



Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROJECTE FINAL DE CARRERA

RECONSTRUCIO DE REDS 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.....	7
Resum del Projecte.....	8
Resumen del Proyecto.....	9
Abstract.....	10
1.Introducció.....	11
Context del projecte.....	11
Objectius.....	13
Estructura de la memòria.....	15
2.Teoría de Grafs.....	16
Definicions i exemples.....	16
Grau d'un vèrtex: camins.....	17
Exemples d'utilització de grafs.....	19
Isomorfismes de Graf.....	22
Models de grafs	23
3.Els Indicadors.....	26
Betweenness centrality	26
Communicability Centrality.....	27
Communicability Betweenness Centrality.....	28
4.Els Algorismes.....	29
Tipus d'Algorismes.....	29
Simulated Annealing.....	29
Threshold Acceptance.....	31
5.Comparació de grafs.....	34

6.La implementació.....	35
Els principis bàsics.....	35
Tests Unitaris.....	35
Principis de la programació Orientada a Objectes.....	41
El digrama de classes.....	42
Representació matricial.....	42
Posibles millores.....	42
7.Els Resultats.....	43
Betweenness Centrality amb Simulated Annealing.....	43
Betweenness Centrality amb Threshold Acceptance.....	44
Communicability Centrality amb Simulated Annealing.....	45
Communicability Centrality amb Threshold Acceptance.....	46
Communicability Betweenness Centrality amb Simulated Annealing.....	47
Communicability Betweenness Centrality amb Threshold Acceptance.....	48
8.Conclusions	50
9.Apèndix.....	52
10.Referències.....	53
Bibliografia.....	53

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 programar i testear uns algoritmes que, donat graf original, extreure'n unes propietats que el descriuen, i intentar reproduir el graf original.

Resumen del Proyecto

El objetivo del proyecto es programar y testear unos algoritmos que, dado un grafo original, extraer unas propiedades que lo describen, y intentar reproducir el grafo original.

Abstract

The project objective is to programm and test two algorithms that, given an original graph, extract properties that determine it, i play the original graph.

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 **reds 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 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 grafs

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 grafs, associant els nodes i enllaços de la xarxa als vèrtexs i arestes del graf.

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

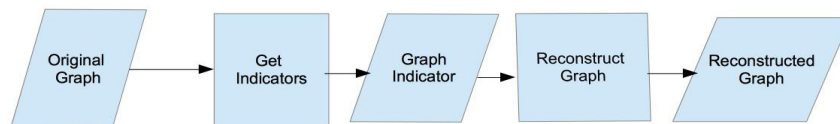
Quan una xarxa és estàtica, per conèixer el seu estat “només” cal conèixer l'estat de cadascun dels seus elements (nodes i enllaços). Si bé és cert que recollir tota aquesta informació en alguns casos pot no ser trivial, en el cas de les xarxes dinàmiques, on els nodes i els enllaços apareixen i desapareixen 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 graf 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 grafs és un dels temes en els que s'està treballant actualment i on es poden trobar aproximacions molt diferents. La necessitat de reconstruir grafs té diverses justificacions. D'entre totes, aquest estudi pretén aportar una **alternativa per l'emmagatzematge de grafs**.

Si som capaços de redimensionar una imatge vectorial, seriem capaços de fer-ho amb un graf?

Objectius

L'objectiu principal d'aquest projecte és avaluar l'ús de dos algorismes coneguts com recuita simulada o *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 indicadors del graf.



El contexte del projecte. A partir d'un graf, extraïem indicadors, com la Betweenness centrality, la Communicability Betweenness Centrality o la Communicability Centrality. A partir d'aquests indicadors, i el nombre de vèrtexs, intentem reconstruir el graf original.

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 random,

- un graf small-world,
- un graf scale-free,
- un regular, circular
- i un graf cluster,

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

L'origen de la teoria de grafs s'associa amb la resolucó que va donar 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 conectada 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 resolvia a aquesta qüestió, sino que va introduir la noció de graf i va resoldre al mateix temps un problema de caràcter més general.

Un **graf no dirigit** $G=(V,E)$ és una estructura combinatoria 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**.

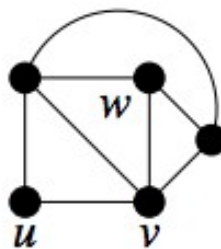


Figura 1.1: Graf amb ordre 5 i tamany 8

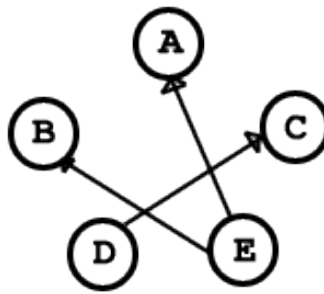


Figura 1.2: Graf amb ordre 5 i tamany 3

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 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.

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

Si existeixen dues o més arestes que uneixen els mateixos vèrtexs parlem de branques paral·leles. un graf que contingui branques paral·leles s'anomena multigraf. Un llaç és una arista que comença i acaba en el mateix vèrtex.

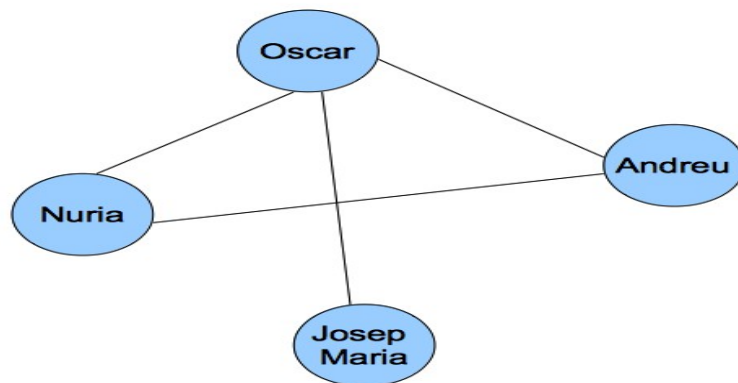
Un **cluster** és un conjunt de vèrtexs entre els que existeixen moltes connexions. El **clustering** n'és un paràmetre relacionat, que mesura la connectivitat local d'un graf. El clustering d'un vèrtex es defineix com la fracció de branques que uneixen les veïns d'aquest vèrtex entre ells, entre la 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.

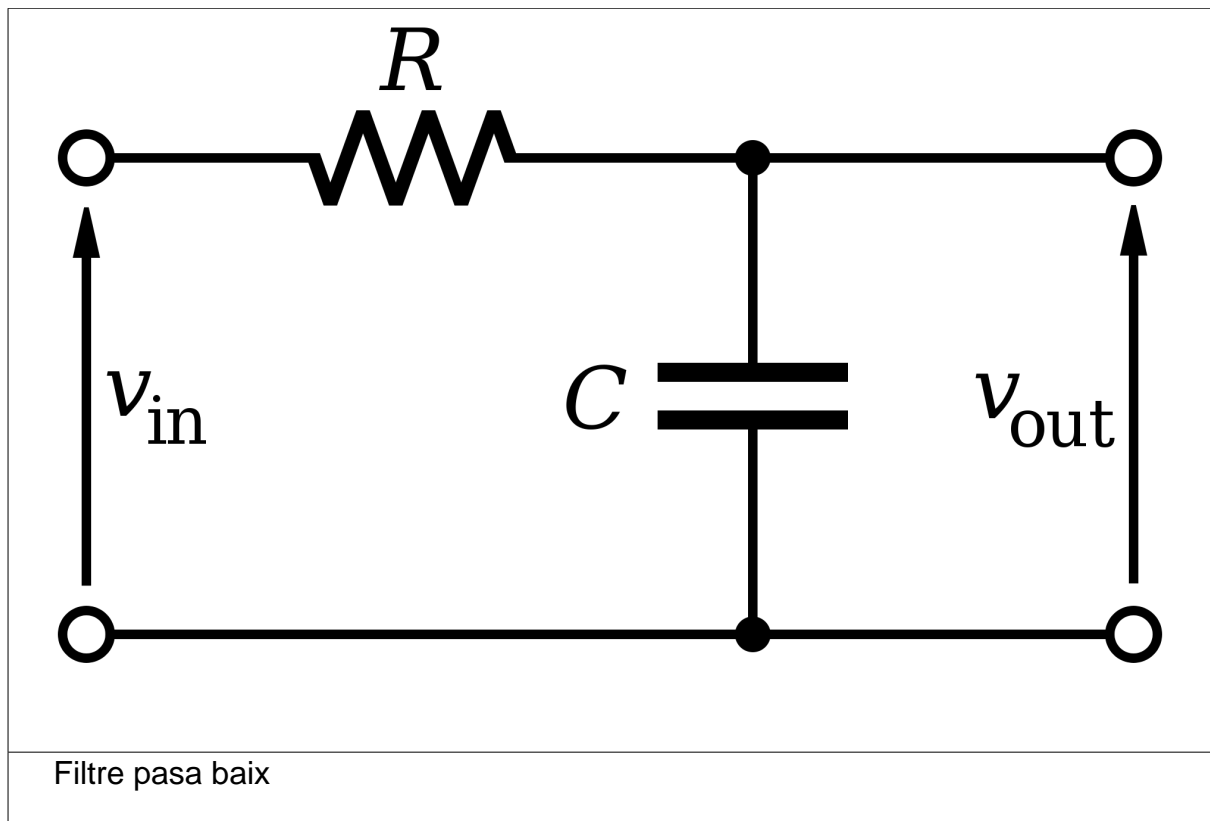
El **grau**, el **clustering**, el **diàmetre** i la **distància** són paràmetres que en els resultats finals ens ajudaran a comparar el graf original amb el reconstruït.

Exemples d'utilització de grafs

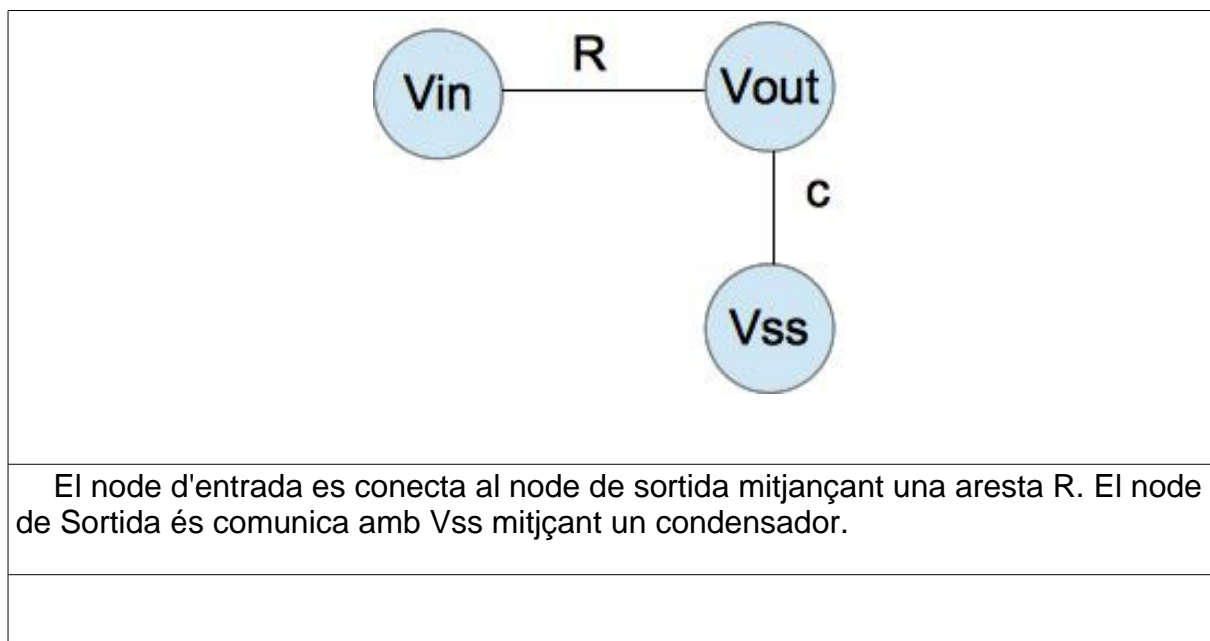
Al final els grafs representen relacions entre elements. Avui en dia podríem parlar de facebook com a exemple de graf. Jo tinc 3 amistats. La Núria, l'Andreu i Josep Maria. La Núria també és amiga de l'Andreu.



Red social representada amb un graf



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





Isomorfismes de Graf

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 hongaresos) 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
Diameter		6
Average Dist		2.89
	min.	1
Degrees	avg.	3.8
	max.	8
Clustering		0.2
Comm. Centr	avg.	6.564
Com. Betw Centr	avg.	0.0894

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

Graf circular

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 vèrtexs, 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
Diameter		10
Average Dist		5.38
	min.	4
Degrees	avg.	4
	max.	4
Clustering		0.5
Comm. Centr	avg.	7.458
Com. Betw Centr	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.

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
Diameter		6
Average Dist		3.31
	min.	3
Degrees	avg.	4
	max.	5
Clustering		0.32
Comm. Centr	avg.	6.709
Com. Betw Centr	avg.	0.0934

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

Graf scale-free

En els grafs scale-free, la distribució dels graus dels vèrtex 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.

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
Diameter		4
Average Dist		2.32
	min.	1
Degrees	avg.	4.95
	max.	17
Clustering		0.26
Comm. Centr	avg.	24.244
Com. Betw Centr	avg.	0.113

Diàmetre baix, i distàncies mitges baixes.

Graf cluster

És un graf que conté grups de nodes molt connectats entre si, aquests grups

s'anomenen clústers.

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

Graf cluster, clustering alt i distàncies baixes.

3. Els Indicadors

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

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

Betweenness centrality

La **centralitat** ens indica quins vèrtexs d'un graf són els més importants. Per exemple en una red social end indicaria quina és la persona que més influència té o en una red 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:

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

Fórmula per calcula la Betweenness centrality

$\rho(i, k, j)$ 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 entre 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 esdevingue 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:

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

Communicability Betweenness Centrality per al node r

El resultat esta 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.

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

Càlcul dels camins que passen de p a q.

Aquesta fórmula convergeix cap a:

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

Convergent la fórmula anterior

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

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

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

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 $e^{\mathbf{A}}$. 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

Evolució dels algorismes

Els algorismes que utilitzem són un seguit de refinaments successius i propostes dels següents algorismes:

Metropolis type Algorithms

Monte Carlo Methods

Simulated Annealing, que és una combinació dels dos anteriors

Threshold Acceptance, que és una variació del Metropolis type Algorithm.

IMATGE AMB TIPUS D'ALGORISME

Simulated Annealing

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 inicials; el refredament lent els hi dona majors probabilitats de recristalitzar en configuracions amb menor energia que la inicial.

Aquest mètode fou descrit independentment per Scott Kirkpatrick en 1985.

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 dendeixi 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 torna a evolucionar ap 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 aleatoria) la solució i calcular la seva funció de cost.
 2. Si és millor, acceptar-la com a nova solució.
 3. Si és pitjor, acceptar-as solament si

$$e^{-\Delta f / T_k} < \text{rand}$$

on Δf és la diferència de cost entre la millor solució i la que s'estar 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 e s 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.

FER LA ANALOGIA ENTRE LA PART TEORICA DEL ANNALING I EL NOSTRE GRAF

Threshold Acceptance

Origen de l'algorisme Threshold Acceptance

Threshold acceptance proposa un algoritme que te moltes de les característiques del simulated annealing però on la regla de modificació es determinística. Els resultats en comparació els dos algoritmes indiquen que la estocicitat de les “Metropolis updating” 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 “local minimum” de la funció de cost. Per escapar d'aquest “local minima”, Kirkpatrick, van utilitzar a stochastic criteria d'acceptació. El Metropolis updating, que pot acceptar una nova configuració de cost major que la pervia, 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.

Per tal de presentar amb èmfasi el suport a la idea que la estocicitat de la regla de modificació no és essencial per al bon performance del simulated annealing, es proposa una regla determinística, la qual accepta arranjaments quan la T és alta, i es redueix 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 l'acceptació del arranjament, ens referim a la equació anterior com a “threshold updating”.

El “threshold updating” és més ràpid que el “Metropolis update” perquè no necessita la generació de nombre aleatori 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.

En resum la idea es que l'èxit del simulated annealing no depèn en gran mesura de la estocicitat de l’“updating rule”. Per els autors, depèn més de la “smoothing of the cost function landscape at high temperature” i la definició del mínim durant el procés de refredament.

Descripció general i pseudocodi de l'algorisme Threshold Acceptance

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

Per més informació sobre la relació entre el Simulated Annealing i el Threshold Acceptance llegir interessant article "Threshold Accepting: A general purpose optimization algorithms appearing superior to Simulated Annealing"

Aplicació específica per la reconstrucció de grafs amb Threshold Acceptance

5. Comparació de grafs

EXPLICAR COM ÉS COMPAREN DOS GRAFS

6. La implementació

Els principis bàsics

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 método en el mòdul, de forma que cada cas sigui independent de la resta.

Beneficis dels Tests Unitaris

L'objectiu del unit testing es aïllar cda 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 sere 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 imposible o difícil de testejar la seva qualitat és baixa, el test unitaris pot forçar als desenvolupadors a estructurar les funcions i els objectes de manera millor.

Les tècniques més extremes de unit testing, el Test Driven development (TDD), que s'utilitza en extrem programming i scrum, els tests unitares es creen abans del propi codi. Si passen els tests es consideren que el codi esta complert. Els tests unitaris s'executen frequentment, 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 métodos així que qualsevol canvi que causi un error, es pugui identificar rapidament. 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.

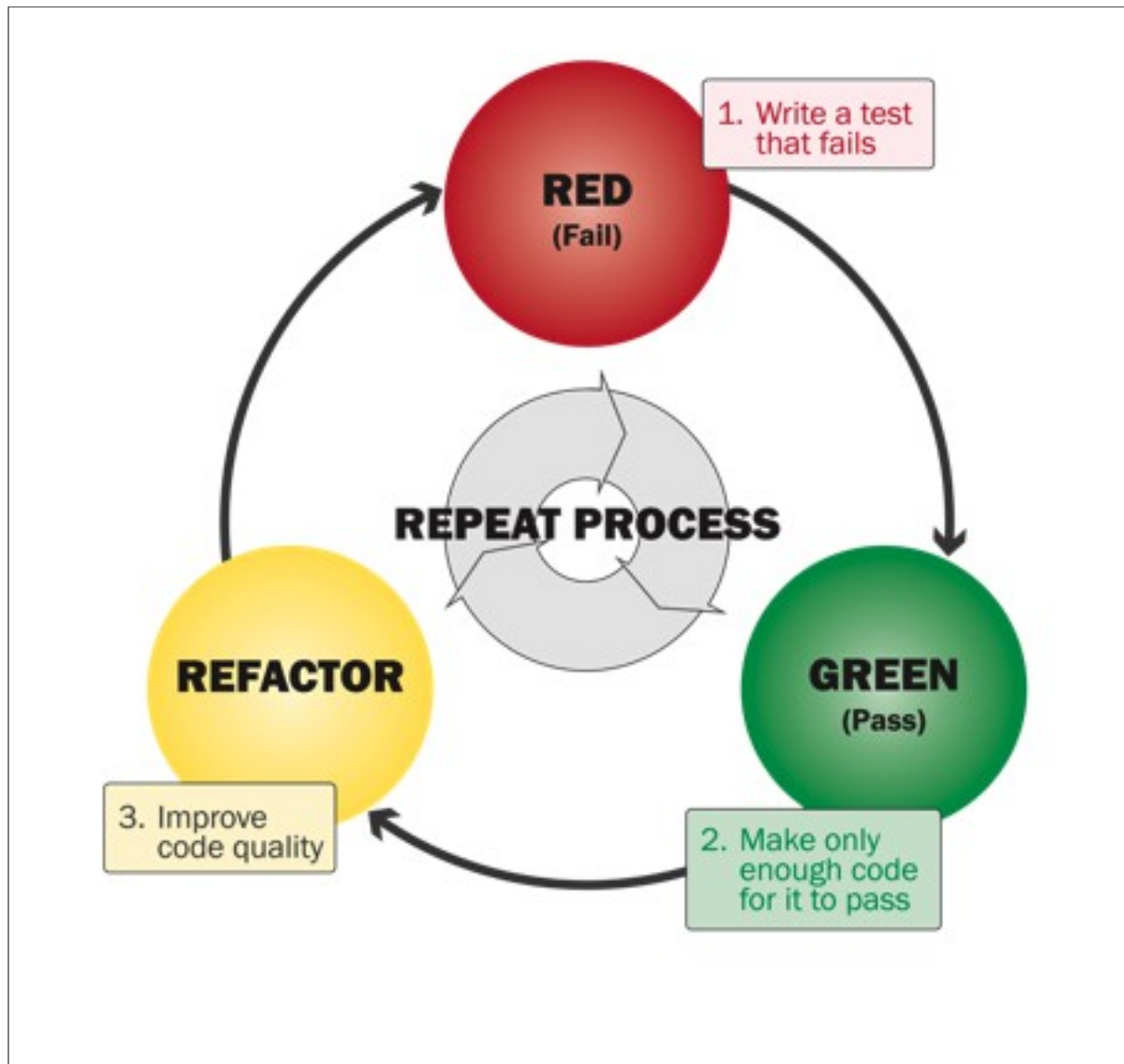
Documentació

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 basic de la interície del mòdul.

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

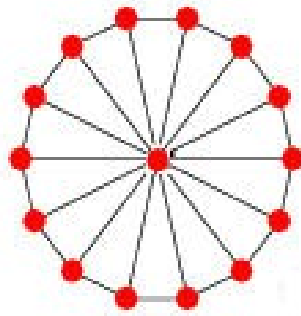
Disseny

Quan el software és desenvolupat utilitzant test-driven development, la cominació 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, metode 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.
- *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, relacionatsa amb llegir un graf desde un fitxer amb tres tests:
 - Si el fitxer no existeix la rutina retorna un error.
 - Si el fitxer existieix 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.

Coverage

El coverage és un indicador que mesura quin és le grau de “cobertura” dels teus tests. En el projecte s'ha executat la utilitat de coverage per demostrar que hi han utilitats de coverage per c, c++ i que és poden arribar a posar en el projecte.

En c,c++ hi ha una utilitat “gcov” que t'indica el coverage dels teus tests.

En la taula de sota és pot veure el coverage de la utilitat analitza, que serveix per processar resultats.

```
[root@b07bf77adda2 utestanalitza]# gcov analitza
File '../analitza/analitza.cpp'
Lines executed:69.59% of 148
Creating 'analitza.cpp.gcov'

File '/usr/include/c++/4.8.2/iostream'
Lines executed:100.00% of 1
Creatina 'iostream.acov'
```

Running coverage from analitza utility, podem veure que esta cobert en un 69.59%

El sistema podria evolucionar cap a un sistema més complert com la pàgina web de Networkx podeu veure on line si la darrera versió del software relacionat amb funcions de grafs estan passant els tests i quin coverage té.

NetworkX

[NetworkX Home](#) | [Documentation](#) | [Download](#) | [Developer \(Github\)](#)

Documentation

If you are new to NetworkX take a look at the [tutorial](#).

The [examples](#) show some simple and complicated ways to use NetworkX.

Full Documentation Versions

Documentation for NetworkX is hosted on [Read the Docs](#). “Stable” is the documentation for the most recent official release, while “Development” corresponds to the documentation currently in development. From that site, you may also select documentation for particular versions of NetworkX for versions 1.10 onward.

- [Stable](#)

Versions

Latest Release

1.10 – 2 August
[downloads](#) | [docs](#)

Development

2.0dev
[github](#) | [docs](#) |
build passing
coverage 93%

Pàgina web de NetworkX on és poden veure dos icones, en verd que indiquen



que els test estan ok i que el coverage és del 93%.

COVERALLS

HOME
FEATURES
PRICING
DOCS
BLOG
SIGN IN

NETWORKX / NETWORKX **93%**

BRANCH: MASTER GITHUB REPO CI PROJECT

LATEST BUILDS

BUILD	BRANCH	COVERAGE	COMMIT	COMMITTER	TYPE	TIME	VIA
#1871	master	93.36	Merge pull request #1722 from chrisnatali/master Added MultiLineString support in reading Graph from shapefile	dschult	push	about 16 hours ago	travis-ci
#1870	master	93.39	Update install documentation	MridulS	PULL #1753	about 23 hours ago	travis-ci
#1869	master	93.38	Update README.rst added installation	rvu95	PULL #1752	a day ago	travis-ci
#1868	master	93.38	Merge pull request #1747 from graal/master dag: add more information to exception message when c	dschult	push	26 Aug 2015	travis-ci

Pàgina web de NetworkX on es poden veure les darreres compilacions amb el percentatge del coverage. És pot veure que el percentatge de coverage del codi ha baixat.

Seria una bona ampliació del projecte generar informes amb la utilitat gcov i mostrar-los via web.

- Integració Continua
 - git
 - futur publicar el codi al git hub i que tothom pugui utilitzar el codi i fer aportacions.
 - vagrand
 - gprof
- Refactorització

Principis de la programació Orientada a Objectes

Object Oriented Bàsics

- Abstracció
- Encapsulació
- Polimorfisme

Design Principles

- Identificacr els aspectes de la teva aplicació que varien i separar-los de les coses que romandran iguals.
- Programa per una interfície, no per una implementació.
- Afavoreix composició per damunt de la herència.
- Esorçat per tenir “loosely coupled designs” entre els objectes que interactuen.
- Utilitza el “Open-Close Principle”. Les classes han d'estar obertes per extensió, pero tancades per modificació.
- “The Dependency Inversion Principles”, depen de les abstraccions, no de les classes concretes.
- Principi del mínim coneixment: “Parla només amb els teus amics immediats”
- The hollywood Principle: No ens cridis, ja et cridarem nosaltres.
- Single Responsibility: Una clase hauria de tenir només una raó per canviar.

Simplicitat

- Executat tots els tests
- Evita la duplicació
- Revela la intenció
- Poques classes i Metodes

SOLID Principles

Design Patterns

El digrama de classes

DIBUIXAR EL DIAGRAMA DE CLASSES DE C++

Representació matricial

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.

Posibles millores

Paralelització i cloud Computing

Base de dades de resultats de indicadors

Utilització de varis Indicadors simultàneament

Altres algoritmes alternatius al Simulated Aleanning i al Threshold acceptance

7. Els Resultats

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											
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

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

Betweenness Centrality amb Simulated Annealing

Betweenness Centrality amb Threshold Acceptance

Communicability Centrality amb Simulated Annealing

Communicability Centrality amb Threshold Acceptance

Communicability Betweenness Centrality amb Simulated Annealing

Communicability Betweenness Centrality amb Threshold Acceptance

8. **Conclusions**

Últim capítol (afegir abans tants capítols com es consideri necessaris).

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.

Isomorfismes:

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
 - Llibreria de Python amb funcions per grafs.
 - <https://networkx.github.io/>
 - Ernesto Estrada, introductor de la communicability
 - https://en.wikipedia.org/wiki/Ernesto_Estrada
 -