



Escola Tècnica Superior d'Enginyeria  
de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# **PROJECTE FINAL DE CARRERA**

## **RECONSTRUCIO DE XARXES A PARTIR DE LA COMMUNICABILITY CENTRALITY DELS NODES**

RECONSTRUCTION OF NETWORKS FROM THE  
COMMUNICABILITY CENTRALITY OF NODES

*Estudis: Enginyeria d' Electrònica*

*Autor: Oscar Raig Colon*

*Director: Francesc Comelles*

*Any: 2015*



## Índex general

Col·laboracions.....	5
Agraïments.....	6
Resum del Projecte.....	7
Resumen del Proyecto.....	8
Abstract.....	9
1.Introducció.....	10
Context del projecte.....	10
Objectius.....	12
Estructura de la memòria.....	14
2.Teoría de Grafs.....	15
Definicions i exemples.....	15
Exemples d'utilització de grafs.....	19
Xarxes en el món real.....	23
Propietats principals de les xarxes.....	26
Models de grafs .....	30
3.Els Indicadors o paràmetres rellevants del graf.....	38
Betweenness centrality .....	39
Communicability Centrality.....	40
Communicability Betweenness Centrality.....	41
4.Els Algorismes.....	43
Simulated Annealing (SA).....	44
Threshold Acceptance (TA).....	46
Paràmetres de l'algorisme TA.....	48
6.La implementació.....	50

Els principis bàsics.....	50
Representació matricial de la classe Graf.....	56
Programes utilitzats per fer el projecte.....	56
Posibles millores en la simulació i obtenció de resultats.....	57
7.Els Resultats.....	58
Communicability Centrality amb Threshold Acceptance.....	60
Communicability Centrality amb Simulated Annealing.....	62
Communicability Betweenness Centrality amb Threshold Acceptance.....	63
Betweenness Centrality amb Simulated Annealing.....	64
8.Conclusions .....	65
9.Apèndix.....	67
10.Referències.....	68
Bibliografia.....	68

# Col·laboracions

Matemàtica Aplicada IV



# Agraïments

Agreixo a Francesc Comelles la oportunitat que m'ha brindat de fer el projecte final de carrera.

El Francesc m'ha obert una porta a tot un seguit de coneixements i tecnologia que ni tant sols sabia que existien.

M'ha omplert de curiositat, Betweenness centrality, Communicability Centrality, Simulated Annealing, Threshold Acceptance, algorismes deterministes, estocàstics, simulacions de hores, dies i mesos.

A més a tirat del "carro" quan jo estava cansat, m'ha motivat i ha tingut molt de "push". M'ha estirat de les orelles, ha estat exigent quan calia i comprensiu quan tocava.

Sincerament,

Moltes Gràcies Francesc.

# Resum del Projecte

L'objectiu del projecte és intentar esbrinar quins són els paràmetres rellevants d'una xarxa. Hem de veure si som capaços de “comprimir” la xarxa d'alguna manera, ja que cada vegada les xarxes són més grans i més volàtils. Per esbrinar-ho, donada una xarxa original, s'extreuen unes propietats o paràmetres que volem saber si descriuen la xarxa, s'executa uns algorismes que, intenten reconstruir el graf original. Si obtenim un graf semblant a l'original sabrem que aquest paràmetre és rellevant per la xarxa.

# Resumen del Proyecto

El objetivo del proyecto es intentar averiguar cuales son los parámetros relevantes de una red. Hemos de ver si somos capaces de “comprimir” la red de alguna manera, ya que cada vez las redes son mas grandes y mas volátiles. Dada una red original, se extraen las propiedades o parametros . Después se ejecutan unos algoritmos que reconstruyen el grafo original. Si obtenemos un grafo parecido a el original sabremos que este parámetro es importante para la red.



# Abstract

The project objective is to try to find out what are the relevant parameters of a network. We must see if we can "compress" the network somehow, because networks are increasingly larger and more volatile. Given an original network, properties or parameters are extracted. After algorithms to reconstruct the original graph are executed. If we get a graph similar to the original we will know that this parameter is important for the network.

# 1. Introducció

## Context del projecte

### Importància de la Teoria de grafs en el contexte tecnològic actual

La teoria de grafs permet modelar de forma senzilla un sistema en el qual existeixi una relació binària entre certs objectes, és per això que el seu àmbit d'aplicació és molt general i cobreix àrees que van des de la mateixa matemàtica, fins l'enginyeria electrònica, les telecomunicacions, la informàtica i la investigació.

Actualment, amb la arribada de les **xarxes socials** (facebook, twitter), la possibilitat de recopilar grans quantitats de dades personals del usuaris d'internet i tractar-les (Big Data), la teoria de grafs esdevé un de les matèries per lligar tota aquesta informació i utilitzar-la com a eina de marketing. El **cloud computing** està sent importantíssim per l'avenç de totes aquestes tecnologies, ja que augmenta les possibilitats de càlcul.

Les aplicacions per mòbils estan substituïnt les aplicacions Desktop a les llars, la gent compra, consulta més sobre els dispositius mòbils originant unes xarxes enormes amb estructures que varien minut a minut.

Ja fa temps que existeixen "graph databases" com Neo4j per donar suport a totes aquestes utilitats, grans companyies com Google treballen amb teoria de grafs, Apache Giraph per processar grafs sobre Big Data, implementacions de map reduces com Apache Hadoop... Són eines, empreses i tecnologies que avui en dia estan apostant per la Teoria de Grafs.

## Justificació de la reconstrucció de xarxes

La proliferació de les xarxes sense fils, el nombre de xarxes locals dinàmiques connectades a la xarxa de xarxes no para de créixer. Aquestes xarxes es poden modelar matemàticament mitjançant gràfics, associant els nodes i enllaços de la xarxa als vèrtexs i arestes del gràfic.

Les xarxes socials són un altre exemple de **xarxa dinàmica**, on les amistats apareixen i desapareixen, les necessitats i els interessos també es poden modelar com un gràfic i aquests últims tenen una volatilitat encara més alta.

Quan una xarxa és estàtica, per conèixer el seu estat “només” cal conèixer l'estat de cadascun dels seus elements (nodes i enllaços). Si bé és cert que recollir tota aquesta informació en alguns casos pot no ser trivial, en el cas de les xarxes dinàmiques, on els nodes i els enllaços apareixen i desapareixen contínuament, aquesta feina pot ser realment complicada, per no dir inviable. Una de les eines que podria ajudar a fer aquest estudi de la xarxa seria un mecanisme que permetés, una vegada obtingut el gràfic que representa la xarxa, **emmagatzemar aquesta informació en un format molt compacte (per poder-lo transmetre ràpidament) i a l'hora fa fàcilment descompactable (per poder recuperar fàcilment tota aquesta informació)**

La reconstrucció dels gràfics és un dels temes en els que s'està treballant actualment i on es poden trobar aproximacions molt diferents. La necessitat de reconstruir gràfics té diverses justificacions. D'entre totes, aquest estudi pretén aportar una **alternativa per l'emmagatzematge de gràfics**.

Per tal d'emmagatzemar aquesta informació hem de saber **quins són els paràmetres rellevants de la xarxa**. Si a partir d'aquests paràmetres som capaços de reconstruir la xarxa original, vol dir que aquest paràmetre és més important que d'altres.

**Si som capaços de redimensionar una imatge vectorial, seriem capaços de fer-ho amb una xarxa?**

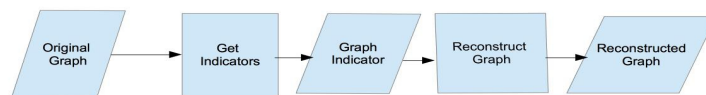
## Objectius

L'objectiu principal d'aquest projecte és esbrinar com de rellevants són els paràmetres d'una xarxa i si ens permeten reconstruir-la a partir d'aquests paràmetres.

Per esbrinar-ho s'ha evaluat l'ús de dos algorismes el *simulated annealing* (SA) i *threshold acceptance* (TA), per reconstruir un graf a partir del *betweenness centrality* (BC), *communicability betweenness centrality* (CBC) o la *communicability centrality* (CC), que són uns paràmetres o indicadors del graf.

El treball posa especial focus en l'algorisme *threshold acceptance* i amb els paràmetres Communicability Betweenness Centrality i Communicability Centrality.

El algorisme *simulated annealing* i el paràmetre Betweenness centrality s'evalua amb menys rigor.



El contexte del projecte.

1. A partir d'un graf, extraïem indicadors, com la Betweenness centrality, la Communicability Betweenness Centrality o la Communicability Centrality.

2. A partir d'aquests indicadors, i el número de vèrtexs, intentem reconstruir el graf original.

3. Si som capaços de reconstruir el graf original és que el paràmetre és rellevant per la xarxa. També ens serveix per comparar quin paràmetre és més rellevant.

Per tant, que hem fet servir una mesura o indicador de cadascun dels vèrtexs del graf (o *vertex* *BC*, *CC*, *CBC*) per calcular la funció de cost que utilitzen els algorismes. Aquesta mesura, com veurem més endavant, dona molta informació del graf, ja que a més de l'ordre el graf, permet classificar els vèrtexs segons el nombre de camins curts que hi passen.

Hem escollit exemples representatius dels principals tipus de grafs que es fan servir per modelar diferents topologies de xarxes i hem realitzat la seva reconstrucció a partir de la llista de valors dels indicadors (*BC*, *CC*, *CBC*).

En concret, hem fet servir:

- un graf aleatori (random),
- un graf petit-món (small-world),
- un graf amb invariància d'escala (scale-free),
- un graf circulant
- i un graf amb agrupament (clustering),

Tots ells amb el mateix ordre  $n=40$ , i per cadascun d'ells s'han realitzat entre 100 - 200 reconstruccions per disposar d'un nombre suficientment gran de grafs reconstruïts i fer una anàlisi estadística dels resultats.

## Estructura de la memòria

Hi han 3 blocs principals:

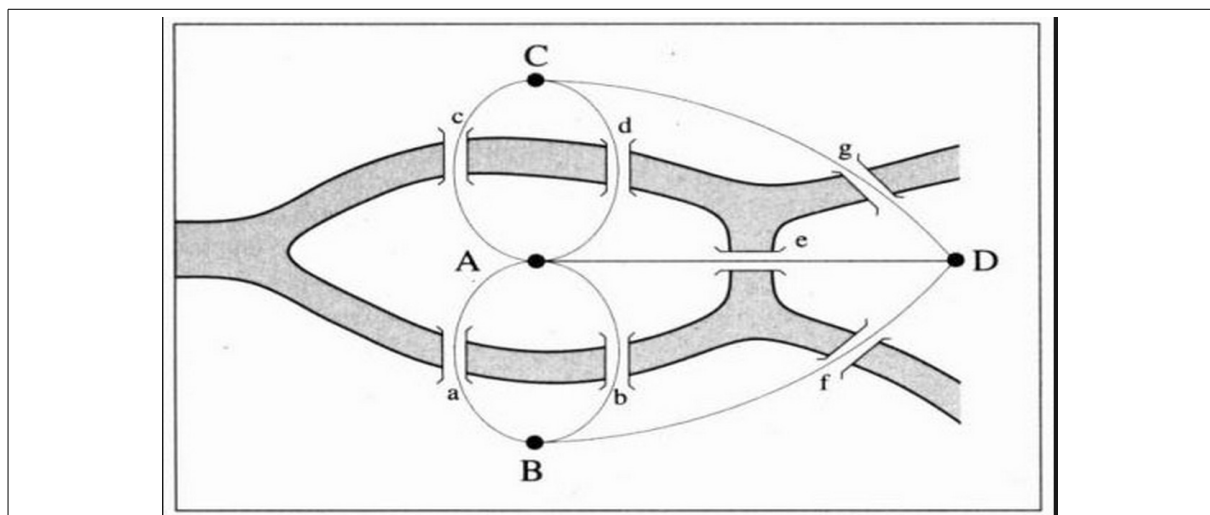
- Conceptes teòrics
  - Teoria de Grafs
  - Indicadors
  - Algorismes
  - Comparació dels grafs
- Implementació
  - Algorisme de reconstrucció
  - Generació de les 100-200 simulacions
  - Recollida de Resultats
- Resultats i Conclusions.

## 2. Teoria de Grafs

### Definicions i exemples

#### Ressenya Històrica : Euler i els ponts de Königsberg

L'origen de la teoria de grafs s'associa amb la resolució que va donar Leonard Euler al anomenat problema dels ponts de Königsberg (1736). En aquesta ciutat hi ha una illa en mig del riu que travessa la ciutat. Aquesta illa està connectada per 7 ponts, el problema intenta passar un sol cop per cada un dels set ponts. La resolució que va donar Euler d'aquest problema no solament resolva a aquesta qüestió, sino que va introduir la noció de graf i va resoldre al mateix temps un problema de caràcter més general.



La illa de la ciutat Königsberg, actualment Kaliningrad. Amb vèrtexs, A, B, C i D i amb arestes a,b,c,d,e,f,g.

#### Definició de grafs dirigits, ordre i mida

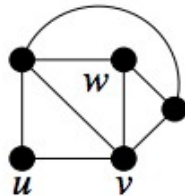
Un **graf no dirigit**  $G=(V,E)$  és una estructura combinatòria constituïda per un conjunt  $V=V(G)$  d'elements anomenats vèrtexs i un conjunt  $E=E(G)$  de parells no ordenats de vèrtexs distints anomenats arestes. Si la aresta  $e=\{u,v\} = uv$  relaciona els vèrtexs  $u$  i  $v$ , es diu que  $u$  i  $v$  són vèrtexs adjacents, de un altre mode, els vèrtexs

es diuen independents.

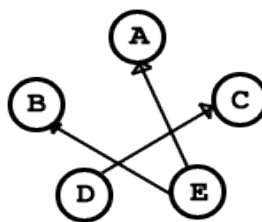
El **grafs dirigits** les arestes són parells ordenats. En aquest document sempre parlarem de grafs dirigits.

El nombre de vèrtex de ,  $|V(G)|$  , és l'**ordre** del graf i el nombre d'arestes  $|E(G)|$  és el **tamany del graf o la mida del graf**.

Una manera de representar qualsevol relació és llistant els seus elements com a parells ordenats. En aquests cas és més convenient utilitzar una representació com la figura de sota. En aquest graf podem veure que podem anar del vèrtex “u” a “w” a través de v. La visualització gràfica és més fàcil de interpretar que els parells ordenats de la relació.



**Figura 1.1: Graf no dirigit amb ordre 5 i tamany 8**



**Figura 1.2: Graf dirigit amb ordre 5 i tamany 3**



## Grau d'un vèrtex, camins

El node és comunica amb altres nodes mitjançant les arestes. El nombre d'arestes que té un node és el **grau del node**.

Un **vèrtex aïllat** és un vèrtex amb grau 0.

Un **vèrtex full** o terminal és un vèrtex amb grau 1.

Dos vèrtexs d'un graf es poden comunicar mitjançant una sèrie d'arestes, que s'anomena **camí**.

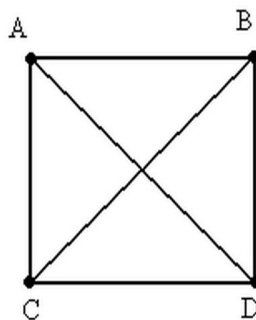
El nombre d'arestes d'un camí s'anomena **longitud** del camí.

La **distància** entre dos vèrtexs és el nombre d'arestes que conté el camí més curt que els enllaça.

El **diàmetre** d'un graf és la màxima distància existent entre totes les parelles de vèrtexs que el formen.

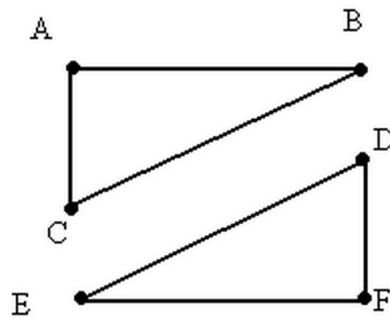
Quan el camí entre dos vèrtexs només passa una vegada per qualsevol d'ells s'anomena **camí simple**.

Un graf no dirigit s'anomena **graf connex** si existeix un camí entre dos vèrtexs distints.



Graf connex o conectat.

Un graf que no sigui connex s'anomena **graf no connex**.



Graf no connex o desconnectat.

Si existeixen dues o més arestes que uneixen els mateixos vèrtexs parlem de **branques paral·les**. un graf que contingui branques paral·leles s'anomena **multigraf**.

Un **llaç** és una aresta que comença i acaba en el mateix vèrtex.

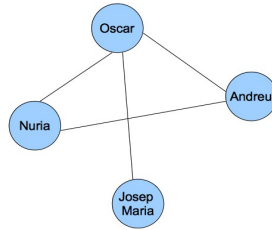
## Clúster i clustering

Un **cluster** és un conjunt de vèrtexs entre els que existeixen moltes connexions. El **clustering** n'es un paràmetre relacionat, que mesura la connectivitat local d'un graf. El **clustering d'un vèrtex** es defineix com la fracció de branques que uneixen les veïns d'aquest vèrtex entre ells, entre la quantitat total possible de branques. El clustering d'un graf és la mitjana dels clusterings dels seus vèrtexs, Si un vèrtex està aïllat o només té un veí, per conveni la seva aportació al clustering global és 1.

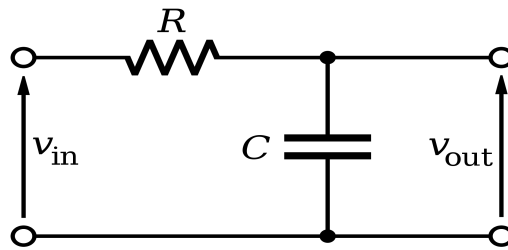
El **grau**, el **clustering**, el **diàmetre** i la **distància** són paràmetres que en els resultats finals ens ajudaran a comparar el graf original amb el reconstruït. Són paràmetres que defineixen la estructura del graf.

## Exemples d'utilització de grafs

Al final els grafs representen relacions entre elements. Avui en dia podríem parlar de facebook com a exemple de graf. Jo tinc 3 amistats. La Núria, l'Andreu i Josep Maria. La Núria també és amiga de l'Andreu.

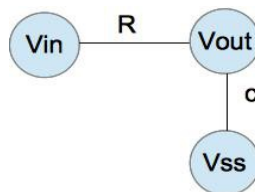


Xarxa social representada amb un graf



Filtre passa baix

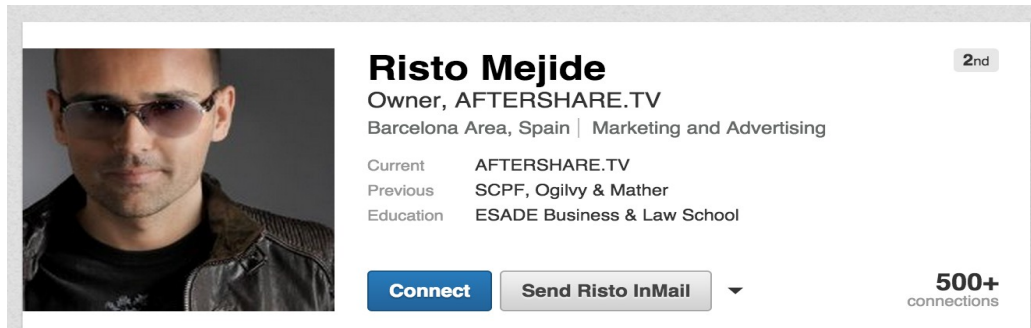
En un circuit elèctric clàssic passa baix, podríem definir els següent nodes:



El node d'entrada es connecta al node de sortida mitjançant una aresta R. El node de Sortida és comunica amb Vss mitjançant un condensador.



Dels “Seven Bridges of Königsberg” als carrers guarnits de les festes de Gràcia, podriem fer un algorisme que passi per tots els carrers guarnits una sola vegada?



**Risto Mejide**  
Owner, AFTERSHARE.TV  
Barcelona Area, Spain | Marketing and Advertising

Current: AFTERSHARE.TV  
Previous: SCPF, Ogilvy & Mather  
Education: ESADE Business & Law School

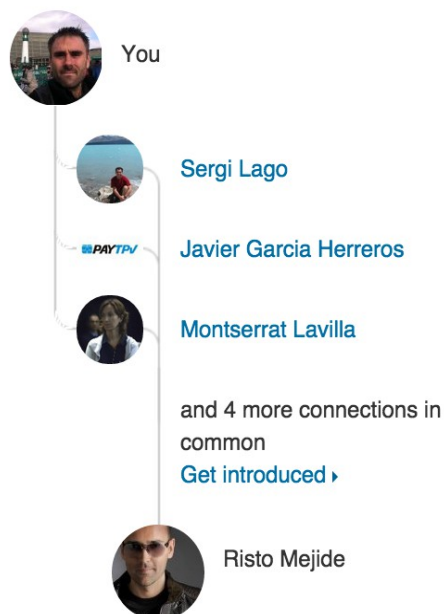
[Connect](#) [Send Risto InMail](#) ▼

**500+**  
connections

2nd

Risto Mejide i l'Oscar Raig els uneix un camí de longitud 2.

### How You're Connected



Els Camins de Longitud 2 que uneixen Oscar Raig i Risto Mejide són 7.

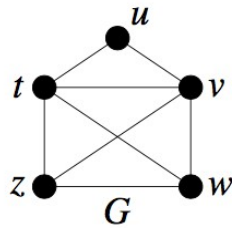
## Matriu d'adjacència d'un graf

Hem parlat de dues maneres de representar un graf, mitjançant una llista de parells ordenats i mitjançant un dibuix, de manera gràfica.

Una tercera manera de visualitzar un graf es mitjançant la matriu d'adjacència.

La matriu d'adjacència d'un graf de  $N$  vèrtexs, és una matriu quadrada  $N \times N$  on:

- L'element  $A_{ij} = 1$  si el vèrtex  $i$  té una aresta amb el vèrtex  $j$ .
- $A_{ij}$  és 0 si no hi ha cap aresta que els connecti.



Graf no dirigit

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Matriu d'adjacència pel graf de la figura d'adalt.

$$A^2 = \begin{pmatrix} 2 & 1 & 2 & 2 & 1 \\ 1 & 4 & 2 & 2 & 3 \\ 2 & 2 & 3 & 2 & 2 \\ 2 & 2 & 2 & 3 & 2 \\ 1 & 3 & 2 & 2 & 4 \end{pmatrix}$$

Càlcul de el nombre de recorreguts (o camins, tots els simples i els que no) de distància 2 entre vèrtex del graf.

## Xarxes en el món real

Durant molt de temps, els estudis que s'havien realitzat en xarxes complexes en basaven en models senzills (grafs aleatoris tipus Erdős-Rényi) que per qüestions de complexitat en els càlculs, i degut a la dificultat de poder fer estudis més exhaustius, no eren molt properes a la realitat. S'ha pogut comprovar que moltes de les xarxes que representen sistemes complexos no tenen l'estructura aleatòria que s'havia suposat fins aleshores.

### Tipus de xarxes científiques

En aquest apartat donarem una visió general de la estructura d'algunes xarxes complexes existents al món real, veurem quines són les propietats més importants. Inspirat per l'article de [Watts and Strogatz] ha hagut un estudi de les xarxes de diferents branques de la ciència, amb èmfasi en les propietats que són comuns a moltes d'elles.

Entre els tipus de xarxes que podem trobar tenim quatre grans grups: xarxes socials, d'informació, tecnològiques i biològiques [Newman]

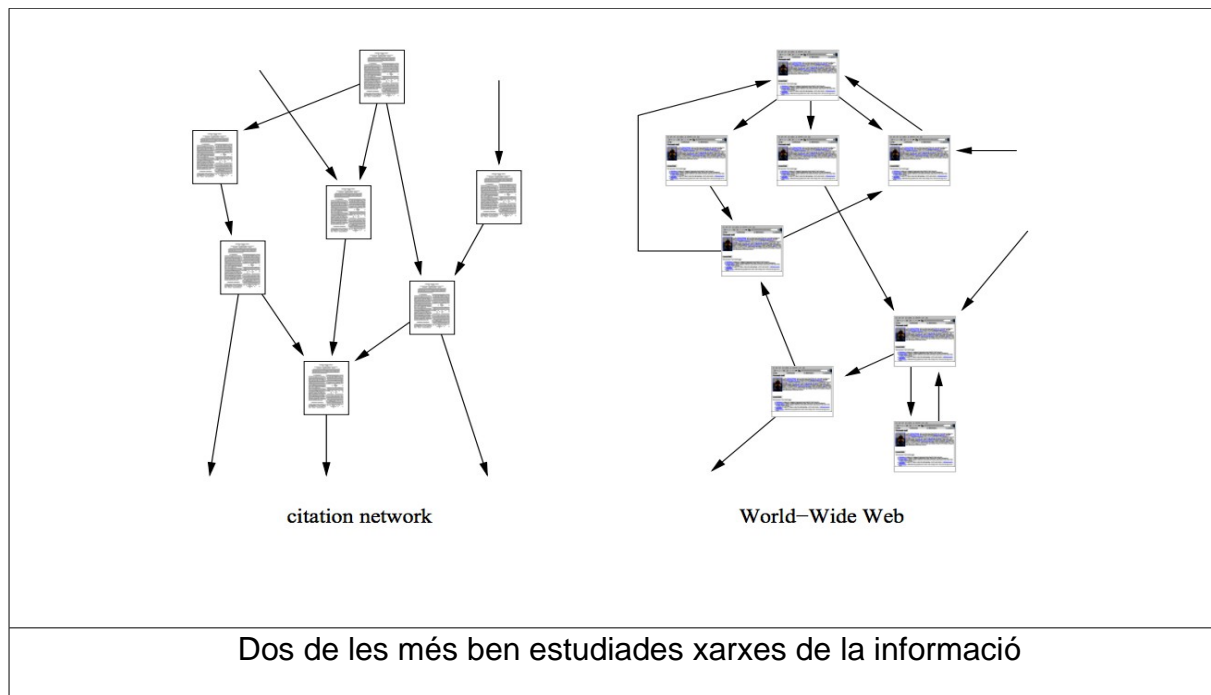
#### Xarxes Socials

Es considera una xarxa social com un conjunt de persones o grups de persones que segueixen un patró de contactes o interaccions entre ells. Els patrons d'amistat entre individus, relacions de negocis entre companyies i casaments entre famílies, són totes elles exemples de xarxes que han estat estudiades en el passat.

Un important conjunt d'experiments són els famosos experiments "petit-món" de Milgram. El experiment va provar la distribució de les longituds de camins en una xarxa demanant als participants que passessin una carta d'una persona a un altre esperant si arribava al seu destinatari. Moltes de les cartes del experiment es van perdre però una quarta part van arribar al destinatari passant per les mans de només sis persones. Aquest experiment va donar lloc al concepte popular dels "sis graus de separació" encara que la frase no apareix en el treball de Milgram, va ser anomenat decades després per Guare.

#### Xarxes d'Informació

També anomenades xarxes de coneixement ("knowledge networks"). Dins de les xarxes d'informació es poden destacar dos exemples:



El primer seria la relació de cites que hi ha entre publicacions acadèmiques, a partir de les publicacions i les referències entre elles. Cal comentar però, que la xarxa resultant és dirigida (document A fa referència al document B) i sense cicles, donat que no es pot fer referència a un article que encara no s'ha escrit.

Un altre exemple és la xarxa www considerant el conjunt de pàgines web i els seus hiperlinks com a vèrtexs i arestes respectivament. En aquest cas la xarxa resultant és també dirigida, amb la diferència que alguns enllaços són bidireccionals i es poden donar cercles. No s'ha de confondre www amb la xarxa d'Internet, que és una xarxa física d'ordinadors enllaçats entre ells per fibra òptica i altre tipus d'enllaç.

### Xarxes tecnològiques

Aquest tipus de xarxes han estat, almenys en principi, dissenyades per l'home per a distribuir recursos com la electricitat o la informació. Hi ha diversos exemples: les xarxes elèctriques, les xarxes de rutes aèries, ferroviàries, xarxa telefònica, correus... fins i tot els circuits electrònics són una forma de xarxa de distribució tecnològica. Un dels casos més interessants és Internet, la red física de connexions entre ordinadors. La xarxa és gran i el nombre d'elements connectats a la xarxa canvia constantment.

Per estudiar la xarxa primer cal tenir en compte una sèrie d'aspectes: La



infraestructura física que permet aquestes connexions no és fàcil d'estudiar ja que el manteniment en depèn de diferents organitzacions. La forma de fer-ho és mitjançant programes que realitzen traces entre dos punts (tracerout). Les dades dels nodes intermedis per els quals passa un paquet de dades és va emmagatzemant fins que es determina l'estructura global de la xarxa. Ara bé, com a conseqüència d'aquesta metodologia sempre hi ha un conjunt de vèrtexs (així com un conjunt de enllaços) que mai són mostrejats, de forma que aquesta aproximació pot arribar a ser prou bona, però no perfecta.

### **Xarxes biològiques**

Hi ha nombrosos sistemes biològics que també formen xarxes complexes.

Una tipus de xarxa biològica és la xarxa reguladora genètica.

Les cadenes alimentícies, en el qual els vèrtexs representen espècies en el ecosistema i una aresta dirigida entre A i B indica que A s'alimenta de B.

Xarxes neuronals són també un altre tipus de xarxa biològica d'importància considerable.

## Propietats principals de les xarxes

Descriurem les principals propietats de les xarxes:

- Distribució potencial de graus (internet traceroute)
- Distribució de les distàncies (petit-món, la carta que passa de mà a mà)
- Correlació entre graus
- Adjunció preferent
- Robustesa i vulnerabilitat

### Distribució potencial dels graus

Durant més de 40 anys, la comunitat científica havia assumit que les xarxes complexes eren aleatòries i seguien el model proposat per Erdős i Rényi l'any 1959.

Partint d'aquestes hipòtesis, a partir d'una distribució aleatòria dels nodes, el sistema resultant hauria de ser molt semblant, en quant a nombre de enllaços per node. D'aquesta manera, la majoria de nodes tindrien aproximadament el mateix nombre d'enllaços i seguirien una distribució de Poisson ben determinada, on trobar nodes de grau molt per sota o per sobre de la mitja seria gairebé impossible.

A l'any 1999, però, Réka Albert, Hawoong Jeong i Albert-László Barabási van estudiar la topologia de la World Wide Web amb l'objectiu de determinar les propietats globals de la xarxa. En un principi, les seves conjetures els van dur a pensar que trobarien una xarxa aleatòria, degut al gran nombre de pàgines web i al gran nombre població amb interessos diferents però els resultats però no van ser els previstos.

El mecanisme que s'encarregava de fer el recompte saltava des d'una pàgina web a un altre i recollia tots els enllaços que trobava. Tot i que únicament van poder explorar una part molt petita de la xarxa, va ser suficient per veure que la WWW es sostenia a partir de molt pocs nodes amb un nombre molt elevat d'enllaços. Més del 80% de les pàgines tenien menys de 4 enllaços, i menys d'un 0.01% de tots els nodes tenien més de 1000.

Fent un recompte de quantes pàgines web tenien exactament  $k$  enllaços (links), es va demostrar que l'histograma resultant seguia una distribució de graus

potencial, és a dir, la probabilitat de que un node fos connectat a  $k$  nodes era proporcional a  $1/k_{\text{expN}}$ . El valor de  $n$  per enllaços de entrada era aproximadament 2. En definitiva, era quatre vegades més probable que qualsevol node tingués la meitat del nombre d'enllaços d'entrada que un altre. El resultat era evident, es tractava d'un altre tipus de distribució totalment diferent a la aleatòria, era una xarxa lliure d'escala, i d'aquí el nom graf amb invariància d'escala

Més formalment, el terme invariància d'escala es refereix per totes les funcions  $f(x)$  que resten inalterables, tot i aplicar-les-hi factors multiplicatius o modificadors d'escala.

### **Distribució de les distàncies**

El fet que la distribució de graus en una xarxa complexa sigui amb invariància d'escala no implica que la distribució de la distància entre nodes també segueixi aquesta regla. Aquest efecte es conegut amb el nom de efecte petit-món on certs enllaços entre nodes poden connectar dues parts molt distants en el graf..

L'experiment que hem comentat abans sobre les cartes expliquen el concepte. En el nostre estudi un dels paràmetres que observem en les reconstruccions és la mitja de les distàncies i els diàmetres.

Sovint trobem xarxes que conviuen els dos models el de invariància d'escala i el de "petit-món".

### **Correlació entre graus**

Una altra característica important de les xarxes amb invariància d'escala es veure la relació que hi ha entre nodes del mateix grau.

El clustering del vertex en funció del seu grau. En aquest cas, l'objectiu es veure fins a quin punt valors clustering elevat implica un grau elevat en els nodes veïns.

### **Adjunció Preferent**

Una de les sorpreses més importants que es van donar en l'estudi de les diferents xarxes complexes va ser la existència de nodes altament connectats -també anomenats hubs. Per explicar el motiu d'existència d'aquests hubs es proposava com a exemple el mateix que van fer servir Barabási per a les seves proves.

Des del seu llançament al 1990 amb una única pàgina, la WWW ha experimentat

un enorme creixement. El model de creixement però, s'ha mantingut al llarg dels anys, i el que és més important, la seva distribució potencial no ha canviat.

No tots els nodes són iguals, quan una nova pàgina es crea l'autor pot decidir enllaçar-la allà on vulgui. Si tenim en compte que la majoria de la població coneix només una petita part, la tendència natural serà connectar la web al lloc on sigui més accessible i on tothom la pugui veure. Amb aquest mecanisme, els nous nodes, tendeixen a enllaçar-se als millors connectats, fet que afavoreix encara més aquests hubs, dotant-los de més connexions i fent-los més "atractius" per a nous nodes. Aquesta preferència a l'hora d'escollir el node d'enllaç es coneix amb el nom d'adjunció preferent. Si tenim en compte que es tracta d'un model de creixement, aquesta realimentació constant afavoreix que els nodes més antics tinguin més probabilitat de esdevenir hubs en el futur, tot i que com veurem més endavant no té perquè ser així.

Aquest grau tan alt de creixement, unit a l'adjunció preferent dona una bona raó per entendre l'existència d'un nombre tan elevat de xarxes amb invariàcia d'escala en qualsevol tipus d'entorn natural o artificial.

Investigacions en els diferents tipus de d'adjunció preferent expliquen que el mecanisme de generació de hubs tendeix a ser lineal. Un nou node es connectarà amb probabilitat doble, a un node antic que tingui almenys el doble d'enllaços que el seu veí. D'altra banda si es força a que la probabilitat sigui 4 vegades més gran, el resultat serà un únic hub que compartirà totes les connexions.

### **Resilència i vulnerabilitat**

Aquest tipus de xarxes són a la vegada molt robustes i molt vulnerables, tot depèn del tipus d'atac al que siguin sotmeses. Barabási i Jeong van realitzar proves sobre una xarxa amb invariàcia d'escala amb dos tipus d'atacs. En primer lloc van dirigir-se de forma aleatòria i uniforme als nodes de la xarxa, mentre que en una altra prova van atacar directament als hubs.

En xarxes aleatòries (Poisson) donat que gran part de nodes tenen el mateix nombre d'enllaços, els dos tipus d'atacs els afectaren de la mateixa manera. A les poques iteracions la connectivitat entre els elements restants del graf ja estava molt malmesa i la "caiguda" de la xarxa era inevitable.

Les xarxes amb invariàcia d'escala tenen dos tipus de comportament enfront els atacs. En el cas de ser atacades de forma aleatòria mostren gran robustesa, ja que "per probabilitat" la majoria de nodes tenen molt pocs enllaços (o veïns), i tot i eliminar molts nodes, la seva connectivitat es veu afectada només de forma local.

D'altra banda, si els atacs van dirigits als hubs les conseqüències solen ser desastroses. Únicament eliminant molt pocs nodes la connectivitat de la xarxa passa a un estat crític. Segons les proves realitzades per Barabási, únicament eliminant el 5% dels hubs, la distància per creuar la xarxa es duplicava. A partir d'aquí es va demostrar que eliminant entre el 5 i el 15% dels nodes millor connectats (comencant pel més gran) n'hi hauria prou per fer caure qualsevol xarxa d'aquest tipus.

## Models de grafs

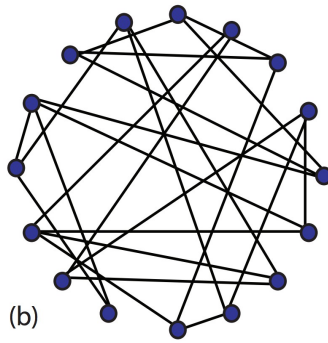
Tal com s'ha dit anteriorment, els grafs es fan servir per modelar matemàticament diferents relacions binàries entre objectes. Per permetre'n adaptar i reproduir diferents situacions de grafs, s'han definit models de grafs a partir de paraïmetres com la distribució dels graus o el mecanisme per triar les associacions entre vèrtexs. A continuació veurem la descripció d'uns quants exemples de grafs que es fan servir per modelar xarxes reals per a les que es pot aplicar la tècnica de reconstrucció descrita en aquesta memòria.

### Graf Aleatori

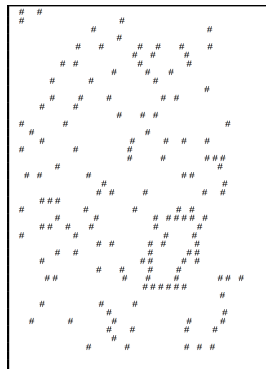
En els grafs aleatoris, com els que s'obtenen seguint el model Erdős-Rényi (dos matemàtics húngars) que hem fet servir per generar el graf aleatori del nostre estudi, per a cada parella de vèrtexs hi ha una aresta que els uneix amb una probabilitat  $p$  fixada i independent de les altres parelles. Donat que hi ha connexions amb la mateixa probabilitat amb qualsevol dels nodes, les distàncies acostumen a ser petites (Average distance 2.89, dels més baixos dels 5 grafs). L'aleatorietat fa que dos vèrtexs adjacents difícilment comparteixen vèrtexs, per la qual cosa el clustering acostuma a ser baix (Clustering = 0.2, el més baix dels 5 grafs).

Nom del Graf		Random
<b>Diameter</b>		6
<b>Average Dist</b>		2.89
	min.	1
<b>Degrees</b>	avg.	3.8
	max.	8
<b>Clustering</b>		0.2
<b>Comm. Centr</b>	avg.	6.564
<b>Com. Betw Centr</b>	avg.	0.0894

Característiques del graf aleatori que utilitzem per fer les proves. Distància i clustering baixos.



Exemple de graf aleatori



Matriu d'adjacència del graf aleatori

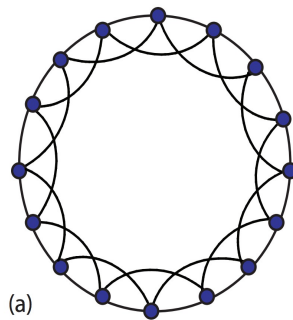
## Graf circulant

En els grafs regulars, tots els vèrtexs tenen el mateix grau. Hi ha grafs regulars amb enllaços aleatoris. En aquest cas (grafs circulars amb enllaços aleatoris) les característiques de clustering i de les distàncies són similars a les dels grafs aleatoris.

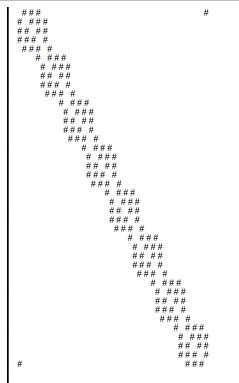
En els grafs fortament regulars, els vèrtexs adjacents comparteixen un cert nombre de veïns, per la qual cosa el clustering és gran (Clustering 0.5 en el nostre cas, el més gran de la sèrie). Per altra banda, per construcció no existeixen enllaços de llarga distància que uneixin parts allunyades de la xarxa, per la qual cosa les distàncies són elevades (5.38 la més gran de la sèrie).

Nom del Graf		Circular
<b>Diameter</b>		10
<b>Average Dist</b>		5.38
	min.	4
<b>Degrees</b>	avg.	4
	max.	4
<b>Clustering</b>		0.5
<b>Comm. Centr</b>	avg.	7.458
<b>Com. Betw Centr</b>	avg.	0.152

Característiques del graf circular/regular que utilitzem per fer les proves. Distàncies llargues. Tots els nodes tenen el mateix grau. Diàmetre, Average Distance i Clustering alts.



Exemple de graf circular



Matriu d'adjacència del graf circular, podem observar com les connexions s'agrupen a la diagonal, per exemple el 3 amb el 2 i el 4 i el 4 amb el 2.



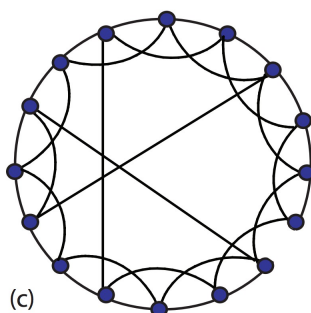
## Graf Small world

Els grafs small world són una modificació dels grafs circulants descrits abans, en els que s'introdueixen alguns enllaços aleatoris.

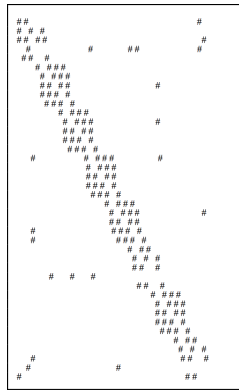
Aquest tipus de graf és d'interès per moltes situacions reals perquè permet modelar alguns aspectes de les xarxes reals, gràcies a que presenta alhora una distància mitjana entre vèrtexs petita, així com un diàmetre petit i un grau d'agrupament de nodes elevat. Aquest tipus de valors són freqüents en sistemes complexos com la xarxa telefònica o Internet.

Nom del Graf		Small World
<b>Diameter</b>		6
<b>Average Dist</b>		3.31
	min.	3
<b>Degrees</b>	avg.	4
	max.	5
<b>Clustering</b>		0.32
<b>Comm. Centr</b>	avg.	6.709
<b>Com. Betw Centr</b>	avg.	0.0934

Característiques del graf small world que utilitzem per fer les proves. Valors ni molt alt ni molt baixos.



Exemple de graf petit-món



Matriu d'adjacència del graf petit-món, molt semblant al graf circulant però amb algunes connexions allunyades de la diagonal, connectant zones llunyanes.

## Graf amb invariància d'escala

En els grafs amb invariància d'escala, la distribució dels graus dels vèrtexs segueix una llei potencial, de manera que molts vèrtexs tenen un grau petit mentre que molt pocs vèrtexs tenen un grau gran. Aquest model també és important a la pràctica perquè reproduïx l'estructura de les connexions d'Internet i de moltes altres xarxes reals.

El terme invariància d'escala s'aplica en general a aquelles xarxes que a mesura que s'afegeixen nodes no canvia el factor d'escala de la seva distribució dels graus. Un exemple de xarxes complexes d'aquest tipus és la WWW, que ha mantingut la seva distribució de graus tot i haver augmentat considerablement el nombre de nodes i enllaços en els últims anys.

Barabasi, A.-L. i Albert R. van proposar el 1999 un algorisme estocàstic per generar grafs invariants d'escala que incorpora dos conceptes generals molt presents a les xarxes reals:

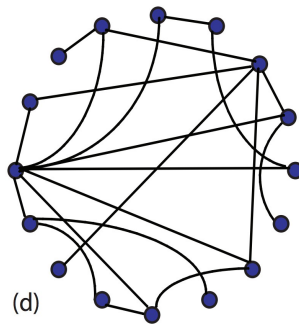
- *El creixement* (o *growth*) indica que el nombre d'enllaços augmenta a mesura que s'afegeixen nous nodes a la xarxa.
- *La preferència d'associació* (*preferential attachment*) indica que, quan un node està molt connectat, és més probable que rebi nous enllaços. Dit d'una altra manera, els vèrtexs de més grau capten més enllaços.

Les característiques d'aquest graf són:

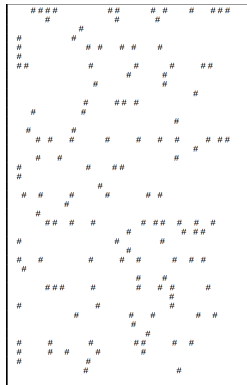
- La distribució de graus no canvia amb més iteracions i queda descrita per una fórmula.
- La distància mitjana creix logàricament amb l'ordre del graf.
- El clustering és una mica més gran que en les xarxes aleatòries.

Nom del Graf		Scale Free
<b>Diameter</b>		4
<b>Average Dist</b>		2.32
	min.	1
<b>Degrees</b>	avg.	4.95
	max.	17
<b>Clustering</b>		0.26
<b>Comm. Centr</b>	avg.	24.244
<b>Com. Betw Centr</b>	avg.	0.113

Diàmetre baix, i distàncies mitges baixes.



Exemple de graf amb invariància d'escala



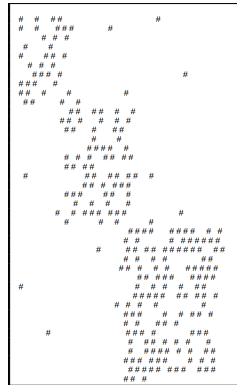
Matriu d'adjacència del graf amb invariància d'escala

## Graf amb agrupament

És un graf que conte grups de nodes molt connectats entre si, aquests grups s'anomenen clústers.

Nom del Graf		Cluster
<b>Diameter</b>		5
<b>Average Dist</b>		2.65
	min.	2
<b>Degrees</b>	avg.	6.3
	max.	13
<b>Clustering</b>		0.37
<b>Comm. Centr</b>	avg.	144.619
<b>Com. Betw Centr</b>	avg.	0.14089

Graf amb agrupament, clustering alt i distàncies baixes.



Matriu d'adjacència del graf clúster, es pot veure com els nodes s'agrupen en tres clústers.

### 3. Els Indicadors o paràmetres rellevants del graf

En aquesta secció definirem els paràmetres rellevants del graf o indicadors que són candidats a definir una xarxa.

Els indicadors que utilitzarem per la funció de cost dels algorismes del projecte són 3:

- Communicability:
  - Communicability Centrality
- Centrality:
  - Betweenness Centrality
  - Communicability Betweenness Centrality

Estrada en el seu llibre situa la CBC com una variant de la Centrality, en pàgines com Networkx, la situen dintre de la communicability.

## Betweenness centrality

La **centralitat** ens indica quins vèrtexs d'un graf són els més importants. Per exemple en una xarxa social ens indicaria quina és la persona que més influència té o en una xarxa de computadors els nodes claus.

Hi han diverses maneres de mesurar la centralitat d'un vèrtex. La betweenness centrality és una d'elles.

La Betweenness centrality mesura el nombre de **vegades que un vèrtex actua com a pont en un camí curt** entre altres dos nodes del graf.

Aquest concepte va se introduït per Freeman, 1979.

La fórmula per calcula la BC és la següent:

$$BC(k) = \sum_i \sum_j \frac{\rho(i, k, j)}{\rho(i, j)}, \quad i \neq j \neq k$$

$\rho(i, k, j)$  són els camins curts que passen per  $k$ .

$\rho(i, j)$  son tots els camins curts que passen per  $i$  i  $j$ .

## Communicability Centrality

La communicability és una mesura introduïda per Ernesto Estrada. En capítol 6 del seu llibre “The structure of Complex Networks” ho explica amb detall i rigor científic.

La comunicació, posem per cas entre dos ciutats no sempre és fa per les carreteres més curtes. Podria ser que el camí més curt és una autopista, però com s'ha de pagar peatge els conductors sovint preferixen passar per camins més llargs.

És possible doncs que una ciutat esdevingui important encara que no estigui en un dels camins curts de la xarxa viària.

La communicability, per tant té en conté la importància de totes les rutes del graf, no només les més curtes.

La communicability centrality en concret és calcula tenint en conte tots els camins tancats que comencen i acaben al node  $n$ .

La **communicability centrality** és pot calcular operant directament amb la matriu d'adjacència que representa un graf. És per això que la implementació del graf que s'ha escollit per aquest projecte **és una implemtació de la estructura graf mitjançant una matriu d'adjacència.**



## Communicability Betweenness Centrality

La communicability Betweenness Centrality va ser proposada per Estrada en el 2009.

La fórmula per calcular la CBC és:

$$BC_r = \frac{1}{C} \sum_p \sum_q \frac{G_{prq}}{G_{pq}}, \quad p \neq q, p \neq r, q \neq r$$

El resultat està dividit per C, que és un factor de normalització. La CBC és un número entre 0 i 1.

$G_{pq}$  són els camins entre p i q,  $G_{prq}$ , els camins entre p i q que passen per r.

$$G_{pq} = \sum_{k=0}^{\infty} \frac{(\mathbf{A}^k)_{pq}}{k!},$$

Càlcul dels camins que passen de p a q.

Aquesta fórmula convergeix cap a:

$$G_{pq} = (e^{\mathbf{A}})_{pq}.$$

$$e^z = \sum_{n=0}^{\infty} \frac{z^n}{n!}.$$

$$e^{At} = \sum_{n=0}^{\infty} \frac{t^n A^n}{n!}.$$

On  $e^A$  és la matriu exponencial, fent una analogia amb les series de Taylor, i amb una matriu  $A$   $n \times n$ .

$$G_{prq} = \left(e^A\right)_{pq} - \left(e^{A+E(r)}\right)_{pq}$$

Càlcul del camins que van de  $p$  a  $q$  passen per  $r$ .

$$BC_r = \frac{1}{C} \sum_p \sum_q \frac{\left(e^A\right)_{pq} - \left(e^{A+E(r)}\right)_{pq}}{\left(e^A\right)_{pq}}, p \neq q, p \neq r, q \neq r$$

El resultat final de la fórmula per la Communicability Betweenness Centrality

Per informació més detallada sobre aquest càlcul, recomano la lectura *The communicability betweenness in complex networks*. Ernesto Estrada, *Physica A*, 388 (2009) 764-774.

La llibreria GNU gsl, ofereix tot de funcions per obtenir  $e^A$ . Es per això que en el projecte s'ha escollit que la implementació dels grafs sigui una matriu en format de la llibreria GNU/gsl, per facilita els càlculs.

## 4. Els Algorismes

Entre els tipus de problemes que ens podem trobar, n'hi ha que són relativament fàcils de resoldre, on el temps de resolució creix de forma lineal amb la quantitat de dades del problema, en els que es poden trobar les solucions ràpidament aplicant una àmplia gamma d'algorismes. Sovint trobem un altre tipus de problemes en que la seva dificultat de **resolució augmenta de forma exponencial** per als quals no es pot garantir una solució òptima en un interval de temps relativament curt.

Hi ha una gran quantitat de problemes que hi ha d'aquest tipus en el dia a dia, i tenim la necessitat de poder-los resoldre de forma ràpida, es van impulsar el desenvolupament de algorismes per trobar bones solucions encara que no fossin les òptimes, però sí bones aproximacions en intervals de temps acceptables.

Aquests metodes, en els que la qualitat de la solució és tan important com el temps que cal per trobar-la són els anomenats metodes heurístics aproximats. Aquests mecanismes proporcionen sovint una bona aproximació a la solució, però no necessàriament la òptima. Lògicament el temps invertit per un mètode exacte per solucionar aquest tipus de problema seria d'una magnitud molt superior, i en alguns casos impracticable.

Un exemple de problema d'optimització combinatoria el trobem en el famós "problema del viatjant", on s'intenta trobar la ruta òptima que ha de seguir un viatjant per passar per totes les ciutats on ha de realitzar vendes sense passar dues vegades per la mateixa i fent la mínima distància possible.

Entre els algorismes existents treballem amb els Simulated annealing o recuita simulada i el Threshold Acceptance.

# Simulated Annealing (SA)

## Origen de l'algorisme Simulated Annealing

El nom ve del procés de recuit (annealing en anglés) d'acer i ceràmiques, una tècnica que consisteix en calentar i després refredar lentament el material per variar les seves propietats físiques. La calor causa que els àtoms augmentin la seva energia i que puguin desplaçar-se a les seves posicions inicial; el refredament lent els hi dona majors probabilitats de recristalitzar en configuracions amb menor energia que la inicial.

Aquest mètode fou descrit per Scott Kirkpatrick a mitjans dels anys 80, s'ha demostrat que és un mecanisme molt potent i molt exitós per resoldre un gran nombre de problemes d'optimització.

## Descripció general i pseudocodi de l'algorisme Simulated Annealing

Traduït als algorismes, per començar ens cal una solució inicial a l'atzar i una funció de cost que n'avaluiï la qualitat. A partir d'aquí, i repetidament, es modifica aleatoriament la solució acceptada en aquell moment i se'n trona a avaluar el cost. El punt clau de l'algorisme és aconseguir que en els primers instants sigui fàcil acceptar una solució que sigui pitjor que l'actual, i que a mesura que avança l'execució sigui cada cop més difícil i es tendeixi a adaptar només els canvis que millorin la resposta acceptada en aquell moment. El paràmetre que regula aquesta evolució s'anomena temperatura.

Pel seu disseny, el simulated annealing evita encallar-se en els mínims locals de la funció de cost. Si cau en un, l'atzar fa que en algun moment s'accepti un canvi a pitjor i es pugui tornar a evolucionar cap a la solució òptima.

L'algorisme, en resum, és el següent:

1. Generar aleatòriament una solució inicial. Fixar la temperatura inicial,  $T_k = T_0$ .
2. Repetir  $N_k$  vegades:
  1. Modificar lleugerament (de forma aleatòria) la solució i calcular la seva funció de cost.
  2. Si és millor, acceptar-la com a nova solució.

3. Si és pitjor, acceptar-la solament si

$$e^{\exp(\Delta f / T_k)} < \text{rand}$$

on  $\Delta f$  és la diferència de cost entre la millor solució i la que s'està revisant i  $T_k$  és la temperatura actual.

3. Disminuir  $T_k$  i repetir 2 fins que  $T_k < T_{\min}$

Cal ajustar els paràmetres  $N_k$ ,  $T_0$ ,  $T_{\min}$ , la forma de disminuir  $T_k$ , etc. Segons la funció de cost i els resultats que s'obtinguin, tant en temps com en qualitat, en cada execució.

## **Aplicació específica per la reconstrucció de grafs amb Simulated Annealing**

El que ens cal veure com podem adaptar al problema que volem resoldre a aquesta descripció general del procés d'optimització per recuita simulada. Per això hem de veure quina analogia podem fer dels conceptes que s'utilitzen en aquest procés, que s'inspira en un procés metal·lúrgic, amb les característiques del procés de reconstrucció dels grafs que és el que ens interessa. A continuació enumerem i descrivim tots i cadascun dels elements clau de la recuita i el relacionem amb el seu equivalent en el context de la reconstrucció de grafs.

1. Generar aleatoriament una solució inicial: En el cas del SA, la Temperatura inicial ha de ser lo bastant alta per acceptar al principi qualsevol canvi. D'aquesta manera l'aleatorietat del resultat pot ser alta, i evitar els mínims locals.
2. Número d'iteracions: Quants intents farem per aconseguir un resultat millor amb una temperatura donada. Aquest paràmetre si és molt alt pot fer que l'algorisme trigui molt, si és molt baix que els resultats no siguin òptims encara que trigarà menys.
3. La temperatura final: La temperatura va disminuint cada  $N$  iteracions, arriba un moment que és difícil superar el resultat actual i la probabilitat d'acceptar resultats no tan bons també. Per tant és posa un límit al nombre d'iteracions amb aquest valor

## Threshold Acceptance (TA)

### Origen de l'algorisme Threshold Acceptance

Threshold acceptance proposa un algorisme que té moltes de les característiques del simulated annealing però on la regla de modificació és determinística. Els resultats en comparació dels dos algorismes indiquen que la estocasticitat dels criteris d'acceptació en el simulated annealing algorithm no juga un rol important en la búsqueda del “near-optimal minima”.

Desde que el **simulated annealing** va ser proposat en 1983 per Kirkpatrick, ha estat un dels més populars algorismes heurístics per trobar solucions near-optimal per problemes d'optimització combinatoria. Kirkpatrick el van aplicar a un layout VLSI i la partició de grafs. L'algorisme normalment es veu atrapat en un mínim local de la funció de cost. Per escapar d'aquest mínim local, Kirkpatrick, van utilitzar un criteri d'acceptació estocàstic, que pot acceptar una nova configuració de cost major que la prèvia, es a dir accepta un resultat pitjor que l'anterior. La probabilitat de acceptar una nova configuració és:

$$P(S) = \exp(-\Delta E/T) .$$

Si  $\Delta E > 0$  la probabilitat es 1.

### Els criteris d'acceptació passen de ser estocàstics a deterministes

Per tal de presentar amb èmfasi el suport a **la idea que la estocacitat de la regla de modificació no és essencial per al bon rendiment del simulated annealing**, es proposa una regla determinística, la qual accepta modificacions quan la T és alta, i es redueixi iterativament a  $T=0$ .

$$P(S)=0 , \text{ si } \Delta E > T$$

és igual a 1 per qualsevol altre.

D'aquesta manera el valor de T és el que determina la acceptació de la modificació, la T esdevé un llindar un “threshold”.

Aquest llindar, aquesta tolerància predefinida va disminuint a mesura que el nombre d'iteracions augmenta. Això vol dir que al principi de les iteracions del

algorisme estem més disposats a acceptar resultats no tan bons. A mesura que van passant les iteracions ens fem més restrictius.

## **Beneficis del threshold**

El càlcul d'acceptació del TA és més ràpid que el del SA perquè no necessita la generació de nombre aleatoris i perquè la evaluació de l'exponencial en la equació, encara que el guany en velocitat és considerablement reduït en aquests problemes **on el càlcul del canvi d'energia és el que consumeix més temps.**

## Descripció general i pseudocodi de l'algorisme Threshold Acceptance

En aquesta taula es veuen els dos algorismes un al costat de l'altre on es pot veure que la diferència és mínima.

<p>SA ALGORITHM FOR MAXIMIZATION.</p> <p>choose an initial configuration choose an initial temperature <math>T &gt; 0</math> Opt: choose a new configuration which is a stochastic small perturbation of the old configuration compute <math>\Delta E := \text{quality}(\text{new configuration}) - \text{quality}(\text{old configuration})</math> IF <math>\Delta E &gt; 0</math>     THEN old configuration := new configuration     ELSE with probability <math>\exp(\delta E/T)</math>         old configuration := new configuration IF a long time no increase in quality or too many iterations     THEN lower temperature <math>T</math> IF some time no change in quality anymore     THEN stop GOTO Opt</p>	<p>TA ALGORITHM FOR MAXIMIZATION.</p> <p>choose an initial configuration choose an initial THRESHOLD <math>T &gt; 0</math> Opt: choose a new configuration which is a stochastic small perturbation of the old configuration compute <math>\Delta E := \text{quality}(\text{new configuration}) - \text{quality}(\text{old configuration})</math> IF <math>\Delta E &gt; -T</math>     THEN old configuration := new configuration IF a long time no increase in quality or too many iterations     THEN lower THRESHOLD <math>T</math> IF some time no change in quality anymore     THEN stop GOTO Opt</p>
Simulated Annealing Algorithm	Threshold Acceptance Algorithm

## Paràmetres de l'algorisme TA

Durant el treball s'han realitzat diferents proves amb diferents paràmetres del algorisme. Que no expliquem en la memòria.

**Llindar d'acceptació:** És el valor que s'utilitza per saber que s'acceptarà un resultat pitjor que l'anterior. Aquest paràmetre normalment és un percentatge. El llindar cada N iteracions, disminueix de valor.

**Nombre d'iteracions:** Es el nombre d'intents per millorar el resultat actual abans de baixar el llindar. És un dels paràmetres més crítics. Si és molt baix l'algorisme és més ràpid però fa pocs intents. Si el nombre és molt alt, el temps d'execució s'allarga. Sovint l'algorisme quan el llindar és molt baix, la probabilitat de millora és baixa si aquest paràmetre és alt podem estar malgastant el temps.

Una millora de l'algorisme és contar el nombre d'iteracions en el qual no s'obtenen millors i posar-hi un màxim.

**Velocitat de refredament:** En cada interacció el llindar es va disminuint per un factor.  $T_k$ . Aquest valor normalment té els valors, 0.98, 0.95, o 0.90. És a dir cada vegada que decidim baixar el llindar ho fem en un 2%, un 5% o un 10%.



Per més informació sobre la relació entre el Simulated Annealing i el Threshold Accepting llegir interessant article “Threshold Accepting: A general purpose optimization algorithms appearing superior to Simulated Annealing” de Gunter Dueck, Tobias Scheuer.

## 6. La implementació

### Els principis bàsics

En aquesta secció explicarem els principis que s'han utilitzat per fer el disseny del software:

- **Tests unitaris:** S'han creat tests unitaris del software que testegen el comportament del propi programa que fa els càlculs. Explico breument que són els tests unitaris, per que serveixen i quines llibreries s'han utilitzat i introdueixo el concepte de coverage.
- A

## Tests Unitaris

En programació una prova unitària és una forma de comprovar el correcte funcionament d'un mòdul de codi. Això serveix per assegurar que cadascun del mòduls funcionen correctament per separat. Després amb les proves d'integració, es poden assegurar el correcte funcionament del sistema o subsistema.

La idea és escriure casos de prova per cada funció no trivial o mètode en el mòdul, de forma que cada cas sigui independent de la resta.

## Beneficis dels Tests Unitaris

L'objectiu del unit testing es aïllar cada part del programa i mostrar quines de les parts individuals són correctes. Un test unitar proporciona un escrit, contracte que cada fragment de codi ha de satisfer. Com a resultat, proporciona diversos beneficis.

### Busca problemes ràpidament

Els test unitaris busquen problemes ràpidament en el cicle de desenvolupament. Això inclou bugs en la implementació del programador i errors i parts de la especificació per al fragment de codi. El procés d'escriure un conjunt de tests força al programador a pensar entrades, sortides i condicions d'errors i aquells tests més detallats defineixen el comportament desitjat. El cost de buscar un bug abans que la programació comenci o quan el codi acaba de ser escrit és considerablament més baix que el cost de detectar-ho, identificar-ho i corregir el bug més tard. Els bugs també poden causar problemes per els usuaris finals del software. El codi que es impossible o difícil de testejar la seva qualitat és baixa, el test unitaris poden forçar als desenvolupadors a estructurar les funcions i els objectes de manera millor.

En les tècniques més extremes de unit testing, el Test Driven development (TDD), que s'utilitza en extreme programming i scrum, els tests unitaris es creen abans del propi codi. Si passen els tests es consideren que el codi està complert. Els tests unitaris s'executen freqüentment, si els tests unitaris fallen alerten el equip de desenvolupadors del problema abans de passar el codi al l'equip de test o als propis clients.

### Facilita el canvi

Els tests unitaris permeten al programador refactoritzar el codi o actualitzar les llibreries posteriorment, i assegurar que el mòdul continua funcionant correctament ( regression testing). El procediment és escriure casos de test per totes les funcions i mètodes així que qualsevol canvi que causi un error, es pugui identificar ràpidament.

Els test unitaris detecten canvis que pot trencar el contracte de disseny.

### **Simplifica la integració**

Els tests unitaris poden ser utilitzats en una aproximació de baix cap a dalt en el estil de test. Testejant les parts del programa primer i després testejar la suma de totes les parts, els test de integració esdevenen molt fàcil.

### **Documentació**

Els tests unitaris proporcionen documentació sobre el sistema. Els desenvolupadors que busquin queina és la funcionalitat que proporciona un mòdul, i com utilitzar-la, poden mirar els tests unitaris per obtenir un coneixement bàsic de la interfície del mòdul.

Els tests unitaris incorporen característiques que son crítiques per al funcionament dels tests. Aquestes característiques iniquen apropiats/inapropiats ús dels mòduls, així com a comportaments negatius que son tractats pel mòdul. Un cas de test unitari, per si mateix, documenta aquestes característiques crítiques.

### **Disseny**

Quan el software és desenvolupat utilitzant test-driven development, la comunicació de escriure test unitaris per especificar la interfície a més de les activitats de refactorització realitzades després que els test passin, donen lloc de un disseny formal. Cada test unitari pot ser vist com un element de disseny especificant classes, mètode i el comportament observat.

A continuació mostro un fragment de codi dels tests unitaris del projecte. En aquest test unitari s'esta testejant la rutina `vertexAreNeighbours`. Aquesta rutina retorna `VERTEX_CONNECTED` si els dos vertexs estan connectats i `VERTEX_DISCONNECTED` si no ho estan. El graf `wheel14` és el següent.



Wheel 14 graph utilitzat en un dels test unitaris, el node 0, es connecta a la resta.

```
void UTest_gslGraph_vertexAreNeighbours(){  
    gslGraph * wheel14Graph = ReadPythonGraphFile::readPythonGraphFile(DIR_GRAPHS "wheel14.txt");  
    int numberOfVertexForWheel14Graph = 14;  
    for ( int i = 1; i < numberOfVertexForWheel14Graph; i++ ){  
        BOOST_CHECK( wheel14Graph->vertexAreNeighbours(0,i) == gslGraph::VERTEX_CONNECTED);  
        BOOST_CHECK( wheel14Graph->vertexAreNeighbours(i,0) == gslGraph::VERTEX_CONNECTED);  
    }  
    BOOST_CHECK( wheel14Graph->vertexAreNeighbours(2,4) == gslGraph::VERTEX_DISCONNECTED);  
    delete wheel14Graph;  
}
```

Exemple de test unitari per la rutina vertexAreNeighbours, per el graf tipo roda.

## Frameworks de tests automatitzats de C++

Hi han diversos frameworks per realitzar tests unitaris en C++, més de 30. En destaquem dos que son els que s'han utilitzat al projecte:

- La llibreria *Boost test library*. Boost és una llibreria de c++ que et dona moltes funcionalitats extres. En el projecte s'utilitza per parsejar els paràmetres d'entrada i per fer tests unitaris.

- *Google test*, que juntament amb la llibreria *Google C++ Mocking Framework de google*, te moltes funcionalitats com mock.

A continuació és mostra la sortida d'un test suite del google test framework del projecte. En aquest cas s'esta testejant la utilitat analitza:

```
DYLD_LIBRARY_PATH=../gtest/libs/ ./utestanalitza_main
[=====] Running 6 tests from 3 test cases.
[-----] Global test environment set-up.
[-----] 3 tests from Llegir_dadaes
[ RUN      ] Llegir_dadaes.if_file_do_not_exists_should_return_error
[      OK  ] Llegir_dadaes.if_file_do_not_exists_should_return_error (0 ms)
[ RUN      ] Llegir_dadaes.if_file_exist_should_return_ok
[      OK  ] Llegir_dadaes.if_file_exist_should_return_ok (1 ms)
[ RUN      ] Llegir_dadaes.circ_graph_should_return_specificvalues
[      OK  ] Llegir_dadaes.circ_graph_should_return_specificvalues (1 ms)
[-----] 3 tests from Llegir_dadaes (2 ms total)

[-----] 2 tests from clustering
[ RUN      ] clustering.circ_graph_should_return_specificvalues
[      OK  ] clustering.circ_graph_should_return_specificvalues (0 ms)
[ RUN      ] clustering.rand_graph_should_return_specificvalues
[      OK  ] clustering.rand_graph_should_return_specificvalues (1 ms)
[-----] 2 tests from clustering (1 ms total)

[-----] 1 test from distances
[ RUN      ] distances.circ_graph_should_return_specificvalues
           Clustering 0.594762
           Diametre 5
           Distancia mitja 0.255128
[      OK  ] distances.circ_graph_should_return_specificvalues (0 ms)
[-----] 1 test from distances (0 ms total)

[-----] Global test environment tear-down
[=====] 6 tests from 3 test cases ran. (3 ms total)
[ PASSED  ] 6 tests.
```

Output dels tests uniaris de la utilitat analitza

Es pot veure com hi han 6 tests , agrupats en 3 test cases.

Els 3 test cases són:

- Llegir dades, relacionats amb llegir un graf desde un fitxer amb tres tests:

- Si el fitxer no existeix la rutina retorna un error.
- Si el fitxer existeix la rutina retorna ok
- Si el fiter existeix es comproven que els valors del graf siguin correctes.
- El segon test case és relacionat amb el clustering
- El tercer test case és relacionat amb les distances.

Visualment mostra força informació:

- Quins test cases hi han
- Quants tests hi han per test case
- Quins tests han passat i quins tests han fallat.

## Representació matricial de la classe Graf

El nucli bàsic del programa és la classe graf. Aquesta classe graf representa una xarxa.

Els grafs de  $N$  vèrtex es poden representar amb una matriu  $M$  de  $N \times N$  elements.

Si un vèrtex  $v$  està connectat amb un vèrtex  $u$  aleshores  $M[u,v]=1$  i  $M[v,u]=1$ , per grafs no dirigits que són els que parlem sempre en aquest treball.

Existeix una llibreria GNU `gsl` que s'ha utilitzat per fer operacions en matrius. Aquesta llibreria permet multiplicar matrius i calcular la matriu exponencial. Aquestes dos operacions són molt importants a l'hora de calcular la CC i la CBC.

Per aquesta raó en la implementació de la matriu de la classe graf està implementada amb una matriu GNU `gsl`.

## Programes utilitzats per fer el projecte

- Git: És un repositori de codi font. En concret he utilitzat un que permet emmagatzemar el codi de manera privada. Bitbucket.
- Vagrant: Eina que serveix per veure si el programa té memory leaks.
- gprof: Serveix per visualitzar el performance de la aplicació.



## **Posibles millores en la simulació i obtenció de resultats**

Després de realitzar el projecte, es podrien tenir en compte aquestes millores en la implementació del software.

### **Paralelització i cloud Computing**

Una de les possibles millores seria que aquest algorisme es pogues executar en varies instàncies del amazon web service de manera paral·lela.

### **Utilització de varis Indicadors simultàneament**

Per tal de reconstruir un graf en comptes de utilitzar un paràmetre es podria utilitzar una funció de cost multivariable amb dos d'ells. Podria donar lloc a un graf més semblant a l'original.

### **Altres algoritmes alternatius al Simulated Annealing i al Threshold acceptance**

Utilitzar algun altre algorisme, no tan genèrics com son Simulated o el Threshold i buscar un que fos més específic del problema de xarxes i que se li treïés més suc als paràmetres BC,CC,CBC amb que es treballen. Es podria combinar amb una tècnica d'algorisme voraç.

## 7. Els Resultats

Output	dash	dash	dash	dash	dash	dash	
<pre>top - 11:03:21 up 25 days, 21:29,  2 users,  load average: 6.01, 6.02, 6.05 Tasks: 357 total,   7 running, 350 sleeping,   0 stopped,   0 zombie %Cpu(s): 25.0 us,  0.0 sy,  0.0 ni, 75.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st KiB Mem: 13202384+total, 6038044 used, 12598580+free,  903616 buffers KiB Swap: 96778240 total,   0 used, 96778240 free. 3328356 cached Mem</pre>							
PID	USER	PR	NI	VIRT	RES	SHR S %CPU %MEM TIME+ COMMAND	
4445	xisco	20	0	7564	828	588 R 100.0 0.0	35634:42 prunatv7
4451	xisco	20	0	7576	832	584 R 100.0 0.0	35634:29 prunatv7
4457	xisco	20	0	7588	844	584 R 100.0 0.0	35633:45 prunatv7
4466	xisco	20	0	7596	860	584 R 100.0 0.0	35633:04 prunatv7
30832	oscar.r+	20	0	29920	10132	2136 R 100.0 0.0	26:38.36 rebuild_graph
30834	oscar.r+	20	0	22280	2480	2080 R 100.0 0.0	3:35.84 rebuild_graph
1	root	20	0	33780	3148	1480 S 0.0 0.0	0:07.15 init
2	root	20	0	0	0	0 S 0.0 0.0	0:00.10 kthreadd
3	root	20	0	0	0	0 S 0.0 0.0	0:00.00 kthreadd

Imatge de les CPUs al 100x100 en un dels servidors.

Les simulacions s'han executat en 3 ordinadors diferents.

- Un Servidor de la UPC Linux Ubuntu de 24 cores 125 Gb de memòria RAM
- Un ios de la UPC amb 8 cores 32 Gb memòria RAM.
- Un portàtil personal ios amb 4 cores. 16Gb memòria RAM.

A continuació es mostren els resultats obtinguts per diferents simulacions.

El procediment per totes elles és idèntic.

Es generen N simulacions per cada graf d'exemple. Els grafs d'exemple són 5:

- Graf aleatori
- Graf amb clusters
- Graf petit-món
- Graf amb invariàcia d'escala
- Graf circulant

## Paràmetres dels algorismes

Els paràmetres que s'han utilitzat per executar els algorismes SA i TA són:

Temperatura Inicial:1.0

Temperatura final: 0.000001

Tasa de refredament: 0.9

Número de passos per iteració:2000

## Paràmetres obtinguts com a resultat de les simulacions

Per cada conjunt de simulacions, es realitzen unes mitjes d'algunes mesures dels grafs resultants. Les mesures són:

**Diàmetre:** Mitja del diàmetres del grafs reconstruït. Al costat de la taula pots veure el diàmetre del graf original. Per exemple. aleatori original 6, aleatori Reconstruït amb CC i TA 5.075.

**Distància mitja:** Mitja de les distàncies dels grafs reconstruït. Per exemple. aleatori Original 2.89, aleatori reconstruït CC i TA, 2.65.

**Degrees o Graus:** Valors Mínim del grau per tots els vèrtexs, la mitja i el valor màxim de Grau. Per exemple, aleatori original 1, 3.8 i 8, i per aleatori Reconstruït CC i TA 1, 3.94 i 12.

**Clustering:** El valor de clustering per el graf original i la mitja del clustering per tots els grafs reconstruïts.

**Comm. Centr.:** El valor del paràmetre rellevant que estem analitzant. Que pot ser La Communicability Centrality, la Communicability Betweenness Centrality o la Betweenness Centrality.

**Delta:** Es un valor que indica la similitud o com d'isomorf son els dos grafs, el que volem reconstruir i el graf original.

## Communicability Centrality amb Threshold Acceptance

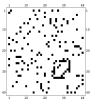
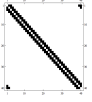
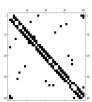
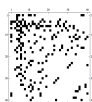
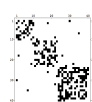
Les simulacions amb el Communicability Centrality suposen la aportació més important del projecte.

També la utilització del Threshold Acceptance com algorisme de reconstrucció de grafs.

La communicability Centrality és ràpida de calcular en comparació al CBC, que és més costosa. És calcula a partir de la matriu exponencial, de la matriu d'adjacència que representa la xarxa.

S'han realitzat unes 200 simulacions diferents, cadascuna generada amb llavors diferents per generar nombres aleatoris diferents a cada simulació.

Els resultats són els següents:

CC Threshold		Aleatori		Circulant		Petit-món		Invariant d'Escala		Agrupament	
											
		Ref.	Recons	Ref.	Recons	Ref.	Recons	Ref.	Recons	Ref.	Recons
Execucions			200		200		200		200		313
Diameter		6	5.075	10	4.115	6	4.545	4	3.975	5	3.354
Average Dist		2.89	2.65	5.38	2.4612	3.31	2.546	2.32	2.205	2.65	2.017
Degrees	min.	1	1	4	3	3	2	1	1	2	1
	avg.	3.8	3.9457	4	4.726	4	4.429	4.95	5.148	6.3	6.675
	max.	8	12	4	6	5	8	17	21	13	20
Clustering		0.2	0.102	0.5	0.0796	0.32	0.078	0.26	0.2447	0.37	0.3287
Comm. Centr	avg.	6.564	6.6952	7.458	7.342	6.709	6.663	24.244	25.72	144.61	163.62
delta	avg.		0.0232		0.0492		0.028		0.0255		0.0598
Resultats del Threshold Acceptance amb Communicability Centrality											

En primer lloc podem dir que l'algorisme Threshold Acceptance funciona força bé.

Com a entrada li donem una xarxa, per exemple la xarxa amb invariàcia d'escala amb una CC de 24.44 i ens retorna una xarxa reconstruïda amb un paràmetre molt semblant 25.72.

Per saber si el paràmetre Communicability Centrality és un paràmetre rellevant hem de mirar la delta que indica l'isomorfisme dels grafs.

**En aquest cas les xarxes que ha reconstruït millor són el aleatori i la xarxa amb invariàcia d'escala.**

Els grafs que han reconstruït pitjor són la xarxa amb agrupament i el circulant.

## Communicability Centrality amb Simulated Annealing

Aquestes simulacions tenen molt poques simulacions tants sols 10 llavors. Serveixen a mode orientatiu, per validar els números d'altres simulacions.

CC Annealing		Aleatori		Circulant		Petit-món		Invariant d'Escala		Agrupament	
		Ref.	Recons	Ref.	Recons	Ref.	Recons	Ref.	Recons	Ref.	Recons
Execucions			10		10		10		10		10
Diameter		6	5.1	10	4	6	4.5	4	4	5	3.4
Average Dist		2.89	2.664	5.38	2.44538	3.31	2.5667	2.32	2.19692	2.65	2.01436
Degrees	min.	1	1	4	4	3	2	1	1	2	2
	avg.	3.8	3.915	4	4.745	4	4.4	4	5.08	6.3	6.685
	max.	8	11	4	6	5	7	17	21	13	19
Clustering		0.2	0.1374	0.5	0.096	0.32	0.0905	0.26	0.26421	0.37	0.32474
Comm. Centr	avg.	6.564	6.6653	7.458	7.411	6.709	6.636	24.244	25.66	144.619	163.294
delta	avg.		0.0214		0.0515		0.0271		0.0309		0.05798

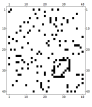
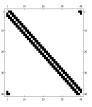
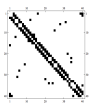
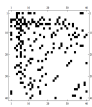
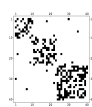
Observem que els números són bastants semblants als de la simulació anterior, podríem dir que realment la incidència en els resultats no depenen del tipus d'algorisme utilitzat, es a dir si utilitzem SA o TA.

## Communicability Betweenness Centrality amb Threshold Acceptance

La Communicability Betweenness Centrality és molt costosa de calcular que la CC o la BC.

El cost del algorisme és proporcional al nombre de nodes de la xarxa. Això vol dir que aproximadament triguem unes 40 vegades més que la CC.

Això dificulta les simulacions, i generar una gran base de resultats.

CBC Threshold		Aleatori		Circulant		Petit-món		Invariant d'Escala		Agrupament	
											
		Ref.	Recons	Ref.	Recons	Ref.	Recons	Ref.	Recons	Ref.	Recons
Execucions			101		101		101		101		101
Diameter		6	5.1485	10	3	6	4.01	4	4.626	5	4.237
Average Dist		2.89	2.5878	5.38	1.897	3.31	2.33	2.32	2.376	2.65	2.143
Degrees	min.	1	1	4	6	3	2	1	1	2	1
	avg.	3.8	4.3178	4	8.222	4	5.18	4	4.736	6.3	6.469
	max.	8	12	4	10	5	8	17	19	13	16
Clustering		0.2	0.1027	0.5	0.1713	0.32	0.0982	0.26	0.212	0.37	0.19
Com. Bet Cent	avg.	0.0894	0.0928	0.152	0.1544	0.0934	0.0951	0.113	0.112	0.1408	0.138
delta	avg.		0.0283		0.1536		0.0523		0.0243		0.055






Els resultats veiem que el CBC, és el que reproduïx millor el scale free, amb un 0.024 de delta, el aleatori també el reproduïx força bé. En canvi el Circulant, el petit-món i el Clustered els reproduïx força malament. Té el millor resultat en les xarxes amb invariància d'escala i el pitjor en Circulant.

## Betweenness Centrality amb Simulated Annealing

La BC, és un indicador fàcil de calcular, l'ordre de càlcul és semblant al del CC.

Hem fet 10 llavors.

Els resultats serveixen per comparar els resultats amb anteriors treballs i assegurar-nos que el Simulated Annealing com la resta de funcionalitats del programa són correctes.

BC Annealing		Aleatori		Circulant		Petit-món		Invariant d'Escala		Agrupament	
											
		Ref.	Recons	Ref.	Recons	Ref.	Recons	Ref.	Recons	Ref.	Recons
Execucions			10		10		10		10		10
Diameter		6	5.6	10	8.8	6	6.6	4	4.4	5	5.5
Average Dist		2.89	2.786	5.38	4.7582	3.31	3.015	2.32	2.348	2.65	2.59808
Degrees	min.	1	1	4	2	3	1	1	1	2	1
	avg.	3.8	3.96	4	2.58	4	3.75	4	5.075	6.3	4.375
	max.	8	12	4	5	5	9	17	21	13	13
Clustering		0.2	0.1004	0.5	0.0313	0.32	0.0714	0.26	0.2171	0.37	0.1264
Betweenness C	avg.	0.049	0.047	0.1154	0.0989	0.0607	0.053	0.034	0.035	0.0433	0.04205
delta	avg.		0.0264		0.074		0.0322		0.023		0.0849



## 8. Conclusions

### Reconstrucció de grafs

No hem conseguit reconstruir el graf original al 100x100, però sí que s'assemblen bastant a l'original en alguns casos, sobretot en xarxa aleatòria, la xarxa amb invariància d'escala i el petit-món.

Els pitjors casos són el circulant o regular i el graf clúster.

Crec que els dos algorismes amb aquestes mesures generen sempre una topologia de graf que estaria a la mateixa distància isomòrfica de les xarxes aleatòries, petit-món i xarxes amb invariància d'escala i que en funció de la CC/BC/CBC amb que estem comparant és belluga cap al graf original però no suficient.

### Simulated Annealing vs Threshold Acceptance

Els dos algorismes el Simulated Annealing i el Threshold Acceptance reproduïen grafs amb la mesura que utilitzaven. Es a dir si tu vols obtenir un graf amb una determinada BC, CC o CBC, els algorismes et generen aquest graf.

Crec que la **diferència entre un algorisme i l'altre (entre SA i TA) no és significativa**. Potser el Simulated Annealing funciona una mica millor en algun cas (SA amb CC graf aleatori 0.21), però el número de simulacions he realitzat amb SA no és conclouent.

### CBC, CC i BC

De les tres mesures estudiades, la rellevància que tenen, també és molt semblant. El que sí sembla que el CC s'aproxima molt més al grafs circulant i la xarxa amb agrupament que el BC. El BC amb el graf circulant és el que s'apropa més al diàmetre.

El CBC obté els millors resultats amb xarxes amb invariància d'escala i els pitjors amb circulant.

El **CC és més regular** amb tots els tipus de graf.

## Milliores

Una possible millora seria que al modificar el graf, en comptes de fer-ho sobre un vèrtex aleatori ho fes sobre el conjunt de vèrtex que tenen pitjor semblança amb el paràmetre BC, CC, CBC del original. S'hauria de testejar. Seria utilitzar una barreja entre SA-TA i un algorisme de tipo greedy.

## 9. **Apèndix**

## 10. Referències

### Bibliografia

#### Teoria de Grafs

Matemáticas discreta y combinatoria, Ralph P. Grimaldi, Ed. Addison-Wesley Iberoamericana

Matemática discreta, Francesc Comellas, Josep Fàbrega, Anna Sanchez, Oriol Sera Ed. Edicions UPC

The Structure of complex Networks, Ernesto Estrada, Oxford Press, 2011

- Chapter 6: Communicability functions in Networks

The communicability betweenness in complex networks. Ernesto Estrada, Physica A, 388 (2009) 764-774

Complex Networks, Structure, Robustness and Function. Shlomo Havlin, Reuven Cohen. Cambridge.

#### Tipus de Xarxes

[Newman] The structure and function of complex networks. Newman.

[Watts, Strogatz] Collective dynamics of "small-world" networks, Nature 393, 440-442.

#### Isomorfismes:

[https://es.wikipedia.org/wiki/Isomorfismo\\_de\\_grafos](https://es.wikipedia.org/wiki/Isomorfismo_de_grafos)

#### Algorismes

Stochastic versus deterministic Update in Simulated Annealing, Pablo Moscato, J.F. Fontanari, 1990, physics Letters A, Volume 146, number 4.

Threshold Accepting: A general Purpose Optimization Algorithm Appearing Superior to Simulated Annealing, Gunter Dueck, Tobias Scheuer, Journal of Computational Physics, 90, 161-175, 1990.

#### Design Patterns

Head First Design Patterns, Eric Freeman, Elisabeth Robson, Bert Bates, Kathy



Sierra. 2004

## **Altres treballs referents a la reconstruccions de grafs**

Reconstrucció Espectral de Grafs, Jordi Díaz Lopez, PFC 2005

Reconstrucció de grafs a partir del grau d'intermediació (betweenness) dels seus vèrtexs, Juan Paz Sanchez

## **Links d'interés**

- Big data i teoria de grafs
  - Graph Theory: Key to Understanding Big Data
    - <http://www.wired.com/insights/2014/05/graph-theory-key-understanding-big-data-2/>
  - Grafs aleatoris
    - [https://en.wikipedia.org/wiki/Erd%C5%91s%E2%80%93R%C3%A9nyi\\_model](https://en.wikipedia.org/wiki/Erd%C5%91s%E2%80%93R%C3%A9nyi_model)
  - Llibreria de Python amb funcions per grafs.
    - <https://networkx.github.io/>
  - Ernesto Estrada, introductor de la communicability
    - [https://en.wikipedia.org/wiki/Ernesto\\_Estrada](https://en.wikipedia.org/wiki/Ernesto_Estrada)