l'estat de cada node utilitzant la fórmula que es mostra a la figura (5):

```
for i in g.nodes():
    for j in g.neighbors(i):
        pesos[i] += g.edges[i, j]['weight']
```

Fig 5. Funció per recalcular l'estat del node

Un cop hem actualitzat el pes del node mirem si supera el llindar de contagi establert per saber si passa d'estat sa a infectat.

Perquè el model funcioni hem d'infectar prèviament algun node del graf perquè pugui infectar als seus veïns i així successivament per veure l'evolució del graf.

Per veure aquesta evolució, assignem a cada node del graf un color diferent segons el seu estat respecte a l'epidèmia. Hem triat tres colors per diferenciar tres tipus diferents d'estat: el groc (individu susceptible), vermell (individu infectat) i verd (individu recuperat).

Fins aquí tot el desenvolupament ha sigut utilitzant la llibreria networkX [2] de Python que ens ajuda i facilita poder treballar amb grafs. A la figura (6) mostro un graf sense aplicar-hi cap propagació (estat inicial). El graf conté 8 individus de dues comunitats diferents i dos membres de comunitats diferents tenen la mateixa edat.

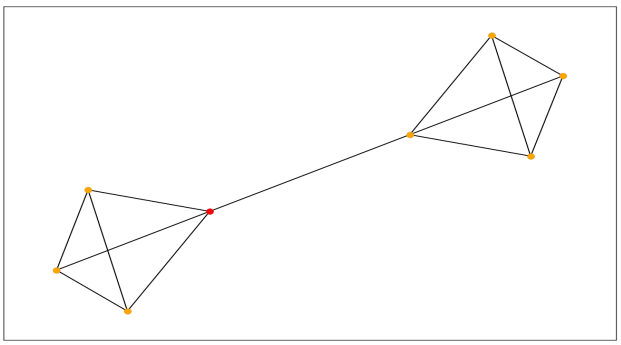

Fig 6. Estat inicial

4.3 Front end del sistema

A partir d'aquí, el desenvolupament es va centrar a fer una interfície d'usuari en la qual l'usuari pot seleccionar diferents paràmetres per fer la simulació.

En aquesta interfície es poden configurar cinc paràmetres: llindar de contagi, índex de contagi alpha, índex de contagi beta, temps de recuperació i índex de recuperació.

La interfície de l'aspecte que mostro a la figura (7):

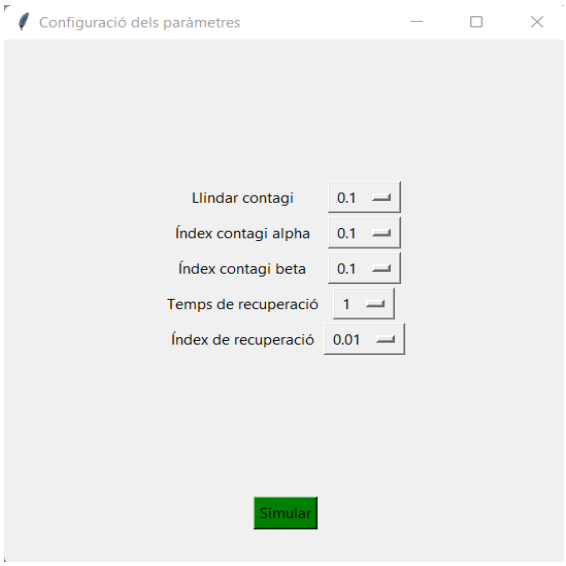

Fig 7. Interfície d'usuari.

En aquesta interfície l'usuari pot configurar els paràmetres i després clicar al botó verd per fer la simulació. Els paràmetres els agafo i els passo a les funcions corresponents per executar el model de propagació.

L' índex de contagi alpha i beta serveixen per inicialitzar el pes de les arestes de graf. Com ja hem dit abans alpha és el pes de les arestes si dos nodes comparteixen comunitat i la beta per si comparteixen edat. En el cas del llindar, temps de recuperació i l'índex de recuperació els passem a la funció de propagació que utilitza el model SIR.

Tant l'índex de contagi alpha, índex de contagi beta i llindar de contagi es poden configurar en els següents valors:

- 0,1 (Per defecte)
- 0,2
- 0,3
- 0,4
- 0,5

El temps de recuperació té els següents valors de configuració:

- 1 (Per defecte)
- 2
- 5
- 10
- 15

L'índex de recuperació té els següents valors de configuració:

- 0,01 (Per defecte)
- 0,02
- 0,03
- 0,04
- 0,05

Aquesta funcionalitat l'he desenvolupat utilitzant la llibreria tkinter [4] de Python que ens ha facilitat bastant el fet d'incorporar un menú al projecte.

Al clicar al botó de simular, el projecte està configurat perquè et mostri 8 imatges on cadascuna es correspon amb un dia diferent de la propagació. Després de l'última imatge es mostra com un resum de la simulació on es veuen les 8 imatges diferents en una mateixa imatge.
El nombre de dies de duració del model de propagació és flexible, però està configurat perquè s'executi 8 vegades.

Per últim, vaig afegir una funcionalitat de crear un vídeo amb els grafs resultants per veure amb més facilitat la simulació. Per fer això, el que vaig fer va ser guardar en una llista les diferents imatges del graf en les diferents etapes de propagació. Una vegada tenim les imatges en una array mitjançant la llibreria de Python openCV [5] les podem concatenar per poder visualitzar-les com si fos un vídeo. La funció d'openCV [5] per crear el vídeo permet configurar els fotogrames per segon i com vols que sigui l'amplada i l'alçada del vídeo entre altres coses.

5 Resultats

A continuació, explico els experiments que he fet i els resultats obtinguts en cadascun d'ells. Per a la realització del projecte he treballat amb una base de dades que la creo amb individus amb atributs aleatoris.

Per fer les diferents simulacions he creat un total de 3 bases de dades que cadascuna conté un nombre d'individus diferent.

La primera és una petita amb 8 individus i dues comunitats diferents només. Aquesta l'he utilitzat principalment per veure que el model de difusió funciona correctament i es poden veure més fàcilment els resultats que en grafs més grans. Hi ha 4 individus de cada comunitat i només un dels individus de la primera comunitat té la mateixa edat que un dels individus de la segona.

Aquesta petita base de dades la mostro a la figura (8):

```
id_individu,comunitat,edat,estat
0,1,20,S
1,2,20,S
2,2,10,S
3,2,15,S
4,1,40,S
5,1,80,S
6,2,50,S
7,1,60,S
```

Fig 8. Base de dades 2 comunitats

Els atributs que he utilitzat a l'hora de crear les relacions han sigut la comunitat i l'edat. He fet un conjunt de simulacions per observar el comportament de l'algorisme i aquests són els resultats.

5.1 Experiments

Experiment 1

La primera simulació després de crear el graf amb les diferents connexions hem introduït els següents paràmetres per a l'execució:

- Llindar de contagi: 0,2
- Índex de contagi (alpha): 0,1
- Índex de contagi (beta): 0,2
- Temps de recuperació: 1
- Índex de recuperació: 0,01

Infectem el node amb identificador igual a 0 per poder aplicar el model de difusió SIR.

Els resultats després d'iterar 8 vegades el model de difusió es mostren a la figura (9):

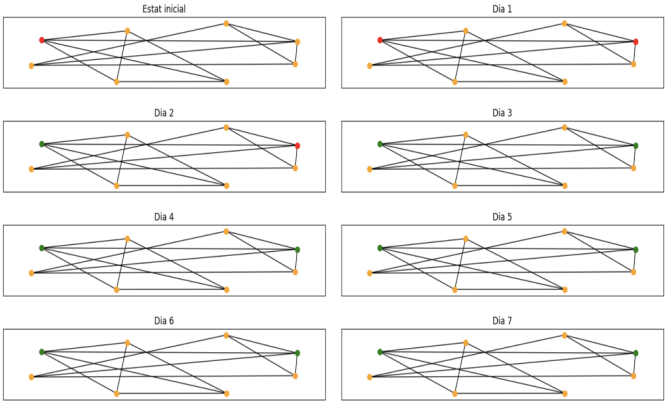

Fig 9. Visualització Experiment 1

Com podem observar els nodes que pertanyen a la mateixa comunitat que el node infectat no s'infecten, en canvi, el node de l'altra comunitat sí que s'infecta.

Això és degut, a què hem configurat el llindar de contagi a 0,2 i com l'índex de contagi (alpha) és igual a 0,1 no transmet el virus a cap individu de la mateixa comunitat.

En canvi, com l'índex de contagi (beta) és igual a 0,2 fa que si dos individus tenen la mateixa edat i aquesta és igual o superior a 20 estableixi aquell pes a l'aresta entre els dos nodes.

Com aquest índex beta és igual o superior al llindar establert, l'individu amb identificador igual 2 dos s'infecta el primer dia.

Pel fet que hem configurat el temps de recuperació a 1 els individus tarden a recuperar-se un dia del virus. Com podem observar l'individu amb identificador 0 el dia 2 ja està recuperat, en canvi, l'individu amb identificador 1 es recupera el dia 3, ja que es va infectar un dia més tard que el primer.

En aquesta simulació, els nodes infectats es recuperen correctament. Podria passar que els nodes amb identificador 1 i 2 no es recuperessin i s'eliminessin del graf. Ara bé, és poc probable, perquè l'índex de recuperació és del 0,01 i això significa que un 99% dels individus infectats es recuperaran satisfactòriament.

Experiment 2

He utilitzat la mateixa base de dades de la figura 5.
Els paràmetres que he introduït per a la simulació són els següents:

- Llindar de contagi: 0,2
- Índex de contagi (alpha): 0,2
- Índex de contagi (beta): 0,1
- Temps de recuperació: 2
- Índex de recuperació: 0,01

Infectem el node amb identificador igual a 0 per poder aplicar el model de difusió SIR.

Els resultats després d'iterar 8 vegades el model de difusió es mostren a la figura (10):

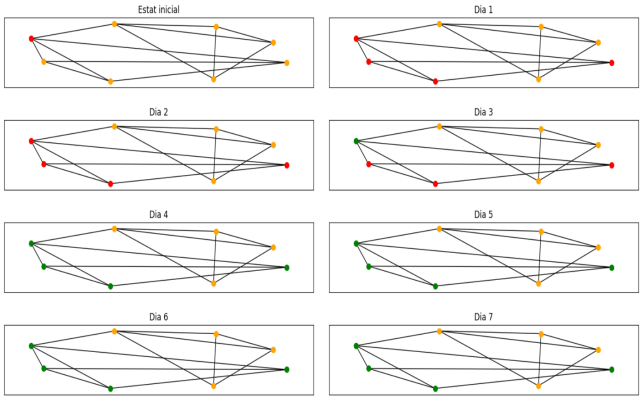

Fig 10. Visualització experiment 2

Com podem observar després de veure la simulació els nodes que s'infecten són els que pertanyen a la mateixa comunitat que el node amb identificador igual a 0. En canvi, els nodes de comunitat dos no s'infecten mai.

Això és degut, a què hem configurat el llindar de contagi a 0,2 i com l'índex de contagi (alpha) és igual a 0,2 els individus que pertanyen a la mateixa comunitat s'infecten. En canvi, l'índex de contagi (beta) és igual a 0,1 llavors els nodes que tenen la mateixa edat que són de comunitats diferents no s'infecten, ja que no supera el llindar establert.

Pel fet que hem configurat el temps de recuperació a 2, els individus tarden a recuperar-se dos dies del virus. Com podem veure l'individu amb identificador 0 el dia 3 ja està recuperat, en canvi, els individus amb identificador igual a 4, 5 i 7 es recuperen el dia 4, perquè es va infectar un dia més tard que el primer.

En aquesta simulació, els nodes infectats es recuperen correctament. Podria passar que els nodes que han estat infectats no es recuperessin i s'eliminessin del graf.

Experiment 3

Per aquest experiment he utilitzat una base de dades de 100 individus separats per 10 comunitats diferents. A la figura (11) es mostra el graf inicial amb les seves relacions.

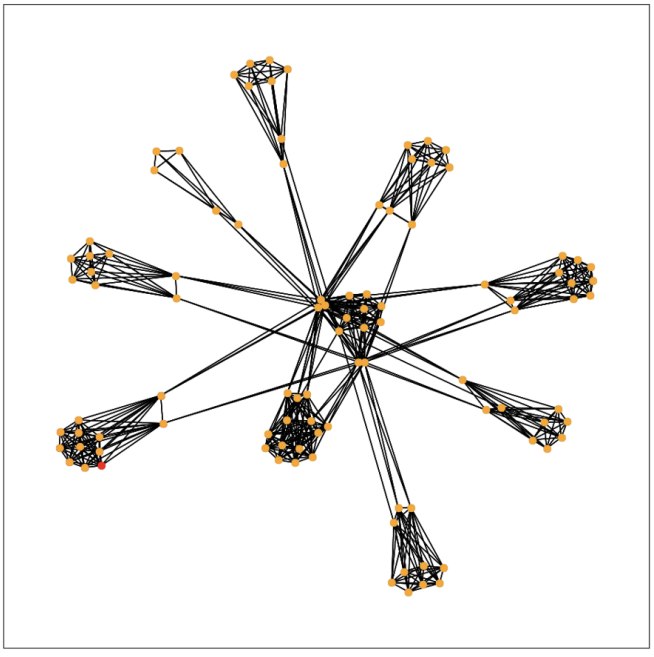

Fig 11. Graf inicial experiment 3

Els paràmetres que he introduït per a la simulació són els següents:

- Llindar de contagi: 0,3
- Índex de contagi (alpha): 0,3
- Índex de contagi (beta): 0,3
- Temps de recuperació: 2
- Índex de recuperació: 0,05

Infectem el node amb identificador igual a 0 per poder aplicar el model de difusió SIR.

Després d'iterar el model de difusió mostrem els resultats del dia 4 i dia 7 després de l'estat inicial a la figura (12) i figura (13) respectivament.

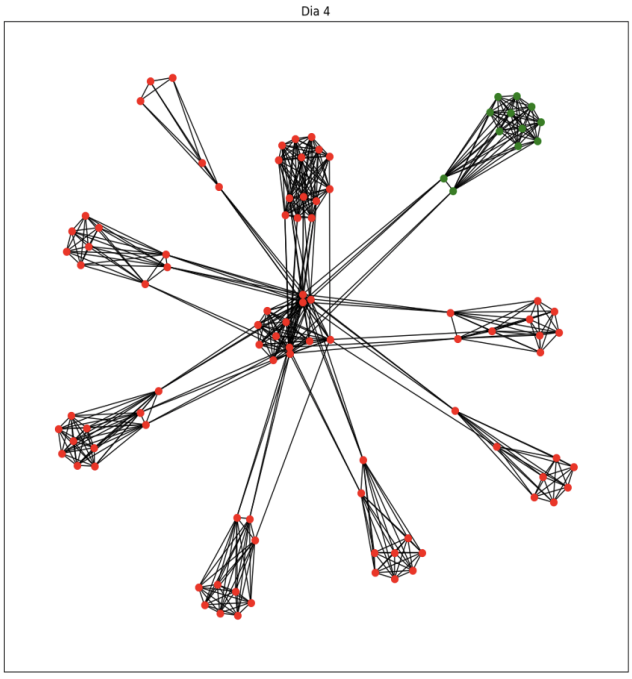

Fig 12. Dia 4 de l'experiment 3

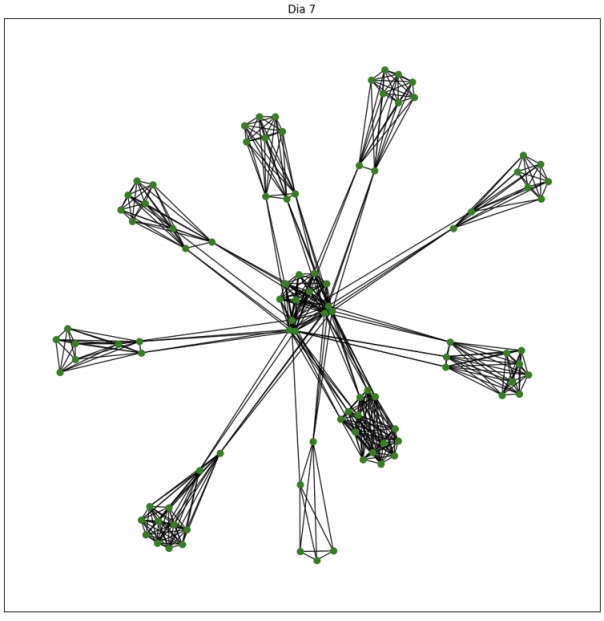

Fig 13. Dia 8 de l'experiment 3

Com podem observar després de la simulació, primer s'infecten els nodes que pertanyen a la mateixa comunitat que el node amb identificador igual a 0. Després els nodes amb edat inferior o igual a 20 infecten als de les altres comunitats i això fa que finalment s'infectin totes les comunitats.

Això és degut, a què hem configurat el llindar de contagi a 0,3 i com l'índex de contagi (alpha) és igual a 0,3 els individus que pertanyen a la mateixa comunitat s'infecten. Com l'índex de contagi (beta) també és igual o superior al llindar de contagi establert, els individus amb edat menor o igual a 15 d'altres comunitats també s'infecten i això fa que com es veu a la figura (9) el dia 4 totes les comunitats tenen tots els individus infectats menys la primera comunitat que els seus individus ja s'han recuperat del virus perquè estem al dia 4.

Com hem configurat el temps de recuperació a dos els individus tarden a recuperar-se 2 dies del virus. Observem que els individus que pertanyen a la mateixa comunitat que l'individu amb identificador igual a 0 ja estan recuperats al quart dia com es veu a la figura (9).

Com podem veure a la figura (10) no tots els nodes es recuperen de forma correcta. Si fem el recompte del total de nodes del graf, ens adonem compte que n'hi falten 3 nodes. Això és degut, a què hem configurat l'índex de recuperació a 0,05 i significa que el 5% dels individus no es recuperaran després de passar el virus. En aquest cas ha sigut 3 el nombre total de nodes eliminats del graf, però com el càlcul es fa de manera aleatòria aquest nombre va canviant per cada simulació que fem.

Experiment 4

Aquest últim experiment es tracta de visualitzar els grafs mitjançant l'eina Gephi[6]que ens permet una visualització més amigable del graf. Per fer això, hem hagut de guardar en graf en l'extensió que demana aquest programari per poder obrir el graf i visualitzar-lo. Hem utilitzat la base de dades de 100 individus per fer l'experiment.

Amb Gephi [6] podem configurar que ens acoloreixi els nodes que pertanyen a la mateixa comunitat de forma molt intuïtiva i fàcil. Hem d'anar l'apartat d'aparença i dins de nodes triar en partició l'atribut que nosaltres volem acolorir. El resultat es mostra a la figura (14):

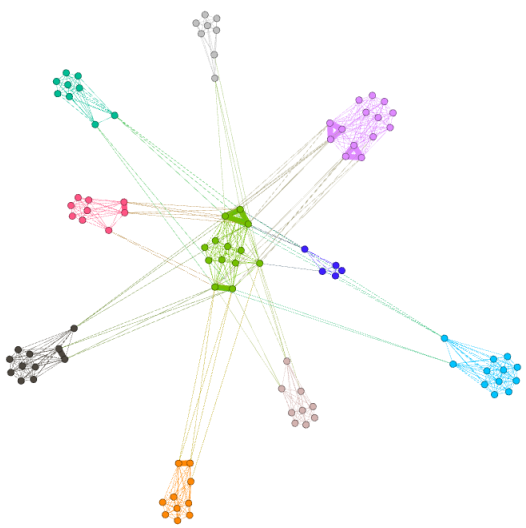

Fig 14. Gephi [6] acolorit per comunitats

També podem acolorir per un altre atribut com per exemple seria l'edat. El resultat es mostra a la figura (15):

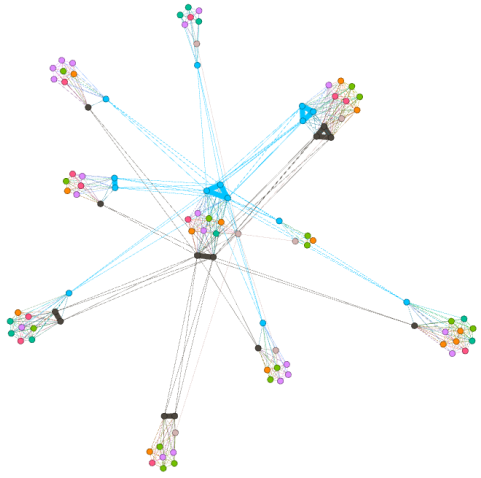

Fig 15. Gephi [6] acolorit per edats

La llegenda corresponent a la figura (15) es mostra a la figura (16):

Fig 16. Llegenda corresponent a la figura (15)

Com podem observar a la figura (15) Gephi [6] ens mostra la llegenda corresponent al graf i a part també ens indica el percentatge que representa cada edat sobre el total d'individus.

Conclusions

Al llarg d'aquest projecte he posat en pràctica diferents eines que he après durant la carrera, com per exemple, estructurar les dades correctament o utilitzar diferents llibreries de Python com numpy o openCV [5]. També he après a dissenyar grafs, crear relacions, assignar pesos als diferents nodes amb la llibreria networkX [2] que no l'havia fet servir amb anterioritat.

També he après a dissenyar el front-end amb Python cosa que no havia fet mai fent servir aquest llenguatge. De cara a fer el vídeo m'ha ajudat els coneixements previs de l'assignatura Visió per Computador, ja que a les pràctiques fèiem servir bastant la llibreria openCV [5].

Respecte als resultats obtinguts puc dir que he assolit bastant els objectius que m'havia proposat l'inici del treball. Com a treball futur una bona millora seria incorporar una nova opció als paràmetres que poguéssim configurar almenys un altre tipus de funció de difusió pel graf.

Bibliografia

[1] Verbel de León, Arturo. 2019.
Todo para trabajar con grafos en Python. http://micaminomaster.com.co/grafo-algoritmo/todo-trabajar-grafos-python/

[2] NetworkX. 2014-2021. Network analysis in Python. https://networkx.github.io/

[3] Pycharm. 2021. IDE per desenvolupar en Python. https://www.jetbrains.com/es-es/pycharm/

[4] tkinter. 2019. Llibreria Python per desenvolupar UI. https://docs.python.org/3/library/tkinter.html

[5]openCV. 2021. Llibreria Python per visió per computador. https://opencv.org/

[6] Gephi. 2009. The Open Graph Viz Platform. https://gephi.org

[7] Albert-Lázló Barabási. 2016.
Network Science.
http://networksciencebook.com