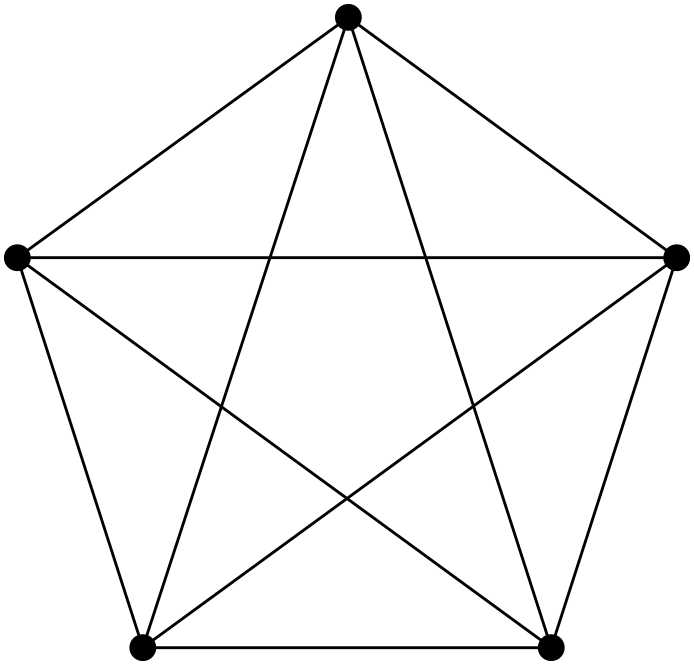
UNIVERSITAT POLITÈCNICA DE CATALUNYA

FIB

Projecte final de carrera - Enginyeria en Informàtica

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



Director: Francesc de Paula Comellas Padró
President: Josep Maria Brunat Blay
Ponent (Secretària): Montserrat Maureso Sànchez
Vocal: Àngel Olivé Duran

Estudiant: Juan Paz Sánchez

Para mi família

"Nunca es tarde si la dicha es buena"

Agradecimientos

Desde aquí quiero agradecer la gran ayuda y los ánimos que he recibido durante estos últimos meses dedicados a la realización del proyecto final de carrera.

En primer lugar a Francesc Comellas por haber confiado en mi para la realización de este proyecto final de carrera. Siempre ha estado dispuesto para resolver las dudas que iban surgiendo, dedicando para ello todo el tiempo necesario, con mucha paciencia y contagiando su interés por la investigación.

También doy gracias tanto a Josep María Brunat como a Montserrat Maureso por todas las facilidades que han puesto a mi disposición con los trámites administrativos. Josep María fue mi profesor de asignatura de matemática discreta y guardo muy buen recuerdo de sus clases. A Montserrat la he conocido al empezar el proyecto y, por lo que la he tratado, creo que tiene una manera de comunicarse con los demás que, como en el caso de Francesc, contagia su ilusión y sus ganas de hacer las cosas bien.

Mi familia, especialmente mi esposa y mi hija, pero también mis padres y mi hermana que en paz descanse, siempre han estado a mi lado y son los que, cada cual a su manera, han logrado motivarme para no tirar la toalla y seguir hasta el final. A ellos les dedico este trabajo.

Algunos buenos amigos, como Miguelón, Dani, Ferran, Eva, Noah, Gloria, Carlota, Álvaro y otros muchos, también han contribuido, de una forma u otra, a hacer que este periodo también tuviera momentos divertidos, agradables y relajados tan necesarios para renovar las ganas de aportar ese granito de arena a la ciencia.

Finalmente debo agradecer a todos mis compañeros de los departamentos ENTEL y MA4 de la UPC que me aprecian y me han dado ánimos para terminar la carrera.

Gracias a todos!

Index general

Capítol 1

Introducció

1.1 Presentació del problema

La branca de les matemàtiques que es dedica a l'estudi dels grafs és la matemàtica discreta. Es considera que la teoria de grafs va neixer arran de l'estudi que Leonard Euler (1707-1783) va fer del problema dels ponts de Königsberg, el qual consistia en trobar un circuit que passés un sol cop per cadascun dels set ponts de la ciutat. Els grafs són considerats actualment un model matemàtic molt important, tant per la seva simplicitat com per la quantitat creixent d'aplicacions que treuen profit del seu estudi teòric, com per exemple el disseny de programes, el modelat de xarxes de transport, xarxes socials, xarxes biològiques, disseny de circuits, etc. Es tracta doncs d'una àrea de coneixement on treballen no només matemàtics sinó també altres científics d'altres branques.

Amb l'arribada de Internet a les llars i 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.

Quan una xarxa es 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 por 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 fàcilment descompactable (per poder recuperar fàcil i ràpidament 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.

1.2 Objectius del projecte

L'objectiu principal d'aquest projecte és avaluar l'us d'una eina d'optimització combinatòria, coneguda com recuita simulada o *simulated annealing*, per reconstruir un graf a partir del grau d'intermediació o *betweenness centrality*, que és un dels invariants del graf. Existeixen estudis previs amb resultats satisfactoris que fan servir aquesta tècnica per la reconstrucció de grafs a partir d'un altre dels invariants com és l'espectre laplacià. La novetat introduïda en aquest project és, per tant, que hem fet servir el grau d'intermediació de cadascun dels vèrtexs del graf (o *vertex betweenness centrality*) per calcular la funció de cost que governa el procés d'optimització per recuita simulada. Aquesta mesura, com veurem més endavant, dóna molta informació del graf, ja que a més de l'ordre, 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 del grau d'intermediació dels seus vèrtexs. En concret, hem fet servir un graf aleatori, un petit mon, un invariant d'escala, un regular i un amb agrupament, tots ells amb el mateix ordre n=40, i per cadascun d'ells s'han realitzat 500 reconstruccions per disposar d'un nombre suficientment gran de grafs reconstruïts i fer una anàlisi estadística dels resultats.

Capítol 2

Conceptes teòrics

2.1 Teoria de grafs

2.1.1 Conceptes bàsics de grafs

Un *graf* G = (V,A) és una estructura combinatòria formada per un conjunt finit i no buit V=V(G) d'elements anomenats vèrtexs i un conjunt A=A(G) de parells no ordenats d'elements diferents de V anomenats arestes ó branques.

Si $a = \{u,v\}$ és una aresta, direm que *u* i *v* son vèrtexs adjacents.

Les relacions d'adjacència entre els vèrtexs d'un graf es veuen millor representant el graf mitjançant un dibuix on els vèrtexs son punts, i les arestes son línies que uneixen vèrtexs adjacents. Val a dir que la representació gràfica d'un graf és completament arbitraria, però la imatge representativa d'un graf és una ajuda per visualitzar i comprendre certes propietats dels grafs.

L'*ordre* d'un graf, $n=|V|$ és el nombre de vèrtexs del graf.

La *mida* d'un graf, $\rightarrow mida(G)=|A|$ és el nombre d'arestes del graf.

El *grau* del vèrtex *v* es denota per *d(v)* i és el nombre d'arestes incidents amb *v*.

Els vèrtexs de grau 0 s'anomenen *vèrtexs aïllats* i els de grau 1, *vèrtexs terminals*.

Un *recorregut de longitud t* és una seqüència de vèrtexs $r=v_0, v_1, \ldots, v_t$ tals que v_{i-1} i v_i són adjacents per tot *i* entre 1 i *t*.

Un *camí* de longitud t és un recorregut que no conté vèrtexs repetits.

Un *cicle* és un camí que comença i acaba al mateix vèrtex.

Un *senderó* de longitud t és un recorregut que no conté arestes repetides.

La *distància* entre una parella de vèrtexs *u* i *v* es denota per *d(u,v)* i és el nombre d'arestes que conté el camí més curt entre *u* i *v*.

El *diàmetre* d'un graf és el màxim entre les distàncies de totes les seves parelles de vèrtexs.

La *distància mitjana* ve donada per la fórmula $\frac{1}{n(n-1)}\sum_{u,v\in V} d(u,v)$

Un graf és *connex* si, per tot parell de vèrtexs *u* i *v*, hi ha un camí de *u* fins a *v*.

En aquest treball ens centrem únicament en grafs connexos

Un agrupament o *clúster* és un conjunt de vèrtexs entre els que hi ha una densitat

elevada d'arestes.

El *clustering* d'un vèrtex és la relació d'arestes que uneixen els veïns d'aquest vèrtex entre ells respecte al nombre total d'arestes possibles.

El *clustering* d'un graf és la mitjana del *clustering* dels seus vèrtexs.

2.1.2 Representacions de grafs

Podem representar els grafs de diverses maneres, entre elles:

1. Com el parell de conjunts finits *(V,A)* on els elements de *A* són de la forma *{u,v}*, amb *u* i *v* vèrtexs diferents de *V*.

2. Mitjançant un dibuix on els punts són els vèrtexs i les línies son les arestes.

El problema que tenen aquestes maneres de representar els grafs és que cap d'elles resulta adient per processar les seves característiques amb un ordinador. A continuació descrivim dues representacions d'un graf que permeten proporcionar tota la seva informació i que, per la seva estructura, si que resulten útils pel seu procés amb ordinadors.

La *matriu d'adjacència* d'un graf $G=(V,A)$ amb conjunt de vèrtexs $V=\{v_1, v_2, \ldots, v_n\}$ és la matriu quadrada $MA(G)=(a_{ij})$ de mida n x n, i definida de la manera següent:

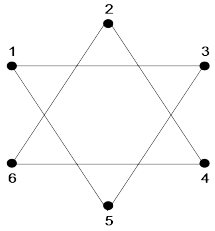
$$a_{ij}=\begin{cases}1, si\ \{v_i, v_j\}\in V\\ 0, altrament\end{cases}$$

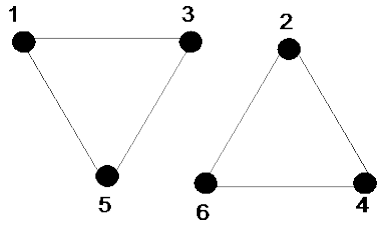
La matriu d'adjacència és simètrica i té elements nuls a la diagonal.

La *llista d'adjacències* és la llista dels veïns de cadascun dels vèrtexs del graf.

A les llistes d'adjacències només es guarda informació dels enllaços existents, la qual cosa permet estalviar espai d'emmagatzematge respecte a la matriu d'adjacència, on es guarda la informació de tots el possibles enllaços del graf.

A continuació mostrem dos representacions gràfiques del mateix graf seguides de les seves matriu d'adjacència i llista d'adjacències.





$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \qquad \begin{matrix} 2 & 4 & 6 \\ 1 & 3 & 5 \\ 2 & 4 & 6 \\ 1 & 3 & 5 \\ 2 & 4 & 6 \\ 1 & 3 & 5 \end{matrix}$$

2.1.3 Tipus de grafs

El *graf nul d'ordre n* es denota per N_n i té n vèrtexs i no té cap aresta. $N_n=(V,A), amb\,|V|=n, A=\emptyset \Rightarrow |A|=0$.

El *graf complet d'ordre n* es denota per K_n, té n vèrtexs i cada vèrtex és adjacent a tots els altres: $K_n=(V,A), amb\,|V|=n, |A|=\frac{n.(n-1)}{2}$.

En un graf *d-regular* tots els vèrtexs tenen grau d.

En un graf *fortament regular* el nombre de veïns comuns que tenen una parella de vèrtexs qualssevol només depèn de si aquests són adjacents entre ells o no.

El *graf camí de longitud n* es denota per P_n i és un graf amb *n* vèrtexs $V=\{v_0, v_1, ..., v_n\}, tals\,que\ \forall\, i,\ 1\le i\le n\ \{v_{i-1}, v_i\}\in A$ i no hi ha més adjacències.

Un graf és *planari* si pot ser dibuixat en el pla sense creuar cap aresta.

2.1.4 Isomorfisme

Dos grafs $G_1=(V_1, A_1)$ i $G_2=(V_2, A_2)$ són isomorfs, i es denota per $G_1\simeq G_2$, si existeix una aplicació bijectiva $f: V_1\to V_2$ tal que es conserven les adjacències, és a dir $\forall\, u, v\in V, u\sim v \Leftrightarrow f(u)\sim f(v)$.

Podem dir, d'una manera intuïtiva, que dos grafs són isomorfs si la seva representació gràfica pot ser la mateixa: canviant únicament l'etiquetatge dels vèrtexs de

G_1 s'obté G_2 . Els vèrtexs de dos grafs isomorfs es poden identificar un a un de manera que les adjacències de l'un i de l'altre siguin les mateixes, és a dir, l'estructura dels dos grafs, un cop identificats els vèrtexs corresponents, és idèntica.

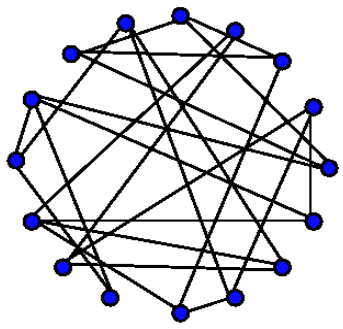
Un invariant per isomorfismes d'un graf G és un nombre o una seqüència de nombres associats a G que pren el mateix valor en tots els grafs isomorfs a G. Així, dos grafs isomorfs tenen el mateix nombre de vèrtexs (ordre) i el mateix nombre d'arestes (mida).

Un sistema complet d'invariants és un conjunt d'invariants que determina un graf llevat d'isomorfismes, és a dir, tots els grafs que tenen associats un mateix sistema complet d'invariants són isomorfs entre ells.

2.1.5 Models de grafs

Tal com s'ha dit anteriorment, els grafs es fan servir per modelar matemàticament diferents d'entorns. Per permetre'n adaptar i reproduir les característiques morfològiques s'han definit models de grafs a partir de paràmetres 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.

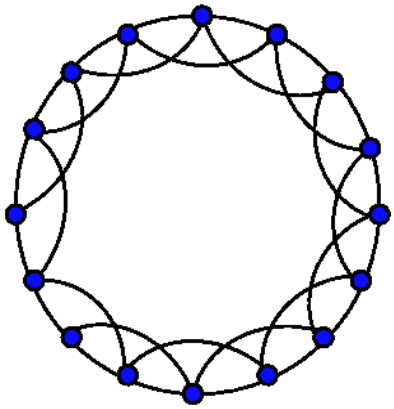
2.1.5.1 Aleatori



En els grafs aleatoris, com els que s'obtenen seguint el model Erdös-Rényi 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, la seqüència de graus segueix la distribució de Poisson amb un pic a $\bar{k}$. La probabilitat de trobar vèrtexs amb k arestes és negligible per $k \ll \bar{k}$ i $k \gg \bar{k}$.

Donat que hi ha connexions amb la mateixa probabilitat amb qualssevol dels nodes, les distàncies acostumen a ser petites. L'aleatorietat fa que dos vèrtex adjacents difícilment comparteixen veïns, per la qual cosa el clustering acostuma a ser baix.

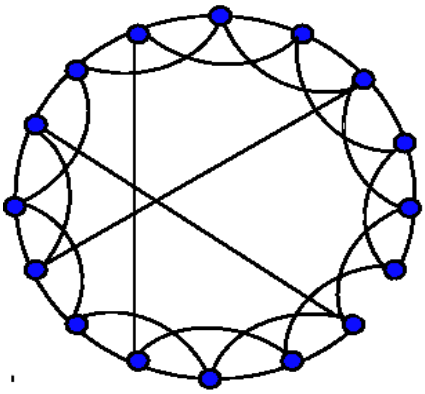
2.1.5.2 Regular (circultant)



En els grafs regulars, tots els vèrtexs tenen el mateix grau. Hi ha grafs regulars amb enllaços aleatoris. En aquest cas 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. 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.

2.1.5.3 Petit mon o *small world*



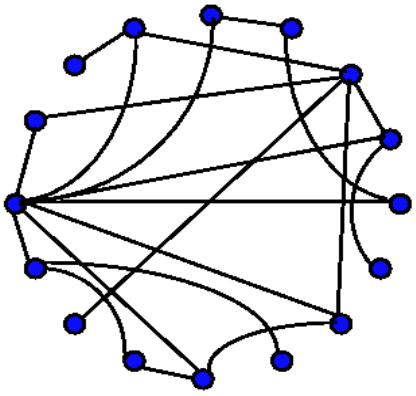
Els grafs petit mon Watts i Strogatz són una modificació dels grafs circulants descrits abans, en els que s'introdueixen alguns enllaços aleatoris. Watts i Strogatz suggereixen el següent mètode per a la construcció de grafs amb les propietats petit mon. Per comoditat de notació en la explicació, etiquetem els vèrtexs del graf de *0* fins a *n-1*, on n és l'ordre del graf, i considerem que el número *i* corresponent a l'etiqueta del vèrtex és *i mòdul n*.

El procediment comença amb un graf circulant $C_{n,\Delta}$, a continuació es tria el vèrtex *0* i la branca adjacent al vèrtex *1*. Amb probabilitat p es reconnecta aquesta aresta a un vèrtex triat a l'atzar entre tots els vèrtex, evitant la duplicació d'arestes. Altrament no es canvia la branca. El procés es repeteix successivament per a tots els vèrtexs des del *0* fins a l'*n-1*, completant una volta. A continuació es realitza el mateix procediment per a les arestes que reconnecten el vèrtex *i* amb amb el vèrtex *i+2*, entenent que tenim un etiquetatge dels vèrtex que va del *0* a l'*n-1*, és a dir que agafem el número del vèrtex mòdul n. Com a la volta anterior, es connecta aleatòriament cadascuna d'aquestes arestes, també amb probabilitat p, i es continua el procés, circulant la llarg de l'anell i seleccionant a

cada volta una aresta més llunyana $(i+2, i+3, \ldots, i+\Delta/2)$, fins que cada branca del graf original $C_{n,\Delta}$ hagi estat considerada una vegada. Així, el procés de connexió s'atura després de $\Delta/2$ voltes. Aplicant aquest procediment, per a *p=0* el graf no queda modificat, mentre que per a *p=1* totes les branques es reconnecten de manera aleatòria, i en aquest cas s'obté un graf aleatori que té una distribució de graus de Poisson. Per a valors intermitgos de *p* s'obtenen diferents tipus de grafs. Amb $p \approx 0{,}01$ s'obtenen grafs petit mon amb un coeficient d'agrupament gran, pròxim al del graf inicial, i amb diàmetre i distancia mitjana similars als valors que tenen els grafs 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 son freqüents en sistemes complexos com la xarxa telefònica o Internet.

2.1.5.4 Amb invariancia d'escala (*scale-free)*



En els grafs invariants d'escala, 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. La distribució dels graus P(k) és proporcional a k^{-b}, on k és el nombre de connexions i b una constant. Aquest model també és important a la pràctica perquè reprodueix 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 número 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 altre manera, els vèrtexs de més grau capten més enllaços.

El procediment comença amb un graf inicial de $m_0 \geq 2$ vèrtexs, cadascun dels quals

té almenys grau 1. Els nous vèrtexs s'afegeixen al graf d'un en un. Cada nou vèrtex es connecta amb $m < m_0$ vèrtexs preexistents. La probabilitat que el nou vèrtex es connecti al vèrtex existent i depèn del seu grau k_i : $P(k_i) = \frac{k_i}{\sum_{j \in V} k_j}$

Els grafs generats així tenen les següents característiques:

- La distribució de graus no canvia amb més iteracions i queda descrita per la llei potencial $P(k) \sim k^{-3}$
- La distància mitjana creix logarítmicament amb l'ordre del graf.
- Encara que no hi ha resultats analítics del coeficients de *clustering* del model Barabasi Albert, els mesurats empíricament son generalment molt més elevats que en les xarxes aleatòries.

2.2 Grau d'intermediació o *betweenness centrality*

La teoria de grafs ofereix diverses mesures de la centralitat d'un vèrtex per determinar la seva importància relativa en el graf. La que hem escollit en aquest estudi és el grau d'intermediació o *vertex betweenness centrality*. Informalment, el grau d'intermediació d'un vèrtex *v* és una mesura del nombre de camins curts que passen per *v*.

Abans de definir rigurosament la *betweenness* cal introduir un parell de conceptes:

Un *camí geodèsic* entre la parella de vèrtexs *u* i *v* és un camí curt que té com a longitud exactament la distància entre *u* i *v*. El nombre de camins geodèsics entre dos vèrtexs u i v es denota per σ_{uv} .

Observació: En els grafs connexos, per tota parella de vèrtex sempre hi ha com a mínim un camí geodèsic. De fet hi ha molts grafs en els que el nombre de camins geodèsics entre dos vèrtex és més gran que 1: $\sigma_{uv} \geq 1$

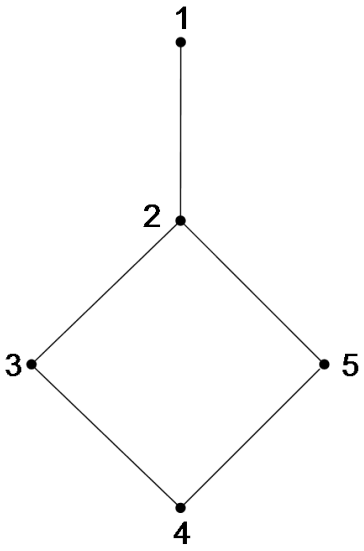
El nombre de camins geodèsics entre dos vèrtexs *u* i *v* que passen pel vèrtex *w* es denota per $\sigma_{uv}(w)$.

Observació: en els grafs connexos, per tota parella de vèrtexs no sempre hi ha un camí geodèsic que passi per un tercer vèrtex determinat. El nombre de camins geodèsics entre *u* i *v* que passa per *w* és, com a molt, el nombre de camins geodèsics que hi han entre *u* i *v:* $0 \leq \sigma_{uv}(w) \leq \sigma_{uv}$.

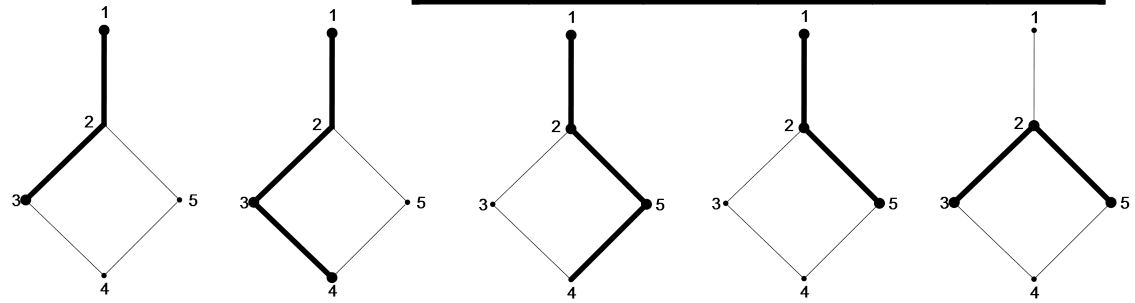
El *grau d'intermediació* o *vertex betweenness centrality* del vèrtex *w*, es denota per $BC(w)$, i és la suma, per a tota parella de vèrtexs *u* i *v* diferents de *w*, dels quocients entre el número de camins geodèsics de tots els parells de vèrtexs *u* i *v* que passen pel vèrtex *w* i el número de camins geodèsics de tots els parells de vèrtexs *u* i *v* (que no tenen perquè passar per *w*). Per normalitzar aquest resultat, l'hem de dividir per *(n-1)(n-2)*, que és el número de parelles de vèrtexs que hi ha en un graf d'ordre *n* en el que no es considera el vèrtex *w*.

$$\overline{BC(w)} = \frac{1}{(n-1)(n-2)} \sum_{u \neq v \neq k \in V} \frac{\sigma_{uv}(w)}{\sigma_{uv}}$$

Aquí tenim una taula en la que resumim els càlculs que s'han de fer per mesurar el grau d'intermediació dels vèrtexs del graf que es mostra a continuació. Es tracta d'un exemple senzill en el que es pot veure la complexitat d'aquest càlcul. Per tal de visualitzar quins son els camins geodèsics que passen per un vèrtex, també hem inclòs una sèrie de representacions del mateix graf en les que assenyalem amb una traçada més gruixuda els camins geodèsics que passen pel vèrtex 2.



	Vèrtex 1	Vèrtex 2	Vèrtex 3	Vèrtex 4	Vèrtex 5	Total
Camí 12	0	0	0	0	0	**0**
Camí 13	0	1/1	0	0	0	**1**
Camí 14	0	2/2	1/2	0	1/2	**2**
Camí 15	0	1/1	0	0	0	**1**
Camí 21	0	0	0	0	0	**0**
Camí 23	0	0	0	0	0	**0**
Camí 24	0	0	1/2	0	1/2	**2**
Camí 25	0	0	0	0	0	**0**
Camí 31	0	1/1	0	0	0	**1**
Camí 32	0	0	0	0	0	**0**
Camí 34	0	0	0	0	0	**0**
Camí 35	0	1/2	0	1/2	0	**2**
Camí 41	0	2/2	1/2	0	1/2	**2**
Camí 42	0	0	1/2	0	1/2	**2**
Camí 43	0	0	0	0	0	**0**
Camí 45	0	0	0	0	0	**0**
Camí 51	0	1/1	0	0	0	**1**
Camí 52	0	0	0	0	0	**0**
Camí 53	0	1/2	0	1/2	0	**2**
Camí 54	0	0	0	0	0	**0**
Total	**0**	**7**	**2**	**1**	**2**	
Total	**0**	**0,58333**	**0,16666**	**0,08333**	**0,16666**	



2.3 Optimització per recuita simulada o *simulated annealing*

La *optimització per recuita simulada* és un meta-algorisme probabilístic genèric que es fa servir a problemes d'optimització global, on el que es pretén és trobar una bona aproximació a l'òptim global d'una funció en un espai de cerca gran. Aquesta idea va ser presentada independentment l'any 1983 per S. Kirkpatrick, C. D. Gelatt y M. P. Vecchi i l'any 1985 per V. Černý.

El nom que inspira aquest procediment prové del procés metal·lúrgic de recuita (*annealing* en anglès), que és una tècnica que consisteix en escalfar i refredar de manera controlada un material per augmentar-ne la mida dels cristalls y d'aquesta manera reduir-ne els defectes. L'escalfament fa que els àtoms tinguin més energia, s'agitin més, surtin de les seves posicions inicials (que són mínims locals d'energia) i es moguin aleatòriament. El refredament redueix la energia dels àtoms. El fet de refredar lentament el material ofereix més probabilitats als àtoms de trobar configuracions amb menys energia interior que l'inicial.

Per analogia amb aquest procés, a cada pas de l'algorisme d'optimització per recuita simulada es reemplaça la solució actual per una solució propera. En el cas que ens ocupa, una solució és un graf G i una solució propera és una altre graf G' que és el resultat de fer una modificació a G. Típicament, G' és el resultat de desconnectar totes les arestes incidents amb un vèrtex de G i tornar a connectar aquest mateix vèrtex a una nova llista de vèrtex). La solució propera es tria amb una probabilitat que depèn de la diferència entre els valors d'una funció de cost i d'un paràmetre global T que decreix gradualment durant el procés. La funció de cost correspon a la energia del procés metal·lúrgic, mentre que el paràmetre T correspon a la temperatura. La probabilitat que la solució actual canvii aleatòriament és menor quan T és més gran i va augmentant a mida que T minva cap a 0. El gran avantatge que ofereix aquest mecanisme, comparant-lo amb altres algorismes voraços, consisteix en el fet que la optimització per recuita simulada aleatòriament pot acceptar com a possible solució un resultat que no redueix el valor de la funció de cost. D'aquesta manera s'evita que el procés de cerca es quedi atrapat en un mínim local, que és el que acostuma a passar en molts algorismes de cerca voraç.

Veiem com funciona de la optimització per recuita en general. El procés comença en un estat inicial triat a l'atzar, que té una determinada energia. La iteració bàsica consisteix en fer un número determinat de passos. A cada pas considerem un veí de l'estat actual, i decidim probabilísticament si deixem el sistema a l'estat *s*, que té associada una energia *E(s)* o el passem a l'estat *s'*, que té associada una energia *E(s')*. Les probabilitats de canvi d'estat són tals que el sistema tendeix finalment a estats amb menor energia. Un cop s'han fet tots els passos, baixem la temperatura i tornem a iterar. Aquestes iteracions generalment es fan fins que s'arriba a un estat suficientment bo per l'aplicació, o la temperatura ha assolit la seva fita inferior o fins que es compleixi un determinat temps de còmput.

El que ens cal és 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.

- Estat: es tracta d'un concepte molt utilitzat en informàtica. Es fa servir per descriure d'alguna manera les característiques d'un sistema. L'estat d'un sistema determina quines accions es poden realitzar. Aquestes accions portaran al sistema a un altre estat fent una transició. Els grafs es fan servir sovint per descriure diagrames d'estats, on els vèrtexs representen el estats i les arestes representen les transicions. El el context del procés d'optimització per recuita simulada que considerem aquí, un estat és un graf. L'estat inicial és doncs un graf inicial. En el cas que ens ocupa, aquest graf inicial és un graf qualsevol, sinó que ha de ser un graf d'un ordre determinat: l'ordre del graf que es vol reconstruir.
- L'energia és el paràmetre que es vol minimitzar en la optimització per recuita simulada. La energia és una característica que es pot mesurar en un estat. El que necessitem doncs és disposar d'una manera de mesurar quant lluny estem del nostre objectiu. Per això definirem una funció de cost que ha de ser una funció que retorni un valor indicatiu a partir del qual determinem si estem més o menys aprop de la solució. Ara ens queda més clar que l'objectiu del procés d'optimització és minimitzar el valor que retorna la funció de cost. Haurem de definir una funció que d'alguna manera

compara el graf que tenim amb el graf que volem generar. Aquí és on intervé el grau d'intermediació dels vèrtex, que és precisament el que coneixem del graf que volem reconstruir i que també podem calcular en el graf que tenim en qualssevol estat.

- Veïnat d'un estat: els veïns de cada estat depenen del problema concret. Usualment un estat veí S' és el resultat d'una petita modificació de l'estat S. Aquí el veí d'un estat és un graf G' que és el resultat d'una petita modificació del graf G. Habitualment aquesta petita modificació és un canvi d'aresta o la reconnexió d'un vèrtex. En el nostre cas, obtenim el graf G' reconnectant un vèrtex de G. La reconnexió d'un vèrtex v consisteix en treure totes les arestes del graf que incideixen amb v, triar un nou grau g i connectar el vèrtex v amb g vèrtexs que també triem aleatòriament. Aquesta transformació de G en G' provoca un canvi d'estat i per tant també comporta un canvi d'energia.

- Probabilitat de canvi d'estat: la probabilitat de fer una transició de l'estat actual s al nou estat s' és una funció P que depèn de dos paràmetres: la diferència d'energies entre l'estat s i l'estat s' i la temperatura. Sembla raonable que quan l'estat s' tingui menys energia que l'estat s, el sistema faci la transició al nou estat. Efectivament. si la transició disminueix la energia, el moviment es acceptat amb probabilitat $P=1$. Però, què passa si la energia del nou estat és més gran que la de l'estat actual? Una qualitat important del mètode és que la probabilitat de transició P és sempre més gran que zero, encara que la diferència d'energia entre l'estat actual i el nou estat sigui positiva. Així, el sistema pot passar a un estat de més energia que l'estat actual que, tal com està pensat el mecanisme, ofereix una solució pitjor que la que dona l'estat actual. Aquesta qualitat és precisament la que fa que el sistema no es quedi atrapat en un òptim local. Abans de veure quina és la incidència de la temperatura en les decisions de canvi d'estat o transicions, hem de conèixer quin és el seu rol en el procés metal·lúrgic i com fem que aquest paràmetre tingui el mateix rol en la optimització per recuita simulada.

- Temperatura: la temperatura en el procés metal·lúrgic és responsable directe de la energia que tenen els àtoms. A més temperatura, més energia. Amb més energia, els àtoms tenen més capacitat de moviment i per tant el material pot experimentar canvis més fàcilment. Per contra, quan els àtoms disposen de menys energia, la seva

mobilitat es redueix i conseqüentment el material resulta menys alterable. Els canvis en el material corresponen a les transicions del procés simulat. Quan la temperatura és alta, la probabilitat que es produeixin canvis és elevada, mentre que quan la temperatura es redueix, la probabilitat de transició minva cap a 0 asimptòticament (és a dir que mai no arriba a ser 0). El procediment de simulació involucra tant la temperatura com la diferència d'energia positiva en la decisió de fer una transició cap a un estat menys favorable. El que fa és calcular una probabilitat de canvi, és a dir un valor entre 0 i 1 que denotem per p_c , mitjançant una funció que fa servir la temperatura T i la diferència d'energia Δ: $p_c = e^{-\Delta/T}$. Si p_c és més gran que un valor aleatori entre 0 i 1, es produeix la transició. D'aquesta manera, a mesura que la temperatura baixa, probabilitat d'acceptar transicions a estats amb més energia redueix.

- La velocitat de refredament: la temperatura disminueix gradualment segons avança el procés. En el procés metal·lúrgic on es treballa a temperatures molt elevades, el refredament s'aconsegueix deixant d'escalfar o escalfant menys del necessari per mantenir la temperatura. En el cas d'un procediment informàtic hi ha moltes maneres de reproduir aquesta disminució de la temperatura, sent el més usual fer servir una funció exponencial, de manera que T disminueix en un factor $k < 1$ a cada iteració.

2.4 Comparació de grafs

Ja hem definit que per determinar si dos grafs G_1 i G_2 són isomorfs hem de trobar una aplicació biunívoca entre els vèrtexs de G_1 i els vèrtexs de G_2. La força bruta és un mecanisme inviable ja que, hi ha n! permutacions possibles, raó per la qual el problema esdevé intractable amb aquesta tècnica, inclús amb grafs relativament petits.

Actualment encara no s'ha descobert cap solució eficient per determinar si dos grafs són isomorfs. Schmidt i Duffel van descriure el 1976 un algorisme que en molts casos permet obtenir el resultat correcte amb un rendiment acceptable, encara que en casos especials de grafs no isomorfs, el rendiment continua sent dolent. Aquesta aproximació aprofita el fet que per ser isomorfs, dos grafs han de tenir la mateixa distribució de graus. El seu funcionament consisteix en classificar els vèrtexs mitjançant les matrius de distàncies i els graus dels

vèrtexs. A partir d'aquí comença a mirar de trobar una correspondència entre els dos grafs. El procediment acaba quan arriba a un dels dos estats següents: o bé troba una correspondència entre els dos grafs (és a dir que els dos grafs són isomorfs i no cal seguir buscant) o bé quan ja ha provat totes les possibles correspondències sense trobar-ne cap que demostri que els dos grafs són isomorfs.

Encara que l'objectiu és que el graf reconstruït $G_{Reconstruït}$ sigui isomorf al graf que volem obtenir $G_{objectiu}$, ens és de gran utilitat disposar d'algun procediment per a mesurar la similitud entre dos grafs. En aquest sentit, entenem que dos grafs son semblants evidentment quan són isomorfs, però també quan, malgrat no ser isomorfs, tenen topologies properes. Per això hem fet servir un procediment que mesura la similitud de dos grafs mitjançant els valors i vectors singulars de la descomposició de les matrius d'adjacència dels grafs comparats. Ipsen i Mikhailov van proposar aquest mètode l'any 2002 basant-se en el fet que els autovalors de les diferents matrius associades a un graf contenen informació sobre la seva estructura.

Repassem els conceptes necessaris per explicar el mètode per comparar grafs.

El *valor propi* (o *autovalor*) d'una matriu quadrada A de n x n sobre un cos K és un escalar $\lambda \in K$ tal que existeix un vector no nul $v \in K^n$ que compleix $Av = \lambda v$. Tot vector que satisfà aquesta relació s'anomena *vector propi* (o autovector) de la matriu A pertanyent al valor propi λ.

Son equivalents les següents afirmacions:

- $\lambda \in K$ és un valor propi de A,
- la matriu $M = \lambda I - A$ és singular, és a dir, $|M| = 0$,
- λ és una arrel del polinomi característic de A.

Segons el teorema de la descomposició en valors singulars, una matriu A es pot descompondre en dues matrius, U i V i una matriu diagonal Σ de valors singulars, tals que es compleix $A = U \Sigma V^T$ i $\Sigma = U^T A V$. Per comparar dos grafs G_1 i G_2 amb matrius d'adjacència A_1 i A_2, podem definir la matriu F en funció de A_1 i de A_2 de la manera següent: $F = F(A_1, A_2) = U_1 \Sigma_2 V_1 T = U_1 U_2^T A_2 V_2 V_1^T$, que es construeix a partir dels vectors singulars de G_1 i G_2 . Si els dos grafs són isomorfs i les seves matrius d'adjacència difereixen només per l'etiquetatge dels seus vèrtexs, aleshores tindrem que $A_1 = F = F(A_1, A_2)$; mentre que si els dos grafs no són isomorfs, les components de F

prendran valors reals més propers als valors de A_1 quant més semblants siguin els dos grafs. D'aquesta manera definim $\Delta = A_1 - F$ i, per mesurar la semblança entre dos grafs, usarem la norma $\delta = \sqrt{\sum_{i,j} \Delta_{ij}^2 / n}$. Es pot comprovar fàcilment que, si els grafs comparats són isomorfs, llavors $F = A_1$, $\Delta_{ij} = a_{ij} - f_{ij} = 0$, i per tant $\delta = 0$.

Capítol 3

Anàlisi

3.1 Anàlisi de requeriments

Abans d'implementar una aplicació per avaluar l'ús de la mesura del grau d'intermediació dels vèrtexs en el procés de reconstrucció de grafs mitjançant la optimització per recuita simulada, hem analitzat quins són els requeriments, és a dir, quines son les capacitats i les condicions que ha de complir el sistema.

3.1.1 Fiabilitat

La fiabilitat dels resultats és, potser, el requeriment més important. Es tracta de veure quin és el nivell de reconstrucció que es pot esperar de la optimització per recuita simulada fent servir la vèrtex betweenness centrality. No volem obtenir uns resultats només per treure'n unes conclusions. El que volem és disposar d'un sistema que implementi la optimització per recuita simulada. Encara que hi al altres mètodes de optimització combinatòria que també es poden fer servir pel mateix objectiu, aquests no són els que volem fer servir aquí. Tanmateix, el paràmetre que volem fer servir com a base per la reconstrucció és el grau d'intermediació dels vèrtexs d'un graf. Es poden fer servir moltes altres mesures, però tampoc són l'objecte del nostre estudi. El principal requeriment que hem perseguit en aquest projecte és oferir uns resultats fiables per tal que les conclusions que se n'extreguin siguin perfectament vàlides.

3.1.2 Rendiment

El rendiment del sistema també és un requeriment important. Abans d'arribar a qualssevol conclusió, s'han de fer moltes proves. Encara que amb unes poques proves es poden treure unes conclusions vàlides, també és cert que la seva validesa es pot veure reforçada per un conjunt de proves tant intens (en el sentit de fer moltes reconstruccions) com extens (en el sentit de fer reconstruccions de diferents grafs). En aquest sentit, disposar d'un sistema que fa les simulacions molt de pressa permet donar un millor suport a les conclusions a les que s'arribi. La velocitat de les simulacions no depèn únicament de la velocitat de procés de la CPU de l'equip en el que es realitzen les simulacions. La eficiència del codi també es un factor molt important. En aquest sentit, és cert que la tasca del compilador pot tenir una incidència determinant. Un compilador que no treu partit de l'arquitectura de l'equip en el que s'executarà el programa que genera pot no ser la millor opció. En aquest sentit, el maquinari i

el programari escollits per l'entorn de simulació també poden jugar papers de gran rellevància a l'hora d'avaluar el seu rendiment durant l'execució de les simulacions.

3.1.3 Requeriments funcionals

Quant als requeriments funcionals, es vol que el sistema permeti incorporar fàcilment altres funcionalitats. Hi ha variants del procediment d'optimització per recuita simulada que es poden fer servir per reconstruir grafs. Des de variacions de la funció de cost fins a la optimització per cerca tabú, passant per la selecció arbitrària del veí a cada pas de la recuita. Es pretén que el sistema sigui fàcilment modificable i serveixi d'eina de optimització combinatòria bàsica per avaluar altres mètodes de reconstrucció de grafs que facin servir alguna variant del mecanisme que fem servir en aquest estudi.

3.2. Funcionament del sistema

El sistema funciona de la manera següent. El primer paràmetre del programa pot ser o bé una llista d'adjacències del graf que es vol reconstruir o bé la llista de valors corresponents al grau d'intermediació dels seus vèrtexs .

- Quan al programa li arriba un graf en forma de llista d'adjacències, el primer que fa és crear una representació interna del graf d'entrada i generar un fitxer de sortida amb la llista d'adjacències del graf que el programa ha interpretat. Fet això, ja disposem de l'ordre del graf i calculem la llista de graus d'intermediació dels seus vèrtexs.
- Si el que li arriba ja és una llista de graus d'intermediació, el que fa és calcular l'ordre del graf.

A partir d'aquí el funcionament és igual, sigui quin sigui el tipus d'entrada del programa.

A més de la informació relativa al graf que es pretén reconstruir, el programa pot rebre de manera opcional una sèrie de valors que permeten caracteritzar el procés de optimització per recuita simulada.

- Els quatre paràmetres que defineixen el comportament de la optimització per recuita simulada:

- ✔ la temperatura inicial (T_0)
- ✔ la temperatura mínima (T_{min})
- ✔ el número de combinacions que s'han de fer a cada pas (N_{max})
- ✔ la tassa de refredament (k)

- Els tres números enters que faran de llavors del generador de números aleatoris

Quan el programa no reb algun d'aquests paràmetres, fa servir uns valors prefixats, que seran sempre els mateixos.

Denotarem per LGI la Llista de Graus d'Intermediació dels vèrtexs d'un graf. L'ordre del graf que hem de reconstruir l'obtenim gràcies a la llista d'adjacències o la LGI rebudes com a paràmetres. El primer que fem és generar aleatòriament un graf del mateix ordre que el graf que es vol reconstruir i en calculem la LGI. Com que al principi és l'únic que tenim, la LGI d'aquest primer graf també és la LGI_millor i també és la LGI_actual. La funció de cost ens dóna una idea de quan diferents son la LGI_actual i la LGI_objectiu. En aquest cas, la funció de cost és la suma dels quadrats de la diferència de valors entre la LGI_actual i la LGI_objectiu. La finalitat del procediment de optimització per recuita simulada és trobar un graf amb una LGI que la funció de cost no pugui diferenciar la LGI_objectiu. És per això que sempre guardarem el cost_millor i el graf_millor.

A partir d'aquí és quan comencem a fer les iteracions i els passos de la optimització per recuita.

Fer

Repetir Nmax vegades

Modificar el graf_actual, per obtenir un nou graf_actual

Calcular LGI_actual i cost_actual

Si cost_actual < cost_millor aleshores

el graf_millor serà el graf_actual

el cost_millor serà el cost_actual

Si cost_actual = 0 aleshores

hem acabat

Altrament

Continuar amb el graf_actual

Fi si

Altrament

Si $e^{-\Delta/T}$ > número_aleatori *aleshores*

Continuar amb el graf_actual

Altrament

el graf_actual serà el millor_graf

Continuar amb el graf_actual

Fi Si

Fi si

Fi Repetir

Baixem la temperatura

Mentre la temperatura sigui més gran que Tmin

3.3 Descripció de les funcions rellevants

En el sistema que acabem de descriure hi ha funcions que cal mirar una mica més de prop per entendre com funciona.

3.3.1 La matriu de distàncies

Una de les necessitats que hem tingut a l'hora de calcular un número de camins i la connectivitat d'un graf era conèixer la distància entre dos vèrtexs. Per a tals efectes hem implementat una matriu de distàncies.

La *matriu de distàncies* d'un graf $G=(V,A)$ amb conjunt de vèrtexs $V=\{v_1, v_2, \ldots, v_n\}$ és la matriu quadrada $MD(G)=(a_{ij})$ de mida n x n, tals que $a_{ij}=d(v_i, v_j)$.

La matriu de distàncies, com la matriu d'adjacències, és simètrica i té elements nuls a la diagonal.

Per generar aquesta matriu hem fet servir una implementació molt intuïtiva. Aprofitant la simetria de la matriu, els recorreguts sempre els fem únicament d'una de les meitats de la matriu. Primer omplim totes les posicions de la matriu amb un valor que no es pot donar per qualssevol parella de vèrtexs d'un graf connex: n, que és l'ordre del graf. Donat que la distància entre una parella de vèrtexs qualssevol en un graf d'ordre n és, com a molt, n-1, si en una posició de la matriu de distàncies trobem un valor superior a aquesta fita màxima, vol dir que no hi ha cap camí que uneixi els dos vèrtexs. Seguidament posem zeros a la diagonal (la distància d'un vèrtex a ell mateix és nul·la). Continuem posant la distància als veïns, que és 1. Finalment fent una repassada de les distancies basada en el fet que la distancia entre dos vèrtexs *u* i *v* ha de ser inferior o igual a la distancia entre *u* i un tercer vèrtex *w* més la distància entre *w* i *v*: $d(u,v) \leq d(u,w)+d(w,u)$ (desigualtat triangular). Mentre en aquesta repassada fem alguna modificació, haurem de repetir el procés.

Posar a n tots els elements

Posar a 0 els elements de la diagonal

Posar a 1 els veïns de cada vèrtex

Fer

Si *d(u,v)>d(u,w)+d(w,v)* *aleshores*

```
d(u,v)=d(u,w)+d(w,v)
```

Fi Si

Mentre fem modificacions

El cost d'aquest algorisme és cúbic (n^3) sobre l'ordre del graf.

3.3.2 El grau d'intermediació

Segons hem vist anteriorment, el valor del grau d'intermediació d'un vèrtex *w* és la suma, per tota parella de vèrtexs *u* i *v* diferents del vèrtex *w*, dels quocients entre el número de camins geodèsics entre totes les parelles de vèrtexs *u* i *v* que passen pel vèrtex *w,* que denotem per $\sigma_{uv}(w)$, i el número de camins geodèsics entre totes les parelles de vèrtexs *u* i *v,* que denotem per σ_{uv} . Aquest resultat, per estar normalitzat l'hem de dividir per *(n-1)(n-2)*.

Hem implementat dos procediments completament diferents per calcular el grau d'intermediació dels vèrtexs d'un graf. Encara que els dos procediments obtenen el mateix resultat, la primera versió és la que resulta de la implementació de la definició, mentre que la segona és una implementació molt més eficient donat que és la que el 2001 va proposar Ulrik Brandes.

3.3.2.1 Implementació de la definició

Aquesta implementació és la primera que vam fer i en la que vam codificar els conceptes necessaris pel càlcul dels camins geodèsics.

La definició de grau d'intermediació d'un vèrtex que hem implementat és la que correspon a la versió normalitzada: $\overline{C_B(w)}=\frac{1}{(n-1)(n-2)}\sum_{u\neq v\neq k\in V}\frac{\sigma_{uv}(w)}{\sigma_{uv}}$

Per calcular el número de camins geodèsics entre una parella de vèrtexs *u* i *v* fem servir un procediment recursiu (també anomenat *backtracking*), fitat per la distància entre els dos vèrtexs. Els fonaments de la recursivitat són els mateixos que es fan servir a les

demostracions per inducció. Es tracta d'anar reduint la longitud del camí geodèsic a analitzar fins que els vèrtexs *u* i *v* siguin veïns. En aquestes condicions, el nombre de camins geodèsics és 0. Per reduir la longitud del camí, cerquem tots els veïns u_i de *u* que poden formar part d'algun dels camins geodèsics entre *u* i *v*. Això ho fem comprovant, per cadascun dels veïns de *u*, que $d(u,v)=1+d(u_i,v)$. El nombre de camins geodèsics entre *u* i *v* és doncs la suma dels camins geodèsics entre cadascun dels u_i que satisfan la condició anterior i *v*.

Per calcular el número de camins geodèsics entre *u* i *v* que passen per un tercer vèrtex *w* fem servir una estratègia similar a la que acabem de descriure, i que consisteix en sumar per una banda el número de camins geodèsics entre *u* i *w* i per altra banda el número de camins geodèsics entre *w* i *v*.

Un cop aclarits aquests dos càlculs, és molt fàcil veure com funciona la resta. Per evitar unes quantes iteracions, guardem resultats parcials en vectors auxiliars. Quan diem afegir, ens referim a que augmentem una variable (inicialitzada a 0 al principi del procés) amb el valor descrit.

Calcular el nombre de camins geodèsics entre totes les parelles de vèrtexs del graf.

Calcular, per cadascun dels vèrtexs del graf, el nombre de camins geodèsics entre totes les parelles de vèrtexs que passen per ell.

Repetir per tots els vèrtexs v_i

Repetir per tots els camins c_{jk}

afegir el nombre de camins geodèsics `entre j i k` *que passen per* v_i *dividit pel nombre de camins geodèsics entre j i k*

Fi repetir

Fi repetir

3.3.2.2 Implementació eficient

La segona versió que hem implementat és la que correspon a la implementació de l'algorisme pel càlcul ràpid de la betweenness que l'any 2001 va publicar en Ulrik Brandes al *Journal of Mathematical Sociology* 25 i que permet fet el mateix càlcul i obtenir els mateixos resultats, però en una fracció del temps. Aquesta implementació és el resultat d'un procés d'optimització de codi i la seva anàlisi minuciós queda fora de l'abast d'aquest estudi.

3.3.3 El generador de números aleatoris

La optimització per recuita és un procés que es basa en seleccions aleatòries d'elements. Per tal de poder disposar del factor aleatori en aquest procés, hem implementat una funció que genera un número aleatori en el interval [0,1) seguint el mètode Wichmann-Hill. La seqüència de números que genera aquesta funció ve determinada per tres llavors. Aquest procediment depèn doncs dels tres números que fa servir com al llavor inicial. Encara que això pugui semblar un inconvenient perquè sembla que obliga a l'usuari a introduir aquests tres números, la qual cosa no resulta còmode, té una justificació. El que fem és oferir la possibilitat (que no la obligació) a l'usuari d'incorporar un joc de llavors particular el en moment de cridar al programa. Si el programa no reb les llavors, farà servir uns valors predeterminats, que sempre seran els mateixos. Aquesta manera de cridar al programa té dos avantatges. En primer lloc permet personalitzar l'execució de cadascuna de les simulacions amb el conjunt de llavors que es vulgui. La segona gran avantatge és permetre reproduir els resultats en qualsevol ordinador en el que es pugui generar un programa executable a partir del nostre codi font.

El cost d'aquest algorisme és constant.

3.3.4 El generador de grafs aleatoris

La optimització per recuita requereix un element inicial. En el cas que ens ocupa, com que el que volem generar és un graf, el primer element que necessitem és un graf. El graf inicial pot ser qualsevol graf, sempre que sigui del mateix ordre que el graf que volem reconstruir. La selecció d'aquest primer graf no es pot fer de manera arbitrària sinó que s'ha de fer de manera aleatòria. Per tal d'obtenir un graf aleatori d'ordre n, el que fem és escollir, per cada vèrtex, un grau a l'atzar entre *1* i *n-1*, de tal manera que el graf resultant sempre sigui connex. Un cop escollit el grau g del vèrtex v, escollim a l'atzar cadascun dels g veïns de v de manera que no hi hagi branques paral·leles. En el cas dels vèrtexs que no tenen grau zero, el nou grau serà un número entre el grau actual i *n-1*.

Fer

Repetir *ordre vegades*

Triar un grau nou_grau aleatòriament entre 1 i n-1

Repetir *nou_grau vegades*

Triar un veí nou_veí (ni veí ni ell mateix)

Connectar el vèrtex inconnex al seu nou veí

Fi Repetir

Fi Repetir

Mentre *el graf sigui no connex*

El cost d'aquest algorisme és quadràtic (n^2) sobre l'ordre del graf.

3.3.5 El modificador de grafs

La opitimització per recuita requereix escollir un veí de l'element actual. En el cas que ens ocupa, com que el que volem generar és un graf, el veí que necessitem és un graf, entenent com a veí d'un graf aquell que resulti d'una petita modificació del graf actual. La selecció d'aquest nou graf s'ha de fer de manera aleatòria. Per tal d'obtenir aquest graf, el que fem és escollir al graf que ja tenim un vèrtex v a l'atzar i li traiem totes les arestes. Fet això, triem un nou grau a l'atzar entre 1 i n-1, de tal manera que el graf resultant sigui connex. Un cop escollit el grau, escollim a l'atzar cadascun dels vèrtexs als que associar v, comprovant que no hi hagi branques paral·leles. En aquesta desconnexió i connexió d'arestes pot donar-se el cas que algun vèrtex es quedi desconnectat. En aquest cas, es tria un nou grau a l'atzar entre 1 i n-1 i es creen les arestes adients Aquest últim pas es repeteix fins que no quedi cap vèrtex aïllat.

Aquest és el pseudocodi associat a aquest procediment:

Escollir el vèrtex v que anem a modificar.

Desconnectar v de tots el seus veïns

Fer

Triar un grau nou_grau aleatori entre 1 i n-1

Repetir nou_ordre vegades

Triar un veí nou_veí (que no sigui ja veí ni el mateix vèrtex)

Connectar el vèrtex inconnex al seu nou veí

Fi Repetir

Mentre el graf no sigui connex

El cost d'aquest algorisme és quadràtic sobre l'ordre del graf.

3.3.6 La funció de cost

La funció de cost és un dels paràmetres més importants de la optimització per recuita. En concret aquesta funció d'alguna manera és la que governa tot el procés, ja que és la responsable d'avaluar si s'ha trobat el graf que es volia reconstruir. El resultat d'aquesta funció és un número decimal positiu o nul que correspon a la suma, per tots els vèrtexs del graf, dels quadrats de les diferències entre els valors de les LGI_millor i les LGI_actual.

El cost d'aquesta funció és lineal (n) sobre l'ordre del graf.

Capítol 4

Disseny

4.1 Decisions de disseny

La primera decisió que vam haver de prendre va ser on fer el desenvolupament, tenint en compte l'entorn on s'hauran de fer les simulacions. El grup de grafs del departament de Matemàtica Aplicada 4 disposa de diferents equips per fer simulacions. En primer lloc disposa de dos servidors de càlcul que són Pcs amb molta memòria RAM i processadors ràpids. Aquests PCs funcionen amb Linux. En segon lloc aquest grup també disposa d'una petita granja de 4 servidors Apple Xserver també amb memòria i processadors molt ràpids. En aquest cas els processadors no són Intel x86 sinó G5. A més, aquests equips funcionen amb MAC OS X Server. Per tal de poder fer servir el programa en aquestes dues plataformes i a l'hora poder disposar d'un entorn de desenvolupament domèstic, vam escollir el C++ com a llenguatge de programació. Es tracta d'un llenguatge de programació orientat a objectes que no analitzarem ara, però que en definitiva es fa servir molt en aplicacions en les que es vol obtenir un programa eficient i pel que hi ha disponible un bon compilador tant per Linux com per MAC OS X: el g++.

Entrant en detalls d'implementació, per aconseguir els objectius que s'han definit, hem definit una estructura de dades per representar els grafs adient tant pel càlcul del grau d'intermediació com pel procés de optimització per recuita. Hem dissenyat una classe per representar els grafs i una altre classe per representar els vèrtex, de manera que els grafs disposen, entre d'altres, d'un vector de vèrtex que en definitiva és un vector d'apuntadors a aquests objectes. Aquesta organització permet identificar clarament quines son les operacions que s'han d'implementar i en el cas que es vulgui treballar amb arestes con comptes de vèrtexs, es pot definir una classe aresta que accedeixi al graf com ho fa la classe vèrtex.

Per facilitar el càlcul del grau d'intermediació hem introduït una sèrie d'atributs a la classe graf com són la matriu de distàncies i la matriu d'adjacències juntament amb les seves corresponents operacions d'actualització, que donat que tenen un cost de còmput elevat, no es criden més que quan és estrictament necessari.

4.2 Metodología

Una part important del temps d'implementació ha estat dedicat a la programació del càlcul del grau d'intermediació dels vèrtexs. Tal com hem mencionat anteriorment, hem implementat dos procediments diferents que obtenen els mateixos resultats. La idea inicial era fer-ne un de sol, que era el que resulta de la transcripció de la definició d'aquesta mesura. Un cop vam tenir una primera versió, vam voler validar els resultats obtinguts i vam fer servir una llibreria de *Python*: el *NetworkX*. Amb grafs petits, els resultats que obteníem eren el mateixos, però vam estar fent proves amb grafs tipus escala i vam veure que hi havien petites diferencies que tanmateix no podien provenir d'errors d'arrodoniment. Vam estar modificant el nostre programa i detectant casos especials que no havíem previst al principi i que podien provocar errors de càlcul, però vam arribar a un punt que no enteníem d'on venien les diferencies en els resultats. Aleshores és quan vam decidir implementar l'algorisme per al càlcul eficient del grau d'intermediació dels vèrtexs que va proposar Ulrik Brandes l'any 2001. Els resultats que obteníem amb la implementació de la definició coincidien amb els que obteníem amb la implementació eficient, però no coincidien amb els resultats de la llibreria de *Python*. Amb el codi de la llibreria *NetworkX* davant, vam descobrir que la mateixa llibreria ofereix dues funcions amb noms indicatius molt semblants per calcular la *betweenness*: *betweenness_centrality* i *brandes_betweenness_centrality*. Resulta que, per una banda, els resultats que donava la funció *betweenness_centrality* no coincidien amb els que nosaltres generàvem. Per altre banda la versió eficient que fa el càlcul mitjançant l'algorisme d'en Brandes no estava accessible des del *Python* per un error en la publicació dels noms de funcions de la llibreria *NetworkX*. Un cop resolt aquest problema en l'ordinador de desenvolupament, vam comprovar que els resultats que havíem estat generant eren els mateixos que obteníem amb la funció de *NetworkX*. Ens vam posar en contacte amb *NetworkX* per comentar el que ens havia passat i com ho havíem resolt. Ens van respondre molt ràpidament agraint la nostre col·laboració, que inclouran les modificacions necessàries per la correcta publicació de les funcions a la següent versió i dient que la funció *betweenness_centrality* que havíem estat utilitzant no calculava la mesura que nosaltres voliem.

No es pot negar que va estar una experiència enriquidora i que va posar a prova la nostre paciència. Aquest problema es va resoldre a finals d'abril i ha provocat un endarreriment de prop de un mes. Aquesta és la raó per la qual no hem disposat de temps suficient per posar en marxa algunes de idees que ens hauria agradat avaluar i han hagut de quedar com a suggeriemnts per treballs futurs.

Capítol 5

Resultats

Ara que ja sabem què fa el sistema, arriba l'hora de posar les màquines a calcular.

Hem fet proves amb grafs d'ordre 10 i el sistema els ha pogut reconstruir en molt poc temps. De fet, fer 10! comprovacions és poca cosa. Comparar 3628800 grafs és una tasca que es pot fer en un temps relativament curt. El que no és tan fàcil és reconstruir un graf de gaires més vèrtex. Amb una mica més de paciència hem aconseguit reconstruir algun graf de 12 vèrtexs.

Per fer l'anàlisi del funcionament del sistema hem triat cinc exemples de grafs per fer les simulacions, cadascun d'ells representatiu d'una família de grafs de les que es fan servir per modelar les diferents topologies de xarxes. Tal com vam anunciar quan vam descriure els diferents tipus de grafs que es fan servir per modelar xarxes, els grafs que hem dissenyat per fer les simulacions són dels tipus aleatori, amb agrupament, amb invariància d'escala, petit mon i regular. Per tal de poder comparar els resultats de cadascuna de les simulacions en les mateixes condicions, hem escollit els cinc grafs del mateix ordre: 40.

La complexitat del problema és força gran. Reconstruir un graf de 40 vèrtexs amb la força bruta és inviable. Hi ha 815915283247897734345611269596115894272000000000 (és a dir 40!) grafs d'ordre 40. El Mare Nostrum trigaria uns quants milions d'anys en fer totes les proves. Es tracta doncs d'un repte computacional.

No pretenem impossibles, sinó que volem veure fins a quin punt la optimització per recuita simulada és un mètode vàlid per aquest tipus de problema. Ja hem dit que el sistema permet reconstruir grafs de fins a 10 vèrtex en qüestió de segons. Sabem que la reconstrucció no sempre genera un graf isomorf al graf que es pretén reconstruir. El que volem analitzar és fins a quin punt la reconstrucció s'acosta al que volem. Per això hem analitzat els resultats de les simulacions i n'hem extret les característiques dels grafs que ens haurien de donar una idea del nivell de reconstrucció que podem esperar amb aquest sistema.

Els paràmetres que s'han fet servir per caracteritzar la recuita són els següents:

Temperatura inicial: 1,0

Temperatura final: 0,000001

Tasa de refredament: 0,9

Número de passos per iteració: 2000

A la taula de la pàgina següent trobem resumits els resultats de les més de 300 simulacions que hem realitzat per cadascun dels grafs. No hem limitat el temps de simulació per cadascuna de les reconstruccions sinó que hem llençat 500 simulacions per cada graf, de tal manera que fem servir les mateixes llavors del generador de nombres aleatoris per cadascun dels grafs.

En aquesta taula trobem, per cadascun dels grafs que hem reconstruit, unes quantes mesures que permeten estimar la qualitat de les reconstruccions com son el diàmetre, la distància mitjana, els graus dels vèrtexs i l'agrupament. A més d'aquestes mesures clàssiques dels grafs hem afegit algunes d'especial rellevància en aquest estudi com són el cost, el grau d'indeterminació dels vèrtexs i la delta.

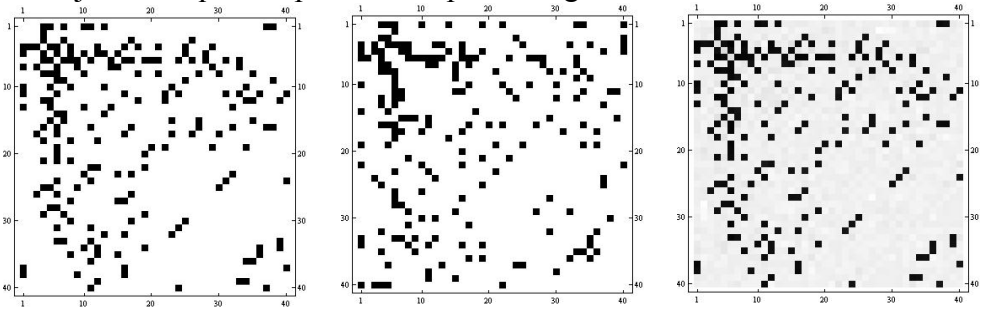
El cost és el valor de la funció de cost corresponent al graf reconstruït, que mesura la similitud de les llistes del grau d'intermediació del grafs que es vol reconstruir amb el graf efectivament reconstruït. Un cost nul només es dóna quan el procés d'optimització per recuita simulada no sap diferenciar el graf objectiu de l'actual, la qual cosa no vol dir que el graf reconstruït sigui isomorf al graf que volíem reconstruir. Quan el cost és més gran que 0, com passa sempre en els casos que hem analitzat, vol dir que la simulació ha acabat abans de reconstruir el graf desitjat, i que el resultat és un graf que no pot ser exactament igual al que es volia reconstruir.

La delta és un valor que obtenim d'acord amb el mètode de descomposició en valors singulars, que és el que hem escollit per comparar el nivell de similitud que tenen dos grafs: el que volem reconstruir i el que ha generat el sistema.

Donat que els grafs tenen 40 vèrtexs, a la taula només mostrem els valors màxim, mitjà i màxim de les mesures dels vèrtexs com son el grau i el grau d'intermediació.

		Aleatori		Circulant		Petit mon		Invariant escala		Agrupat	
		Ori	Rec	Ori	Rec	Ori	Rec	Ori	Rec	Ori	Rec
Diàmetre	Min	6	5,0000	10	8,0000	6	5,0000	4	4,0000	5	4,0000
	Avg		5,8700		8,7700		6,6870		4,4751		5,2710
	Max		7,0000		11,000		9,0000		6,0000		7,0000
Distància mitjana	Min	2,89	2,6400	5,38	4,2400	3,31	2,8500	2,32	2,2100	2,65	2,4590
	Avg		2,7800		4,6700		3,0280		2,3400		2,5780
	Max		2,9300		5,1400		3,2630		2,4600		2,7320
Grau	Min	1	1,0000	4	2,0000	3	1,0000	1	1,0000	2	1,0000
	Avg	3,8	3,9500	4	2,5700	4	3,6900	4,95	5,0960	6,3	4,4640
	Max	8	16,000	4	5,0000	5	10,000	17	22,000	13	15,000
Agru-pament	Min	0,2	0,0400	0,5	0,0000	0,32	0,0210	0,28	0,1340	0,37	0,0750
	Avg		0,1950		0,0300		0,1150		0,2620		0,2001
	Max		0,3200		0,1000		0,2490		0,4180		0,3488
Grau d'inter-mediació	Min	0,0000	0,0000	0,1154	0,0480	0,0132	0,0004	0,0000	0,0003	0,0003	0,0000
	Avg	0,0498	0,0469	0,1154	0,0966	0,0608	0,0534	0,0346	0,0353	0,0433	0,0415
	Max	0,2865	0,2819	0,1154	0,1495	0,1235	0,1231	0,3127	0,3144	0,2048	0,2027
Cost	Min		0,0390		0,1497		0,0600		0,0220		0,0600
	Avg		0,0534		0,1930		0,0806		0,0320		0,0816
	Max		0,0840		0,2320		0,1020		0,0540		0,1100
Delta	Min		0,0120		0,0590		0,0190		0,0140		0,0576
	Avg		0,0260		0,0740		0,0290		0,0235		0,0816
	Max		0,0620		0,0878		0,0500		0,0490		0,1070

A títol representatiu, a continuació mostrem unes imatges de les matrius d'adjacència que hem generat a partir dels resultats obtinguts. La primera imatge correspon al graf inicial, la segona al graf reconstruït i la tercera mostra en grisos el producte de la transformació de les matrius d'adjacència que ens permet comparar els grafs.



Capítol 6

Conclusions

La durada d'aquest projecte ha estat de gairebé 5 mesos. La primera reunió que vam tenir per parlar del projecte va ser el dijous 8 de febrer de 2007 i la defensa està prevista pel 5 de juliol de 2007.

6.1 Valoració econòmica

En la valoració d'aquest projecte s'ha de considerar que es tracta d'un treball de recerca en el que han participat principalment dues persones. En primer lloc en Francesc Comellas ha estat el director del projecte, per la qual cosa ha fet un seguiment continuat de la evolució i els avanços, aclarint els dubtes i orientant cap el camí correcte sempre que hi ha estat necessari a en Juan Paz, l'alumne que ha realitzat aquest estudi. La tasca realitzada per na Montserrat Maureso ha estat la de supervisar el treball realitzat per tal de validar que es pot considerar con un projecte final de carrera segons els criteris definits per la Facultat d'Informàtica de Barcelona, per la qual cosa ha hagut d'avaluar la càrrega que suposava la feina. L'equipament informàtic que hem fet servir en aquest projecte no es van comprar exclusivament per aquest propòsit. L'ordinador en el que s'ha fet tot a excepció de les simulacions ha estat el portàtil personal de l'estudiant. En quant a les simulacions, hem utilitzat equips de simulació amb molta potència de càlcul que es van comprar pel projecte de recerca "Grandes redes tolerantes a fallos. Nuevos modelos, diseño, anásis y algoritmos" TEC2005-03575 dirigit per en Francesc Comellas i en el que s'emmarca aquest projecte final de carrera. En aquest sentit es pot fer una estimació del cost econòmic associat a l'ús de l'equipament informàtic que s'ha fet servir basada en el temps d'utilització i el nombre de persones que han compartit el recurs.

Ordinador portàtil:

Preu de compra: 800 €

Període d'amortització estimat: 5 anys

Dedicació durant el projecte: 100% durant els 5 mesos

El cost associat a l'ús de l'ordinador portàtil és de:

800 x 5 / 60 = 67 €

Els servidors de càlcul, fora del període de optimització no s'han fet servir més que per comprovar que el programa es podia compilar i executar sense problemes. Encara que la

utilització d'aquest tipus de servidors normalment es mesura en temps de CPU, entenem que una aproximació més realista consisteix en repartir el cost entre els diferents usuaris que han fet servir la potència de càlcul durant el període de simulacions.

Servidors de càlcul:

Preu de compra: 22.000 € (inclou sistema operatiu i manteniment)

Període d'amortització estimadat: 5 anys

Dedicació durant el projecte: 50% durant dos mesos

El cost associat a l'ús del servidor de càlcul és de:

22.000 x 50 x 2/ (60 x 100) = 367 €

El cost associat a l'ús d'equipament informàtic és de 434 €

En quant a la valoració econòmica associada al personal, la feina realitzada podria formar part d'una tasca de recerca d'un estudiant de doctorat. Entre els diferents tipus de beques que atorga l'estat hem escollit com a referència les FPI (Formació de Personal Investigador), la remuneració de la qual es de 1120 € al mes. Encara que la duració del projecte ha estat de 5 mesos, la dedicació no ha estat completa. Fent balanç del temps dedicat al projecte, es pot dir que s'han dedicat una mitja de 20 hores setmanals, és a dir mitja jornada, que fan un total de 420 hores.

El cost associat al personal resultant és: 1120 x 5 / 2 = 2.800 €

6.2 Balanç i futur

Els resultats obtinguts es poden considerar bons donat que el sistema que hem implementat sempre genera un graf que, encara que no sempre sigui la millor solució, si que serà topològicament molt semblant al que volem reconstruir. La optimització per recuita simulada és un bon mètode per abordar problemes difícils, ja que permet obtenir resultats satisfactoris en un temps raonable. La variabilitat dels resultats és un fet, però sens dubte es tracta d'una característica inherent del tipus d'eina que fem servir. El balanç d'aquest estudi és que la reconstrucció de grafs és una tasca que requereix molts esforços i aquest treball n'és una prova.

Aquest mètode es pot millorar modificant la funció de cost de tal manera que, a més de comparar les llistes de graus d'intermediació globalment, oferís informació dels vèrtexs amb graus d'intermediació més semblants i més diferents a la llista de valors de referència corresponents al graf desitjat. Es tracta d'una petita modificació que no suposaria afegir gaire operacions a la funció de cost. És precisament cap aquest sentit on pensem que s'ha d'investigar. Donat que el grau d'intermediació dels vèrtexs és la mesura en la que es basa la funció de cost, sembla raonable fer servir aquesta informació per escollir el vèrtex que modificarem en el graf.

Una altre aproximació que es pot analitzar consisteix en fer que el mecanisme de modificació del graf no desconnecti totes les arestes incidents amb un vèrtex sinó només una part. Així els grafs serien més semblants i la diferència d'energia no serà tan gran d'un pas a l'altre. Com a conseqüència, tant l'aproximació com l'allunyament de la solució serà més progressiva, la qual cosa permet un millor encaminament cap a la solució. En aquest cas, també sembla raonable fer servir una mesura que no doni informació del vèrtexs sinó de les arestes, de manera que també es puguin ajudar a triar les modificacions que es fan del graf a cada pas de la recuita en funció d'una mesura que considera en primera instància les arestes en comptes dels vèrtexs.

Bibliografia

Ulrik Brandes: "A Faster Algorithm for Betweenness Centrality". *Journal of Mathematical Sociology* **25**(2), (2001) 163-177.
http://www.inf.uni-konstanz.de/algo/publications/b-fabc-01.pdf

Mads Ipsen, Alexander S. Mikhailov: "Evolutionary reconstruction of networks".
http://es.arxiv.org/abs/nlin/0111023

Shawn Carlson: "Algorithm of the Gods". *Scientific American* 3 (1997).
http://www.sciam.com/0397issue/0397amsci.html

M. Girvan, M.E.J. Newman: "Community structure in social and biological networks". *Proc. Natl. Acad. Sci.* USA 99 (2002) 7821-7826.
http://www.pnas.org/cgi/reprint/99/12/7821

Barabasi, A., Albert R.: "Emergence of scaling in random networks". *Science* **286**, (1999) 509–512.
http://www.answers.com/topic/ba-model

Francisco J. Abad Cerdá: "Estructura básica de un proyecto final de carrera"
http://www.dsic.upv.es/~fjabad/pfc/estructura.pdf

F. Comellas, J. Fàbrega, A. Sànchez, O. Serra: *Matemàtica discreta.* Edicions UPC 1994. ISBN 84-7653-413-2