

# CONSTRUCCIÓ DE XARXES D'INTERCONNEXIÓ DENSES I "CAGES" MITJANÇANT LA RECUITA SIMULADA

Autor: Xavier Guarch Domingo  
Director: Francesc Comellas

PFC- ETSETB 12 de Gener de 2000

A en Francesc,  
per l'entusiasme que transmet

Als meus pares,  
que sempre m'han empès a créixer

Als meus amics,  
que sempre aposten per mi

A la Vida,  
que sempre em dóna més

i a tots aquells  
que busquen la Veritat.

## ÍNDEX

<b>1. INTRODUCCIÓ .....</b>	<b>5</b>
<b>2. TEORIA DE GRAFS I ELS PROBLEMES NP-COMPLETS .....</b>	<b>9</b>
2.1. NOCIONS BÀSIQUES DE TEORIA DE GRAFS .....	9
2.2. ELS PROBLEMES NP-COMPLETS .....	13
<b>3. ALGORISMES DE CERCA .....</b>	<b>18</b>
3.1. INTRODUCCIÓ.....	18
3.2. ELS ALGORISMES .....	18
3.2.1. Cerca exhaustiva.....	18
3.2.2. Cerca local o aleatòria.....	19
3.2.3. Algorismes heurístics .....	19
3.2.4. Algorismes naturals.....	22
3.2.4.1. Les Xarxes Neuronals.....	23
3.2.4.2. Els Algorismes Genètics.....	24
3.2.4.3. Els Algorismes Multiagent .....	25
3.2.4.4. La Cerca Tabú .....	25
3.2.4.5. La Recuita Simulada.....	26
<b>4. CONVERGÈNCIA DE L'ALGORISME DE LA RECUITA SIMULADA.....</b>	<b>28</b>
<b>5. XARXES D'INTERCONNEXIÓ AMB GIRTH FIXAT: LES CAGES.....</b>	<b>36</b>
5.1. DESCRIPCIÓ.....	36
5.2. ALTRES MÈTODES UTILITZATS FINS ARA .....	39
<b>6. XARXES D'INTERCONNEXIÓ AMB DIAMETRE FIXAT: EL PROBLEMA (<math>\Delta</math>,D).....</b>	<b>45</b>
6.1. DESCRIPCIÓ.....	45
6.2. ALTRES MÈTODES UTILITZATS FINS ARA .....	49
<b>7. IMPLEMENTACIÓ DELS ELEMENTS DE LA R.S. PER ALS DOS PROBLEMES TRACTATS.....</b>	<b>50</b>
7.1. L'ÀMBIT DEL PRESENT PROJECTE.....	50
7.2. ELS ELEMENTS RELLEVANTS EN LA IMPLEMENTACIÓ.....	51
7.2.1. Els tres elements principals de l'algorisme .....	53
7.2.1.1. Els mecanismes de construcció dels grafs .....	53
7.2.1.2. Els mecanismes d'alteració del graf.....	68
7.2.1.3. La funció objectiva.....	70
7.2.2. La resta de paràmetres de l'algorisme .....	82
<b>8. SIMULACIONS I RESULTATS.....</b>	<b>87</b>
8.1. VERSIONS DE L'ALGORISME .....	87
8.1.1. Algorisme A: Arbre variable.....	87
8.1.2. Algorisme B: Arbre fix.....	88
8.1.3. Algorisme C: Cercle fix amb cordes.....	92
8.1.3.1. Les cages .....	93
8.1.3.2. El problema ( $\Delta$ ,D) .....	97
8.2. RESUM DE RESULTATS .....	102
<b>9. CONCLUSIONS .....</b>	<b>105</b>
<b>10. ANNEX 1 – DESCRIPCIÓ GENERAL DEL PROGRAMA.....</b>	<b>110</b>
10.1. ESTRUCTURA LÒGICA DEL PROGRAMA (EN PSEUDO-CODI) .....	110
10.2. CONSTANTS DE CONFIGURACIÓ DE L'ALGORISME (FITXER CAPÇALERA) .....	112
10.3. FITXER DE DADES.....	115

<b>11.</b>	<b>ANNEX 2 – CODI PER AL PROBLEMA DE LES CAGES.....</b>	<b>117</b>
11.1.	ANNEAG.H .....	117
11.2.	ANNEAG.C.....	122
<b>12.</b>	<b>ANNEX 3 – CODI PER AL PROBLEMA (<math>\Delta</math>,D).....</b>	<b>156</b>
12.1.	ANNEAD.H .....	156
12.2.	ANNEAD.C.....	161
<b>13.</b>	<b>ANNEX 4 – PROGRAMA D'ANÀLISI DE PROPIETATS DE GRAFS .....</b>	<b>189</b>
13.1.	MOSTRA_G.C .....	189
<b>14.</b>	<b>BIBLIOGRAFIA.....</b>	<b>198</b>
<b>15.</b>	<b>LLISTA DE FIGURES.....</b>	<b>203</b>
<b>16.</b>	<b>LLISTA DE TAULES.....</b>	<b>204</b>

## 1. INTRODUCCIÓ

En els darrers anys s'ha produït una enorme explosió, com tots sabem, en el món de les telecomunicacions, que ve capitanejada per un nom "Internet", i que neix d'una massiva integració entre computadores i sistemes de comunicacions. Però "Internet" no n'és l'únic exponent. Hi ha actualment milers de xarxes d'interconnexió de sistemes esteses per tot el món, amb topologies ben diferents, que busquen resoldre necessitats de comunicació també ben diverses. Necessitats que normalment es tradueixen en un conjunt més o menys complex de requeriments i d'especificacions a complir (de qualitat de les comunicacions, de temps de resposta del sistema, d'interconnectivitat dels equips, de costos, etc.) i que l'enginyer encarregat del disseny de la xarxa ha de resoldre de forma òptima en tots i cada un dels dissenys. I, com molt bé sap aquell qui ha treballat o treballa en aquest camp, sempre hi ha requeriments contraposats i irreconciliables: sempre cal arribar a un disseny de compromís, un disseny òptim en el sentit que compleixi "al màxim" tots els requisits.

Però la complexitat de les xarxes creix al mateix ritme que la tecnologia: es demanda que integrin cada vegada un major nombre d'ordinadors i, sobretot, que la resposta global de la xarxa sigui més òptima. Això fa que la capacitat dels enginyers per trobar aquest disseny òptim es vegi posada a prova cada vegada amb més freqüència. I, de vegades, hi ha dissenys que requereixen de quelcom més que de l'enginy, l'experiència i el coneixement tècnic clàssic de telecomunicacions.

En el present projecte tractem d'aportar un mètode que ens permeti, amb eficàcia, resoldre aquest tipus de problemes d'optimització complexos, de

disseny, quan la dimensió dels mateixos creix més enllà del que la capacitat humana pot resoldre mitjançant mètodes tradicionals. I ens referim a problemes impracticables d'abordar no només per la nostra limitada ment, sinó fins i tot per a la tecnologia més potent que hem sigut capaços d'idear i que podríem dir que és gairebé una extensió de la nostra pròpia ment, la supercomputació paral·lela. Per als problemes que aquí tractem, coneguts en l'àmbit matemàtic com a NP-Complets (que significa *no polinòmics determinístics*), qualsevol enfoc clàssic (lineal) d'intent de resolució condueix ràpidament al fracàs en quant la dimensió d'aquests augmenta una mica, perquè la seva complexitat creix de forma no-polinomial a la mida dels mateixos.

Per això, pretenem mostrar en aquest projecte la necessitat i, alhora, la validesa d'un enfoc alternatiu al clàssic per a la resolució d'aquest tipus de problemes d'enginyeria: l'ús d'algorismes no-lineals que són imitació d'alguns mecanismes que utilitza la Natura per a la resolució de determinats "problemes". Aquest és el cas de l'algorisme de la Recuita Simulada (*Simulated Annealing*), que està inspirat en el fenomen de la Cristal·lització. No obstant, l'ús d'aquest tipus d'algorismes no treu que utilitzem els recursos de la computació com a mitjà per assolir aquest objectiu.

En concret, volem mostrar la capacitat d'aquest algorisme per resoldre dos problemes d'optimització combinatòria de Teoria de Grafs, que modelen, cada un, un cas diferent de disseny de xarxes d'interconnexió: l'un, **el problema  $(\Delta, D)$** , relacionat amb xarxes denses en que el màxim retard permès per a les comunicacions entre dos nodes qualsevol de la mateixa està prefixat; i l'altre, **el problema de les cages**, relacionat amb xarxes que conserven camins de retorn de la informació d'una longitud mínima especificada.

L'objectiu és descobrir si aquest mètode, la Recuita Simulada, que ha sigut ja aplicat amb èxit en nombrosos àmbits de l'enginyeria, pot ser útil i eficient per a la resolució de problemes de disseny de xarxes d'interconnexió com els aquí estudiats.

L'enfoc d'aquesta cerca que seguirem en el present projecte és des del punt de vista i en el context de la Teoria de Grafs i de l'Algorísmica moderna, per tal de donar-li la consistència i solidesa matemàtiques que requereix una tesi universitària en telecomunicacions. No obstant, hem intentat donar-li també un enfoc pràctic i mostrar el procés que hem seguit al llarg del projecte en la implementació de l'algorisme per assolir l'objectiu marcat, de tal manera que pugui servir també de guia de procediments per a l'enginyer que desitgi utilitzar la Recuita Simulada en el seu treball de disseny de xarxes.

Per això, en el proper capítol (capítol 2) comencem fent una breu exposició de les nocions de Teoria de Grafs relacionades amb el tema; i en el capítol 3 repassem els diversos mètodes d'optimització ideats per l'Algorísmica al llarg de la història i mostrem les limitacions dels enfoc clàssics en aplicar-los als problemes que aquí ens ocupen, així com una descripció dels "nous" algorismes com la Recuita Simulada (RS). Per demostrar-ne la validesa com a mètode, en el capítol 4 fem una anàlisi qualitativa del bon comportament de

la RS davant d'un problema estàndard d'optimització, tot comparant-lo al comportament d'algorismes clàssics.

I a continuació, en els capítols 5 i 6, descrivim les característiques matemàtiques dels dos problemes estudiats, les *cages* i el problema  $(\Delta, D)$ , respectivament; així com l'estat de les investigacions relacionades amb tots dos dins del sí de la comunitat matemàtica en la que s'inscriu el nostre projecte.

A partir d'aquí, al capítol 7 és on fem l'exposició de tots els elements que cal tenir en compte alhora d'adaptar l'algorisme a un problema concret i dels passos en la implementació pràctica que hem seguit al llarg del projecte. I al capítol 8 detallem el conjunt de casos estudiats d'ambdós problemes, i els resultats obtinguts.

Finalment, el capítol 9 recull les conclusions que es deriven dels resultats obtinguts i les possibilitats que aquests resultats ofereixen de cara a implementacions immediates en l'àmbit del disseny de xarxes o en d'altres àmbits de les enginyeries de telecomunicacions i telemàtica.

Com a annexos, s'adjunta el codi implementat per a cada un dels problemes tractats, breument explicat de forma global al Annex 1, i també el codi del programa *mostra\_g* d'anàlisi de les propietats matemàtiques que ens interessen dels grafs, capaç de llegir i escriure el descriptor dels grafs en diversos formats que són d'ús comú i estàndard en la comunitat matemàtica mundial.

## 2. TEORIA DE GRAFS I ELS PROBLEMES NP-COMPLETS

En aquest capítol donem una visió general dels conceptes fonamentals de la Teoria de Grafs que tenen relació directa amb el present projecte, aprofundint en les propietats més rellevants dels grafs que ens són necessàries de conèixer per a la comprensió del treball aquí realitzat.

A continuació, fem una exposició de les característiques dels problemes NP-Complets, que formen un subconjunt diferenciat del tipus de problemes tractats a Teoria de Grafs, i dels principals conceptes relacionats amb els mateixos.

### **2.1. Nocions bàsiques de Teoria de Grafs**

Un **graf**  $G=(V,E)$ , consisteix en un conjunt  $V$  finit no buit d'elements nomenats **vèrtexs** i en un conjunt  $E$  de parells no ordenats d'elements de  $V$  anomenats **branques**.

S'anomena **ordre** al nombre de vèrtexs del graf i **mida** al nombre de branques.

Si  $(x,y)$  és una branca del conjunt  $E$  indica que el vèrtex  $x$  és **adjacent a**  $y$ , i recíprocament, que  $y$  és adjacent a  $x$ . Si existeixen **dues o més** branques que uneixen els vèrtexs  $x$  i  $y$  entre sí, es diu que aquestes són **branques paral·leles**.

S'anomena **grau** d'un vèrtex  $v$  de  $G$  al nombre de branques que hi incideixen, i se sol representar per  $d(v)$ . Un graf  $G$  es diu que és **regular** o **k-regular** si tots els seus vèrtexs tenen el mateix grau  $k$ .

Dos grafs  $G_1$  i  $G_2$  són **isomorfs** si existeix una correspondència biunívoca entre els vèrtexs de  $G_1$  i els de  $G_2$  i amb la propietat de que si dos vèrtexs són adjacents a  $G_1$ , els corresponents vèrtexs de  $G_2$  també ho són. Se sol dir que  $G_1$  i  $G_2$  difereixen en **l'ordenació** o en **l'etiquetat** dels seus vèrtexs . La Figura 2.1 mostra un exemple de dos grafs isomorfs.

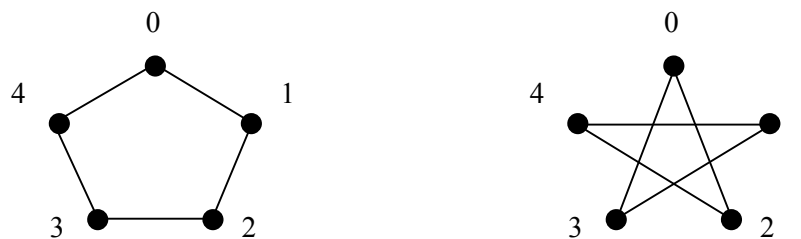


Figura 2.1 Grafs isomorfs

La **matriu d'adjacència** d'un graf es la representació en forma de matriu del graf que n'indica els vèrtexs que són adjacents entre sí.

Així, donat un graf d'ordre  $n$  sense branques paral·leles, la seva matriu d'adjacència correspon a un matriu de  $n \times n$  definida sobre  $\mathbb{N}$  de la següent manera:

$$a_{ij} = \begin{cases} 1 & \text{si hi ha una branca entre els vèrtexs } i \text{ i } j. \\ 0 & \text{altrament.} \end{cases}$$

En destaquen les següents propietats:

- i) És simètrica.

- ii) Si tots els  $a_{ii}=0$ , no hi ha autollaços al graf.
- iii) El grau d'un vèrtex és igual al número de 1 en la seva fila o columna ( si no té autollaços).
- iv) Una reordenació (re etiquetat) dels vèrtexs, es tradueix en una simple permutació de files (i de les corresponents columnes).
- v) Dos grafs són isomorfs si i només si les seves matrius d'adjacència  $A(G_1)$  i  $A(G_2)$  es relacionen per:

$$A(G_2) = P^{-1} A(G_1) P$$

on  $P$  correspon a una matriu de permutacions.

Les propietats de la matriu d'adjacència s'extenen també a les seves potències.

Així, per a la potencia 2 es té:

- $(A^2)_{ii}$  és el grau del vèrtex  $v_i$ , si no hi ha autollaços.

$$(A^2)_{ii} = \sum_{k=1}^n a_{ik} a_{ki} = \sum_{k=1}^n a_{ik}^2 = \sum_{k=1}^n a_{ik} = d(v_i).$$

- $(A^2)_{ij}$  és igual al nombre de camins diferents de longitud 2 que hi ha entre els vèrtexs  $v_i$  i  $v_j$ .

I de forma general, per a qualsevol potència  $k$  de  $A$  es demostra que:

TEOREMA:

- $(A^k)_{ii}$  és igual al nombre de circuits de longitud  $k$  que contenen el vèrtex  $v_i$ .
- $(A^k)_{ij}$  és igual al nombre de seqüències diferents de  $k$  branques entre els vèrtexs  $v_i$  i  $v_j$ .

Els **autovalors** d'un graf  $G$  són els autovalors de la seva matriu d'adjacència

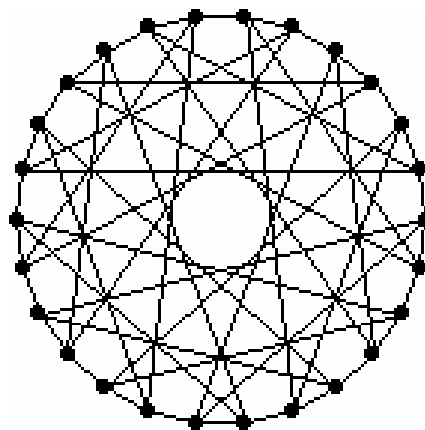
$A(G)$ . El conjunt d'autovalors és **l'espectre** de  $G$ .

**TEOREMA:** Donat un graf  $G$  amb matriu d'adjacència  $A$  i amb grau mínim i màxim  $\delta$  i  $\Delta$ , respectivament, es demostra [9] que:

- Els autovalors d' $A$  satisfan que  $|\lambda| \leq \Delta$ .
- $\Delta$  és autovalor de  $A$  si i només si  $G$  és  $\Delta$ -regular, i en aquest cas, la multiplicitat de l'autovalor  $\Delta$  és 1.

La **representació gràfica d'un graf** és la forma de dibuixar els seus vèrtexs, branques i la forma en que aquests estan relacionats entre sí. La convenció més corrent per a grafs no dirigits és la de representar els vèrtexs com a punts i les branques com a línies que uneixen aquests punts.

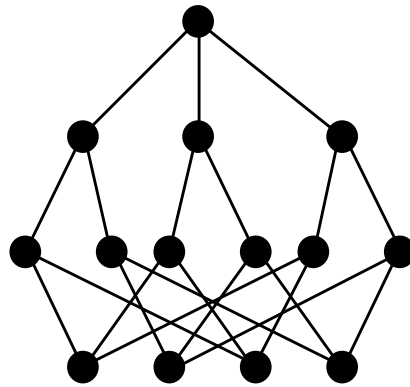
La **representació circular** d'un graf  $C=(V,E)$  és aquella en que el conjunt de vèrtexs  $V$  estan situats sobre una circumferència imaginària i el conjunt de branques  $E$  es situen sempre a l'interior de la mateixa. La Figura 2.2 mostra la representació d'un graf en la seva forma circular.



*Figura 2.2 Representació circular d'un graf*

La **representació en arbre** d'un graf  $C=(V,E)$  és aquella en que s'inicia la

representació partint d'un dels vèrtexs (anomenat *arrel* i indexat com a vèrtex "0", zero) i es dibuixen les seves branques i els seus veïns immediats obrint-se en "vano", normalment en direcció descendent. I a continuació es dibuixen els altres veïns d'aquests vèrtexs, seguint la mateixa direcció; i així fins a completar tot el graf. La Figura 2.3 mostra la representació d'un graf en la seva forma d'arbre.



*Figura 2.3 Representació en arbre d'un graf*

## **2.2. Els problemes NP-Complets**

En el camp de la Teoria de Grafes es tracten una gran diversitat de problemes, la complexitat dels quals es mesura sovint mitjançant la complexitat o l'eficiència dels algorismes que millor els resolen.

En l'Algorísmica clàssica, la complexitat d'un algorisme es definia per la seva complexitat temporal en termes de màquines de Turing [17]. Actualment, s'ha introduït un concepte més informal, però igualment vàlid per a definir-la, que té a veure amb el nombre d'operacions a realitzar en l'algorisme en funció de la mida del problema.

En aquest sentit, s'introdueix la següent notació per a estimar l'eficiència d'un algorisme:

Sigui  $f$  una funció de  $\mathbb{N}$  en  $\mathbb{N}$ . Diem que  $f(n)$  és  $O(g(n))$ , o **d'ordre  $g(n)$**  si existeix una constant  $k$  positiva tal que  $f(n) \leq k \cdot g(n)$  per a tot  $n \in \mathbb{N}$  (llevat possiblement d'un nombre finit d'excepcions).

Per exemple, un algorisme que requereixi de l'ordre de 100 o 200 operacions quan  $n=100$ , i 1000 o 2000 quan  $n=1000$ , és  $O(n)$ ; mentre que un altre que requereixi de l'ordre de 10.000 en el primer cas i 1.000.000 en el segon, és  $O(n^2)$ .

És partint d'aquesta definició que es fa la següent classificació dels tipus de problemes de decisió existents:

- Un problema es diu que és de tipus **P** si per a la seva resolució calen algorismes que realitzen un nombre d'operacions bàsiques  $O(p)$ , on  $p$  és un polinomi en la mida del problema.
- En contraposició amb aquests, estan els de tipus **NP**, que vol dir *no determinístic polinòmic*. Aquests problemes poden ser resolts en temps polinòmic amb una màquina de Turing no determinista, que pot ser definida com un conjunt de màquines de Turing deterministes processant la cinta en paral·lel. En aquest cas, la complexitat temporal en funció de la longitud  $n$  de l'entrada ve donada pel nombre de passos màxim que pot fer, considerant totes les possibles entrades de longitud  $n$  i prenent com a temps corresponent a una entrada el de la màquina de Turing determinista que ha trigat menys que totes les que computaven en paral·lel. No obstant, una solució d'un problema **NP** pot ser comprovada en temps polinòmic.

Per a molts problemes **NP** es coneixen fites lineals eficients, tanmateix les fites conegudes per al cas pitjor són exponencials.

En aquest marc, es diu que un problema *X* es pot *transformar polinòmicament* en un altre problema *Y* si, donades dues entrades *A* i *B*, que poden construir-se una a partir de l'altra en temps polinomial respecte de l'original, *A* fa que *X* tingui resposta afirmativa si i solament si *B* fa que *Y* tingui resposta afirmativa.

Amb aquesta noció es pot introduir la definició de problema **NP-C** o **NP-Complet**. Un cert problema *X* és del tipus NP-Complet si, a més de ser NP, tots els altres problemes NP s'hi poden transformar polinòmicament. Es demostra (Cook, 1971; Levin, 1973) que si se sap resoldre de forma eficient un dels problemes NP-Complet queden resolts tots. De la mateixa manera, si es demostra que un qualsevol dels problemes d'aquesta categoria no és tractable, tots els altres tampoc ho són.

D'entre el total de problemes tractats en Teoria de Grafs, els anomenats NP-Complets s'inclouen dins del conjunt de problemes d'**optimització combinatòria**.

En els problemes d'optimització combinatòria es tracta sempre de trobar la millor solució (o l'òptima) d'entre totes les possibles solucions que pot tenir el problema considerat. El conjunt de possibles solucions s'anomena **espai d'estats**. A cada element d'aquest espai d'estats se li assigna un **cost** segons una funció anomenada **funció objectiva** o **funció de cost**, així doncs, es tracta de trobar aquella solució que tingui el major (o menor) cost possible.

D'entre tots els problemes de tipus NP-Complets existents en els diversos àmbits de la ciència i l'enginyeria, el primer que es va estudiar és el del Viatjant (*traveling salesman*), exposat per T.P. Kirkman i W.R. Hamilton el 1856, impulsors de moltes de les bases de l'actual Teoria de Grafs. Aquest problema a sigut, des de llavors, el més conegut i més estudiat, amb especial intensitat a la dècada passada.

En la formulació més coneguda d'aquest problema, un venedor parteix de la ciutat on viu i ha de visitar totes les ciutats que té en una llista, passant únicament una vegada per cadascuna i retornant finalment a la seva ciutat. La qüestió està en trobar aquell recorregut que faci mínima la distància recorreguda.

El problema, com es veu, pot resultar senzill de resoldre per a un nombre reduït de ciutats a recórrer, 5 o 10, però resulta cada vegada més complex conforme va augmentant el nombre de ciutats, 20 o 30, i esdevé impossible d'abordar mitjançant mètodes convencionals de càlcul quan la dimensió del problema esdevé de l'ordre de 50, 100 o 500 ciutats a recórrer, donat que l'espai d'estats creix de forma no-polinomial.

D'altres problemes de la mateixa índole dins de la Teoria de Grafs són, per exemple, el de coloració de grafs, el d'aplanament de grafs, ..., així com també el problema de les *cages* i el problema  $(\Delta, D)$ .

I en l'àmbit de les telecomunicacions i l'enginyeria, l'assignació de freqüències en xarxes cel·lulars de mòbils, mecanismes de propagació d'informació tipus *broadcast* eficients, formació de codis de xifrat per a comunicacions, i generació automàtica de xarxes, entre d'altres, que es corresponen en molts casos amb algun problema equivalent en el camp de la Teoria de Grafs.

En el proper capítol s'exposa una visió amplia del ventall d'algorismes que s'han anat descobrint i desenvolupant al llarg de la història de la matemàtica fins arribar als més recents, entre els quals es troba la Recuita Simulada, que resulten especialment eficients per a la resolució de problemes del tipus NP-Complets.

### 3. ALGORISMES DE CERCA

#### **3.1. Introducció**

Com hem pogut veure en el capítol anterior, el tipus de problemes que abordem en aquest projecte s'inclouen dins dels anomenats NP-Complets.

Això significa que, per les seves característiques, no tots els tipus d'algorismes coneguts són adequats per a la resolució d'aquests problemes.

D'entre tots els algorismes que s'han anat ideant al llarg de la història de la matemàtica només n'hi ha alguns que resultin vàlids i eficients. A continuació mostrem doncs en detall les possibilitats i limitacions dels diversos tipus d'algorismes existents en aplicar-los als problemes NP-Complets.

#### **3.2. Els algorismes**

Prenent com a punt de mira en tot moment la resolució eficient del problema, és a dir, que el temps de càlcul que es requereix per a trobar una solució satisfactòria sigui raonable, és evident que l'enfoc que s'utilitzi en la cerca, l'algorisme que s'apliqui, ens aproparà o ens allunyarà d'aquest objectiu.

##### 3.2.1. Cerca exhaustiva

D'aquí que la utilització de mètodes basats en una cerca exhaustiva i sistemàtica de tot l'espai d'estats, que constitueixen els algorismes més simples, és evident que resulta impracticable per les pròpies característiques dels problemes NP-Complets.

### 3.2.2. Cerca local o aleatòria

Mentre que, a l'altre extrem, mètodes basats en una exploració reduïda i a l'atzar de l'espai d'estats, com solen fer alguns algorismes de cerca local reduïda o de cerca aleatòria, tampoc ens acosten a l'objectiu, doncs no aporten cap garantia de conduir-nos a una solució bona.

Alguns algorismes de cerca local consisteixen en explorar solament els estats d'una zona limitada de l'espai de solucions i considerar com a millor solució la que s'obtingui d'avaluar el cost de la funció objectiva en els estats d'aquell grup reduït de solucions. Però difícilment podem assegurar que la zona escollida contingui la millor solució.

Mentre que la cerca aleatòria realitza un "mostreig" a l'atzar de l'espai d'estats i avalua el cost de cadascuna de les solucions triades, quedant-se amb la millor. I, per tant, resulta inútil per a grans dimensions en l'espai d'estats.

Per aquests mètodes, fins i tot encara que s'augmenti bastament el nombre d'estats explorats en detriment de l'eficiència, res ens assegura que trobarem una solució, ni tan sols sub-òptima, del problema.

### 3.2.3. Algorismes heurístics

Una altra opció que resulta molt més òptima per aconseguir l'objectiu desitjat d'eficiència i èxit consisteix a utilitzar algorismes heurístics especialment adaptats al problema que abordem.

No obstant, sovint no és gens fàcil d'aconseguir trobar un mètode que condueixi eficaçment vers una solució sense el perill de que s'encalli en un mínim local del qual no pugui sortir-se'n. I, per la pròpia naturalesa d'aquest tipus d'algorismes, l'estratègia utilitzada difícilment pot ser aplicada a d'altres problemes d'optimització combinatòria. Això fa que l'opció

d'utilitzar algorismes heurístics, a la pràctica, requereixi d'una idea brillant i, sovint, d'un estudi matemàtic intensiu de les propietats del problema concret i específic amb que volem treballar per tal de trobar claus que ens condueixin cap a una solució. I, per tant, un enfoc d'aquest tipus no ens assegura l'èxit de la temptativa. Podríem dir que és ineficient pel que fa al temps d'anàlisi prèvia del problema que es requereix fins a trobar un mètode realment adaptat i eficient, que pot no existir. I que a més, difícilment serà adaptable a cap altre problema, de vegades inclusiu a una instància del mateix problema que difereixi només en alguna de les seves dimensions.

No obstant, pels problemes que ens ocupen, hi ha nombrosos treballs en que s'han aplicat algorismes d'aquesta classe amb èxit. Entre ells es donen tres tipus de mètodes ben diferenciats: els constructius, els de partició i els d'aproximació successiva.

Els primers consisteixen a construir la solució directament, mitjançant l'aplicació d'unes regles de construcció específiques que cal seguir. La clau d'aquests algorismes rau en trobar unes regles de construcció veritablement "equilibrades": suficientment "restrictives", per tal que en cap cas ens puguin desviar de la solució òptima, però alhora prou senzilles i "amplies", per tal que puguem anar-les aplicant a cada pas de la construcció sense la possibilitat de quedar encallats en cap punt degut a que poden més les restriccions que la construcció.

Els mètodes de partició es fonamenten en trobar una divisió del problema o de la possible solució en parts més petites i abordar la resolució òptima d'aquests sub-problemes. Un cop trobades les solucions per a cada un dels

sub-problemes, cal però unir-les entre sí per formar la solució global del problema original, i això últim no sempre és possible fer-ho sense que aquesta unió afecti a la qualitat de les solucions parcials trobades. De fet, això només és possible quan els sub-problemes són disjunts, la qual cosa no acostuma a ser fàcil d'obtenir.

Els mètodes del tercer tipus es dediquen a realitzar una cerca progressiva de la solució, partint d'una solució inicial qualsevol i movent-se en tot moment cap a la millor solució veïna, fins que no se'n trobi ja cap de millor. Així, per a l'aplicació d'aquests algorismes, es requereix, d'una banda, de la definició d'una funció que ens permeti avaluar en tot moment la "qualitat" de qualsevol de les solucions de l'espai d'estats d'una forma "objectiva" (en el sentit que ha de ser independent del camí traçat i dels veïns que pugui tenir); i ,d'altra banda, de l'establiment d'unes regles de transformació de la solució actual per obtenir les possibles solucions veïnes. En aquests mètodes, destaca doncs la importància de la selecció d'una "bona" funció de cost, així com de les regles de transformació que ens permetin arribar a la solució òptima partint de qualsevol estat inicial.

El més representatiu i àmpliament utilitzat d'aquests mètodes és l'anomenat *down-hill*. En ell, a cada iteració el cost de la funció objectiva minva respecte el cost anterior, però fàcilment pot conduir-nos a una solució sub-òptima si es queda encallat en un mínim local de la funció de cost. I, per tant, no ens assegura l'èxit en la cerca. És per això que, sovint, se sol aplicar l'algorisme varies vegades partint d'una solució inicial diferent cada cop escollida aleatòriament. I finalment, de totes les solucions obtingudes, s'escull la de menor cost.

La diferència d'aquest tercer tipus d'algorismes heurístics, els d'aproximació successiva, respecte dels altres dos tipus és que són fàcilment aplicables a un ampli ventall de problemes d'optimització, degut a la generalitat i senzillesa del seu enfoc. Això els fa molt més "útils" que la resta de mètodes que hem descrit fins al moment, o usant el terme que s'utilitza per a descriure aquesta característica, més robustos. Tot i així, és evident que per a instàncies grans dels problemes, i més encara per als de tipus NP-Completo que ens ocupen, tampoc aquests algorismes resulten eficients. No només perquè esdevenen totalment impracticables en temps d'execució, sinó sobretot perquè no ens aporten cap garantia de conduir-nos cap a la solució òptima. Donada la seva definició, no tenen "capacitat" per assegurar que amb un temps d'execució tendint a infinit obtindríem la solució òptima. No tenen "**direcció**".

#### 3.2.4. Algorismes naturals

Per tot l'exposat anteriorment, hi ha d'haver un altre enfoc que ens dugui a la consecució del nostre objectiu, la resolució eficient del problema, d'una forma molt més fiable. Aquest és l'enfoc d'un conjunt d'algorismes que tenen en comú el fet que es fonamenten tots ells en la imitació d'algun mecanisme o model funcional utilitzat per la natura per a la resolució de determinats "problemes". Al conjunt heterogeni d'aquest tipus d'algorismes ens referirem aquí sota el nom d'**algorismes naturals**, per tal de distingir-los de la resta d'algorismes.

L'avantatge d'aquests algorismes naturals és que, a més de ser realment eficients, com demostra la pròpia Natura, pel fet de ser conceptualment tant senzills són fàcilment aplicables a tot tipus de problemes d'optimització i

resulten relativament fàcils d'implementar.

Entre ells es troben els algorismes de Xarxes Neuronals, els Algorismes Genètics, els algorismes basats en models multi-agent (com el de "Les Formigues", *Ants*), la Cerca Tabú i la Recuita Simulada (*Simulated Annealing*). Exposem a continuació una breu descripció de cada un d'ells.

### **3.2.4.1. Les Xarxes Neuronals**

Els algorismes de **Xarxes Neuronals** estan basats en el model de J.J. Hopfiel i D.W.Tank [28] que consisteix en construir una xarxa formada per un bon nombre d'elements o "neurons" artificials. Cada neurona funciona com una unitat que rep un estímul d'entrada a través de les connexions d'entrada, el processa segons una funció pròpia de resposta i genera un senyal de sortida que distribueix per totes les seves connexions de sortida cap a les seves veïnes.

La xarxa neuronal més simple està formada per tres capes de neurones: una capa de neurones d'entrada, una capa de neurones internes i una tercera capa de neurones de sortida. Cada neurona d'entrada està connectada amb les neurones internes mitjançant un factor de ponderació que varia per a cada connexió. Les neurones internes sumen els estímuls ponderats que reben de totes les seves connexions, responen a l'estímul total mitjançant la seva funció de resposta i la distribueixen per les seves connexions amb les neurones de la capa de sortida que tenen també un factor de ponderació. La resposta global de la xarxa està formada pel conjunt de senyals de sortida i els seus nivells.

L'interessant de les xarxes neuronals és la capacitat que tenen d'aprendre. Per fer-ho, es fan variar els valors de ponderació de les connexions de la xarxa per tal que la resposta global s'aproximi cada cop més a la òptima. I, és

per això, que en aquests algorismes cal definir-hi una mesura de la "qualitat" de la resposta a l'estímul global d'entrada aportat a la xarxa. Llavors, en funció d'aquesta "qualitat" de la resposta és que s'adaptin les ponderacions mitjançant un algorisme de "propagació cap enrera", fins assolir una xarxa que respongui òptimament a l'estímul d'entrada o "problema" plantejat.

### **3.2.4.2. Els Algorismes Genètics**

Uns altres algorismes que han tingut una gran acceptació i que han sigut aplicats a una ampla gamma de problemes d'optimització són els anomenats **Algorismes Genètics**. Partint del model proposat per J.H.Holland als anys 60, n'han anat apareixent nombroses variants que s'han aplicat amb èxit en problemes molt diversos (veure [24] i [27] per una descripció exhaustiva).

En un algorisme genètic el punt de partida és una col·lecció, que rep el nom de població, de possibles solucions (o individus) generades aleatòriament. Es fa una valoració del cost de cada una d'elles per tal de determinar les millors solucions. Llavors es procedeix a crear una nova generació (iteració), això s'aconsegueix a través d'una reproducció, un encreuament i una mutació de les millors solucions de cada generació. Normalment es pren constant el nombre d'individus de cada generació. A partir d'una generació es construeix una nova generació mitjançant l'encreuament de solucions de la generació antiga anomenada pare. Aquest encreuament es realitza de manera que dues solucions pares donen dues solucions fills mitjançant l'intercanvi aleatori de fragments de les solucions pare. Això mateix es fa per a tota la població. A més, una altra aleatorietat s'introdueix mitjançant el mecanisme anomenat "mutació" que modifica lleugerament les solucions generades i assegura que l'algorisme no es quedi atrapat en un mínim local.

Les operacions d'encreuament i mutació es realitzen d'acord amb una certa probabilitat establerta. Això garanteix que algunes solucions de la generació actual continuaran presents a la següent generació. Un cop la nova generació s'ha creat, el cost de totes les solucions o individus es torna a avaluar i el procés es repeteix. En cada generació es guarda la millor solució. L'algorisme finalitza quan el resultat s'estabilitza o es troba la solució òptima si aquesta pot ser identificada.

#### **3.2.4.3. Els Algorismes Multiagent**

Els **algorismes multiagent** tenen el seu major exponent en el model presentat per J.Ozón i F.Comellas el 1995 anomenat *Ants*. Aquest aborda la resolució del problema global d'optimització mitjançant la col·laboració simultània d'un cert nombre de sub-programes o agents anomenats "formigues" que s'encarreguen de realitzar una optimització local. Aquests agents s'ocupen d'alterar localment la solució vers una solució de menor cost i després es "desplacen" des del lloc on es troben cap a l'indret veí que té major cost local o bé cap a un altre veí, amb una certa probabilitat. I un cop s'ha desplaçat, realitza altra cop la mateixa tasca d'optimització local. D'aquesta manera, l'acció conjunta d'optimització local de tots els agents condueix vers la solució globalment òptima.

#### **3.2.4.4. La Cerca Tabú**

La **Cerca Tabú** utilitza un esquema molt similar al dels algorismes de cerca successiva amb la variant que incorpora una llista dels darrers estats que s'han explorat, de manera que en escollir el següent estat cap on desplaçar-se, exclou aquells que estan presents a la llista. Lògicament, la llista ha de tenir una longitud limitada pel fet que una llista excessivament llarga, d'una

banda alentirà considerablement el temps trigat a repassar-la i, d'altra banda suposarà un elevat cost en memòria durant l'execució. És per això que aquesta és alhora la seva clau i la seva limitació més important en el moment de la seva implementació i execució.

### **3.2.4.5. La Recuita Simulada**

Finalment, la tècnica de la **Recuita Simulada**, en la qual centrarem el nostre projecte, va ser introduïda per S.Kirkpatrick, C.D.Gelatt i M.P.Vecchi l'any 1983 en [30] i deriva d'un algorisme conegut per Metrópolis [31]. De fet pot considerar-se també com una variant de l'algorisme *down-hill* ja que accepta sempre canvis de reducció del cost, no obstant, per evitar caure en un mínim local, accepta canvis d'augment del cost amb una certa probabilitat donada per l'exponencial de Boltzman  $\exp \{ \Delta E / k \cdot t \}$ , on  $k$  és la constant de Boltzman i el paràmetre  $t$  és conegut per temperatura. La temperatura en un inici és alta, després baixarà suaument estabilitzant-se i conduint l'algorisme vers la solució cercada.

Com es pot observar, el seu mecanisme té un paral·lelisme total amb el fenomen físic de la **crystal·lització** de substàncies líquides. Aquesta s'esdevé justament per un refredament progressiu de la substància, des d'una temperatura inicial alta, a la qual es troba en estat líquid, i cap a la temperatura de congelació, a la qual el cos acaba formant un cristall perfecte. Al principi, totes les molècules es troben en un estat "caòtic", un estat de molta energia on es distribueixen lliurement. En aquest estat, qualsevol variació en la posició d'una molècula és pràcticament inapreciada pel que fa a la variació en l'energia global del cos i, per tant, és gairebé sempre "acceptada". Però, a mesura que la temperatura es fa disminuir, les molècules

van integrant-se de forma natural (pel principi de mínima energia) en una estructura conjunta que condueix a un estat cristal·lí, on l'energia global és molt menor a la de l'estat líquid. Conforme ens apropem a la temperatura de congelació, la variació en la posició d'una molècula afecta de forma cada cop més important en l'energia global del cos i, per tant, és acceptada de manera més restrictiva. Aquest mecanisme de restricció progressiva de les variacions és el que es simula mitjançant l'ús de l'exponencial de Boltzman, amb la qual s'obté la probabilitat d'acceptació dels increments de temperatura deguts a les variacions en l'estructura.

Com podeu veure, el seu enfoc és tan simple i consistent en sí mateix que el fa totalment independent del problema tractat i, per tant, adaptable pràcticament a qualsevol problema d'optimització. A més, es demostra matemàticament que l'esquema essencial de l'algorisme correspon a una seqüència de cadenes de Markov, que portada fins a l'infinit **assegura** la convergència vers la solució òptima [1].

L'adaptació a cada problema es realitza a través de la selecció adequada d'una funció de cost que representi per un cost nul (o mínim absolut) l'estat òptim del sistema, és a dir, l'equivalent a l'estructura de **crystal·l perfecte**.

Al proper capítol exposem amb més detall les característiques d'aquest algorisme. I mostrem, a través d'una anàlisi qualitativa de la seva resposta enfront d'un problema "tipus", la seva capacitat per a convergir vers la solució en aplicar-lo a problemes d'optimització.

Al capítol 7 mostrarem ja en tot detall la implementació pràctica i concreta d'aquest algorisme, la Recuita Simulada, aplicat al problema de les *cages* i al problema ( $\Delta, D$ ) tal com l'hem desenvolupada en aquest projecte.

#### 4. CONVERGÈNCIA DE L'ALGORISME DE LA RECUITA SIMULADA

En aquest capítol farem una interessant exposició comparativa de perquè la Recuita Simulada resulta tan eficient per a la resolució de problemes d'optimització complexos amb independència de les característiques que presenti l'espai d'estats amb el que es tracti o, millor dit, la topologia que es deriva de la funció de cost utilitzada.

La Recuit Simulada, per les seves característiques, presenta un component de tipus estocàstic que permet fer una anàlisi teòrica de la seva convergència assintòtica. Aquesta anàlisi estableix el paral·lelisme de l'algorisme amb una seqüència de cadenes de Markov, i demostra que aquest té les característiques necessàries per convergir vers la solució òptima conforme el paràmetre de control (que aquí correspon al llindar d'acceptació d'empitjoraments en el cost d'una nova solució) tendeix a zero. No obstant, aquí no farem l'anàlisi des del punt de vista teòric, sinó des d'un punt de vista qualitatiu, mostrant com és capaç de progressar eficaçment davant d'un problema de cerca genèric.

L'algorisme de la Recuita Simulada s'inclou dins d'una classe d'algorismes de cerca local anomenats **algorismes de llindar** (*threshold algorithms*). Per a descriure'ls utilitzarem les següents consideracions:

Suposem que  $(S, f)$  sigui una instància d'un problema de minimització combinatoria amb el conjunt de solucions  $S$  y funció de cost  $f: S \rightarrow \mathbb{R}$  (nombres reals). A més, definirem  $V: S \rightarrow \mathcal{F}(S)$  com una funció de veïnatge, que defineix per cada  $i \in S$  un conjunt  $V(i) \subset S$  de solucions veïnes. La qüestió

consisteix a trobar la solució òptima  $i^* \in S$  que minimitza el cost de totes les solucions sobre  $S$ .

Aquests tipus d'algorismes tenen un mecanisme que es pot descriure mitjançant el següent pseudo-codi:

```
procediment ALGORISME_LLINDAR
inici
    INICIALITZAR( $i_{inici}$ );
     $i := i_{inici}$ ;
     $k := 0$ ;
    repetir
        GENERAR( $j$  de  $V(i)$ );
        si  $f(j) - f(i) < t_k$  llavors  $i := j$ ;
         $k := k + 1$ ;
    fins ATURAR;
final;
```

El procediment INICIALITZAR escull una solució inicial de  $S$ , el procediment GENERAR escull una solució del veïnatge de la solució actual, y el procediment ATURAR avalua un criteri d'aturada que determina la finalització de l'algorisme. Els algorismes de llindar seleccionen tota l'estona un veí de la solució actual i comparen la diferència en cost entre aquestes solucions amb un llindar. Si la diferència en el cost és menor que el llindar, el veí substitueix la solució actual. Altrament, la cerca segueix amb la solució actual. La seqüència ( $t_k \mid k=0,1,2,\dots$ ) denota els llindars, on  $t_k$  és utilitzat a la iteració  $k$  de l'algorisme de cerca local.

Distingirem entre els següents tres tipus d'algorismes de lllindar depenent de la naturalesa del lllindar:

\* *Millora iterativa*:  $t_k = 0, k= 0, 1, 2, \dots$ . Clarament aquesta és una variant de la clàssica cerca local per gradient on només els veïns que redueixen el cost són acceptats. L'exemple més típic d'aquesta classe d'algorismes és l'anomenat *down-hill*.

\* *Acceptació per lllindar*:  $t_k = c_k, k=0,1,2, \dots$ , on  $c_k \geq 0, c_k \geq c_{k+1}$   $\lim_{k \rightarrow \infty} c_k = 0$

L'acceptació per lllindar utilitza una seqüència no-creixent de lllindars determinístics. Degut a l'ús de lllindars positius, solucions veïnes amb cost major s'accepten d'una forma limitada. En el decurs de l'execució de l'algorisme, els valors del lllindar es fan disminuir progressivament, eventualment fins a 0, moment en el qual només millores són acceptades.

\* *La Recuita Simulada*:  $t_k =$  una variable aleatòria amb valor esperat  $E(t_k)=c_k \in \mathbb{R}, k=0,1,2, \dots$ . Els  $t_k$  segueixen una funció de distribució de probabilitat  $F_{c_k}$  sobre  $\mathbb{R}$ . La Recuita Simulada utilitza lllindars aleatoris amb valors entre zero i infinit, i la probabilitat de que un lllindar  $t_k$  sigui al menys  $y \in \mathbb{R}$  ve donada per  $\mathbf{P}_{t_k} \{t_k \leq y\} = F_{c_k}(y)$ . Això implica que tota solució veïna pot ser escollida amb una probabilitat positiva per substituir a l'actual solució. A la pràctica, la funció  $F_{c_k}$  s'escull de manera que solucions que corresponen a grans increments del cost tenen una baixa probabilitat de ser acceptades, mentre que solucions que corresponen a petits increments del cost tenen una major probabilitat de ser acceptades.

En la versió original de la Recuita Simulada, Kirpatrick, Gelatt i Vecchi [30]

prenen per a  $F_{c_k}$  la distribució exponencial negativa amb paràmetre  $1/c_k$ .

Aquesta elecció és idèntica al següent criteri d'acceptació:

Per dues solucions  $i, j \in S$  la probabilitat d'acceptar  $j$  partint de  $i$  a la  $k$ -èssima iteració ve donada per:

$$P_{c_k} \{ \text{acceptar } j \} = \begin{cases} 1 & \text{si } f(j) \leq f(i) \\ \exp (f(i) - f(j) / c_k) & \text{si } f(j) > f(i) \end{cases}$$

El paràmetre  $c_k$  és utilitzat a la Recuita Simulada com a *paràmetre de control*, i juga un paper molt important en l'anàlisi de la convergència de l'algorisme. A partir d'aquí traurem el subíndex  $k$  del paràmetre de control si no és estrictament necessari.

Per analitzar el comportament d'aquests *algorismes de llindar* considerarem el següent problema d'optimització combinatoria, que és una versió simplificada del problema introduït per Lundy i Mees [31]. Fem que el conjunt de solucions vingui donat per  $S = \{0, 1, \dots, N\}$  amb  $1 \ll N$ , i la funció de cost  $f: S \rightarrow \mathbb{R}$  vingui donada per:

$$f(i) = i - \lfloor i/n \rfloor \delta,$$

amb  $n \in \mathbb{V}$ ,  $1 \ll n \ll N$ ,  $\delta \in \mathbb{R}$ , i  $1 < \delta < n$ . La Figura 4.1 il·lustra la funció de cost per  $\delta = 2$ . El problema consisteix a trobar l'element de  $S$  amb mínim cost.

Aquesta formulació del problema pot ser generalitzada a conjunts de solucions de  $d$  dimensions, de la següent manera: Si  $S = \{0, 1, \dots, N\}^d$ , la funció de cost ve donada per  $f(i_0, i_1, \dots, i_{d-1}) = r - \lfloor r/n \rfloor \delta$ , amb  $r = \max\{i_0, i_1, \dots, i_{d-1}\}$ .

Lundy i Mees [31] discuteixen una versió bidimensional d'aquest problema.

Definim a més una funció de veïnatge donada per  $V(i) = \{i-1, i, i+1\}$  per  $i \in \{1, 2, \dots, N-1\}$ ,  $V(0) = \{0,1\}$  i  $V(N) = \{N-1, N\}$ . A continuació considerem el comportament dels *algorismes de llindar* en aplicar-los a aquest problema, per a dos eleccions de  $\delta$ , diguem  $\delta \approx 1$  i  $\delta \gg 1$ , corresponent a situacions on petits o grans deterioraments en el cost son necessaris per a escapar d'un mínim local, respectivament.

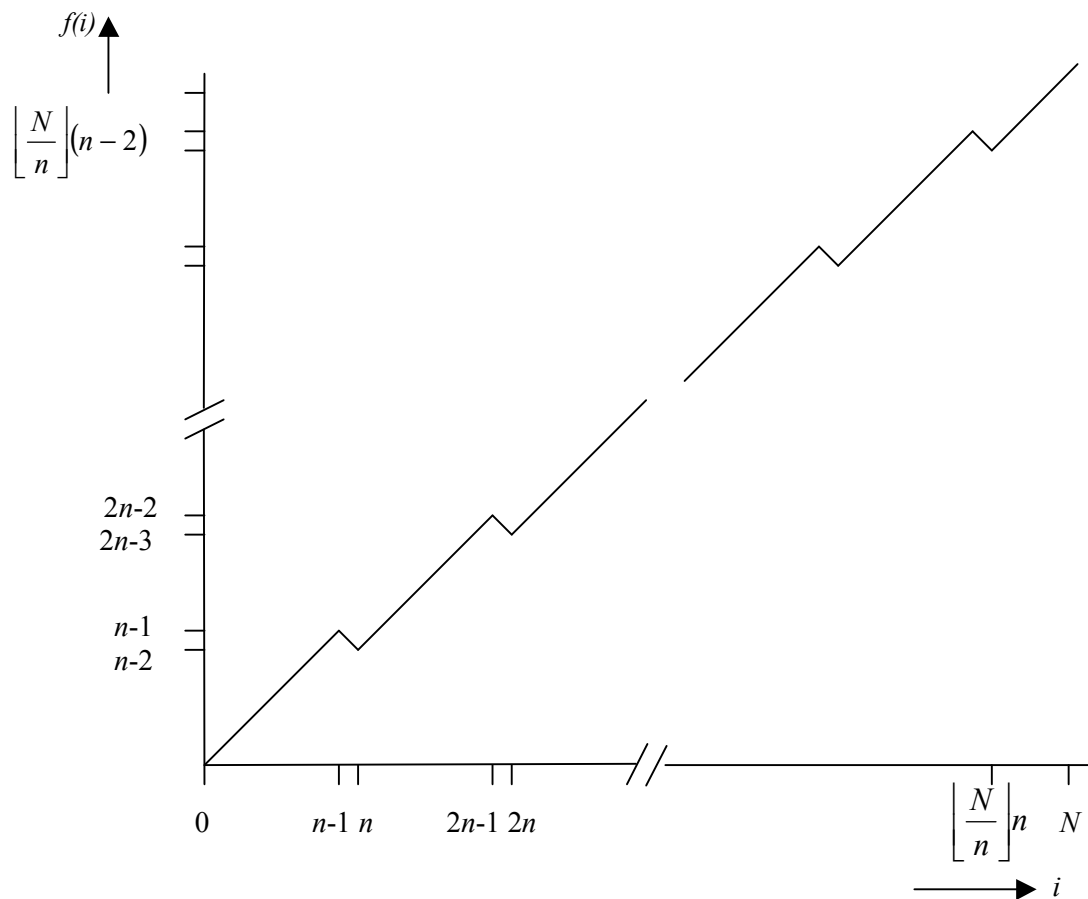


Figura 4.1 Funció de cost  $f(i)$  per  $\delta = 2$

El comportament de la *millora iterativa amb múltiples re-inicis* és idèntica per ambdós casos. Una sola repetició es descriu de la següent manera. Aquesta escull aleatòriament una solució inicial  $i$  de  $S = \{0, 1, \dots, N\}$  i finalitza en un mínim local donat per  $i' = \lfloor i/n \rfloor n$ . Clarament, l'algorisme únicament troba un òptim global en una iteració donada si la solució inicial es troba a  $\{0, 1, \dots, n-1\}$ . El nombre esperat d'iteracions per obtenir un òptim global es demostra ser  $O(M)$ , amb  $M = N/n$ , per al cas unidimensional, i  $O(M^d)$  per al cas amb  $d$  dimensions. Això il·lustra que la *millora iterativa amb múltiples re-inicis* sempre pot trobar un òptim global, però que l'esforç requerit pot ser molt gran, equivalent a enumerar totes les solucions de  $S$ .

Considerem ara el comportament de *l'acceptació per llindar* i la *Recuita simulada* per al cas  $\delta \approx 1$ .

Per a l'algorisme *d'acceptació per llindar*, considerarem tres situacions, depenent del valor dels llindars  $t_k$ . Si  $t_k \geq 1$ , cada transició d'una solució  $i$  a un dels seus veïns  $i-1$ ,  $i+1$  és acceptada, i l'algorisme realitza un camí aleatori per l'espai de solucions  $S$ . Si  $\delta-1 \leq t_k < 1$ , l'algorisme només accepta transicions des de la solució  $i$  al seu veí  $i-1$ . Fins i tot en aquest cas, l'algorisme es comporta com un algorisme d'optimització. Si  $t_k < \delta-1$ , l'algorisme es comporta com la *millora iterativa* i acaba al mínim local  $i'$  que correspon a la solució actual  $i$ .

Amb aquest exemple podem observar com *l'algorisme d'acceptació per llindar* es pot comportar molt bé si s'escolleixen adequadament els llindars  $t_k$ . Si  $\delta-1 \leq t_k < 1$ , l'algorisme troba un òptim global en  $O(N)$  transicions o, assumint que  $n$  és constant, en  $O(M)$  transicions. Noteu que aquest nombre de transicions no depèn del nombre de dimensions. En conseqüència, si el llindar és escollit adequadament, *l'acceptació per llindar* es comporta

considerablement millor que *la millora iterativa amb múltiples re-inicis*.

Per al cas en que  $\delta \approx 1$ , la *Recuita simulada* es comporta de forma similar a *l'acceptació per llindar*. Això és fàcil de mostrar de la següent manera. Si  $c \geq 1$ , la *Recuita simulada* realitza més o menys un camí aleatori, però amb una lleugera preferència per solucions amb un cost baix. Si  $\delta - 1 \leq c \ll 1$ , l'algorisme pràcticament només accepta transicions des de la solució  $i$  cap al seu veí  $i-1$ . En aquest cas, l'algorisme també es comporta com un algorisme d'optimització, requerint  $O(N)$  transicions per obtenir un òptim global. Si  $c < 1$ , l'algorisme probablement acaba en el mínim local  $i'$  que correspon a la solució actual  $i$ , encara que també existeix una petita probabilitat obtenir una solució millor.

A continuació considerem el cas en que  $\delta \gg 1$ . En aquest cas, *l'acceptació per llindar* en promig no pot trobar millors solucions que la *millora iterativa (amb un sol intent)*. Això es pot mostrar de la següent manera. Si  $t_k \geq \delta - 1$ , l'algorisme accepta qualsevol transició proposada. I conseqüentment, pel fet que  $t_k \geq \delta - 1$ , *l'acceptació per llindar* realitza un camí aleatori pel conjunt de solucions. Si  $t_k < \delta - 1$ , l'algorisme es comporta al menys tant malament com la *millora iterativa (amb un sol intent)*. Sigui  $i$  la solució actual en aquest instant. Llavors, la millor solució que es pot obtenir és  $i' = \lfloor i/n \rfloor$ . L'algorisme fins i tot té una probabilitat positiva d'acabar en un mínim local  $j'$  amb  $j' > i'$ . Donat que, en promig, la solució inicial és idèntica a la solució obtinguda després del camí aleatori, arribem a la conclusió de que, en promig, *l'acceptació per llindar* no pot obtenir millors solucions que la *millora iterativa (amb un sol intent)*. I a més, fins i tot si a l'algorisme se li dóna temps suficient i els llindars són escollits acuradament, *l'acceptació per llindar* no pot garantir trobar un mínim global, mentre que la *Recuita simulada* sempre té una probabilitat positiva d'assolir un òptim global, quan se li dóna temps

suficient. L'estudi estocàstic demostra que la *Recuita simulada* asimptòticament troba un òptim global amb probabilitat 1.

En resum, aquest estudi qualitatiu ens duu a les següents conclusions:

- \* Per alguns problemes d'optimització, el nombre esperat de transicions necessari per assolir un mínim global és molt menor per a la *Recuita simulada* i per a l'*acceptació per llindar* que per a la *millora iterativa amb múltiples re-intents*.
- \* Per d'altres problemes d'optimització, el cost esperat per a una solució final obtinguda amb l'*acceptació per llindar* no és millor que el cost esperat d'una solució final obtinguda per la *millora iterativa (amb un sol intent)*.

Així doncs, l'interès en la *Recuita simulada* ve motivat pel fet que, en comparació amb la *millora iterativa amb múltiples re-intents* i l'*acceptació per llindar*, el seu comportament és menys depenent de la topologia específica del "paisatge de la funció de cost". I aquest fet li confereix una enorme capacitat d'adaptabilitat a tot tipus de problemes, una enorme *robustesa*.

En el capítol 7 veurem quina ha sigut l'estratègia seguida en aquest projecte per aplicar la *Recuita simulada* a l'obtenció de topologies de xarxes d'interconnexió amb característiques molt específiques, com les de les *cages* i les de les solucions al problema  $(\Delta, D)$ .

## 5. XARXES D'INTERCONNEXIÓ AMB GIRTH FIXAT: LES CAGES

L'objectiu del projecte, com s'ha dit, se centra principalment en l'anàlisi i en la cerca de noves solucions pel problema de les *cages cúbiques* i per al problema  $(\Delta, D)$ , mitjançant l'aplicació de la tècnica de la Recuita Simulada.

En aquest capítol presentem la descripció matemàtica del problema de les *cages cúbiques*, així com també quin era l'estat de les investigacions relacionades amb aquest problema, i les fites assolides fins al moment d'iniciar el nostre projecte. Més tard, mostrem també una descripció general d'alguns dels mètodes emprats per part de diversos investigadors d'arreu del món per a l'obtenció d'algunes d'aquestes fites i solucions, amb la intenció de mostrar els diversos tipus d'enfocs que s'havien aplicat al problema fins llavors.

### 5.1. *Descripció*

S'anomena *cage* (que traduït literalment significa "gavia") o, més específicament,  $(k, g)$ -*cage* al graf regular de grau  $k$  i *girth*  $g$  amb menor nombre de vèrtexs possible. És a dir, el graf de menor ordre que compleix aquestes dues condicions de grau i *girth*.

És possible, però, per la pròpia definició de *cage*, que hi hagi en realitat més d'una *cage* per a un grau i *girth* donats. Tots les *cages* d'un grau i *girth* tindran necessàriament el mateix ordre, però diferiran entre elles en les unions entre els seus vèrtexs. És a dir, no existirà cap isomorfisme mitjançant el qual puguem convertir una de les *cages* en una altra. Aquest punt és important

per a la determinació de quin és el nombre real de grafs diferents que corresponen a *cages* d'un grau i *girth* determinats.

En el nostre cas, ens hem centrat en les  $(3,g)$ -*cages*, o *cages* cúbiques, sense que això tregui generalitat a l'aplicació feta. Entre aquestes, el nostre treball s'ha aplicat principalment a l'anàlisi de les *cages* de *girths* entre 8 i 16.

Per a les *cages* de qualsevol grau es pot trobar fàcilment una fita inferior ( $n_0$ ) de l'ordre que han de tenir les *cages* en funció del *girth*, simplement contant el nombre de vèrtexs en la matriu de distàncies del graf respecte d'un vèrtex. El càlcul difereix lleugerament depenent de si el *girth* és parell o senar:

Si el *girth* és senar, llavors  $g=2r+1$  per algun enter positiu  $r$ . Sigui  $v \in V(G)$ . Per  $1 \leq i \leq r$ , el número de vèrtexs a distància  $i$  de  $v$  és  $k(k-1)^{(i-1)}$ . Així doncs,

$$n_0 = 1 + k + k(k-1) + k(k-1)^2 + \dots + k(k-1)^{r-1} = 1 + k[(k-1)^r - 1] / (k-2).$$

Si el *girth* és parell, llavors  $g=2l$ , on  $l \geq 2$ . Sigui  $e=uv \in E(G)$ . Per  $1 \leq i \leq l-1$ , el número de vèrtexs a distància  $i$  de  $u$  o  $v$  és  $2(k-1)^i$ . Doncs,

$$n_0 = 2 + 2(k-1) + 2(k-1)^2 + \dots + 2(k-1)^{l-1} = 2[(k-1)^l - 1] / (k-2).$$

Per tant, per a *girths* senars, es té:

$$n_0 = 1 + k[(k-1)^{(g-3)/2}] / (k-2).$$

i per a *girths* parells:

$$n_0 = 2[(k-1)^{g/2-1}] / (k-2).$$

Qualsevol *cake* que realment assoleixi aquesta fita, anomenada *fita de Moore*, correspon de fet a un graf molt especial: a un *graf de Moore*, si el *girth* és senar, i a un Polígon generalitzat, si el *girth* és parell.

A la pràctica, hi ha molt poques *cages* que acompleixin aquesta fita (per a les *cages* cúbiques, només les de *girth* 3, 4, 5, 6, 8 i 12), i de fet, en general, sabem molt poc sobre l'ordre i les característiques de les *cages* de cada *girth*.

Fins al moment, no s'ha trobat encara la forma matemàtica teòrica de demostrar, per a qualsevol *girth* donat, ja no quina és l'estructura de les seves *cages*, sinó ni tan sols quin és necessàriament el seu ordre.

Això només s'ha pogut determinar de forma exacta (en molts casos, però, mitjançant l'ús de mètodes computacionals de cerca exhaustiva, no pas mitjançant mètodes teòrics) per a les *cages* (cúbiques) de *girths* entre 3 i 12. A partir de *girth* 13, únicament ha sigut possible trobar una fita inferior de l'ordre de les *cages*, per sota de la qual es demostra que és impossible l'existència d'una solució. Però, per cap d'aquests *girths* grans no s'ha trobat cap *cake* que assoleixi aquesta fita inferior en el seu ordre. I, per tant, tan sols es coneix el rang de valors que pot prendre l'ordre de les seves *cages*, que es troba sempre entre aquesta fita inferior i l'ordre de la millor solució trobada fins al moment. En cap d'ells s'ha pogut demostrar encara que la solució trobada correspon efectivament a una de les *cages* per aquell *girth*.

**Donant-se aquesta situació, resulta especialment interessant poder trobar un mètode o algorisme que resulti eficient per a abordar el problema i que pugui ser aplicat amb èxit per a qualsevol instància del mateix. Aquest és**

## el repte del present projecte.

Les fites assolides fins a l'inici d'aquest projecte es poden veure en la taula següent. Aquesta taula és la recopilació de tots els treballs apareguts sobre el tema feta per G.Royle, professor del departament de *Computer Science* de la *University of Western Australia*.

The cages	n(3,g)	Range	Number	Reference	Notes
<a href="#">(3,3)-cages</a>	4	4	1	K_4	-
<a href="#">(3,4)-cages</a>	6	6	1	K_3,3	-
<a href="#">(3,5)-cages</a>	10	10	1	Petersen	<a href="#">more</a>
<a href="#">(3,6)-cages</a>	14	14	1	Heawood	<a href="#">more</a>
<a href="#">(3,7)-cages</a>	22	24	1	McGee graph	<a href="#">more</a>
<a href="#">(3,8)-cages</a>	30	30	1	Tutte's 8-cage	<a href="#">more</a>
<a href="#">(3,9)-cages</a>	46	58	18	Brinkmann/McKay/Saager	<a href="#">more</a>
<a href="#">(3,10)-cages</a>	62	70	3	O'Keefe/Wong	<a href="#">more</a>
<a href="#">(3,11)-cages</a>	94	112	1+	McKay/Myrvold	<a href="#">more</a>
<a href="#">(3,12)-cages</a>	126	126	1	Generalized hexagon	<a href="#">more</a>
<a href="#">(3,13)-cages</a>	190	202-272	1+	McKay/Myrvold - Hoare	<a href="#">more</a>
<a href="#">(3,14)-cages</a>	254	258-406	1+	McKay - Biggs	<a href="#">more</a>
<a href="#">(3,15)-cages</a>	382	384-620	1+	Biggs	<a href="#">more</a>
<a href="#">(3,16)-cages</a>	510	512-990	1+	Biggs	<a href="#">more</a>
<a href="#">(3,17)-cages</a>	766	768-6072	1+	Biggs	<a href="#">more</a>
<a href="#">(3,18)-cages</a>	1022	1024-4218	1+	Hoare	<a href="#">more</a>
<a href="#">(3,19)-cages</a>	1534	1536-4324	1+	H(47)	<a href="#">more</a>
<a href="#">(3,20)-cages</a>	2046	2048-14910	1+	S(71)	<a href="#">more</a>

## 5.2. Altres mètodes utilitzats fins ara

Per a l'obtenció d'aquestes fites i de les *cages* per a cada *girth* obtingudes fins al moment, s'han utilitzat varietat de mètodes i d'enfocs. Entre ells es troben mètodes de tipus més teòric amb altres de més heurístics o, fins i tot, alguns de purament exhaustius. Descrivim a continuació de forma breu algunes característiques d'aquests mètodes, que si bé parteixen d'enfocs totalment diferents a l'utilitzat en aquest projecte, empen tècniques molt enginyoses

que exploren a fons les propietats d'aquests grafs i que ens han suggerit alguna de les tècniques que hem utilitzat per a millorar l'eficiència del nostre algorisme.

G.Exoo [22] exposa un mètode per a l'obtenció de *cages* de *girths* 14, 15 i 16, que parteix d'un determinat nombre d'arbres cúbics idèntics d'ordre molt menor al de les *cages* buscades. Els arbres cúbics són arbres en els quals cada vèrtex té grau 3 (vèrtexs interns) o grau 1 (vèrtexs terminals). El mètode consisteix a establir el patró que segueixen les unions dels vèrtexs terminals dels arbres dels uns als altres, per formar així els grafs definitius, d'ordre molt major, amb el *girth* desitjat. Per exemple, per obtenir la *cage* de *girth* 14 parteix de 82 arbres (numerats de l'1 al 82) de 6 vèrtexs cada un, i estableix que els increments en l'índex dels arbres amb que cal unir cada vèrtex terminal són: 1 per al vèrtex 0; 22 per al vèrtex 1; 9 per al vèrtex 2 i 34 per al vèrtex 3 (mòdul 82). És a dir, el vèrtex 0 de cada arbre s'ha d'unir al vèrtex 0 del següent, el vèrtex 1 al vèrtex 1 de 22 arbres més enllà, i així successivament. No obstant, cal tenir en compte que per fer-ho, s'estableix a més a més un etiquetat dels vèrtexs dels arbres de 6 vèrtexs originals que difereix segons l'índex de l'arbre sigui parell o senar.

Dos altres mètodes són exposats per G.Brinkmann, B.McKay i C.Saager [11] per a la determinació de l'ordre exacte de les *cages* de *girth* 9 i l'obtenció de totes les 18 possibles *cages* per aquest *girth*, demostrant també que aquestes són les úniques existents. El segon dels mètodes és aplicat també a l'obtenció de fites inferiors per als *girths* 11 i 13.

El primer mètode (aplicable per a *girths* menors o iguals a 9) parteix aquest cop d'un sol arbre d'ordre major que 48 i diferencia els vèrtexs entre vèrtexs interns, fulles i vèrtexs externs. A continuació estableix, mitjançant un mètode exhaustiu ideat per McKay per a l'obtenció d'isomorfismes, una reducció de totes les possibles combinacions de branques (no-equivalents entre sí), tot agrupant-les en tres grups: branques que uneixen fulles entre sí, branques que uneixen fulles amb vèrtexs externs i branques que uneixen vèrtexs externs entre sí. A partir d'aquí, realitza una cerca exhaustiva traient totes les branques *intermitges* i *superiors*, modificant o eliminant una o dues branques *inferiors* i tornant a posar les branques *intermitges* i *superiors* en totes les combinacions possibles. L'obtenció dels 18 *cages* mitjançant aquest mètode va requerir de 4 anys de càlculs en múltiples ordinadors.

En el segon mètode es defineix una família de "representacions" del graf cúbic com a una llista de subllistes, amb una subllista per cada vèrtex no-intern construït en una forma particular. D'entre totes les possibles representacions corresponents a diferents numeracions del graf, aquella que és lexicogràficament menor és un invariant de la classe d'isomorfismes i se l'anomena la representació *canònica*. El programa genera representacions de forma sistemàtica utilitzant una simple tècnica de "ramificació i tall", buscant representacions *canòniques*. Com és usual en els programes de "ramificació i tall", l'eficiència depèn summament de quan d'aviat podem reconèixer i eliminar les branques de l'arbre de cerca que no poden contenir representacions *canòniques*.

La clau de les representacions resideix en els *noms* assignats als vèrtexs durant el procés de construcció. Utilitzant la nomenclatura vista abans per a

la classificació dels tipus de branques i vèrtexs del graf, els vèrtexs *interiors* de l'arbre no canvien durant tot el procés de construcció i, doncs, no cal donar-los nom. Als altres vèrtexs se'ls assigna un nom que consisteix en una parella de valors que indiquen si són externs o fulles, el primer, i en quin ordre han sigut units a l'arbre, el segon. Per a cada vèrtex tenim una subllista de noms formada per ell i els seus veïns immediats; i el conjunt de totes les subllistes forma la representació del graf a partir de la qual el podem reconstruir completament.

Partint de l'arbre bàsic, s'inicia el procés de construcció branca a branca, assignant el nom corresponent a cada nou vèrtex unit i discriminant en tot moment si l'estructura construïda fins al moment té possibilitats de conduir-nos a una solució vàlida o no. Per realitzar aquesta discriminació es defineixen diversos "criteris de tall" que serveixen per eliminar, de forma anticipada a la construcció completa del graf, les solucions no òptimes. Entre els criteris utilitzats, uns tenen en consideració si la ramificació actual ens pot conduir a una representació canònica o no, i uns altres si el graf pot complir amb el *girth* requerit o no.

Entre els del primer tipus, el més important s'encarrega de comprovar en tot moment que no existeixi cap elecció del vèrtex arrel i de la numeració subsegüent dels vèrtexs que ens pogués dur a una representació parcial que sigui menor (segons el criteri de canonicitat exposat) a l'actual. Per implementar aquest criteri de tall es busca, per a la part inicial de la seqüència de vèrtexs que ja tenen grau 3 i cada vegada que es completa un nou vèrtex a grau 3, els primers noms dels vèrtexs que donen la menor representació possible respecte dels segons noms.

L'exemple principal de l'altre tipus de criteris utilitza una simple revisió dels vèrtexs que encara no tenen grau 3. Cada vèrtex d'aquests i cada parella

d'aquests vèrtexs ha de poder ser unit a uns altres vèrtexs d'aquest tipus sense crear cicles curts. Per realitzar aquesta comprovació de forma fàcil, es manté la matriu de distàncies del conjunt d'aquests vèrtexs encara no completats i s'actualitza cada vegada que una nova branca es escollida.

Aquests criteris són utilitzats per a tots els *girths* amb que es treballa, però hi ha d'altres criteris addicionals que són específics d'algun dels *girths*. Per exemple, per al cas de les *cages* de *girth* 9, s'utilitza encara un altre criteri que té a veure directament amb el *girth*, i pel qual es calculen totes les seqüències de 34 vèrtexs amb valència 3 com a màxim, però amb 12 de valència exactament 3, i contra elles es contrasta la llista de deficiències de valència dels vèrtexs que encara no han sigut completament units a l'estructura per veure si coincideix amb alguna d'aquestes seqüències per tal d'acceptar o descartar la construcció.

Hem volgut aquí exposar algunes de les tècniques utilitzades fins ara en la cerca d'algunes de les *cages* per tal de mostrar la mena d'operacions que s'han arribat a implementar per tal d'assolir-les. Seguint el que hem vist en el capítol 3 relatiu al tipus d'algorismes, es pot observar fàcilment com aquests mètodes són de tipus plenament heurístic, donat que utilitzen mecanismes de construcció i cerca de les solucions que es fonamenten en propietats molt específiques i particulars només de *cages* amb *girths* molt similars, però que mai poden extrapolar-se a *cages* d'altres *girths*. Aquest fet fa molt interessant la cerca de mètodes que per la seva simplicitat i generalitat puguin aplicar-se sense cap limitació a *cages* de qualsevol *girth*. Aquest és doncs, un dels objectius que en aquest projecte es pretén cobrir: mostrar la capacitat de cert tipus d'algorismes, en concret de l'algorisme de la Recuita Simulada, que pel seu enfoc global permeten aplicar-los amb èxit a problemes com el de les

*cages* amb independència de la dimensió del problema. Això s'aconsegueix fent que l'algorisme prengui en consideració en la seva cerca d'un òptim només aquelles propietats de les *cages* que són precisament globals a totes elles i independents del *girth*.

## 6. XARXES D'INTERCONNEXIÓ AMB DIAMETRE FIXAT: EL PROBLEMA ( $\Delta, D$ )

Exposem a continuació el segon problema en que s'ha centrat el projecte, el **problema ( $\Delta, D$ )**, sobre el qual hem aplicat també amb èxit el treball realitzat per al problema de les *cages*, tot realitzant en l'algorisme una simple adaptació de les característiques dels tipus de grafs cercats.

### **6.1. Descripció**

El problema ( $\Delta, D$ ) consisteix a trobar el graf de grau  $\Delta$  i diàmetre  $D$  d'ordre major possible.

Com es pot veure, el tipus de graf cercat no s'assembla gaire a las característiques de les *cages*, i fins i tot es tracta d'un problema d'optimització invers en quant que cerquem el graf d'ordre major possible, enlloc del graf d'ordre menor possible, com és el cas de les *cages*.

De la mateixa manera que per a les *cages*, per al problema ( $\Delta, D$ ) es pot trobar fàcilment una fita superior per a les solucions, anomenada (també) *fita de Moore*, que s'obté de contar simplement el número de vèrtexs de la matriu de distàncies d'un dels vèrtexs fins a distància igual al diàmetre,  $D$ . L'expressió que s'obté és:

$$n(\Delta, D) = [\Delta(\Delta-1)^D - 2] / (\Delta-2)$$

Aquesta expressió pot demostrar-se fàcilment utilitzant el següent raonament:

Si fixem un vèrtex, a distància 1 d'ell hi ha com a màxim  $\Delta$  vèrtexs, a

distància 2,  $\Delta(\Delta-1)$ , etc.

a) Si  $\Delta \geq 3$ :  $n(\Delta, D) = 1 + \Delta(1 + (\Delta-1) + \dots + (\Delta-1)^{(D-1)}) = [\Delta(\Delta-1)^D - 2] / (\Delta-2)$

b) Si  $\Delta = 2$ :  $n(\Delta, D) = 2D + 1$

Els grafs que assoleixen aquesta fita són anomenats *grafs de Moore*, però, es demostra que, per  $D \geq 2$  y  $\Delta \geq 3$ , només poden existir els *grafs de Moore* amb  $D=2$  i  $\Delta=3, 7$  i  $57$ .

Exposem a continuació en detall la demostració d'aquest teorema.

Pel cas concret  $D=2$ ,  $n(\Delta, D) = \Delta^2 + 1$ . Si existeix aquest graf, donats dos vèrtexs qualsevol ha d'existir un sol camí entre ells de longitud menor o igual a  $D$ . Donat que  $\mathbf{A}$ , matriu d'adjacència del graf, ens indica els diferents camins de longitud 1 entre parells de vèrtexs, i  $\mathbf{A}^2$  els recorreguts de longitud 2 (els elements  $ii$  d'aquesta matriu indiquen els recorreguts d'anada i tornada dels del vèrtex  $i$ ) haurà de verificar-se que:

$$\mathbf{I} + \mathbf{A} + \mathbf{A}^2 - \Delta \mathbf{I} = \mathbf{J}$$

on  $\mathbf{J}$  és la matriu amb valor 1 per tot  $i$  i  $j$ .

La matriu d'adjacència d'un graf és simètrica, així és diagonalitzable, es a dir  $\mathbf{A} = \mathbf{B} \mathbf{\Lambda} \mathbf{B}^{-1}$ , i donat que  $\mathbf{I} = \mathbf{B} \mathbf{I} \mathbf{B}^{-1}$ ,  $\mathbf{A}^2 = \mathbf{B} \mathbf{\Lambda}^2 \mathbf{B}^{-1}$  i  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_s)$  l'equació queda de la forma:

$$\mathbf{B}(\mathbf{\Lambda}^2 + \mathbf{\Lambda} - (\Delta-1)\mathbf{I})\mathbf{B}^{-1} = \mathbf{B} \mathbf{\Lambda}' \mathbf{B}^{-1} = \mathbf{J}$$

Per calcular  $\mathbf{\Lambda}'$  hem de trobar els valors propis de  $\mathbf{J}$ . Aquests són 0 (zero), amb multiplicitat  $n-1$  i  $n$  amb multiplicitat 1.

Com se sap, si el graf és regular, un autovalor de la seva matriu d'adjacència és precisament el grau, i la seva multiplicitat és 1. D'aquesta manera deduïm

que  $\Delta^2 + \Delta - (\Delta-1)1=n$ , és a dir  $\Delta^2+1=n$ . Els altres autovalors de la matriu d'adjacències hauran de verificar que  $\lambda_i^2 + \lambda_i - (\Delta-1)=0$  que condueix als dos valors següents:

$$\lambda_1 = \frac{-1 + \sqrt{4\Delta-3}}{2}, i$$

$$\lambda_2 = \frac{-1 - \sqrt{4\Delta-3}}{2}$$

Si les multiplicitats respectives són  $m_1$  i  $m_2$ , aquestes han de verificar que  $m_1+m_2=n-1$ .

Si considerem la traça de  $\mathbf{A}$  (recordem que no hi ha autollaços),  $\text{tr}(\mathbf{A}) = \text{tr}(\Lambda) = \Delta + m_1\lambda_1 + m_2\lambda_2 = 0$ . Substituint en aquesta equació els valors de  $\lambda_1$  i  $\lambda_2$  obtinguts abans tindrem una equació que ens determinarà  $m_1$  i  $m_2$ :

$$m_{1,2} = \frac{\Delta^2 \pm (\Delta(\Delta-2))}{\sqrt{4\Delta-3}}$$

Donat que tant  $m_1$  com  $m_2$  han de ser enters i  $\Delta$  també ho és, per  $\Delta > 2$ ,  $4\Delta-3$  ha de ser quadrat perfecte. Si definim  $\sqrt{4\Delta-3} = x$  (enter) deduïm a partir de  $\Delta + m_1\lambda_1 + (\Delta^2 - m_1)\lambda_2 = 0$  l'equació per a  $x$ :

$$6x^9 + x^5 + x^4 - 2x^2 + (9 - 32m_1)x - 15 = 0$$

Donat que  $x$  ha de ser enter, només són possibles els valors  $x=1, 3, 5$  i  $15$  que condueixen a uns valors vàlids per al grau  $\Delta=3, 7, 57$ .

El de  $\Delta=3$  correspon al graf de *Petersen*, el de  $\Delta=7$ , al graf de *Hoffman-Singleton* de 50 vèrtexs i el de  $\Delta=57$  correspon a un graf amb 3250 vèrtexs del qual no se n'ha provat encara l'existència.

Per tant, en aquest context és de gran interès trobar grafs que per un grau i diàmetre donats tinguin un número de vèrtexs tant proper com sigui possible a la *fita de Moore*.

A la taula següent mostrem l'estat de les investigacions relacionades amb el problema  $(\Delta, D)$ , en la versió actualitzada de juny'99.

Taula 6-1 Darreres fites per al problema  $(\Delta, D)$

$\Delta \backslash D$	2	3	4	5	6	7	8	9	10
3	<u>10</u>	<u>20</u>	<u>38</u>	<u>70</u>	<u>132</u>	<u>190</u>	<u>330</u>	570	950
4	<u>15</u>	<u>41</u>	<u>96</u>	364	<u>740</u>	<u>1.155</u>	<u>3.080</u>	<u>7.550</u>	<u>17.604</u>
5	<u>24</u>	<u>72</u>	<u>210</u>	<u>558</u>	<u>2.760</u>	<u>5.500</u>	<u>16.956</u>	<u>53.020</u>	<u>164.700</u>
6	<u>32</u>	<u>110</u>	<u>380</u>	<u>1.395</u>	<u>7.908</u>	<u>19.279</u>	<u>74.800</u>	<u>294.679</u>	<u>1.211.971</u>
7	50	<u>148</u>	<u>672</u>	<u>2.756</u>	<u>11.220</u>	<u>52.404</u>	<u>233.664</u>	<u>1.085.580</u>	<u>5.243.030</u>
8	57	<u>253</u>	<u>1.081</u>	<u>5.050</u>	<u>39.671</u>	<u>129.473</u>	<u>713.539</u>	<u>4.039.649</u>	<u>13.964.808</u>
9	74	585	<u>1.536</u>	<u>7.884</u>	<u>75.696</u>	<u>270.048</u>	<u>1.485.466</u>	<u>8.911.766</u>	<u>25.006.478</u>
10	91	650	<u>2.211</u>	<u>12.788</u>	<u>134.395</u>	<u>561.949</u>	<u>4.019.489</u>	<u>13.964.808</u>	<u>52.029.411</u>
11	<u>98</u>	715	3.200	<u>18.632</u>	156.864	<u>970.410</u>	<u>5.211.606</u>	<u>48.626.760</u>	179.755.200
12	133	780	4.680	<u>29.435</u>	<u>358.183</u>	<u>1.900.319</u>	10.007.820	<u>97.386.380</u>	466.338.600
13	<u>162</u>	845	6.560	<u>39.402</u>	531.440	<u>2.901.294</u>	<u>15.733.122</u>	<u>145.880.280</u>	762.616.400
14	183	<u>912</u>		<u>56.325</u>	<u>812.924</u>	6.200.460	29.992.052	<u>194.639.900</u>	1.865.452.680
15	186	1.215	11.712	<u>73.984</u>	1.417.248	7.100.796	<u>45.000.618</u>	282.740.976	3.630.989.376
16	<u>198</u>	1.600	14.640	132.496	1.771.560	14.882.658	86.882.544	585.652.704	7.394.669.856

Aquesta taula és mantinguda actualitzada per Francesc Comellas i és accessible per Internet a la URL [http://www-mat.upc.es/grup\\_de\\_grafs/](http://www-mat.upc.es/grup_de_grafs/) ; i

paral·lelament Charles Delorme del *Laboratoire de Recherche en Informatique* d'Orsay, en manté una copia en format [LaTeX](#).

En el present projecte hem centrat la nostra cerca en els casos de  $\Delta = 3$ , i entre ells especialment en els diàmetres 5, 8, 9 i 10.

## **6.2. Altres mètodes utilitzats fins ara**

Un dels mètodes utilitzats, ideats per R.Bar-Yehuda i T.Etzion [3], parteix de dos cicles de longitud  $n$  (esdevenint  $2n$ , per tant, l'ordre del graf final) i estableix les connexions entre els vèrtexs dels dos cicles amb una periodicitat de  $s$ , divisor de  $n$ . L'avantatge d'aquest mètode és que redueix considerablement el nombre de variants a explorar, així com també el nombre de vèrtexs a avaluar a cada iteració, doncs es fa suficient avaluar el diàmetre per a  $2s$  vèrtexs, donat que la resta segueixen exactament el mateix patró de connexió.

## 7. IMPLEMENTACIÓ DELS ELEMENTS DE LA R.S. PER ALS DOS PROBLEMES TRACTATS

El cor del projecte ha consistit en l'aplicació de l'algorisme de la Recuita Simulada (*Simulated Annealing*) als casos concrets de les *cages cúbiques* i del problema  $(\Delta, D)$ .

Com ja hem comentat en capítols anteriors, de bon començament vàrem decidir-nos per abordar el problema de les *cages* principalment pel dinamisme que viu aquest tema des de fa uns anys en l'àmbit de l'aplicació de l'algorísmica moderna a problemes del tipus NP-Complets.

Així doncs, com veureu, aquest problema és el que ha centrat en aquest projecte la major part del desenvolupament i adaptació de la Recuita Simulada a un cas concret. I ha sigut a partir de l'obtenció dels primers resultats per al problema de les *cages* que vàrem ampliar i modificar el disseny de l'algorisme per adaptar-lo també al cas particular del problema  $(\Delta, D)$ .

Tot i que aquest ha sigut el procés seguit durant el projecte, aquí exposem la implementació final de l'algorisme en tots els aspectes que són globals a ambdós problemes, doncs la major part del desenvolupament és comú per a tots dos. I particularitzem per a cada un d'ells per separat allà on cal. És a dir, especialment en tractar la definició de la funció de cost, que és lògicament on es dona la principal diferència en la implementació pràctica de l'algorisme.

### **7.1. L'àmbit del present projecte**

Com ja hem dit anteriorment, en realitzar l'adaptació de l'algorisme als problemes tractats, d'entre l'enorme ventall casos possibles a estudiar ens

hem centrat en un àmbit concret, comú tant per a les *cages* com per al problema  $(\Delta, D)$ : els grafs de grau 3.

Hem escollit els grafs **cúbics** perquè, en tots dos problemes, aquests grafs han sigut els més àmpliament estudiats. La majoria d'investigadors d'ambdós problemes han centrat els seus estudis en els grafs de grau 3 perquè permeten l'anàlisi completa dels respectius problemes<sup>1</sup>, però amb el màxim de simplicitat estructural i amb el mínim de cost en les dimensions dels mateixos. Això últim, a més, és d'especial rellevància quan és pretén utilitzar mètodes computacionals per a la cerca de les solucions.

## **7.2. Els elements rellevants en la implementació**

Com ja hem vist en els capítols 3 i 4, l'esquema general de l'algorisme de la Recuita Simulada és molt simple i prou conegut en els àmbits matemàtics. El mecanisme de refredament que simula és pot descriure fàcilment en forma de pseudo-codi de la següent manera:

---

<sup>1</sup> Noteu que això no és possible, per exemple, amb les *cages* de grau 2, donat que per aquest grau la solució és trivial: la *cage* per a un *girth* donat és sempre senzillament el graf circulant d'ordre igual al *girth*. I, d'altra banda, treballar amb grau igual a 4 o superior no afegix cap element al problema que no tingui ja el cas grau=3.

## ESQUEMA GENERAL

**0.** Inicialitzar  $T_k = T_o$ .

**1.** **Construir** una solució qualsevol de l'espai d'estats, de forma aleatòria.

**2.** Repetir  $N_k$  vegades:

*Alterar lleugerament la solució.*

*Evaluar el cost de la nova solució.*

*Guardar-la, si és la millor obtinguda fins al moment.*

*Si millora respecte l'anterior, acceptar-la sempre.*

*Si empitjora respecte l'anterior, acceptar-la només si:  $\exp(-\Delta f / K T_k) < \text{rand}()$ .*

**3.** **Disminuir**  $T_k$  i repetir **2** fins que  $(T_k < T_{fi})$  o bé **decidim aturar-lo** per algun altre criteri.

*Esquema de funcionament general de l'algorisme RS.*

En aquest esquema general de l'algorisme, hem marcat en **negreta** les parts que varien en funció de quin és el problema que es vol resoldre. Com es pot observar aquestes són (per ordre d'aparició a l'esquema anterior): **la temperatura inicial ( $T_o$ )**, **el mecanisme de construcció** de les possibles solucions, **el número d'iteracions** a cada temperatura ( $N_k$ ), **el mecanisme d'alteració** d'una solució per passar a una altra, i **la funció objectiva** d'avaluació de l'energia de cada solució, **el mecanisme de refredament** de la temperatura, **la temperatura de congelació ( $T_{fi}$ )**, i la definició dels criteris d'aturada de l'algorisme.

### 7.2.1. Els tres elements principals de l'algorisme

D'entre tots aquests elements constituents que són variables, n'hi ha tres que formen, podríem dir, el cor de l'algorisme, en quant a que contenen tota la definició del problema concret al qual es vol aplicar l'algorisme i de l'aproximació o estratègia aplicada; i porten el pes principal de l'execució del mateix. Aquests tres elements principals són:

- 1) **el mecanisme de construcció** d'una solució,
- 2) **el mecanisme d'alteració** d'una solució per passar a una altra, i
- 3) **la funció objectiva** d'avaluació de l'energia de cada solució.

Aquests han sigut, doncs, els aspectes més treballats en el projecte i els que constitueixen la clau de l'èxit del mateix i de l'eficiència obtinguda en l'aplicació de l'algorisme.

Exposem ara les estratègies utilitzades en aquests tres aspectes de l'algorisme. I abordarem, més tard, els criteris i valors de la resta de paràmetres que intervenen en la seva implementació: la temperatura inicial, el mecanisme de refredament, la temperatura de congelació, el número d'iteracions i la definició dels criteris d'aturada.

#### **7.2.1.1. Els mecanismes de construcció dels grafs**

Al llarg del projecte s'han implementat dues alternatives per la forma en la que són construïts els grafs: la primera està inspirada en la **representació en arbre** d'un graf i la segona, en la **representació circular**.

Cada un d'aquests dos mecanismes de construcció implica, alhora, una estratègia ben diferent en la forma d'abordar els problemes tractats. Aquesta estratègia o enfoc, és la que s'exposa a continuació.

El mecanisme de construcció en arbre

La primera estratègia, que ha donat lloc al **mecanisme de construcció en arbre**, va néixer de l'observació de la completa simetria que han de presentar els grafs solució. Això és que, "partint" des de qualsevol dels seus vèrtexs, l'estructura del graf ha de ser exactament la mateixa. Per mostrar-ho, en la següent exposició ens centrarem només en el cas de les *cages* cúbiques, però la mateixa anàlisi es pot fer per al problema ( $\Delta, D$ ).

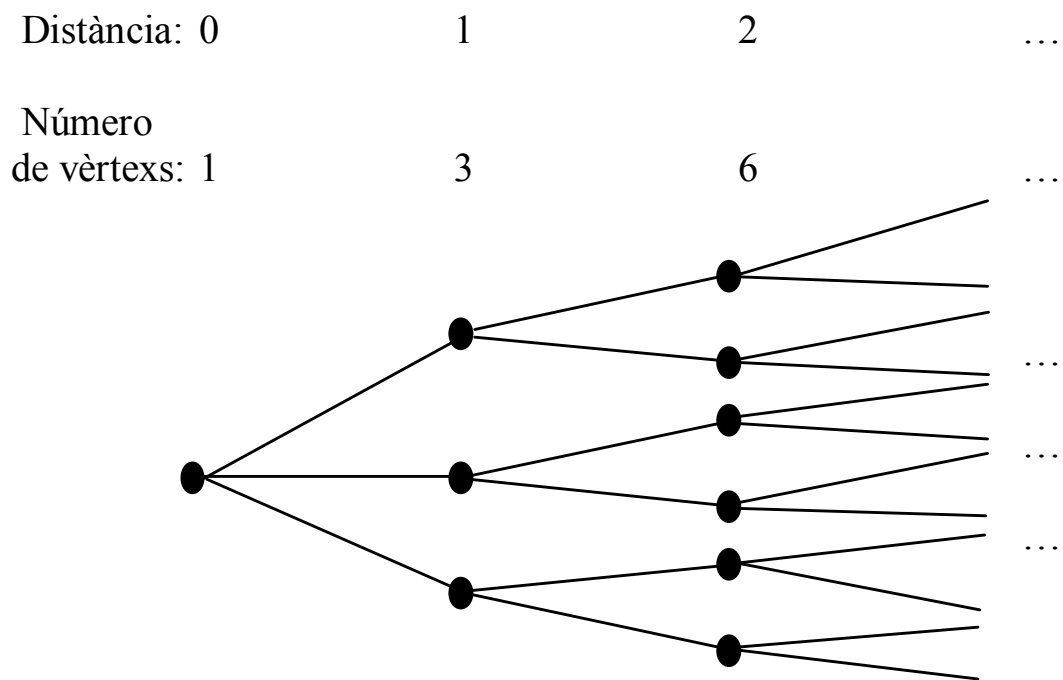
Una forma de plasmar aquesta estructura, que podríem definir com l'estructura que "veu" cada vèrtex, és contant el número de veïns del vèrtex que es troben a distància 1, els que es troben a distància 2, i així fins assolir tots els vèrtexs del graf (donat que el graf és connex). L'aplicació d'aquest enfoc a la que seria la "*cage* perfecta" (amb l'ordre menor possible) ens duu directament a l'obtenció del següent *patró d'expansió*: a partir de la distància 1 (distància a la qual cada vèrtex lògicament té 3 veïns per tractar-se de *cages* cúbiques en el nostre cas) el número de veïns es multiplica per 2 en augmentar en 1 la distància. Això dóna lloc a la següent *matriu model* de vèrtexs a cada distància, que anomenarem *patró d'expansió*:

Taula 7-1 *Patró d'expansió*

<b>Distància</b>	0	1	2	3	...
<b>Vèrtexs a aquesta distància</b>	1	3	6	12	...

(on l' '1' que apareix a la distància zero correspon lògicament al vèrtex del qual partim).

Podem observar doncs com l'estructura que "veu" un vèrtex s'assembla molt a un arbre que neix d'un vèrtex arrel i es va ramificant uniformement, fins assolir la distància límit; tal com mostra la figura següent.



*Figura 7.1 Estructura d'un graf cúbic vista des d'un dels seus vèrtexs*

En base a lo exposat, sembla raonable pensar que un mecanisme de construcció del graf que segueixi directament aquest patró pot ser una bona base de la qual partir per trobar solucions adequades al problema.

Aquesta *patró d'expansió* és el que ha constituït *la base* del primer dels mètodes de construcció utilitzats: *el mecanisme de construcció en arbre*.

En ell, partim d'un vèrtex arrel (que sempre numerarem com vèrtex zero), i l'unim a tants veïns com indica el seu grau (en el nostre cas 3), els seus veïns immediats. I a continuació anem unint cada vèrtex que pertany a l'estructura

a tants nous veïns com branques li queden "lliures" a aquest vèrtex. D'aquesta manera, donat que a partir del vèrtex arrel tot nou vèrtex ha quedat ja unit a l'estructura per una de les seves branques, encara disposa de dos branques més que unirem amb dos nous veïns, que pertanyeran al "següent nivell".

Anomenem "nivell" al conjunt de vèrtexs que es troben a la mateixa distància del vèrtex arrel. D'aquesta manera, al nivell 0 tenim el vèrtex arrel (el vèrtex 0, zero), en el nivell 1 els seus veïns immediats (els vèrtexs etiquetats com a 1, 2 i 3), al nivell 2 tenim 6 vèrtexs, i així fins arribar a una distància límit.

La distància límit fins a la qual es manté aquest patró depèn directament del *girth* i varia segons aquest sigui parell o senar. És la distància que en el programa C hem anomenat "nivell\_base". El seu càlcul i valors són els que s'indiquen a continuació:

Senar:  $nivell\_base=(girth-1)/2$

Parell:  $nivell\_base=(girth-2)/2$

*Taula 7-2 Nivell base*

<i>girth</i>	5	6	7	8	9	10	11	12	13
<i>nivell_base</i>	2	2	3	3	4	4	5	5	6

És fàcil de veure com n'està aquest enfoc d'estretament relacionat amb el càlcul de la *fit a de Moore* que hem vist al capítol 5. I també, com la representació en arbre del que seria el *graf de Moore* (si existís) corresponent a

cada *girth* varia també segons el *girth* sigui parell o senar. En concret la diferència es troba justament a partir del *nivell base*.

Per als *girths* senars, aquest és l'últim nivell en que hi ha vèrtexs, donat que el cicle més curt (de longitud igual al *girth*) que podem formar en aquest *graf de Moore* el podem fer seguint aquest recorregut: sortir del vèrtex arrel, anar fins un vèrtex del *nivell base*, saltar d'aquest a un altre vèrtex del *nivell base* i tornar al vèrtex arrel per un altre camí. Amb això, la longitud d'aquest cicle és:  $nivell\_base + 1 + nivell\_base$ ; que ha de ser igual al *girth*. Per comprovar-ho:

$$girth = nivell\_base + 1 + nivell\_base = (2 \cdot nivell\_base) + 1;$$

que sempre serà senar, donat que  $(2 \cdot nivell\_base)$  és, evidentment, parell.

I, aïllant *nivell\_base*, obtenim:

$$nivell\_base = (girth - 1) / 2;$$

com ja havíem avançat abans.

Per tant, el *grafs de Moore* de *girth* senars haurien de tenir l'aspecte mostrat a la següent figura:

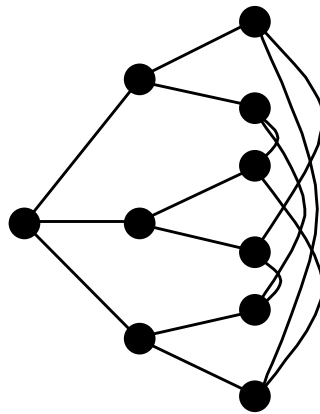


Figura 7.2 Graf ideal per a *girth* senar (*girth* = 5)

En canvi, per a *girths* parells, l'estructura ha de variar, presentant a més un últim nivell de vèrtexs per damunt del *nivell base*, però amb un nombre menor de vèrtexs que aquest. I el recorregut a seguir per completar el cicle de menor longitud és ara: sortir del vèrtex arrel, anar fins un vèrtex del nivell que està per damunt del *nivell base* i tornar al vèrtex arrel per un altre camí. Amb això, la longitud d'aquest cicle és:  $(nivell\_base+1) + (nivell\_base+1)$ ; que ha de ser igual al *girth*. Per comprovar-ho:

$$girth = nivell\_base + 1 + nivell\_base + 1 = (2 \cdot nivell\_base) + 2;$$

que en aquest cas és sempre parell.

I, aïllant *nivell\_base*, obtenim:

$$nivell\_base = (girth - 2) / 2;$$

Demostrada aquesta igualtat, resulta ara interessant de determinar quina

deu ser l'estructura d'aquest "graf ideal"<sup>1</sup> a l'últim nivell, el que es troba per damunt del *nivell base* en el cas de *girth* parell. És a dir, quants vèrtexs el formen. Per determinar-ho només cal fer el següent raonament: si a tots els vèrtexs del *nivell base* els queden dues branques lliures després d'unir-los a l'arbre, hi ha un total de  $2 \times (\text{número de vèrtexs al nivell base})$  branques per unir. I, donat que cap d'aquestes pot posar-se unint vèrtexs del *nivell base* (doncs llavors existirien cercles de longitud menor al *girth* indicat i, per tant, el graf deixaria de tenir aquest *girth*), totes han d'unir-se als vèrtexs del *nivell base+1*. I tots els vèrtexs d'aquest últim nivell, han de tenir també necessàriament totes les seves branques unides als vèrtexs del *nivell base*. Per tant, el total de branques a unir des del *nivell base* ha de ser igual al total de branques a unir des del *nivell base+1*. I donat que els vèrtexs del *nivell base+1* tenen encara totes tres branques per unir, obtenim que,

$$2 \cdot N_{\text{nivell base}} = 3 \cdot N_{\text{nivell base+1}}$$

d'on:

$$N_{\text{nivell base+1}} = (2/3) \cdot N_{\text{nivell base}}$$

Aquest resultat ens permet representar perfectament l'estructura del "graf ideal" per a *girth* parell, que mostrem a la següent figura.

---

<sup>1</sup> Recordem que en el cas de *girth* parell el "graf ideal" no correspon a un *graf de Moore* (com en el cas de *girth* senar), sinó a un Polígon generalitzat.

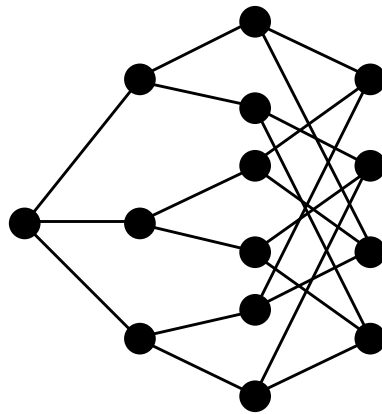


Figura 7.3 Graf ideal per a girth parell ( $girth = 6$ )

És per aquesta diferencia en l'estructura que el mètode de construcció de l'arbre varia també segons el *girth* sigui parell o senar, canviant l'estratègia en assolir el *nivell base*. I és per això que en el present projecte hem decidit centrar el nostre estudi en només un dels tipus de construcció en arbre, el relacionat amb els grafs de *girth* senar. Però l'estudi es podria haver realitzat amb la mateixa validesa escollint el de *girth* parell, sense treure generalitat als resultats.

Centrant-nos ara en els *girths* senars, hem vist com al *nivell base* el patró de propagació s'acaba, i a més aquest resulta ser l'últim nivell existent. Però cal recordar que això només és així en el cas que existeixi per aquest *girth* el graf de Moore corresponent (que és sobre el qual hem desenvolupat abans la nostra dissertació) i que, per tant, té l'ordre mínim possible que pot tenir un graf d'aquest *girth*. Així doncs, si aquest no existeix (com ja hem comentat que succeeix per als *girths* 7, 9, 10, 11 i a partir del 13) les *cages* tindran ordre superior i, per tant, presentaran necessàriament vèrtexs més enllà del *nivell base*. És a partir d'aquest nivell que la forma d'unir els vèrtexs a l'estructura i entre ells pot variar. Fins aquest nivell tenim el que hem anomenat l'*arbre*

*base*, el qual és fix i invariable per un *girth* donat. I comença la "copa", que és el nom que donarem al conjunt de branques i vèrtexs que formen la part superior (i variable) de l'arbre. El nombre total de vèrtexs que formen aquesta *copa* i la manera en la qual els vèrtexs que la formen han d'unir-se entre sí és la que ens falta per descobrir.

Gràcies a aquest enfoc, el problema que inicialment implicava a un número més o menys elevat de vèrtexs i branques, queda reduït ara a un número molt menor: només els de *la copa*.

En concret, el número de vèrtexs implicats en *la copa* inclou com a mínim tots els que formen el *nivell base*, doncs recordem que aquests han sigut units a l'estructura per una de les seves branques, però tenen encara dos branques "lliures". Així doncs, la manera d'unir-se ells entre si ja forma part del que anomenem *la copa*, tinguem o no més vèrtexs per unir a l'estructura.

Per els diversos *cages* estudiats, la reducció en el número de vèrtexs i branques amb les que s'ha de treballar en comparació al total de les del graf es pot observar a la següent taula:

*Taula 7-3 Reducció en el nombre de vèrtexs sobre els que cal avaluar la funció de cost*

<i>girth</i>	5	7	9	11	13	15
Nº de vèrtexs (graf de Moore)	10	22	46	94	190	382
Vèrtexs a avaluar (nivell base)	6	12	24	48	96	192
Nº de branques	15	33	69	141	285	573

(graf de Moore)						
Branques a avaluar (nivell base)	9	18	36	72	144	288

### El mecanisme de construcció en cercle

La segona estratègia, que va dur-nos a la definició del **mecanisme de construcció en cercle**, va sorgir a partir de resultats obtinguts per K.Piwakowski [35] en els seus treballs d'investigació relacionats amb grafs de Ramsey amb el propòsit de trobar noves fites inferiors per als clàssics números de Ramsey. K.Piwakowski se centra en l'ús de grafs circulants i, utilitzant l'algorisme de Cerca Tabú, descobreix l'existència de grafs cíclics (que repeteixen patrons de connexió) que encaixen bé en el tipus de grafs de Ramsey, i milloren les fites trobades fins al moment.

La representació més usual dels grafs circulants consisteix en mostrar-los com un cercle format per tots els seus vèrtexs, de forma que cada vèrtex s'uneix per un costat al d'índex immediatament inferior i per l'altre al d'índex immediatament superior. I la branca que queda lliure s'uneix a qualsevol altre vèrtex del cercle, formant una malla central de connexions, anomenades *cordes*. Aquesta és la **representació circular** que hem vist en el capítol 2.

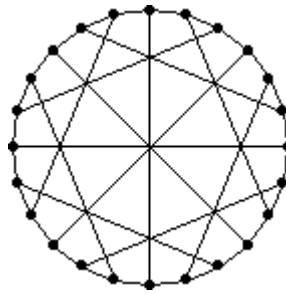


Figura 7.4 Graf circular

Els resultats obtinguts per Piwakowski ens van animar a investigar si els grafs circulars i la representació circular podien presentar propietats interessants per a la construcció de *cages* i, alhora, per al problema  $(\Delta, D)$ .

I, efectivament, aquest enfoc va resultar ser aplicable de forma immediata per a les ja conegudes  $(3,6)$ -*cage* d'ordre 14 i  $(3,8)$ -*cage* d'ordre 30.

Així, la  $(3,6)$ -*cage* d'ordre 14 es pot representar mitjançant la repetició 7 vegades d'un mateix patró de connexió format per 2 vèrtexs i la  $(3,8)$ -*cage* d'ordre 30 mitjançant la repetició 5 vegades d'un mateix patró de connexió format per 6 vèrtexs. Les configuracions trobades són les següents:

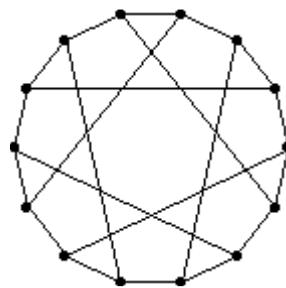


Figura 7.5  $(3,6)$ -*cage* d'ordre 14

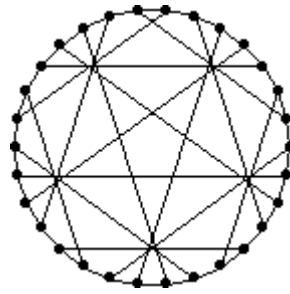


Figura 7.6 (3,8)-cage d'ordre 30

Aquestes propietats tan avantatjoses que ofereix la representació circular provenen del fet que la seva simetria permet que un subconjunt dels vèrtexs del graf esdevinguin totalment equivalents. I, per tant, del total dels vèrtexs, finalment només cal estudiar-ne uns quants, doncs la resta en són una rèplica exacta. D'aquí la possibilitat de trobar grups de vèrtexs que es repeteixen en les seves característiques. El cas extrem d'aquesta propietat es dona quan tots els vèrtexs del graf són equivalents i, doncs, el graf és vèrtex-simètric.

A partir d'aquest descobriment, va semblar raonable pensar que configuracions similars podien conduir a bones solucions per a *cages* de *girths* majors i pel problema  $(\Delta, D)$ . I, a més, amb l'avantatja afegida de que en presentar-se patrons repetitius, la dimensió del problema es redueix dràsticament.

L'únic que feia falta era escollir la mida dels grups patró per cada ordre del graf a investigar. Això s'ha resolt partint de la premissa de que el número de vèrtexs en el *grup patró* ha de ser parell, donat que en el *grup patró* es troben sempre els dos extrems de qualsevol de les seves cordes de connexió.

Potser l'exemple més il·lustratiu sigui justament el de la (3,6)-cage d'ordre 14

mostrat en la Figura 7.5, on podem observar com el *grup patró* (de 2 vèrtexs) inclou tant la corda en la direcció "positiva" ( +5 ) com la seva parella associada, que té "signe negatiu" ( -5 ). Amb aquesta premissa clara, únicament cal trobar tots els divisors de l'**ordre** del graf i formar tots els parells ordenats possibles de nombres naturals  $(g,m)$  amb el següent significat:

$(g,m) = (n^\circ \text{ de repeticions del grup}, n^\circ \text{ de vèrtexs del grup patró}),$

que compleixin que el  $n^\circ \text{ de vèrtexs del grup patró}$  sigui múltiple de 2 i que multiplicats donin, lògicament, l'**ordre** del graf.

L'existència d'aquesta condició ens permet arribar a una conclusió important al respecte dels possibles valors en l'**ordre** que poden prendre les *cages* que cerquem amb aquest mètode: han de ser necessàriament **parells**. Aquesta conclusió és fàcil de deduir a partir del que hem comentat abans:

$$\begin{aligned} \text{ordre} &= n^\circ \text{ de repeticions del grup} \cdot n^\circ \text{ de vèrtexs del grup patró}, \quad \text{i} \\ n^\circ \text{ de vèrtexs del grup patró} &= 2 \cdot a, \quad \text{per } a \in \mathbb{N}. \end{aligned}$$

Llavors,

$$\text{ordre} = 2 \cdot a \cdot n^\circ \text{ de repeticions del grup}$$

que és evidentment parell.

No obstant, en realitat aquest és un principi que és vàlid per a les *cages* cúbiques amb independència del mètode utilitzat en la construcció d'aquestes, donat que té el seu fonament en la pròpia definició de les *cages*: les *cages* són grafs *regulars*. I, donat que tota branca té necessàriament dos vèrtexs que la comparteixen, tenim la següent igualtat:

Si l'ordre és  $n$  i el grau és  $k$ , tindrem  $(n \cdot k)/2$  branques al graf.

Donat que el nombre de branques ha de ser enter, deduïm que  $(n \cdot k)$  ha de ser múltiple de 2. I en el cas que ens ocupa,  $k = 3$ . Per tant, si  $(n \cdot k)$  és parell i  $k$  és senar, això obliga a que  $k$  ha de ser parell; tal com volíem demostrar.

És doncs una condició intrínseca a la definició de *cage*. Senzillament ho hem exposat aquí perquè ha vingut al cas en mostrar les característiques dels parells de valors vàlids per al mètode de construcció en cercle utilitzant un *grup patró*. Tornant a aquesta exposició, un exemple pot resultar més clar: per la cerca de les *cages* de *girth* igual a 10, que són ja conegudes i d'ordre 70, els parells ordenats possibles són:

**Girth:** 10

**Ordre:** 70

**Divisors:** 1, 2, 5 i 7. ( $70=2 \cdot 5 \cdot 7$ )

**Parells:**

(1,70)

(5,14)

(7,10)

(35,2)

D'entre ells, els que són realment òptims són (5,14) i (7,10) donat que impliquen treballar amb un *grup patró* de només 14 o 10 vèrtexs, respectivament. El cas (1,70) no suposa optimització alguna en quant al número de vèrtexs a manejar en els càlculs, però s'ha estudiat també a mode de referència per poder contrastar amb ella optimització aconseguida amb els altres parells. A més, ofereix també la possibilitat d'obtenir, si existissin, altres resultats que no es ceneixin necessàriament a l'esquema presentat.

En els casos analitzats mitjançant aquest mètode, senzillament s'ha realitzat una cerca per cada un dels parells ordenats possibles per un **ordre** donat.

En la taula següent mostrem la millora que s'obté en l'eficiència pel fet de treballar amb un *grup patró* d'un nombre reduït de vèrtexs, en comparació amb el total de vèrtexs del graf:

*Taula 7-4 Reducció mitja en el nombre de vèrtexs sobre els que cal avaluar la funció de cost (reducció mitja/vèrtexs totals x 100)*

<i>Girth</i>		<b>6</b>	<b>8</b>	<b>10</b>	<b>12</b>	<b>14</b>	<b>15</b>
Ordre de les <i>cages</i>		14	30	70	126	258-406	384-620
G R U P P A T R Ó	20 vèrtexs	x	x	x	x	93,7%	95,8%
	16 vèrtexs	x	x	x	x	94,9%	96,6%
	14 vèrtexs	0%	x	80%	89%	95,6%	97,0%
	12 vèrtexs	x	x	x	x	96,2%	97,5%
	10 vèrtexs	x	66%	85,7%	x	96,8%	97,9%
	8 vèrtexs	x	x	x	x	97,5%	98,3%
	6 vèrtexs	x	80%	x	95,2%	98,1%	98,7%

Com podem veure, conforme treballem amb grafs cada vegada més grans, la reducció que s'obté és més espectacular. Per tant, si amb aquest mètode de construcció s'obtenen resultats, aconseguim un significatiu augment en l'eficiència de l'algorisme gràcies a aquest simple canvi d'enfoc.

### **7.2.1.2. Els mecanismes d'alteració del graf**

El següent aspecte important a determinar és el mètode o mètodes d'alteració que s'utilitzaran per anar explorant tot l'espai d'estats del problema.

Aquest punt té estreta relació amb el mètode que s'hagi utilitzat per a la construcció dels grafs de l'espai de solucions. I, per tant, en el nostre cas hem de diferenciar de nou entre l'estratègia seguida quan utilitzem la construcció en arbre i aquella que seguim quan l'estructura utilitzada és la circular.

No obstant, hi ha una sèrie de principis que s'han pres com a base i que s'han mantingut en ambdues estratègies:

- 1) No es permet que hi hagi mai un vèrtex que pertanyi al graf i que estigui desconnectat del mateix.
- 2) És més, no es permet que hi hagi mai un vèrtex amb alguna de les seves branques "lliure". Tot vèrtex ha de tenir sempre les seves tres branques unides a algun altre vèrtex del graf.
- 3) Així mateix, no es permet que entre dos vèrtexs existeixi més d'una branca de connexió directa. És a dir, no es permet l'existència de branques paral·leles.

D'aquesta manera, s'assegura en tot moment que cada vèrtex està unit sempre a tres veïns diferents, i que el graf sempre és connex.

Aquests principis s'han pres, d'una banda, per reduir el camp de solucions que per avançat sabem que no són vàlides (recordem el *patró d'expansió* {1,3,6,...}, que aplica de la mateixa manera per als grafs en arbre com per als grafs en cercle), i, d'altra banda, per facilitar la implementació de la funció de cost òptima al problema, que veurem més endavant.

Per tal de mantenir aquests principis a lo llarg de tota l'execució del programa, el mecanisme d'alteració ha d'assegurar que no els infringeix en cap cas i, també, que no altera en cap moment l'estructura del graf en sí mateixa. D'aquesta manera, el mecanisme utilitzat consisteix sempre en: seleccionar a l'atzar dues branques diferents del graf que pertanyin a vèrtexs distints i intercanviar entre sí els vèrtexs als quals "apunten" (aquesta és només una forma d'expressar-ho que no té res a veure amb les propietats de direcció de les branques en els digrafs. Recordem que en el present projecte treballem amb grafs, i no amb digrafs).

Sobre *la base* d'aquests principis, s'ha adequat el procediment d'alteració a l'estructura amb la que es treballa en cada cas.

En el primer cas, amb la estructura en arbre, el procediment d'alteració es redueix a lo ja exposat. Amb això el que s'aconsegueix és mantenir sempre un mateix número de connexions entre els vèrtexs de *la base* i els de *la copa*, i entre els vèrtexs de *la base* entre sí o entre els de *la copa* entre sí, si n'hi ha.

Inicialment, es va implementar un sistema d'alteració molt més sofisticat: es va dissenyar el programa per que en realitzar l'alteració seleccionés entre aquesta alteració simple i l'afegit o l'eliminació de vèrtexs del graf, variant així l'ordre del graf a lo llarg de la simulació (versió que hem anomenat **Algorisme A**). Amb aquest mètode es van obtenir els primers resultats per a la *cage* de *girth* 7, però la complexitat en la forma d'expandir l'arbre i l'enorme impacte que això tenia en la funció de cost (per més que aquesta era independent del número de vèrtexs del graf) va fer que finalment s'optés per fixar l'ordre del graf a estudiar durant tota la simulació, i realitzar simulacions diferents per a cada ordre.

En el segon cas, amb l'estructura circular, el procediment d'alteració incorpora algunes variants addicionals:

- D'una banda, en intercanviar els destins de les branques del *grup patró*, decideix primer a l'atzar a quin grup veí es connectarà cada branca. És a dir, si la branca del 3r vèrtex del *grup patró* apuntava al 6è vèrtex de 4 grups més enllà (increment intra-grup=3; increment total=(4·m)+3; on m és el nombre de vèrtexs que formen el *grup patró*) i passarà a apuntar al 8è vèrtex del grup, el mecanisme d'alteració li assignarà de forma aleatòria el número A (enter) de grups que se li han de sumar a l'increment intra-grup (=5); obtenint doncs un increment total=(A·m)+5.

- D'altra banda, amb una probabilitat fixada abans de l'execució del programa (anomenada *rotació* en el codi del programa), en lloc de realitzar un intercanvi entre dues branques el que es fa és variar aquest número A de grups que se li sumen a l'increment intra-grup, mantenint aquest últim inalterat. D'aquesta manera, l'impacte de l'alteració és menor, la qual cosa és molt convenient conforme es van assolint temperatures cada vegada menors. Després, una vegada realitzada l'alteració, ja sigui d'un tipus o de l'altre, en les branques del *grup patró*, es reconstrueix de nou tot el graf, simplement alterant les branques equivalents a aquestes a la resta de grups del graf.

### **7.2.1.3. La funció objectiva**

L'elecció de la funció de cost òptima a cada problema, com s'ha comentat, és un punt crucial de la implementació de l'algorisme de la Recuita Simulada i que impacta directament en l'eficiència del mateix. La definició de la funció de cost és el mecanisme mitjançant el qual se li fa conèixer, per dir-ho així, a l'algorisme el problema amb el qual ha de treballar. És a través d'ella que se

li dóna la descripció del tipus de solucions cercades: la descripció de "com és la solució perfecta".

És per això que, en aquest punt de l'algorisme, l'enfoc per resoldre el problema de les *cages* i el que cal utilitzar per al problema  $(\Delta, D)$  difereixen àmpliament. I, per tant, els veurem a continuació per separat. No obstant, es podrà observar com ambdós tenen una gran similitud entre sí. Aquesta és de fet una de les raons que ens va impulsar a tractar tots dos problemes dins del mateix projecte, donat que hi ha una gran part en comú entre tots dos.

Per al problema de les *cages*:

Per abordar el problema de les *cages*, en el present projecte s'ha utilitzat una funció de cost que té en consideració dos aspectes propis d'aquestes que la solució ha de complir, o, millor dit, que cada un dels seus vèrtexs ha de complir. El primer és lògicament el *girth*, el *girth* que "té" o que "veu" cada vèrtex. El segon, deriva directament del *patró d'expansió* que, com s'ha comentat amb anterioritat en aquest mateix capítol, han de "complir" necessàriament tots els vèrtexs de la solució fins una certa distància.

El principi aplicat aquí és que, si aquest patró ha de mantenir-se al menys fins la distància *nivell\_base* per tots els vèrtexs del graf solució, qualsevol vèrtex que s'allunyi d'ell incorpora una "distorsió" en el graf. Parlant en els termes de la tècnica de la Recuita Simulada, aquest vèrtex té, per sí mateix, una energia major a la mínima i, en conseqüència, augmenta l'energia global del graf. Aquest vèrtex s'allunya de l'estructura cristal·lina perfecta.

Aquest segon element, que denominarem *Distorsió en l'expansió* introdueix un major grau de sensibilitat en la funció de cost que el degut al *girth*. Això

és degut a que el *girth* aporta una informació "filtrada", podríem dir, o "resumida" donat que indica la longitud del cicle més curt que hi ha en tot el graf, però no ens diu res, per exemple, sobre el número d'aquests cicles o d'altres cicles majors que existeixen. En canvi, la *Distorsió en l'expansió* descriu molt més fidelment, sense seleccions, com està l'estructura cristal·lina del graf.

A més, a nivell pràctic, incorporar aquesta *Distorsió en l'expansió* en la funció de cost representa un increment petit en el número de càlculs a efectuar, doncs per a l'obtenció del *girth* que "es veu" des de cada vèrtex es fa necessari construir aquesta matriu d'expansió del vèrtex al menys fins la distància *nivell\_base*.

La funció de cost que s'ha utilitzat ha sigut la següent:

$$f(g) = (k_1 \cdot \text{Distorsió en l'expansió} + k_2 \cdot \text{Distorsió deguda al girth}) / (k_4),$$

on:

$$\text{Distorsió deguda al girth} = \frac{\sum_{i=0}^{m-1} (\text{Girth de la cage} - \text{girth vist pel vèrtex } [i])}{N^\circ \text{ vèrtexs evaluats}}$$

que sempre serà positiva perquè només la considerarem quan el *girth vist pel vèrtex [i]* sigui menor que el *Girth esperat de la cage* (un *girth* major no afecta al *girth* global del graf, donat que es pren sempre el mínim trobat). I,

*Distorsió en l'expansió* =

$$\frac{\sum_{i=0}^{m-1} \sum_{j=0}^{\text{nivell\_base}} (\text{Taula Patró de } v_{txs} \text{ a la distància } [j] - \text{taula de distàncies del } v_{tx} [i] \cdot \text{a la dist } [j])}{N^{\circ} \text{ vèrtexs avaluats}}$$

*Taula Patró de vèrtex a la distància [j]* (que en el programa hem anomenat *distab\_patro[l]*) és la matriu que conté el *patró d'expansió*, indicant el número de vèrtexs que s'han de trobar a cada distància *j*, fins a la distància *nivell\_base*. La diferència respecte del patró se suma per a tots els vèrtexs sobre els quals s'avalua la funció de cost. I finalment, es divideix pel número de vèrtexs sobre els quals s'ha avaluat la funció de cost (*Nº vèrtexs avaluats*) per tal que el resultat sigui independent d'aquesta variable. Així, en realitat es treballa sempre amb la mitja de la distorsió que tenen tots els vèrtexs avaluats, i d'aquesta manera aconseguim mantenir sempre valors semblants de distorsió per qualsevol nombre de vèrtexs a *la copa*, per exemple, o bé al *grup patró*.

Un aspecte important, doncs, de la funció de cost és la preponderància que es dóna a la *Distorsió deguda al girth* respecte a la *Distorsió en l'expansió*. És a dir, els valors dels paràmetres  $k_1$  i  $k_2$ . Una tasca important del projecte, prèvia a la cerca de noves solucions als problemes plantejats, ha consistit doncs en l'obtenció dels valors òptims per a aquests paràmetres. S'ha realitzat amb aquest motiu tota una estadística de l'eficàcia de l'algorisme en variar la raó ( $k_1/k_2$ ), els resultats de la qual és poden veure en el capítol següent. La raó òptima ( $k_1/k_2$ ) finalment obtinguda ha sigut:

$$k_1/k_2 = 4$$

S'han realitzat també proves amb  $k_2=0$ , és a dir, sense incorporar a la funció de cost el terme de la *Distorsió en el girth*. Però això conduïa a *falses* solucions pel fet que la *Distorsió en l'expansió* només s'avalua fins a la distància *nivell\_base* (donat que més enllà desconexem l'estructura que el graf ha de tenir) i, per tant, no dóna una informació completa. Només la incorporació d'ambdós termes aporta l'eficàcia desitjada.

Com ja hem comentat anteriorment, també hem assajat una versió de l'algorisme que contemplava la variació dinàmica en l'ordre del graf. Aquesta versió afegeix un tercer terme en la funció de cost, anomenat *Distorsió en l'ordre*, amb una constant  $k_3$  que pondera la penalització en el cost global deguda a l'ordre del graf, fent augmentar el cost com més gran es l'ordre. Aquest terme és el que permetia a les simulacions realitzades amb aquesta versió de l'algorisme tendir en tot moment cap a la solució amb menor ordre possible. Com es pot veure en el codi del programa, aquesta versió només va ser implementada per al mecanisme de construcció en arbre, però no per al circular.

La versió definitiva de la funció de cost utilitzada per al problema de les *cages* incorpora, a més a més, una millora addicional que ha demostrat accelerar força la convergència cap a solucions amb cost mínim i estructura òptima. Aquesta millora consisteix en la utilització d'una ***taula de ponderació*** de la *Distorsió en l'expansió* (anomenada en el programa *k[I]*). En concret, aquesta taula conté un factor multiplicatiu de la distorsió respecte del *patró d'expansió* que s'ha trobat a cada distància de la *taula de distàncies del vèrtex [i]*. De tal manera que penalitza molt més la distorsió quan es dóna a distàncies molt properes al vèrtex perquè això indica que l'estructura està

més lluny de l'esperada, ja des dels primers veïns del vèrtex. La ponderació finalment escollida parteix del *nivell\_base* i va multiplicant per 4 la penalització conforme es va apropant a la distància zero.

Es comença a ponderar des del *nivell\_base* cap a la distància zero perquè, d'aquesta manera, els valors de distorsió que se solen donar durant l'execució de l'algorisme es mantenen pràcticament idèntics, aconseguint una major independència del *girth* amb el que es realitza la simulació. Sobretot en les temperatures finals de la simulació en les quals treballem ja amb grafs que difereixen de l'òptim només en les últimes distàncies (properes a *nivell\_base*) i on la precisió de la funció de cost és molt més rellevant, doncs qualsevol petit canvi afecta molt més al resultat final. En aquestes temperatures, si tinguéssim un comportament molt divergent en la funció de cost segons el *girth* faríem molt difícil l'obtenció de resultats en canviar de *girth* i trencaríem la capacitat d'escalat que atorga l'algorisme de la Recuita Simulada. És a dir, la capacitat d'utilitzar exactament el mateix algorisme, amb idèntica funció de cost, etc., sigui quin sigui el *girth* amb el qual treballem.

Amb aquesta ponderació, la *Distorsió en l'expansió* passa a obtenir-se de la següent manera:

*Distorsió en l'expansió* =

$$\frac{\sum_{i=0}^{m-\text{Nivell\_base}} \sum_{j=0} (Taula Patró de vtxs a la distància [j] - taula de dist. del vtx [i] \cdot a la dist [j]) \cdot k[j]}{N^{\circ} \text{vèrtexs evaluats}}$$

on, per *girth* igual a 10 per exemple, la *taula de ponderació* és:

$$k [j] = \{256,64,16,4,1\}$$

Aquesta ponderació afecta també al pes global que passa a tenir la *Distorsió en l'expansió* respecte a la *Distorsió deguda al girth*. De manera que cal aplicar una correcció en el factor de ponderació de la *Distorsió en l'expansió*,  $k_1$ , per tal que es mantingui l'òptim en la preponderància d'ambdós termes que havíem trobat quan no hi havia ponderació  $k[]$ . Ara cal dividir per un factor 10 el valor òptim trobat per a  $k_1$ , quedant per tant una raó "aparent" de:

$$k_1/k_2 = 0.4$$

però on cal tenir en compte que ara els valors de la *Distorsió en l'expansió*, s'han vist multiplicats, en mitja, per 10.

Per obtenir aquest nou valor, s'han repetit les estadístiques del cas millor buscant que els valors de la *Distorsió en l'expansió* després de multiplicar-la per  $k_1$  fossin similars als que teníem en la versió sense la *taula de ponderació* ( $k[]$ ).

Per al problema ( $\Delta, D$ ):

La funció de cost utilitzada per aquest problema lògicament difereix de la de les *cages*, però parteix de la mateixa filosofia: contempla els aspectes que han de complir tots els vèrtexs d'un graf que sigui solució a aquest problema i introdueix dos termes: un d'avaluació global o "resumida" i un altre de molt més descriptiu de l'estructura del graf a cada iteració. El primer correspon evidentment al diàmetre que "té" cada vèrtex, o *Distorsió deguda al diàmetre*. I el segon té relació amb el *patró d'expansió* que han de "veure" els vèrtexs de la solució, de forma similar a com es fa en el problema de les *cages*. Ara, però, el *patró d'expansió* que han de complir els vèrtexs té una definició molt més

simple, i que deriva directament de la restricció que imposa el diàmetre: el patró és, simplement, que a partir de la distància igual al diàmetre, en la matriu de distàncies del vèrtex no ha d'aparèixer cap nou veí. D'altra manera, significarà que aquell vèrtex té algun veí que es troba a una distància d'ell major a la del diàmetre requerit i, per tant, aquest vèrtex "tindrà" un diàmetre major. I en conseqüència, tot el graf també. Aquest terme és el que hem anomenat *Distorsió en les distàncies*, per diferenciar-lo del de *Distorsió en l'expansió* utilitzat en el cas de les *cages*.

En la següent figura podem veure un exemple de graf (suposant  $D=2$ ) on es destaquen en puntejat els vèrtexs que queden més lluny de la distància diàmetre. Mostrem l'estructura en arbre perquè és més fàcil d'observar-hi aquest aspecte, però l'estructura utilitzada (en arbre o circular) no altera la definició i la validesa del mateix.

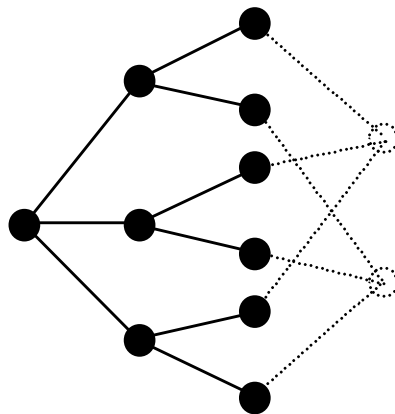


Figura 7.7 Vèrtexs més enllà de la distància diàmetre = 2.

Per tal que es vegi més clar com actua aquest terme, mostrem a continuació una possible *taula de distàncies* dels vèrtexs que en una estructura circular formarien part del *grup patró*, i indiquem quina és la *Distorsió en les distàncies* que se'n deriva (Exemple per a  $D=3$ ).

*Taula 7-5 Exemple de la Distorsió en les distàncies (per D=3)*

<i>Taula de distàncies</i>							
<i>Distància</i>	0	1	2	3	4	5	6
<i>Vèrtex</i>	<i>Nombre de veïns a cada distància</i>						
[vèrtex 0]	1	3	6	12	2	2	0
[vèrtex 1]	1	3	6	10	4	2	0
[vèrtex 2]	1	3	6	11	3	2	0
[vèrtex 3]	1	3	6	11	4	1	0
[vèrtex 4]	1	3	6	12	3	1	0
[vèrtex 5]	1	3	6	10	4	2	0

<i>Distorsió en les distàncies</i>					20	10	0
------------------------------------	--	--	--	--	----	----	---

Com podem observar, aquest terme introdueix efectivament més "detall", més "sensibilitat" a la funció de cost pel fet que es capaç de tenir en compte no només la distància màxima a la que els vèrtexs tenen veïns, sinó el nombre de veïns que té cada un dels vèrtexs més enllà de la distància *diàmetre* i, per tant, quant de lluny està d'aconseguir tenir el diàmetre esperat. Així, si un vèrtex (com el vèrtex 0 de l'exemple) té 4 veïns més enllà de la distància 3 (=diàmetre esperat), està més prop de tenir diàmetre 3 que un vèrtex (com el vèrtex 1) que tingui 6 veïns. Però aquest terme de *Distorsió en les distàncies* ens permet fins i tot diferenciar, d'entre tots els vèrtexs que tenen 4 veïns "de distorsió", quins són millors, considerant quants en tenen a cada distància. I, d'aquesta manera, el vèrtex 4 de l'exemple serà considerat millor que el vèrtex 0, encara que tots dos tenen 4 veïns "de distorsió", pel fet

que el vèrtex 4 en té només 1 a distància 5, mentre que el vèrtex 0 en té 2. Aquest nivell de precisió s'aconsegueix si utilitzem el mecanisme de ponderació de la *Taula de distàncies* que hem vist en la descripció feta per a les *cages*, però ara per a les distàncies més enllà del *diàmetre* i en ordre creixent de penalització.

Així doncs, en el cas del problema  $(\Delta, D)$  la funció de cost utilitzada ha sigut:

$$f(g) = (k_1 \cdot \text{Distorsió en les distàncies} + k_2 \cdot \text{Distorsió deguda al diàmetre}) / (k_4),$$

on:

$$\text{Distorsió deguda al diàmetre} = \frac{\sum_{i=0}^{m-1} (\text{Diàmetre esperat} - \text{diàmetre vist pel vèrtex } [i])}{N^\circ \text{ vèrtexs evaluats}}$$

i,

$$\text{Distorsió en les distàncies} = \frac{\sum_{i=0}^{m-1} \sum_{j=D+1}^{\infty} (\text{taula de distàncies del vtx } [i] \cdot a \text{ la dist } [j])}{N^\circ \text{ vèrtexs evaluats}}$$

### Nombre d'avaluacions de la funció de cost:

Un altre punt molt important, comú a tots els problemes d'optimització combinatòria, i que té impacte directament en l'eficàcia i la rapidesa de l'algorisme per a trobar una solució és el número de vegades que s'ha

avaluar aquesta funció de cost després de cada alteració. És a dir, del número de vèrtexs per als quals s'ha de recalcular cada vegada. Per aquest motiu, tota estratègia que permeti la reducció del número de vèrtexs que considerem alhora avaluar serà de gran rellevància en l'aplicació de l'algorisme.

En el present projecte, aquest ha sigut un dels aspectes que s'ha tingut més en compte alhora de definir l'estratègia d'implementació pràctica de l'algorisme. Aquest aspecte ha sigut fonamental, en concret, en l'elecció de les dues estructures de construcció dels grafs que s'han utilitzat. En ambdues, una part de l'estructura s'ha fet fixa per tal que "l'ordre efectiu" del problema, la part variable, fos reduïda al màxim.

D'una banda, en l'estructura en arbre es fixen tots els vèrtexs i les connexions entre ells des del vèrtex *arrel* fins a l'últim vèrtex del que hem anomenat *la base* (vèrtexs del *nivell base*). La cerca queda així reduïda a considerar només els vèrtexs de *la base*, els de *la copa* i a les connexions entre ells, i és únicament sobre aquests vèrtexs que s'efectuen les alteracions.

D'altra banda, l'elecció de l'estructura circular com a segona alternativa de treball es va realitzar tenint en consideració els patrons repetitius que s'hi poden donar i que, com ja s'ha comentat amb anterioritat en aquest mateix capítol, s'havien observat ja per als *girths* 6 i 8. I l'existència d'aquests patrons ha sigut la base del mecanisme utilitzat per a la construcció i la cerca de les solucions: la repetició d'un *grup patró*.

Així doncs, per a l'estructura en arbre l'avaluació de la funció de cost és

limita a considerar els vèrtexs de *la base* i els de *la copa*, que representen un conjunt força reduït respecte del total. Mentre que per a l'estructura circular, només és tenen en compte els vèrtexs del *grup patró*, donat que la resta de vèrtexs són senzillament una copia d'aquests i, per tant, no aporten informació addicional sobre la proximitat o llunyania del graf respecte de l'estructura cristal·lina òptima. Utilitzant aquesta estructura, doncs, la reducció pot arribar a ser encara major que per a l'estructura en arbre.

A continuació és mostren les reduccions en el número de vèrtexs "significatius" (a avaluar la funció de cost) que s'obtenen a cada *girth* utilitzant una i altra estructura (mostrem per a l'estructura circular els valors típics amb que s'han obtingut resultats d'èxit en aquest projecte):

*Taula 7-6 Nombre de vèrtexs a avaluar segons el mètode de construcció*

<i>Girth</i>	Ordre de la <i>cage</i>	Vèrtexs a avaluar	
		E.en arbre ( <i>base+copa</i> )	E.en cercle <i>grup patró</i>
8	30	20 (12+8)	6
9	58	38 (24+12)	-
10	70	48 (24+24)	10
11	112	66 (48+18)	-
12	126	80 (48+32)	14
13	272	178 (96+82)	16
14	406	312 (96+216)	-
15	620	430 (192+238)	10

### 7.2.2. La resta de paràmetres de l'algorisme

A més de les parts de l'algorisme que hem descrit, que podríem dir que formen el cor del mateix, s'han de tenir en compte la resta de paràmetres que el constitueixen i per als quals cal determinar els valors òptims que han de prendre per a obtenir la màxima eficiència en la seva execució. Per a tots ells, a continuació detallem els criteris utilitzats en la seva determinació i els valors finalment escollits.

#### Temperatura inicial

La temperatura inicial se selecciona tenint en compte que sigui suficientment alta com per que al principi de la simulació s'acceptin gairebé tots els canvis que es realitzen en el graf. En concret han resultat ser òptims valors entre el 80 i el 90% de canvis acceptats a la temperatura inicial.

En el cas de la utilització de l'estructura circular, el % d'acceptacions al principi de l'execució es veu fortament afectat per l'elecció del nombre de vèrtexs que forma el *grup patró*, per més que la temperatura inicial escollida sigui la mateixa. Això es fàcil d'entendre donada la pròpia forma de construcció que utilitzem en aquesta estructura: pel fet que en tot moment reconstruïm el graf complet a partir de la repetició del *grup patró*, com més petit és el nombre de vèrtexs en el *grup patró* i, per tant, més gran el nombre de repeticions d'aquest grup que fem per obtenir el graf total, més impacte té en l'estructura global del graf qualsevol canvi en una de les branques del *grup patró*. Per això, quan treballem amb l'estructura circular hem optat per incloure com a paràmetre d'entrada variable a cada simulació una temperatura inicial específica per a cada valor del *nombre de vèrtexs del grup patró*. Mostrem a continuació alguns dels valors utilitzats en les simulacions

realitzades per a les cages de girth 14 mitjançant la tècnica de construcció en cercle:

*Taula 7-7 Temperatures inicials en funció del n° de vèrtexs al grup patró*

<i>Nombre de vèrtexs del grup patró</i>	<i>Temperatura inicial òptima</i>
8	9.0
10	6.0
12	4.0
14	2.0

#### Refredament de la temperatura

El refredament progressiu de la temperatura es realitza sempre multiplicant la temperatura actual per un factor R menor que 1. És a dir,

$$T_k = R \cdot T_{k-1}$$

Això fa que aquesta disminueixi més bruscament al principi de la simulació i, conforme ens anem apropant a la temperatura de congelació, el refredament vagi esdevenint cada vegada més suau. Aquest és un requisit indispensable per a la convergència de l'algorisme i per alinear-lo amb l'efecte de l'exponencial de Boltzman que va modificant el llinard d'acceptació d'augment en l'energia d'una forma molt més acusada conforme més ens apropem a la temperatura de congelació.

En aquest projecte hem mantingut en totes les simulacions un factor de multiplicació de 0,95 .

#### Temperatura de congelació

Aquest ha sigut també un valor inalterat a lo llarg de tot el projecte, però

diferenciat també segons l'estructura de construcció utilitzada. Així, per a la construcció en arbre hem utilitzat la temperatura 0,001. Mentre que, per a la circular, l'hem situat en un valor de 0,1.

#### Número d'intents a cada temperatura

Com veurem en el proper apartat sobre els criteris d'aturada, el nombre d'intents a cada temperatura pot no ser constant, sinó anar augmentant conforme ens apropem a la temperatura d'aturada. De fet, aquest comportament millora la convergència de l'algorisme donat que permet que també a les temperatures baixes (on s'accepten cada vegada menys alteracions) s'accepti un nombre suficient de canvis que assegurï haver assolit l'estabilitat necessària per poder tornar a refredar la temperatura. D'altra manera, un refredament produït massa aviat en aquestes temperatures faria que l'algorisme s'encallés molt fàcilment en un mínim local. Veieu [35] per a més detalls sobre aquest aspecte de l'algorisme.

#### Criteris d'aturada

Donat que l'objectiu d'aquest projecte és el de mostrar l'aplicabilitat i l'eficiència de l'algorisme de la Recuita Simulada a l'obtenció de *cages* i de solucions al problema  $(\Delta, D)$ , ja siguin conegudes o fins i tot noves fites no obtingudes abans, les úniques solucions que ens seran vàlides són les que corresponguin a grafs amb cost nul.

Donada la forma en la qual ha sigut definida la funció de cost, un graf final que tingui cost no-nul no pot correspondre mai a un *cage*. I, per tant, aquest graf no correspondrà ni tan sols a una solució sub-òptima del problema: senzillament, no el podrem considerar un resultat útil per a nosaltres.

No obstant, el fet que l'únic resultat vàlid sigui l'obtenció d'un graf amb cost

nul, no implica que aquest sigui l'únic criteri d'aturada a utilitzar. Cal considerar d'altres criteris per als casos en que no s'obté una solució, donat que si l'algorisme s'encalla en un mínim local, esperar a l'assoliment de la temperatura de congelació per aturar la cerca és un procediment altament ineficient. Per això, utilitzem també com a criteri el principi segons el qual, si a una certa temperatura el nombre d'alteracions del graf acceptades després de realitzar el nombre d'alteracions inicial,  $L$ , estipulat per a cada temperatura és inferior a un llindar, llavors l'algorisme s'encarrega d'augmentar en un tant per cent el nombre d'alteracions a realitzar. Si s'efectuen les noves alteracions i encara no s'ha assolit el llindar mencionat, l'algorisme torna a realitzar aquest procediment altre cop. I així successivament fins a un màxim d'alteracions igual al doble de les inicialment requerides. És a dir,  $2L$ . Si en assolir aquest màxim d'alteracions, el número de canvis acceptats encara no ha superat el llindar mencionat, llavors l'algorisme considera que difícilment podrà sortir del mínim local en el qual és troba i, per tant, s'atura.

El llindar que es sol prendre per a l'aplicació d'aquest criteri és:

$$\text{Llindar} = L/20.$$

És a dir que imposem que a cada temperatura al menys un 5 per cent ( $Llindar/L$ ) de les alteracions realitzades hagin sigut acceptades per tal de continuar. I en el pitjor dels casos, en seran un 2'5 per cent ( $Llindar/2L$ ). Sinó, aturem l'execució.

Construcció de xarxes d'interconnexió denses i "cages" mitjançant la Recuita Simulada

## 8. SIMULACIONS I RESULTATS

En el capítol anterior hem mostrat, de forma conceptual, l'estratègia utilitzada en l'adaptació de l'algorisme a l'obtenció de *cages* i al problema  $(\Delta, D)$ : les estructures concebudes, els mecanismes utilitzats i les funcions de cost implementades, així com els valors òptims per als paràmetres globals de l'algorisme.

En aquest capítol mostrem l'estructura concreta del programa implementat, incloent totes les seves variants i la forma en que aquestes variants se seleccionen per tal d'escollir la versió de l'algorisme amb la que treballem. I mostrem també els *girths* i els diàmetres estudiats amb cada versió i, finalment, els resultats obtinguts.

### **8.1. Versions de l'algorisme**

Per al problema de les *cages* hem aplicat totes les diverses estratègies que hem anat implementant al llarg del projecte, mentre que al problema  $(\Delta, D)$  hi hem aplicat ja només la versió més eficient. Descriuim a continuació les diferents versions de l'algorisme utilitzades, els casos analitzats amb cada una d'elles i els resultats obtinguts.

#### 8.1.1. Algorisme **A**: *Arbre variable*.

Vàrem començar, en un principi, centrant-nos en l'estratègia de construcció en arbre i en la versió de l'algorisme que contempla la **variació dinàmica de l'ordre** del graf durant l'execució. En aquesta versió es defineix, abans de començar la simulació, una fita inferior per a l'ordre (MINORD, al programa) i una fita superior (MAXORD).

Aquesta versió la vàrem aplicar als *girths* 7, 9 i 11 amb els següents

paràmetres: per al *girth* 7 els valors de MINORD i MAXORD amb que es va treballar van ser 22 i 28, respectivament; per al *girth* 9 es van usar els valors 46 i 66; i per al *girth* 11, els valors 94 i 120.

En el cas del *girth* 7, vàrem obtenir el primer èxit, trobant amb una relativa facilitat la *cake* ja coneguda per a aquest *girth*.

Però la **variació dinàmica de l'ordre** va resultar ser massa inestable per a *girths* majors, a partir del propi *girth* 9; pel fet que cada nova incorporació/eliminació de vèrtexs a/de *la copa* suposa un impacte important en tota l'estructura de *la copa* i en conseqüència, de tot el graf. Això feia que per a *girths* grans es fes molt més difícil d'aconseguir estabilitzar el curs de la simulació, conduint sovint a que l'algorisme s'encallés en mínims locals, amb valors per a l'ordre que no eren òptims.

Així per al *girth* 9, per exemple, en variar lleugerament la preponderància de la *Distorsió en l'expansió* (factor de ponderació PES\_GRAU) respecte la *Distorsió deguda a l'ordre* (PES\_ORDRE), es passava de quedar-se encallat en l'ordre mínim possible (46) a començar a augmentar l'ordre fins arribar al màxim donat (66) i ja no sortir-ne.

Per aquests motius, a continuació vàrem implementar una nova versió de l'algorisme que treballa amb un ordre prefixat per a les solucions i el manté durant tota la simulació.

### 8.1.2. Algorisme **B**: *Arbre fix*.

Aquesta versió treballa igualment amb la construcció en arbre, però manté fix l'ordre de les solucions durant tota la simulació.

Per aquesta nova versió, en el programa senzillament es va establir que MAXORD=MINORD i PES\_ORDRE=0.

Amb ella, vàrem estudiar de nou les *cages* de *girths* 7, 9 i 11, obtenint aquesta vegada un millor comportament i trobant una solució també per al *girth* 9. Cal tenir en compte que coneixíem per avançat l'ordre de la millor solució trobada fins al moment per a cada *girth*, i aquesta solució va ser la que vàrem buscar i obtenir en tots els casos.

No obstant, per al mateix *girth* 11 i en intentar aplicar aquesta versió a *girths* majors (mantenint-nos en l'elecció dels *girths* imparells pels motius que hem explicat en el capítol anterior) ens vàrem topar amb una altra limitació.

La limitació sorgeix en aquesta ocasió per la manera en que hem definit el **mecanisme de construcció en arbre**. Tal com el vàrem concebre, el mecanisme parteix del vèrtex arrel (vèrtex zero) i va construint tots els nivells de vèrtexs que es troben a igual distància d'aquest de forma successiva, un rera l'altre, fins assolir el nivell que hem anomenat *nivell base* o simplement *la base*. Aquest nivell és el que completa l'ordre definit per la *fita de Moore* per al *girth* que estiguem analitzant en cada cas. Per tant, donat que l'ordre dels grafs amb els que treballem està sempre per damunt de la *fita de Moore* (com hem vist al capítol 5), la resta de vèrtexs que hem d'afegir al graf inicial (fet que en aquesta versió de l'algorisme es realitza just abans d'iniciar la simulació) han de situar-se necessàriament per damunt d'aquest nivell base, formant el que hem anomenat *la copa*. Fins aquí, el mecanisme de construcció no té cap dificultat.

A partir d'aquí, doncs, el mecanisme inicia la construcció de la copa. Comença a construir-la en aquest nivell (el que està just per damunt de *la base*) i ho fa afegint-hi vèrtexs de 2 en 2; per tal de seguir mantenint les branques de tots els vèrtexs connectades. Això aparentment no presenta

tampoc cap dificultat, i de fet dóna bons resultats per a grafs de *girths* baixos. Però la problemàtica sorgeix quan treballem amb grafs d'ordres cada vegada majors, i bé donada pel fet que no coneixem realment res sobre l'estructura que ha de tenir *la copa* (a diferència del que passa amb la resta de l'arbre). De fet, qualsevol nou vèrtex que s'introdueix a *la copa* no sabem si ha de situar-se a aquest mateix primer nivell o a un nivell per damunt d'aquest, ni tenim manera de saber en quin moment hem de passar a un nou nivell de *copa*. Això, doncs, posa un límit en la capacitat d'aquest mètode de construcció en arbre per aplicar-lo a recerques de *girths* creixents, pel fet que cada vegada l'ordre de les *cages* s'allunya més de la *fita de Moore*, l'estructura del graf es fa més complexa, i *la copa* es fa cada vegada més gran (arribant a fer-se fins i tot major en nombre de vèrtexs que *la base*) i, per tant, més complexa també, i pot molt ben ser que no tots els vèrtexs de *la copa* s'hagin de situar en un sol nivell per poder obtenir el *girth* esperat.

En concret, les fites inferior i superior per a l'ordre amb que s'ha de treballar en l'estat actual de la cerca de les *cages* són els que hem mostrat a la taula 5.1 al capítol 5, i que impliquen treballar amb les estructures en arbre "teòriques" que mostrem a continuació:

*Taula 8-1 Matriu de distàncies "ideal" d'algunes cages actuals.*

<i>Girth</i>	Fita de Moore	Ordre de les actuals <i>cages</i>	Estructura "teòrica" de les <i>cages</i>
7	22	24	1, 3, 6, 12   2
9	46	58	1, 3, 6, 12, 24   12
11	94	112	1, 3, 6, 12, 24, 48   18
13	190	272	1, 3, 6, 12, 24, 48, 96   82

15	382	620	1, 3, 6, 12, 24, 48, 96, 192   238
----	-----	-----	------------------------------------

on separem amb un "|" la base de la copa.

En aparença, segons la taula anterior, el cas per al *girth* 11 no sembla pas gaire diferent del de *girth* 9. Però el fet és que no ens va ser possible amb aquesta versió de l'algorisme obtenir de cap manera la *cage* ja coneguda per a aquest *girth*; per més simulacions que vàrem intentar. Els resultats ens van fer pensar que possiblement la raó residia en el que hem exposat, i l'estructura de la *copa* d'aquesta *cage* devia diferir del "model pla" (d'un sol nivell de *copa*) que estàvem construint.

Arrel d'aquesta situació va ser que ens vàrem animar a investigar quina era "l'estructura real" de les *cages* actuals. Gràcies a que G.Royle ha inclòs a la seva pàgina web sobre les *cages* també la descripció de moltes d'aquestes, ens va ser possible, utilitzant el programa *mostra\_g* ideat per nosaltres (veure l'Annexe3), d'esbrinar la seva estructura interna fins l'últim detall, "vista" des de tots els seus vèrtexs. Això ens va conduir als resultats per als *girths* 7, 9, 11, 13 i 15, que podem observar a la següent taula. En ella mostrem per a la *cage* de cada *girth* totes les diferents estructures "vistes" pels seus vèrtexs:

Taula 8-2 Matriu de distàncies d'algunes *cages* conegudes.

<i>Girth</i>	Fita de Moore	Ordre de les actuals <i>cages</i>	Estructures "reals" de les <i>cages</i>
7	22	24	1, 3, 6, 12   2
9	46	58	1, 3, 6, 12, 24   22
11	94	112	1, 3, 6, 12, 24, 48   18 1, 3, 6, 12, 24, 48   16, 2

			1, 3, 6, 12, 24, 48   12, 4, 2
13	190	272	1, 3, 6, 12, 24, 48, 96   74, 8
15	382	620	1, 3, 6, 12, 24, 48, 96, 192   188, 44, 6

Com podem observar, efectivament a partir del *girth* 11 inclòs, l'estructura de *la copa* deixa de ser per a tots els seus vèrtexs "l'estructura plana" d'un sol nivell i passa a tenir-ne més d'un.

Aquest descobriment va fer que centréssim finalment la resta de la cerca de solucions a les *cages* i al problema  $(\Delta, D)$  en l'ús del mecanisme de construcció en cercle, que és la base de la tercera versió de l'algorisme que vàrem desenvolupar.

### 8.1.3. Algorisme **C**: Cercle fix amb cordes.

Aquesta versió es fonamenta en l'ús del **mecanisme de construcció en cercle** i, al igual que la versió B, també manté fix l'ordre de les solucions durant tota la simulació.

Segons l'exposat per al mecanisme de construcció en cercle en el capítol anterior, abans d'iniciar l'execució cal escollir un valor "adequat" per al nombre de vèrtexs del *grup patró*, mantenint un compromís: que no sigui "massa petit", perquè llavors no tindrà prou branques per disposar d'un ventall de possibilitats suficientment ampli; ni tampoc "massa gran", perquè llavors perdem gran part de l'eficiència que es deriva de la reducció en el número d'avaluacions de la funció de cost que ens aporta el treballar amb un *grup patró* que es repeteix (veure la Taula 7-4).

Amb aquesta versió, hem treballat sobre els següents casos: les *cages* de *girth* 8, 10, 11, 12 i 14, que exposem a continuació; i per al problema  $(\Delta, D)$ , els diàmetres 5, 8, 9 i 10, que veurem més endavant.

### **8.1.3.1. Les cages**

En el cas de les *cages*, els *girths* 8 i 10 els hem utilitzat a mena de comprovació de la validesa de l'estratègia seguida en aquesta versió de l'algorisme. I, un cop obtinguts els primers resultats, han servit també per a afinar els valors dels paràmetres de configuració de l'algorisme ( $T_0$ ,  $k_1/k_2$ , etc.), mitjançant la realització sistemàtica de repetides simulacions per tal obtenir una certa estadística de l'impacte de cada un dels valors variats.

#### **Cas Girth=8**

Lògicament en aquest cas vàrem provar directament amb el parell de valors  $(g,m)=(5,6)$  que ja coneixíem com a bons. I l'algorisme, efectivament, no va trigar gaire a trobar la solució.

#### **Cas Girth=10**

Aquest ha sigut el cas amb el qual hem treballat més a fons per a la optimització de l'algorisme i els seus paràmetres. Els valors possibles per al parell  $(g,m)$  en aquest cas són:

(35,2)

(7,10)

(5,14)

(1,70)

dels quals només els (7,10) i (5,14) ens resulten interessants. Per a tots dos parells s'han obtingut *cages* òptimes en un nombre d'iteracions força baix. Les solucions obtingudes són:

Parell (7,10)

**Cordes:**

9 23 -34 -28 -23 28 21 -21 34 -9

**Graf:**

;n70g69 35 1;27 2;63 3;12 4;56 5;40 6;24 7;51 8;69 9;18 10;45 11;37 12;13;22 14;66  
15;50 16;34 17;61 18;19;28 20;55 21;47 22;23;32 24;25;60 26;44 27;28;29;38 30;65  
31;57 32;33;42 34;35;36;54 37;38;39;48 40;41;67 42;43;52 44;45;46;64 47;48;49;58  
50;51;52;53;62 54;55;56;57;58;59;68 60;61;62;63;64;65;66;67;68;69;.

Parell (5,14)

**Cordes:**

(Solució 1) 15 -15 35 -27 27 -23 15 -15 -9 -35 23 -27 27 9

(Solució 2) 13 19 29 -29 33 21 -19 -13 13 -33 -27 27 -21 -13

**Graf:**

(Solució 1) ;n70g69 15 1;56 2;37 3;46 4;31 5;52 6;21 7;62 8;69 9;44 10;33 11;54  
12;39 13;22 14;29 15;16;51 17;60 18;45 19;66 20;35 21;22;23;58 24;47 25;68 26;53  
27;36 28;43 29;30;65 31;32;59 33;34;49 35;36;37;38;61 39;40;67 41;50 42;57  
43;44;45;46;47;48;63 49;50;51;52;53;54;55;64  
56;57;58;59;60;61;62;63;64;65;66;67;68;69;.

(Solució 2) ;n70g69 13 1;20 2;31 3;44 4;37 5;26 6;57 7;64 8;21 9;46 10;53 11;38  
12;61 13;14;27 15;34 16;45 17;58 18;51 19;40 20;21;22;35 23;60 24;67 25;52  
26;27;28;41 29;48 30;59 31;32;65 33;54 34;35;36;49 37;38;39;66 40;41;42;55 43  
;62 44;45;46;47;68 48;49;50;63 51;52;53;54;55;56;69  
57;58;59;60;61;62;63;64;65;66;67;68;69;.

A més hem realitzat amb tots dos parells la següent estadística d'avaluació del valor òptim per a la raó  $k_1/k_2$ , que determina (com hem vist a l'apartat 7.2.1.3 del capítol 7 en parlar de la funció de cost) la preponderància de la *Distorsió en l'expansió* respecte de la *Distorsió en el girth*.

Taula 8-3 % d'èxits en funció de la raó  $k_1/k_2$

$k_1/k_2$	Parell (5,14)		Parell (7,10)	
	% d'èxits	Nº mig d'iteracions	% d'èxits	Nº mig d'iteracions
0.25	10%	198605	0%	-
1	50%	210524	10%	204608
4	60%	214825	20%	132153
9	40%	154832*	10%	96990

[\* No obstant, sobre aquest valor cal remarcar que la **desviació mitjana de l'estadística** en aquest cas és exageradament més gran que en la resta de casos, donant-se un cas d'èxit amb tan sols 19.400 iteracions amb d'altres de 268.214. Això és degut al que s'explica a continuació].

Com es pot observar en aquesta taula, hi ha un punt d'inflexió en el valor  $k_1/k_2 = 4$ , a partir del qual torna a disminuir la probabilitat d'èxit en la cerca, tot i que el nombre d'iteracions mig quan la raó val 9 es significativament menor al de raó igual a 4. Això és degut a que certament incrementar el pes de la *Distorsió en l'expansió* respecte de la *Distorsió en el girth* fa que la "topologia" de la funció de cost es torni molt més abrupta (recordem que aquest primer terme de la funció de cost té molta més "sensibilitat" que el relacionat amb el *girth*, donat que aquest últim correspon a una característica global; dóna una informació "filtrada", podríem dir). I, per tant, d'una banda això fa que els òptims (mínims absoluts de la funció de cost) destaquin molt

més respecte de la resta de solucions i, per tant, "el camí de baixada" sigui molt més ràpid; però d'altra banda, també fa que la resta de sub-òptims (mínims locals de la funció de cost) destaquin més respecte els seus veïns locals i, en conseqüència, sigui molt més fàcil que l'algorisme quedi encallat en un d'ells; disminuint així les probabilitats d'èxit.

### Cas Girth=11

Per al *girth* 11 s'ha demostrat (per mètodes computacionals) que l'ordre mínim possible per a les seves *cages* és 112, i fins al moment, només se n'ha trobat una; tot i que n'hi poden haver més.

En aquest cas, donat que l'ordre és  $112 = 2^4 \cdot 7$ , els valors assajats de  $(g,m)$  han sigut: (14,8) i (8,14). Però amb cap dels dos ens ha sigut possible obtenir aquesta *cage* o una de nova. I, per tant, hem deduït que probablement aquesta *cage* no s'ajusti a les característiques de repetibilitat periòdica, que venen implícites amb l'ús de la construcció en cercle amb repeticions d'un *grup patró*.

### Cas Girth=12

Per aquest *girth*, no hi ha tampoc possibilitat de millora donat que la *cage* òptima és coneguda i única, i a més coincideix en l'ordre amb el mínim teòric possible per a aquest *girth*, 126. Tenim doncs que  $126 = 2 \cdot 3^2 \cdot 7$ , i els parells possibles són: (3,42), (7,18), (9,14) i (21,6). No obstant, tampoc en aquest cas ens ha sigut possible trobar la *cage*. I, donat que aquest és un cas que igualment no permet millora, hem passat directament al de *girth* igual a 14.

### Cas Girth=14

Per aquest *girth* la situació ja canvia, doncs existeix aquí un marge de valors

per a l'ordre de les possibles *cages*. Aquest rang ve donat, d'una banda per la darrera fita per sota del qual s'ha demostrat que no en pot existir cap, 258, i d'altra banda per l'ordre de l'última *cake* trobada, 406. Entre aquests dos valors, teòricament, encara seria possible trobar una nova *cake*.

En aquest projecte, el que hem fet ha sigut principalment intentar reproduir la *cake* d'ordre 406, i paral·lelament, cercar noves *cages* d'ordre menor.

Per a l'ordre 406 el parell utilitzat ha sigut el (29,14), però no hem trobat una solució que compleix aquest ordre.

Igualment hem realitzat intents per als ordres 400, 396, 392, 390, 380, 374, 370 i 322. Aquesta és la taula dels parells intentats:

*Taula 8-4 Parells intentats per al girth 14*

<b>Ordre</b>	<b>Parells</b>
406	(29,14)
400	(25,16) (40,10) (50,8)
396	(18,22) (33,12)
392	(28,14) (49,8)
390	(15,26) (39,10) (65,6)
380	(27,14) (38,10)
374	(17,22)
370	(37,10)
322	(23,14)

### **8.1.3.2. El problema ( $\Delta, D$ )**

En el cas de l'aplicació de la Recuita Simulada al problema ( $\Delta, D$ ) ens hem centrat fonamentalment en els casos  $D = 8$  i  $D = 9$  perquè d'entre tots els

casos de  $\Delta=3$  és en aquests són dos dels tres casos on s'havien produït les darreres millores i alhora on també hi ha una possibilitat amplia d'assolir nous òptims, donat que la *fita de Moore* hi queda encara molt lluny. El valor de la *fita de Moore* per aquest dos casos, així com les darreres fites assolides es mostren a la taula següent:

*Taula 8-5 Últimes fites obtingudes per al problema ( $\Delta, D$ ) per  $D=8$  i  $9$ .*

D	Fita de Moore	Última fita
8	766	330
9	1534	570

Hem aplicat, però, l'algorisme també per al cas  $D = 5$  de dimensió menor per tal de posar a punt els paràmetres globals de l'algorisme abans d'abordar els casos de major dimensió.

Exposem a continuació la situació per a cada un dels diàmetres estudiats per separat i el treball realitzat; fent també una breu ressenya de l'autor de la darrera fita assolida a l'inici del nostre projecte i del mètode que havia utilitzat per aconseguir-la, per a cada un dels diàmetres.

### **Cas Diàmetre = 8**

El darrer òptim, de 330 vèrtexs, el va obtenir G.Exoo [23] el 1998 mitjançant l'aplicació d'un algorisme on combinava la Recuita Simulada i la Cerca Tabú. Donat que Exoo va fer públic el graf, ens va ser possible comprovar-ne la veracitat mitjançant un programa auxiliar anomenat *mostra\_G* que hem desenvolupat especialment per aquesta tasca (veieu l'Annex 4 per una

descripció detallada), així com també amb d'altres programes especialitzats. El programa *mostra\_G*, dóna com a resultat la taula de distàncies de cadascun dels vèrtexs, així com el seu diàmetre. En aplicar-lo al graf obtingut per Exoo, ens va permetre observar que per a tots els seus vèrtexs el diàmetre era sempre 8, confirmant, per tant, la validesa del graf. Però també ens va mostrar una certa repetibilitat en els valors de la taula, que de fet són coherents amb la forma en la que va ser construït el graf segons la descripció feta per Exoo. Aquesta repetibilitat mostra grups de 33 vèrtexs amb idèntica matriu de distàncies, corresponents probablement als 10 grups de 33 vèrtexs en què Exoo va basar el seu mètode.

Per tant, va resultar evident que, en el nostre esquema de construcció, el cas amb majors probabilitats d'èxit era el de 33 grups de 10 cordes. Així doncs, aquest va ser el primer a ser explorat, i ens va conduir efectivament a una solució de 330 vèrtexs. No obstant, a posteriori vàrem intentar també els parells (15,22) i (11,30) a mode de comprovació, sense obtenir cap resultat.

En la solució trobada, les cordes són:

-126 -84 34 -155 126 -96 -34 84 155 96

Estar per demostrar si la solució trobada és idèntica (isomorfa) amb la trobada per Exoo o no.

#### Nous intents:

A continuació vàrem abordar grafs d'ordre major, en l'intent de millorar aquesta fita.

El primer cas que vàrem intentar va ser el de 336 vèrtexs, donat que és el

següent ordre més proper a 330 que es pot formar amb grups de 8, 12 o 14 vèrtexs. Casualment, a demés, tots tres tipus de grups coincideixen en aquest ordre, podent-se formar amb ells tres parells vàlids, que són:

(42,8)

(28,12)

(24,14)

Per tant, es va posar en marxa una simulació amb aquests tres parells a ser assajats, però no es va obtenir èxit en aquest cas.

També es van provar a posteriori els ordres 340 (34·10), 344 (43·8) i 350 (35·10), però tampoc s'han obtingut resultats vàlids.

### **Cas Diàmetre = 9**

La última fita en aquest cas té 570 vèrtexs, i va ser trobada per Jean-Jacques Quisquater. No obstant, l'autor no ha donat mai una demostració del mètode utilitzat ni tan sols la descripció del graf. Això fa que aquest cas sigui, doncs, més interessant encara.

L'objectiu inicial va ser, per tant, provar obtenir una solució amb aquest mateix nombre de vèrtexs.

570 es pot descomposar com  $2 \cdot 5 \cdot 57$ , i així, les configuracions possibles per a aquest ordre són: (57,10) i (5,114).

El primer és el que dóna una reducció en el nombre de nodes a estudiar suficientment bona, mentre que el segon resulta molt menys òptim en aquest sentit. Per tant, hem provat directament amb el primer parell.

Doncs bé, l'execució de l'algorisme realitzada amb el parell (57,10) ha conduït

a una solució, que ve descrita per les cordes:

-255 -58 -164 58 -237 255 -47 237 164 47

I hem comprovat que, efectivament, la solució té diàmetre 9. I, per tant, aquest és el primer graf trobat de 570 vèrtexs que resol el problema  $(\Delta, D)$  per a  $\Delta=3$  i  $D=9$  i que pot ser comprovat; donat que Quisquater no ha donat mai el graf suposadament trobat per ell. Aquest resultat és doncs, ja per sí sol, el resultat més manifest de l'èxit del projecte, de l'enfoc utilitzat i de tota l'estratègia seguida. I a més és un resultat que pot ser presentat a nivell de la comunitat matemàtica mundial per a la seva corroboració.

Nous intents:

A continuació, varem provar també els següents ordres a aquest que resulten interessants a priori. Exposem a continuació una taula il·lustrativa dels valors útils i dels parells en que es poden descomposar, on hem ressaltat en negreta aquells que resulten útils per a l'enfoc del nostre projecte:

*Taula 8-6 Parells intentats per al Diàmetre 9*

<b>Ordre</b>	<b>Descomposició</b>	<b>Parells</b>	<b>Útils</b>
572	$2^2 \cdot 143$	(143,4)	No
<b>574</b>	$2 \cdot 7 \cdot 41$	(41,14)	<b>SI</b>
<b>576</b>	$2^6 \cdot 3^2$	(72,8) (48,12) (36,16)	<b>SI</b>
578	$2 \cdot 17^2$	(289,2)	No
<b>580</b>	$2^2 \cdot 5 \cdot 29$	(58,10)	<b>SI</b>
582	$2 \cdot 3 \cdot 97$	(97,6)	No
<b>584</b>	$2^3 \cdot 73$	(73,8)	<b>SI</b>

588	$2^2 \cdot 3 \cdot 7^2$	(49,12)	SI
-----	-------------------------	---------	----

De tots aquests, els que hem intentat han sigut els ordres 574, 576 i 580, però cap d'ells ens ha dut a noves solucions vàlides.

## 8.2. Resum de resultats

El conjunt d'aquests resultats els podem sintetitzar en els següents punts:

- S'ha demostrat que, de les dues formes de construcció ideades, construcció en arbre i construcció en cercle, **la construcció en cercle** dels grafs resulta ser un enfoc molt encertat per encarar tots dos problemes, doncs no només ens ha conduït a solucions òptimes en tots dos, sinó que, a més, ens ha permès fer-ho en temps de computació molt inferiors als utilitzats fins al moment amb qualsevol altres mètodes (veure [3], [11] i [22]). Això s'ha aconseguit, d'una banda gràcies a l'ús de la Recuita Simulada, i d'altra banda gràcies a la reducció en el nombre de vèrtexs a analitzar a cada iteració obtinguda en construir les solucions com a repetició d'un grup reduït de vèrtexs (*grup patró*). Amb aquesta estratègia hem aconseguit que la mida (en termes de "vèrtexs a contemplar en els càlculs") d'una instància del problema, que inicialment és igual a l'ordre dels grafs amb que es treballa, quedi reduïda dràsticament, arribant fàcilment a un 5% de la mida original; o fins a un 1,3% en els millors casos estudiats. I consegüentment, l'espai d'estats "virtual" amb el que treballem encara es veu més fortament reduït, donat que per als problemes que tractem la proporció és no-polinomial.
- Hem pogut igualar amb aquesta tècnica algunes de les millors solucions

trobades fins al moment per als problemes  $(\Delta, D)$  i de les *cages* per part d'altres investigadors. En concret, hem corroborat la validesa dels actuals òptims per als casos  $D = 5$  (solució de 70 vèrtexs),  $D = 8$  (330 vèrtexs) i  $D = 9$  (570 vèrtexs) en el problema  $(\Delta, D)$  i per als casos *girth* = 8 (30 vèrtexs) i 10 (70 vèrtexs) per al de les *cages*.

- Els resultats mostren una important diferència entre el nivell d'èxit obtingut en el problema  $(\Delta, D)$ , i l'obtingut en el problema de les *cages*. Mentre per al problema  $(\Delta, D)$  s'han aconseguit els òptims existents per a tots els casos aquí analitzats (fins i tot per al cas de diàmetre 9 de 570 vèrtexs), per a les *cages* no hem pogut obtenir èxits més enllà del graf de 70 vèrtexs de *girth* 10. L'anàlisi dels resultats (veure 8.1.2 i 8.1.3) ens porta a la conclusió que aquesta diferència és conseqüència directa de la definició de les funcions de cost escollides per a cada un dels problemes. En concret, la funció de cost per al de les *cages* ha resultat inadequada per a *girths* grans pel fet que la seva avaluació només avarca fins a una distància límit de la taula de distàncies de cada un dels seus vèrtex (distància *nivell base*), però "perd la visibilitat" més enllà. Per això, conforme la dimensió dels grafs solució creix en créixer el *girth*, hi ha cada vegada una major apart de vèrtexs que queden "sense avaluar" (tots els de *la copa*). En canvi, per al problema  $(\Delta, D)$  la funció de cost avalua fins a la última distància a la que cada vèrtex hi tingui encara algun veí i, per tant, "veu" i "penalitzat" perfectament tots els vèrtexs que generen distorsió en l'estructura.
- S'ha aconseguit també (veure Taula 8-3) trobar l'òptim en els termes de la funció de cost, per tal que hi hagi l'equilibri necessari entre **sensibilitat**

(*Distorsió en l'expansió* o *Distorsió en la distància*, segons es tracti de les *cages* o del  $(\Delta, D)$ , respectivament) i **estabilitat** (*Distorsió deguda al girth* o *Distorsió deguda al diàmetre*, respectivament), de manera que el "paisatge" que presenta la funció no sigui ni massa abrupte, facilitant d'encallar-se en mínims locals; ni massa pla, fent que l'algorisme doni voltes sense arribar enlloc.

## 9. CONCLUSIONS

Del conjunt de resultats obtinguts se'n deriven les conclusions que exposem a continuació:

- L'algorisme de la Recuita Simulada pot ser efectivament adaptat per a la cerca de solucions tant per al problema de les *cages* com per al problema  $(\Delta, D)$ . I, a més, resulta molt fàcil d'adaptar per aplicar-lo a nous problemes similars un cop ja se n'ha fet una implementació que resulti eficaç; tal com hem fet en aquest projecte per passar de la implementació per a les *cages* a la del problema  $(\Delta, D)$ .
- La RS ha demostrat ser un mètode útil i eficient en molts dels casos estudiats, que condueix a resultats òptims en temps de computació relativament curts si es realitza una bona elecció dels elements principals que el formen: la forma de construcció dels grafs, el mètode de transformació dels mateixos i una funció objectiva que sigui independent de l'ordre dels grafs estudiats.
- S'ha demostrat, per tant, l'enorme "rendibilitat", per dir-ho així, de l'algorisme de la RS en quant a que el seu esquema tant senzill i genèric, tan "robust" per sí mateix, un cop aplicat a un problema concret fa que resulti igualment eficaç per a qualsevol dimensió del problema sempre que se seleccionin, és clar, els elements adequats que el componen. En el nostre cas, un cop definida la forma de construcció dels grafs més adequada, **la construcció en cercle**, el mètode de transformació dels mateixos, **l'intercanvi de cordes**, i una funció objectiva que sigui independent de l'ordre dels grafs estudiats, **la Distorsió en la matriu de**

**distàncies**, hem pogut aplicar l'algorisme amb un simple canvi dels paràmetres a qualsevol diàmetre que ens ha interessat, en el cas del problema  $(\Delta, D)$ , o a qualsevol *girth*, en el cas de les *cages*.

- Això ens ha conduït a la conclusió que, en especial en el cas del problema  $(\Delta, D)$ , les configuracions que resulten útils tenen sempre una estructura altament simètrica, i que estan directament relacionades amb els nombres primers. És a dir, en utilitzar el nostre mètode de construcció en cercle i haver obtingut resultats vàlids, queda patent d'immediat que per als problemes teòrics aquí tractats, on cada vèrtex i cada branca és exactament idèntic, les millors solucions tenen en molts dels casos patrons de repetició periòdics. Les solucions trobades així ho demostren, com per exemple la configuració de 5 grups de 14 vèrtexs (5 per  $7 \cdot 2$ ) de la  $(3,10)$ -*cage* obtinguda en aquest projecte, o la configuració de 33 grups de 10 vèrtexs ( $3 \cdot 11$  per  $5 \cdot 2$ ) o 57 grups de 10 vèrtexs ( $57$  per  $5 \cdot 2$ ) dels casos  $(3,8)$  i  $(3,9)$  del problema  $(\Delta, D)$ , respectivament.
- De tots els resultats obtinguts, destaquen les solucions (ja conegudes) que hem trobat per als casos de Diàmetre 8 i 9 del problema  $(\Delta, D)$ , que juntament amb la solució de Diàmetre 5 representen un èxit en la consecució de l'objectiu d'aquest projecte: mostrar la capacitat de la Recuita Simulada per a tractar i trobar solucions per a aquest tipus de problemes (NP-Complets) i mostrar-ne també la seva *robustesa*; que es comprova en el fet que hem pogut aplicar-lo amb èxit a una ampla gamma de casos sense haver realitzar adaptacions especials de cap classe, fos quina fos la dimensió tractada. Això significa una avenç respecte els enfoc heurístics que queden limitats a casos molt especials i no poden

aplicar-se més enllà, com és el cas dels mètodes que van utilitzar d'altres investigadors abans que nosaltres per a trobar les mateixes solucions aquí obtingudes (veure [11] i [22], per exemple).

Conclusió final:

Tot plegat ens porta a la conclusió final de que la Recuita Simulada, correctament adaptada, és una eina que pot resultar molt eficient per al disseny de xarxes d'interconnexió de dimensions grans i amb requeriments o especificacions molt restrictives que es fan impossibles de resoldre òptimament per mètodes clàssics d'anàlisi lineal. A més, aporta un mètode global i independent de les dimensions del problema, i fàcilment adaptable a una ampla varietat de problemes, superant així els mètodes heurístics que no disposen de les característiques de versatilitat, robustesa i aplicació sistemàtica que requereix una eina que pugui servir per a l'ús professional per part d'un enginyer dissenyador de xarxes.

Noves perspectives:

Queda ara, doncs, oberta la porta per a seguir utilitzant la Recuita Simulada en la cerca de noves estructures altament simètriques o bé a estructures amb propietats molt singulars i que, per tant, són de gran interès en el camp de les xarxes d'interconnexió en que hem centrat aquest projecte, però també en d'altres camps tant innovadors com l'assignació de freqüències en xarxes de cobertura de telefonia mòbil o la definició d'algorismes d'encaminament per a *routers* en xarxes TCP/IP.

En aquest darrer àmbit, per exemple, el treball aquí realitzat podria aplicar-se com a mètode per a la decisió dels camins òptims d'encaminament de paquets per a xarxes que requereixen una cert nivell de QoS (*Quality of*

*Service*); en la línia del que pretén el darrer protocol estàndard que s'està desenvolupant com a successor del RIP (*Routing Information Protocol*): l'OSPF. L'OSPF (*Open Shortest Path First* - El camí més curt primer) utilitza algorismes basats en vectors-distància (*vector-distance algorithms*), que es podrien implementar en el codi del nostre projecte en la forma de pesos per a les branques que uneixen cada parell de nodes, on els nodes representen aquí tant *gateways* com xarxes locals. Aquests dos tipus de nodes estan units entre si pel fet que cada *gateway* uneix dues xarxes locals i cada xarxa pot estar unida a més d'un *gateway* tal com mostra la següent figura (veure [18]):

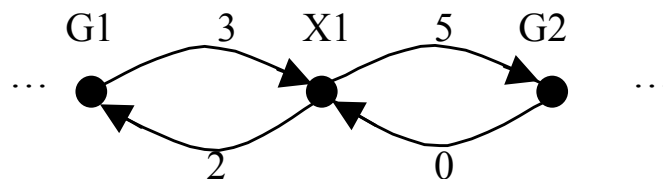


Figura 9.1 Model de la xarxa en que es basa l'OSPF

Sobre aquest esquema, els pesos en les branques - que aquí passen a ser branques dirigides i, per tant, passem a treballar amb digrafs - corresponen al retard en la comunicació introduït per aquell enllaç. L'OSPF defineix el protocol que hauran d'utilitzar i entendre la nova generació de *routers* (ja en comercial) per intercanviar-se aquest tipus d'informació, de forma que tots els *routers* de la xarxa puguin confeccionar les seves taules d'encaminament òptim i modificar-les de forma dinàmica conforme van canviant els valors dels retards. L'objectiu de l'OSPF és doncs la cerca dinàmica del camí més

curt en termes de retard entre dos nodes qualsevol de la xarxa. I que, per tant, pot no sempre correspondre's amb el camí més curt en termes de nombre de nodes. És a dir, pot haver-hi un camí que uneixi dos nodes-xarxa que passi només per 2 nodes-*gateway* i un node-xarxa intermig, per exemple, però que tingui uns retards que el facin més lent que un altre camí que passi per 4 nodes-*gateway* i 3 nodes-xarxa.

Per tant, podríem adaptar fàcilment el treball aquí realitzat perquè tingués en compte, en primer lloc el treball amb digrafs, i en segon lloc la *taula de distàncies* en termes de "branques ponderades" (retards) enlloc de "nombre de branques" (sota aquesta visió ampliada, podríem dir que el valor de ponderació de les branques que hem fet servir en el present projecte ha sigut sempre igual a '1').

Esperem que aquest projecte hagi servit per mostrar la potència de la Recuita Simulada com a algorisme de resolució per a problemes complexos i que la forma en la que l'hem exposat (inclòs el codi dels programes) faciliti a d'altres enginyers el camí cap a la seva utilització en aquestes i moltes altres aplicacions que pugin ser útils per a la societat.

L'essència de l'algorisme de la Recuita Simulada, que està inspirada en el fenomen de la cristallització, demostra una vegada més que la clau de l'èxit en aquest món - tant a les matemàtiques, l'enginyeria, com la Natura - es troba sempre en utilitzar l'enfoc més senzill.

Com més descobrim, més ens adonem de tot el que encara ens queda per descobrir...

## 10. ANNEX 1 – DESCRIPCIÓ GENERAL DEL PROGRAMA

### **10.1. Estructura lògica del programa (en pseudo-codi)**

- 1) Guardar variables per a la capçalera, entre elles l'instant inicial d'execució.
- 2) Construir els noms dels fitxers de resultats i traces.
- 3) Si és en cercle, Recollir\_dades des del fitxer de dades:
  - a) Recollir el nombre de parells que s'han d'assajar.
  - b) Recollir tots els parells, amb les seves temperatures inicials respectives.
- 4) Repetim per a cada parell, si treballem en cercle; o bé, una sola vegada, si treballem en arbre:
  - a) Inicialitzem les variables i estructures d'entorn.
  - b) Obrim els fitxers de resultats i traces.
  - c) Hi escrivim la capçalera.
  - d) Iniciem l'execució de la part Principal de l'algorisme, que repetirem SIMULACIONS vegades:
    - i) *Si treballem en arbre:*
      - (1) *Construir\_arbre\_basic*
      - (2) *Construir\_copa*

*Sinó:*

      - (1) *Construir\_graf\_circular*
    - ii) *Salvar graf inicial.*
    - iii) *Inicialitzar la temperatura inicial  $t=t_0$ .*
    - iv) *Iniciar el Procés de Recuita i Refredament progressiu.\**
    - v) *Un cop finalitzat o aturat aquest, guardar l'instant de finalització.*
    - vi) *Recalculat la taula de distàncies per a tots els vèrtexs de la millor solució trobada.*

vii) *Mostrar-ne les propietats, per pantalla i als fitxers de resultats.*

viii) *Escriure el graf en format compacte i estàndard al fitxer de solucions trobades.*

ix) *Escrivim les dades de final de la simulació als fitxers de resultats.*

x) *Guardar en memòria els resultats d'aquesta simulació per al resum final.*

e) Un cop finalitzades totes les simulacions, mostrem el resum de totes elles als fitxers de resultats.

5) Tanquem finalment els fitxers de resultats.

(\*) Detall del:

**Procés de Recuita i Refredament** (punt 4.d.iv):

A) Inicialitzem els valors del nombre d'intents mínim a realitzar a aquesta temperatura, el màxim i el de grafs acceptats fins al moment.

B) Avaluem la funció de cost per al graf inicial.

C) Iniciem les iteracions a la temperatura actual, i per a cada iteració:

C.1) Guardem les característiques (ordre, branques,...) del graf actual abans d'alterar-lo.

C.2) Segons treballem amb arbre o cercle:

- Alterar\_arbre:

De forma aleatòria:

- Intercanviar dues branques, o bé
- Augmentar l'ordre del graf (si es pot), o bé
- Disminuir l'ordre del graf (si es pot).

- Alterar\_cercle:

De forma aleatòria:

- Intercanviar dues branques, o bé
- Variar només l'increment intergrup d'una de les branques.

C.3) Avaluem la funció de cost per al nou graf alterat.

C.4) Si l'energia del nou graf és menor:

- Acceptar-lo, i
- Si és la millor trobada fins el moment:
  - Guardar-lo com a millor solució, i
  - Mostrar-ne les característiques als fitxers de resultats i traces.

Sinó, si l'energia és major:

- Acceptar-lo amb una probabilitat donada per:

$$\text{prob} = \exp(-\Delta E/t).$$

C.5) Si el nou graf ha sigut acceptat:

- incrementar el comptador d'acceptats,
- copiar-lo com a nou graf inicial, i
- posar-ne les característiques (ordre, branques,...) com a les actuals.

Sinó,

- restaurem les característiques del graf inicial.

C.6) Comprovem el tant per cent de grafs acceptats respecte dels intents a

aquesta temperatura, i si **no** s'ha assolit el mínim

(intents/FRACCIO), llavors:

Si encara no hem assolit el màxim d'intents permès a aquesta temp.:

- Augmentar el nombre d'intents a fer en un tant.

Si hem assolit el màxim:

- Aturem l'actual simulació per aquest parell per considerar que ens hem encallat en un mínim local.

## **10.2. Constants de configuració de l'algorisme (fitxer capçalera)**

Com podem veure en aquest esquema general del programa, les diverses opcions de construcció, alteració i avaluació es troben integrades dins del

mateix cos del codi. Per seleccionar-les hi ha un conjunt de constants que es defineixen en temps de compilació (mitjançant *#defines*) i que exposen a continuació indicant-ne el seu significat (mostrem aquí les del programa per a les *cages*):

**1. Dimensions del problema tractat (*cage* cercada):**

```
#define GIRTH      7      /* Girth de la cage que volem trobar. */  
#define MINORD    22     /* n° de nodes per començar a treballar. */  
#define MAXORD    28     /* n° de nodes màxim a manejar. */
```

Els paràmetres MINORD i MAXORD serveixen per definir el rang de valors per a l'ordre amb que volem treballar i també les dimensions de les estructures que cal reservar per als grafs, taules, etc.

**2. Selecció del tipus d'estructura amb que treballem:**

```
#define CIRCULAR    0
```

El valor 0 (zero) indica que utilitzarem la construcció en arbre i el valor 1, la construcció en cercle.

**3. Constants de configuració dels diversos mecanismes:**

**a) Mecanisme d'Avaluació (Funció de cost):**

```
#define TIPUS_DE_COST      DISTORSIO  
#define EXCENTR_USADA     COMPLERTA
```

Seleccionem el tipus de distorsió utilitzada entre:

**b) Mecanisme d'Alteració:**

```
#define EXPANSIO 15 /* Grau d'expansibilitat */  
#define ROTACIO 10 /* Grau de rotació sense creuar branques */
```

**c) Nombre d'intents a cada temperatura:**

```
#define K_INTENTS 2.0  
#define K_MAXT 4.0  
#define FRACCIO 10 /* Fracció d'intents que han de tenir èxit */
```

**4. Selecció de les constants de ponderació de les diverses components de la funció de cost:**

```
#define PES_DISTORSIO_GRAU 8 /* Per a la Distorsió en l'expansió */  
#define PES_DISTORSIO_GIRTH 20 /* Per a la Distorsió deguda al girth */  
#define PES_NODES_EXTRA 0 /* Per a la Distorsió deguda a l'ordre */
```

**5. Temperatures inicial i final:**

```
#define TINI 0.8 /* Només si treballem amb la construcció en arbre */  
#define TFIN 0.001
```

**6. Selecció de les traces a mostrar:**

```
#define PANTALLA 1  
#define MOSTRATOT 0  
#define MOSTRATEST 1  
#define MOSTRAMUT 1  
#define GUARDARES 1  
#define GUARDATEST 1  
#define GUARDACABRI 1
```

## 7. Altres paràmetres de l'execució:

```
#define OF_CADA_COP    1    /* Obrir i tancar fitxers de resultats abans i  
després de cada traça. */  
  
#define ITERACIO      5    /* Identificador de l'executable */
```

### 10.3. Fitxer de dades

Per a la l'execució del programa en la versió C (construcció en cercle) tant per a les *cages* com per al problema  $(\Delta, D)$ , a més de definir tots aquests paràmetres que acabem de veure, cal també preparar un fitxer de dades on s'indiquen els parells  $(g, n)$  a assajar i les temperatures inicials a utilitzar en cada cas. El format és:

**<nº de parells que venen a continuació>**

**$g_1 \ m_1 \ t_{ini1}$**

**$g_2 \ m_2 \ t_{ini2}$**

...

Un exemple per a la *cage* de *girth* 10 és:

**2**

**5 14 0.8**

**7 10 0.8**

Una altra possibilitat que ofereix el programa, i que afecta al format del fitxer de dades, és la de re-iniciar l'execució a partir d'un graf determinat, del que disposem de la seva descripció de cordes. En aquest cas, cal posar el **#define DAT** a **1** i el format del fitxer d'entrada és ara:

**$g \ m \ corda_1 \ corda_2 \ \dots \ corda_m$**

Un exemple que tindria relació amb l'exposat abans podria ser el següent:

*5 14 15 -15 17 29 -29 -17 31 -31 43 -43 45 57 -57 -45*

on indiquem al programa que iniciï l'execució prenent per graf inicial un graf format per 5 grups de 14 cordes, i les cordes venen descrites a continuació (-15, 15, ...).

Cal destacar que en aquest ús del programa, la temperatura inicial es pren del fitxer capçalera, enlloc del fitxer de dades.

## 14. BIBLIOGRAFIA

- [1] E.Aarts, J.Korst, J.M.van Laarhoven. "Simulated Annealing", Local Search in Combinatorial Optimization. Edited by E.Aarts and J.K.Lenstra. John Wiley & Sons Ltd., 1997, pp. 91-120.
- [2] A.T.Balaban. "Trivalent graphs of *girth* nine and eleven and relationships among the cages", Rev. Roumaine Math 18, 1973, pp. 1033-1043.
- [3] R.Bar-Yehuda, T.Etzion, "Connections between two cycles – A new design of dense processors interconnection networks", Discrete Applied Mathematics, 37/38 (1992). pp. 29-43.
- [4] J.-C.Bermond, C.Delorme and J.J.Quisquater. "Table of large  $(\Delta,D)$ -graphs". Discrete Applied Mathematics, 37/38 (1992), pp. 575-577.
- [5] N.Biggs. "Algebraic Graph Theory", Cambridge Math. Library ISBN 0-521-45897-8 pbk, (1974, 1993 (2nd edition)).
- [6] N.L.Biggs and M.J.Hoare. "A trivalent graph with 58 vertices and *girth* 9", Discrete Math 30, 1980, pp. 299-301.
- [7] N.L.Biggs and M.J.Hoare. "The sextet construction for cubic graphs". Combinatorica 3, 1983, pp. 153-165.
- [8] N.L.Biggs. "Cubic Graphs with Large *Girth*", Combinatorial Mathematics: Proceedings of the Third International Conference, Annals of the New York Academy of Sciences 555, 1989, pp. 56-62.
- [9] B.Bollobas. "Graph Theory - An Introductory Course", Springer-Verlag, New York, Heidelberg, Berlin, 1979.
- [10] J.Bray, C.Parker, P.Rowley. "Graphs Related to Cayley Graphs and Cubic Graphs of Large *Girth*", Preprint April 1998.

- [11] G.Brinkmann, B.D.McKay, C.Saager. "The smallest cubic graphs of *girth* nine", *Combinatorics Probability and Computing*, Vol.5 (1995), pp. 1-13.
- [12] A.E.Brouwer, A.M Cohen, A.Neumaier. "Distance regular graphs", Springer-Verlag, 1989.
- [13] N.J.Calkin, P.Erdős, C.A.Tovey. "New Ramsey bounds from cyclic graphs of prime order", *SIAM Journal on Discret Mathematics*, Vol.10, No.3, pp. 381-387, August 1997.
- [14] F.Comellas, M.A.Fiol. "Using Simulated Annealing to Design Interconnection Networks". DMAT Report 05-290, January 1991.
- [15] F.Comellas, J.Gómez. "New large graphs with given degree and diameter.", *Graph Theory, Combinatorics and Algorithms*, Vol.1, Yousef Alavi and Allen Schwenk (Eds.), John Wiley & Sons, Inc.; New York (1995) pp.221-233. ISBN 0-471-30437-9. (Proc. Of the Seventh Quadrennial International Conference on the Theory and Applications of Graphs, Kalamazoo, MI, USA, June 1992.)
- [16] F.Comellas, E.Pallarès. "Optimització combinatoria i disseny de xarxes d'interconnexió", *Butlletí de la Societat Catalana de Ciències*, Vol.XIV, Núm.2, 1994.
- [17] F.Comellas. "Apunts del curs sobre Algorismes Evolutius", impartit al Departament de Matemàtica Aplicada i Telemàtica, UPC, període 1998-1999.
- [18] D.E.Comer and D.L.Stevens. "Interworking with TCP/IP". Vol.2. Design, Implementation and Internals. Ed.Prentice Hall, 1999.
- [19] M.Conder. "Small trivalent graphs of large *girth*", Department of Mathematics of the University of Auckland, Private Bag 92019, Auckland, New Zealand, June 1997.

- [20] F.Darema, S.Kirkpatrick i V.A.Norton. "Parallel algorithm for chip placement by simulated annealing". IBM J. Res. Develop., vol. 31, pp. 391-402, May 1987.
- [21] M.J.Dinneen & P.Hafner. "New results for the degree/diameter problem". Networks, 24 (1994) pp. 359-367.
- [22] G.Exoo (exoog@schneider.com). "A Simple Method for Constructing Small Cubic Graphs of *Girths* 14, 15 and 16". Electronic Journal of Combinatorics, Vol. 3, 1996.
- [23] G.Exoo. "A family of graphs and the degree/diameter problem". Preprint. Dept. of Math. and Comp. Science. Indiana State University. Communicated May, pp. 19-22, July, 1 1998
- [24] D.E.Goldberg. "Genetic Algorithms in Search, optimization, and Machine Learning", Addison-Wesley, 1989.
- [25] J.Gómez, I.Pelayo and C.Balbuena. "New large graphs with given degree and diameter six". Networks, 34 (1999), pp. 154-161.
- [26] P.R.Hafner (hafner@math.auckland.ac.nz). "Large Cayley graphs and digraphs with small degree and diameter". Report CDMTCS-005, June 1995. Wieb Bosma and Alf van der Poorten (eds) Computational Algebra and Number Theory Mathematics and Its Applications 325; Kluwer Academic Publishers Dordrecht/Boston/London (1995), pp. 291-302; ISBN 0-7923-3501-5
- [27] J.H.Holland. "Genetic Algorithms", Scientific American, v. 267, July 1992, pp. 44-91.
- [28] J.J. Hopfield i D.W. Tank. "Neural computation of decisions in optimization problems", Biol. Cybernet., vol. 52, pp. 141-152, 1985.
- [29] R.M.Karp. "Complexity of Computer Computations", R.E. Miller i J.W. Thatcher, Eds. New York, Plenum, 1972.

- [30] S.Kirkpatrick, C.D.Gelatt i M.P.Vecchi. "Optimization by Simulated Annealing". Science, vol. 220, pp. 671-680, 1983.
- [31] M. Lundy, A.Mees. "Convergence of an annealing algorithm". Mathematical Programming 34, pp. 111-124.
- [32] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller i E. Teller. "Equation of state calculations by fast computing machines". Journal of Chemical Physics, vol.21, pp. 1087, 1953.
- [33] M.O'Keefe and P.K.Wong. "A smallest graph of *girth* 10 and valency 3". Journal of Combinatorial Theory (B) 29, 1980, pp. 91-105.
- [34] R.H.J.M. Otten i L.P.P.P. van Ginneken. "The Annealing Algorithm". Kluwer Academic Publishers, Boston, 1989.
- [35] E.Pallarès, F.Comellas. "Aplicació de la recuita simulada a aplanament de grafs". Projecte Final de Carrera de E.T.S. d'Enginyeria de Telecomunicacions de Barcelona. Novembre 1992.
- [36] K.Piwakowski. "Applying Tabu Search to Determine New Ramsey Graphs", The electronic journal of combinatorics, Vol.3 (1996) #R6.
- [37] J.J.Quisquater. "Chordal rings". Communicated February 27, 1998.
- [38] F. Romeo i A. Sangiovani-Vincentelli. "A theoretical framework for simulated annealing". Algorithmica, 6, pp. 302-345, 1991.
- [39] M.Sampels (Sampels@Informatik.Uni-Oldenburg.DE) & S.Schöf. "Massively parallel architectures for parallel discrete event simulation". Proceedings of the 8th European Simulation Symposium (ESS'96), vol.2, (1996), pp. 374-378.
- [40] P.N. Strenski i S. Kirkpatrick. "Analysis of finite length annealing schedules", Algorithmica, 6, pp. 346-366, 1991.

- [41] O.Wohlmuth. "A New Dense Group Graph Discoverd by an Evolutionary Approach". Paralleles und Verteiltes Rechnen Beitrage zum 4. Workshop ueber Wissenschaftlichen Rechnen, pp. 21-27, Shaker Verlag 1996.ISBN 3-8265-1826-8.
- [42] P.K.Wong. "Cages - a survey", Journal of Graph Theory 6, 1982, pp. 1-22.

## 15. LLISTA DE FIGURES

Figura 2.1 Grafs isomorfs .....	10
Figura 2.2 Representació circular d'un graf .....	12
Figura 2.3 Representació en arbre d'un graf.....	13
Figura 4.1 Funció de cost $f(i)$ per $\delta = 2$ .....	32
Figura 7.1 Estructura d'un graf cúbic vista des d'un dels seus vèrtexs .....	55
Figura 7.2 Graf ideal per a girth senar (girth = 5).....	58
Figura 7.3 Graf ideal per a girth parell (girth = 6) .....	60
Figura 7.4 Graf circulant .....	63
Figura 7.5 (3,6)-cage d'ordre 14 .....	63
Figura 7.6 (3,8)-cage d'ordre 30 .....	64
Figura 7.7 Vèrtexs més enllà de la distància diàmetre = 2.....	77
Figura 9.1 Model de la xarxa en que es basa l'OSPF.....	108

## 16. LLISTA DE TAULES

Taula 6-1 Darreres fites per al problema ( $\Delta, D$ ) .....	48
Taula 7-1 Patró d'expansió.....	54
Taula 7-2 Nivell base.....	56
Taula 7-3 Reducció en el nombre de vèrtexs sobre els que cal avaluar la funció de cost.....	61
Taula 7-4 Reducció mitja en el nombre de vèrtexs sobre els que cal avaluar la funció de cost (reducció mitja/vèrtexs totals x 100).....	67
Taula 7-5 Exemple de la Distorsió en les distàncies (per $D=3$ ) .....	78
Taula 7-6 Nombre de vèrtexs a avaluar segons el mètode de construcció ..	81
Taula 7-7 Temperatures inicials en funció del $n^{\circ}$ de vèrtexs al grup patró....	83
Taula 8-1 Matriu de distàncies "ideal" d'algunes cages actuals. ....	90
Taula 8-2 Matriu de distàncies d'algunes cages conegudes. ....	91
Taula 8-3 % d'èxits en funció de la raó $k_1/k_2$ .....	95
Taula 8-4 Parells intentats per al girth 14 .....	97
Taula 8-5 Últimes fites obtingudes per al problema ( $\Delta, D$ ) per $D=8$ i $9$ .....	98
Taula 8-6 Parells intentats per al Diàmetre 9 .....	101

## Agraiments

Vull agrair tota l'ajuda prestada per Francesc Comellas al llarg del projecte, i l'entusiasme que ha demostrat en tot, fent de la investigació una veritable aventura.

Agrair també al Departament de Matemàtica Aplicada i Telemàtica de la UPC tot el suport rebut, a través de l'ampli equipament en computadores de que disposa el laboratori, principalment en estacions Sun, que és on s'han realitzat la majoria de les simulacions dels algorismes desenvolupats en el projecte.

I aprofito aquestes línies per donar el meu agraiment especial al professor de l'assignatura Processament del Señal, Gregorio Vázquez, gràcies al consell del qual en cert moment dels meus estudis he arribat aquí on sóc ara, a l'últim pas de la carrera.

També als meus pares, el meu agraiment per la labor que van realitzar durant tants anys en procurar formar-me en tot, i que ara culmina una etapa tant important.

I sobretot el meu reconeixement i el meu afecte de tot cor a la colla d'amics, que han demostrat en aquests anys de projecte una enorme confiança en mi i s'han ofert en tot el possible a permetre'm el poder-m'hi dedicar el màxim de temps, rellevant-me en d'altres tasques que sovint són tant importants com aquesta però sempre menys reconegudes. Gràcies perquè sense tots ells, aquest projecte no seria ara una realitat.