Universitat de Barcelona

Treball Final de Grau

GRAU DE MATEMÀTIQUES

Facultat de Matemàtiques
Universitat de Barcelona

Estudi de la Planaritat i Número de Talls d'un Graf

Adrià Màdico Ferrer

Tutor: F. Javier Soria de Diego
Departament de Matemàtica Aplicada i Anàlisi
Barcelona, primavera 2013

Índex

Agraïments

Per començar, vull agrair a les persones que han sigut peça clau en el resultat final de treball, el Javier Soria i l'Alejandro de Miquel. El primer per aconsellar-me correctament en certs aspectes fent que el projecte hagi millorat globalment i al segon per confiar-me la seva brillant idea que ha acabat en aquest resultat final.

També m'agradaria tenir unes paraules per tots companys de facultat que m'han acompanyat durant aquests quatre anys de carrera i m'han donat l'oportunitat de viure experiències molt divertides i enriquidores.

Alhora, vull referir-me als meus pares, avis i parella per tot el suport que m'han donat per seguir endavant en tot moment. Per últim, agraeixo especialment als meus avis Joan i Ramon per haver-me cuidat durant aquests 22 anys ja que sense ells res hagués sigut el mateix.

Background and Introduction

Sometimes, despite the simplicity of a question, the answer is not as easy as could be expected. This fact occurs normally in graph theory although a graph is only a set of points and edges that link one point to another. One of the unsolved problems in this area is to know how to draw a graph in such a way that it has the least possible number of crossings among its edges. To a certain point, this is normal because if the graph is really big then the question seems quite complex, but this problem is harder than expected. As an example, nowadays it is still unknown which is the minimum value for a graph of 13 vertexes all of them joined together. Moreover, the problem has been experimenting low improvements since the day that it was settled, so it is not because of a lack of interest that the solution has not been found yet. In the next paragraph, we show part of a Paul Turán experience that he had while he was working in a camp during the Second World War. This extract was in *A Note of Welcome* inside the first volume of the *Journal of Graph Theory* [13]:

> We worked near Budapest, in a brick factory. There were some kilns where the bricks were made and some open storage yards where the bricks were stored. All the kilns were connected by rail with all the storage yards. The bricks were carried on small wheeled trucks to the storage yeards. All we had to do was to put the bricks on the trucks at the kilns, push the trucks to the storage yards, and unload them there. We had a reasonable piece rate for the trucks, and the work itself was not difficult; the trouble was only at the crossings. The trucks generally jumped the rails there, and the bricks fell out of them; in short this caused a lot of trouble and loss of time which was rather precious to all of us (for reasons not to be discussed here). We were all sweating and cursing at such occasions, I too; but *nolens-volens* the idea occurred to me that this loss of time could have been minimized if the number of crossings of the rails had been minimized.

In this project, we will work on two types of graphs, the complete and the complete bipartite graph. They are two important structures in graph theory and are defined in 1.2.1. We show the proofs for each case for which the number corresponding to the minimum cuts of edges is already known. What is more, we also explain some general lower bounds achieved for the minimum. All the proofs can be found

in several original papers and our work fixes all and is a useful guide for having a general reference point.

The fact that this is not a recent problem can be noticed because the first one who wrote about this was Zarankiewicz in 1954. He used only mathematical arguments [18] like other mathematicians that in the sixties and seventies developed their own methods. In 1993, Woodall started using computational algorithms [17] and finally, Farahani in 2013 published the last paper explained in this work [2].

The last part of the project is about some algorithms we have done that generated our initial interest for this topic. Although they do not give unknown results, they are expected to open a new path that will help us to reach new objectives.

To conclude, we finally see that this is an open problem and it is difficult to determinate how the improvements will come. The lasts proofs that use the computational power and have achieved better bounds fall easily with bigger graphs because this is an NP-complete problem. In my opinion, better results will arrive when we manage to reduce the number of cases limiting the huge magnitude of the problem, otherwise, there should be extremely good improvements in the actual algorithms to offset the complexity.

Antecedents i Introducció

De vegades, per molt senzilla que sigui una pregunta, la resposta a aquesta no ens reserva la mateixa sort. Aquest fet succeeix habitualment en teoria de grafs malgrat que un graf és senzillament una sèrie de punts i arestes que els uneixen. Un dels problemes oberts en aquesta matèria és saber com hem de dibuixar un graf per aconseguir que les arestes es tallin el mínim de cops possibles. Fins a cert punt pot semblar normal que donat un graf qualsevol costi calcular-ho ja que aquest pot ser molt gran, però és que realment aquest problema és molt més difícil del que un pot pensar a priori. Per posar un exemple, actualment no es coneix quin és el mínim per un graf amb 13 vèrtexs que s'uneixen tots entre tots. A més, no ha sigut pas per falta d'interès en el problema que aquest no s'hagi pogut solucionar ja que podem trobar petits avenços des de gairebé el dia que es va proposar el problema. A continuació, mostrem parts d'unes reflexions de Paul Turán que trobem en *A Note of Welcome* del primer volum del *Journal of Graph Theory* [13] i que va fer en un camp de treball durant la Segona Guerra Mundial:

> We worked near Budapest, in a brick factory. There were some kilns where the bricks were made and some open storage yards where the bricks were stored. All the kilns were connected by rail with all the storage yards. The bricks were carried on small wheeled trucks to the storage yeards. All we had to do was to put the bricks on the trucks at the kilns, push the trucks to the storage yards, and unload them there. We had a reasonable piece rate for the trucks, and the work itself was not difficult; the trouble was only at the crossings. The trucks generally jumped the rails there, and the bricks fell out of them; in short this caused a lot of trouble and loss of time which was rather precious to all of us (for reasons not to be discussed here). We were all sweating and cursing at such occasions, I too; but *nolens-volens* the idea occurred to me that this loss of time could have been minimized if the number of crossings of the rails had been minimized.

En aquest treball ens centrem en dos tipus de grafs, els complets i els bipartits complets. Si no es coneixen les definicions es podem veure en 1.2.1. D'aquests grafs en donem les demostracions en el cas que es conegui el nombre de talls mínims entre les arestes d'aquests i, a més, també exposem algunes acotacions inferiors generals d'aquest mínim. Les demostracions es troben en múltiples articles i en

aquest projecte unim els arguments principals per poder tenir un lloc de referència general.

Com s'ha comentat amb interioritat, aquest no és un problema recent i aquest fet el denota que el primer que n'escriu al respecte és Zarankiewicz en el 1954 [18] donant arguments purament matemàtics. Més tard d'altres desenvolupen els seus propis mètodes, i és en el 1993 quan es comencen a utilitzar arguments informàtics per Woodall [17]. Finalment acabem concloent el treball amb un resultat d'aquest mateix any 2013 donat per Farahani en el seu article [2].

L'últim bloc del projecte està dedicat a alguns algoritmes propis, els quals van ser els inicialment van generar interès en la matèria. Aquests no assoleixen resultats no coneguts, però ens obren un camí per continuar desenvolupant que esperem que acabi arribant a bon port.

Com a conclusió final podem veure que aquest és un problema obert amb un futur incert. Les últimes demostracions que aprofiten el potencial de computació actual han millorat els resultats però es queden curtes ràpidament al tractar-se d'un problema NP-complet. Personalment, opino que el camí és la reducció de casos intentant acotar la magnitud del problema ja que altrament les millores han de ser exponencials per compensar la complexitat.

Objectius i motivació

Mesos abans de plantejar-me fer aquest treball, un company de facultat, Alejandro de Miquel, va comentar-me la idea de fer un programa per intentar solucionar un problema de la teoria de grafs: el nombre de talls dels grafs complets. Vam començar el projecte amb un bàsic algoritme que estava molt lluny d'aconseguir qualsevol resultat important. Tanmateix, amb el temps vam poder aplicar considerables millores i es van assolir resultats interessants.

En aquest treball espero veure com es treballa actualment en aquesta matèria i poder aplicar aquests coneixements en els meus algoritmes. Encara que sigui molt difícil, l'objectiu ideal seria assolir algun resultat que encara no es conegui o, si més no, trobar idees per poder perfeccionar els programes al màxim.

Capítol 1

Preliminars

En tot aquest primer bloc s'ha seguit com a referència a Harary [6]. Tanmateix els resultats es podem trobar en qualsevol llibre de teoria de grafs general.

1.1 Conceptes bàsics

Definició 1.1.1. *Un **graf** G és una parella formada per un conjunt de **vèrtexs**, V, i per un conjunt d'**arestes**, E. Aquestes últimes són elements que uneixen dos vèrtexs no necessàriament diferents. Notarem el graf de la següent forma $G = (V, E)$.*

Observació 1.1.2. En el treball, considerarem que el nombre de vèrtex i d'arestes és finit degut a que és el context per estudiar els problemes que volem desenvolupar.

Notacions 1.1.3. Durant tota l'exposició utilitzarem les següents notacions:

- $\nu(G)$ serà el **nombre de vèrtexs** del graf G.
- $\alpha(G)$ serà el **nombre de arestes** del graf G.
- Donat que les arestes uneixen dos vèrtexs les podem notar de la següent forma: sigui $e \in E$ i $v, w \in V$ els vèrtex que uneix, llavors $e = \overline{vw}$.

Definicions 1.1.4. *A continuació tenim un seguit de definicions:*

- *Direm que unes arestes són **múltiples** quan uneixen els mateixos vèrtexs. S'ha de tenir en compte que si notem les arestes com a unió se dos vèrtex i el graf en té de múltiples llavors tindrem arestes diferents notades igual.*
- *Direm que una aresta és un **bucle** quan uneix un vèrtex amb si mateix.*
- *Si un graf no té arestes múltiples ni bucles direm que el graf és **simple**.*
- *Sigui $G = (V, E)$ un graf. $G' = (V', E')$ serà un **subgraf** de G si G' té estructura de graf, $V' \subseteq V$ i $E' \subseteq E$.*

- *Dos grafs $G = (V, E)$, $G' = (V', E')$ són **isomorfs** si existeix una bijecció f de V a V' tal que es conserven les arestes, i.e., per tot $e \in E$ on e uneix els vèrtexs $v, w \in V$ llavors existeix $e' \in E'$ tal que uneix $f(v)$ i $f(w)$.*

- *Donats dos vèrtexs direm que són **adjacents** si existeix una aresta que els uneix, a més, direm que l'aresta és **incident** en els vèrtex i recíprocament. També d'una forma semblant direm que dos arestes són adjacents si comparteixen com a mínim un vèrtex.*

- *Donat un vèrtex $v \in V$, en definim el seu **grau**, $\alpha(v)$, com la suma del nombre d'arestes incidents a ell més els bucles. D'aquesta manera és com si consideréssim els bucles doblement incidents.*

- *Considerem una successió de n vèrtexs adjacents, $v_1, ..., v_{n+1}$, definim **camí** de longitud n de v_1 a v_{n+1} com la successió d'arestes que uneixen els vèrtexs adjacents, $\overline{v_1v_2}, \overline{v_nv_{n+1}}$. Direm que aquest és **simple** si totes les arestes són diferents i **elemental** si també ho són els vèrtexs.*

- *Direm que un graf és **connex** si existeix un camí elemental que uneix cada parella de vèrtexs. És fàcil veure que si existeix un camí qualsevol, n'existeix un d'elemental.*

Exemple 1.1.5. En aquest exemple veiem les definicions anteriors.

Per començar podem observar els grafs de la Figura 1.1 i veure que $\nu(G) = 4, \alpha(G) = 6, \nu(H) = \nu(I) = 4$ i $\alpha(H) = \alpha(I) = 2$. A més, en G les arestes b i c són múltiples i a és un bucle, fet que fa que G no sigui simple. En canvi, H i I ho són. H i I també són grafs isomorfs encara que els veiem diferent dibuixats, i ambdós són subgrafs de G.

Per mostrar un exemple d'adjacència i d'incidència podem dir que en el graf G els vèrtexs v_1 i v_2 són adjacents i que f és incident a ells.

Com s'ha definit, podem calcular el grau d'un vèrtex. El grau del vèrtex v_1 en el graf G, és 5 donat que té 4 arestes incidents i una d'elles és un bucle. Un camí en el mateix graf seria el que comença en el vèrtex v_1, passa per les arestes bcf i acaba a v_4, el qual és simple però no elemental. Per últim queda dir que G és connex i H i I no ho són.

1.2 Tipus de grafs

Donada la poca restricció que hi ha en la definició de grafs se'n poden construir molts no isomorfs. Tanmateix, n'hi ha que tenen unes peculiaritats especials i aquests els definirem en aquesta secció.

Definicions 1.2.1.

- *Diem que un graf és un **cicle** de n vèrtexs si cada vèrtex té grau 2 i és connex. El notem de la següent forma: C_n.*

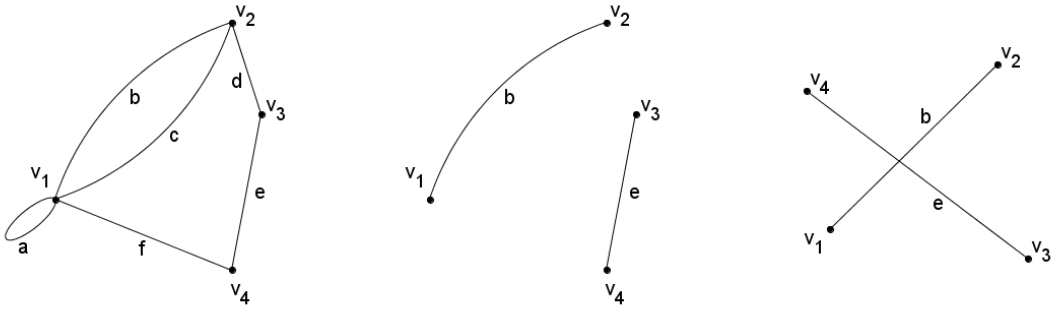

Figura 1.1: Graf G, graf H i graf I, respectivament.

- *Diem que un graf és un* ***graf complet*** *de n vèrtexs si cada un dels vèrtexs és adjacent a tota la resta mitjançant una sola aresta. El notem de la següent forma: K_n.*
- *Diem que un graf $G = (V, E)$ és* ***bipartit*** *m, n si els seus vèrtexs es poden agrupar en dos subconjunts V_1 i V_2 amb m vèrtexs i n vèrtexs respectivament tals que:*
  - $V_1 \cup V_2 = V$.
  - $V_1 \cap V_2 = \emptyset$.
  - $\forall a, b \in V_1, \overline{ab} \notin E$ *i* $\forall c, d \in V_2, \overline{cd} \notin E$.

  A més diem que és un graf ***bipartit complet*** *m, n si una sola aresta uneix cada un dels vèrtexs de V_1 amb tota la resta de V_2, o viceversa i el notarem $K_{m,n}$.*

Notació 1.2.2. Donat un graf bipartit complet $K_{m,n}$ als vèrtexs pertanyen al conjunt dels m vèrtexs els notarem $U = \{u_i : i \in \{1, \ldots, m\}\}$ i als que pertanyen als n vèrtexs $W = \{w_j : j \in \{1, \ldots, n\}\}$.

Exemple 1.2.3. En la primera columna de la Figura 1.2 trobem el C_4 i el C_5, en la segona el K_4 i el K_5 i en la tercera el $K_{2,3}$ i el $K_{3,4}$.

1.3 Representacions de grafs

La manera habitual de representar un graf és sobre el pla. Tanmateix, no ha de perquè ser així i podem considerar els grafs en l'espai, en un tor, en una esfera... En el treball ens centrarem en els grafs sobre el pla i en les maneres de col·locar-los per intentar que les arestes no es tallin unes amb les altres, o que ho facin el mínim possible.

Per començar són necessaris una sèrie de conceptes els quals podem trobar tant en Pan i Richter [11] com en Guy [5] i que a continuació mostrem.

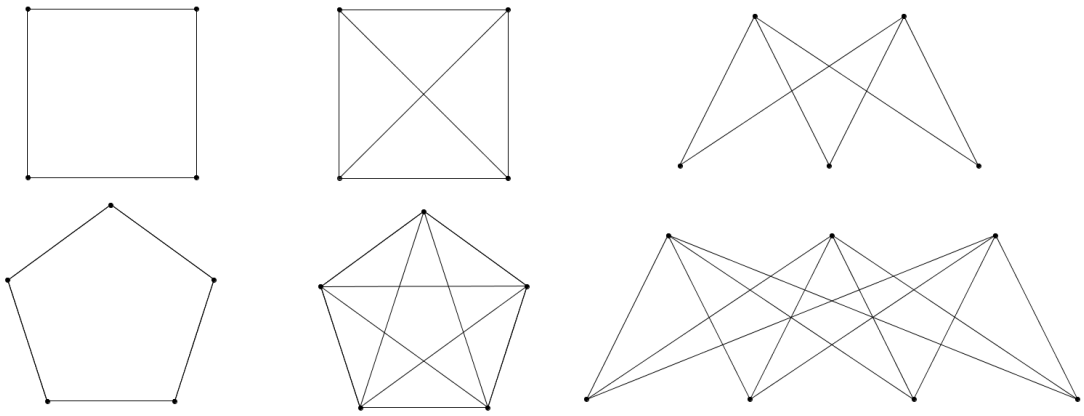

Figura 1.2: Exemples de grafs.

Definicions 1.3.1.

- *Una* ***representació d'un graf*** *consisteix en una sèrie de punts i corbes sobre el pla, un per cada vèrtex i una corba per cada aresta. Les corbes van de vèrtex a vèrtex i no en contenen cap en el seu interior. A més, si un punt no és un vèrtex no pot estar en tres arestes. En general, a una representació en concreta d'un graf G li direm R_G o, simplement R en el cas que estigui clar a quin graf ens referim.*

- *Un* ***tall*** *d'una representació és un punt del pla que pertany a dues arestes i no és un vèrtex. El nombre de talls total d'una representació R el notem com $\mathrm{cr}(R)$ i el conjunt de talls com T_R.*

- *Donat un graf denotem el* ***nombre de talls*** *del graf per $\mathrm{cr}(G)$ al mínim $\mathrm{cr}(R_G)$ de totes les representacions R_G de G.*

- *Direm que un graf G és* ***planar*** *quan es pugui representar en un pla sense que les seves arestes es tallin, és a dir si $\mathrm{cr}(G) = 0$.*

- *Les representacions que assoleixen el nombre de talls del graf són representacions* ***òptimes****.*

- *Direm que una representació és* ***satisfactòria*** *si cada parella d'arestes no múltiples comparteixen com a molt un punt en comú, incloent vèrtexs i talls; i les múltiples només comparteixen els vèrtexs.*

- *Sigui R una representació d'un graf G. Definim el conjunt X dels* ***extrems*** *de R com el que conté els vèrtexs i els talls de R i el conjunt S de les* ***subarestes*** *de R com el que conté els subsegments d'arestes de E que no continguin talls i que uneixin dos extrems.*

- *Siguin R i R' dues representacions d'un mateix graf $G = (V, E)$ amb els seus respectius talls T i T', extrems X i X', i subsegments S i S'. R i R' són* ***isomorfes*** *si existeix una bijecció f de X a X' tal que:*
  - *Si $v \in V$ llavors $f(v) \in V$.*
  - *Si $t \in T$ llavors $f(t) \in T'$.*
  - *Per tot $s \in S$ on s uneix els dos extrems $x_1, x_2 \in X$ llavors existeix $s' \in S'$ tal que uneix $f(x_1)$ i $f(x_2)$*

Exemple 1.3.2. El grafs complets i els seus números de tall són part important del treball, per això mostrem uns exemples que ens il·lustren les anteriors definicions.

Tenim el graf complet K_4 que com podem observar en la Figura 1.3 té un tall i en la segona cap. En aquest cas, com que no pot tenir menys de 0 talls és clar que la segona representació és òptima, i per tant, $\mathrm{cr}(K_4) = 0$.

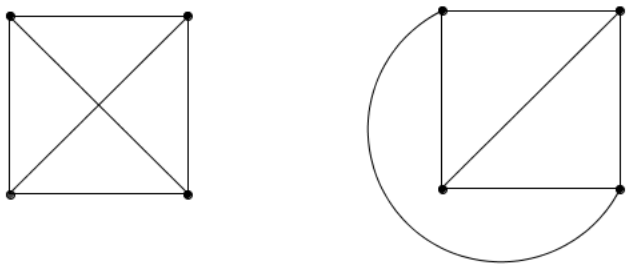

Figura 1.3: K_4.

Del graf complet K_5 en tenim dues representacions en la Figura 1.4. La primera té cinc talls i en la segona un. Contràriament al cas del K_4 aquí no podem dir que la segona és una representació òptima d'una manera directa ja que no sabem si n'hi ha alguna amb 0 talls. En el Teorema 2.2.1 confirmarem que sí que és realment òptima.

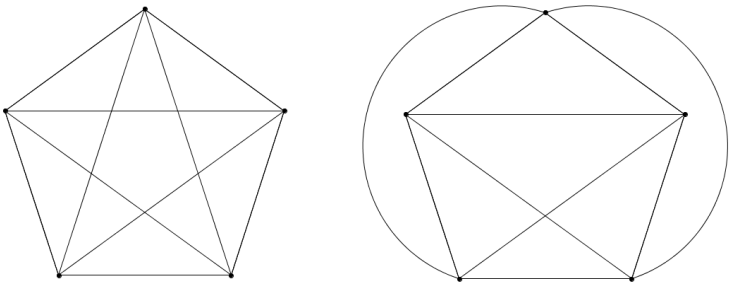

Figura 1.4: K_5.

A continuació, mostrem uns exemples per explicar el significat d'una representació isomorfa. En la Figura 1.5 tenim tres representacions diferents d'un graf. Si

aquestes tinguessin un nombre diferent de talls ja podríem dir directament que no són isomòrfiques, però no és el cas. És més, les dues últimes sí que són isomòrfiques, i en canvi, la primera no ho és amb elles. Per veure aquests fets cal identificat els vèrtexs, talls, i les subarestes tal i com està fet en la imatge.

Per veure que les dues últimes són isomòrfiques podem crear una aplicació que ens enviï els vèrtexs v_i a v_i i t a t i ja ho tenim. Per veure que la primera i la segona no ho són ens podem fixar en què hauríem d'enviar el v_1 i el v_5 del primer graf al v_5 segon i això faria que el que creéssim ja no fos un isomorfisme.

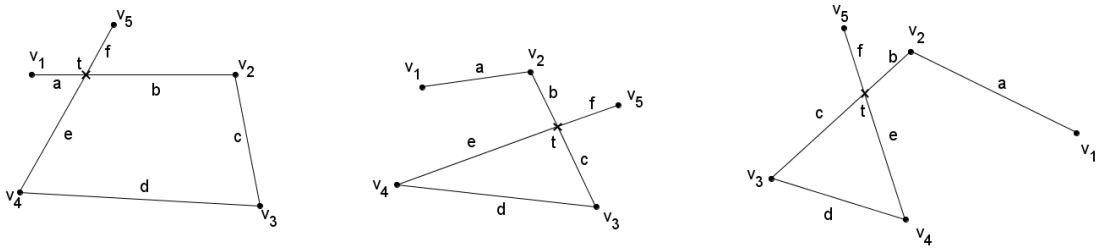

Figura 1.5: Tres representacions diferents d'un graf.

En la Figura 1.6 podem veure un exemple de representació no satisfactòria.

Notació 1.3.3. En general, tant si ens volem referir al nombre de talls d'un graf, d'un subgraf, d'una arestes o segments d'aquestes últimes es hi referirem com $\mathrm{cr}(\cdot)$.

Observació 1.3.4. La motivació de la definició de representació satisfactòria sorgeix de la necessitat de no considerar una sèrie de grafs que fàcilment s'observa que no són òptims. La següent Proposició ens ho mostra.

Proposició 1.3.5. *Si una representació és òptima és satisfactòria.*

Demostració. Aquesta demostració consisteix en construir una representació amb menys talls d'una que no sigui satisfactòria, doncs considerem una representació no satisfactòria R. Aquesta representació tindrà com a mínim un parell d'arestes que comparteixen dos punts o més en comú dels quals un no és vèrtex. A les dues arestes les diem e i f, els dos punts en comú A i B i els segments tancats que van de A a B que pertanyen a les arestes v_1 i v_2, e' a f'.

Ara generarem una representació R' diferent a R amb menys nombres de talls. Per fer-ho observem el nombre de talls dels segments e' i f' i el que en tingui més, suposem que és f', el fem passar al costat de l'altre com es mostra en la imatge prèvia. En el cas que tinguin els mateixos talls és irrellevant moure una aresta o l'altre. Distingim dos casos:

- Si A i B no són vèrtexs, els talls A i B no hi seran, per tant:

$$\mathrm{cr}(R') = \mathrm{cr}(R) + \mathrm{cr}(e') - \mathrm{cr}(f') - 2 \leq \mathrm{cr}(R) - 2 < \mathrm{cr}(R).$$

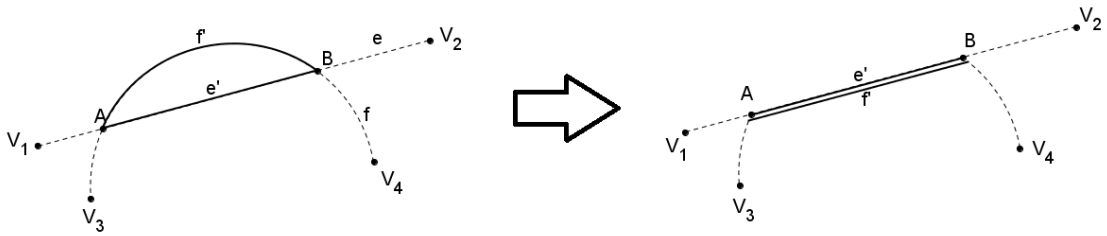

Figura 1.6: Exemple en el cas que cap dels punts sigui vèrtex.

- Si A o B és vèrtex, el tall A o B no hi serà, per tant:

$$\mathrm{cr}(R') = \mathrm{cr}(R) + \mathrm{cr}(e') - \mathrm{cr}(f') - 1 \leq \mathrm{cr}(R) - 1 < \mathrm{cr}(R).$$

En els dos casos un argument similar ens porta a que nombre de talls de R' serà menor. Per tant, R no és òptim. □

Observació 1.3.6. En la Figura 1.4 o 1.3 es pot veure que no el recíproc no és cert.

Observació 1.3.7. Donat que coneixem la Proposició 1.3.5, a partir d'ara per buscar grafs òptims considerarem únicament els que són satisfactoris. És més, mitjançant la repetició de l'anterior construcció podem transformar les representacions no satisfactòries en altres que sí que ho siguin.

Capítol 2

Primers grafs no planars: K_5 i $K_{3,3}$

Saber en general el nombre de talls d'un graf no és una qüestió resolta, és més, aquest problema és considerat NP-complet, fet que s'indica en l'article [3] i que denota la seva gran complexitat. A priori, no trobem millors resultats si ens restringim als grafs complets i bipartits, però degut a la seves estructures en tenim uns resultats més amplis.

En els preliminars 1.3.2 ja hem vist que $\mathrm{cr}(K_4) = 0$. En aquest capítol ens centrarem a en veure que els grafs K_5 i $K_{3,3}$ no són planars. Pels dos casos haurem de definir uns conceptes que ens serviran per la resta de casos i utilitzarem uns arguments semblants i àmpliament coneguts com el Teorema de la característica d'Euler.

2.1 Planaritat i Teorema de la Característica d'Euler

Definició 2.1.1. *Donat un graf planar G i una representació sense cap tall $R \subset \mathbb{R}^2$ definim les **cares** del graf com cada una de les components connexes de $\mathbb{R}^2 \setminus R$. Ho notem com $\zeta(G)$.*

Observació 2.1.2. Un graf amb arestes múltiples i bucles serà planar si el seu subgraf més gran simple és planar. Això es deu a que podem considerar que les arestes múltiples es poden fer passar totes pel mateix lloc i els bucles els podem retraure fins el seu vèrtex incident.

Definició 2.1.3. *Donat un graf planar simple diem que és **maximal** si al afegir-li una aresta més deixa de ser planar o simple.*

Observació 2.1.4. És clar que dir que un graf és maximal és equivalent a dir que totes les seves cares són triangles ja que en cas contrari, o bé es podrien traçar arestes entre vèrtexs no adjacents del polígon que no fos un triangle o el graf no seria simple.

Teorema de la Característica d'Euler 2.1.5.
Donat un graf G connex i planar es compleix la següent igualtat:

$$\nu(G) - \alpha(G) + \zeta(G) = 2.$$

Demostració. Per demostrar el resultat farem inducció sobre el número de vèrtexs.

Si el graf té un vèrtex les úniques arestes que pot tenir són bucles i en la Observació 2.1.4 ja hem vist que això és el mateix que considerar el graf de un sol vèrtex sense arestes. En aquest cas veiem que la equació és certa ja que:

$$1 - 0 + 1 = 2.$$

Suposem que el resultat és cert pels grafs connexes i planars de n vèrtexs. Sigui ara H un graf de $n+1$ vèrtexs connex i planar. Si li trèiem un vèrtexs i les seves arestes incidents complirà la fórmula per hipòtesi d'inducció, volem veure que H també la compleix. Suposem que el vèrtex que hem tret tenia m arestes, per tant, mitjançant un argument de construcció sabem que haurem tret $m-1$ cares. Veiem que H compleix la formula:

$$\begin{aligned}\nu(H) - \alpha(H) + \zeta(H) &= (\nu(G)+1) - (\alpha(G)+m) + (\zeta(G)+m-1) \\ &= \nu(G) - \alpha(G) + \zeta(G) = 2.\end{aligned}$$

□

Corol·lari 2.1.6. *Si G és un graf maximal llavors $\alpha(G) = 3\nu(G) - 6$. A més, si un graf és planar i simple tenim la següent cota màxima del nombre d'arestes en funció del nombre de vèrtexs: $3\nu(G) - 6$.*

Demostració. Com que G és maximal, totes les cares són triangles, per tant, $3\zeta(G) = 2\alpha(G)$. Aplicant això a la formula del Teorema d'Euler tenim:

$$\nu(G) - \alpha(G) + 2\alpha(G)/3 = 2 \Leftrightarrow \alpha(G) = 3\nu(G) - 6.$$

Ara ja sabem que tots els grafs maximals G compleixen que $\alpha(G) = 3\nu(G) - 6$. Com que si en algun graf maximal li afegíssim una aresta aquest deixaria de ser planar o simple el nombre màxim d'arestes d'un graf planar simple és $3\nu(G) - 6$. □

2.2 Nombre de talls de K_5 i $K_{3,3}$ i Teorema de Kuratowski

Teorema 2.2.1. *K_5 no és planar, i en conseqüència* $\mathrm{cr}(K_5) = 1$.

Demostració. Si K_5 fos planar llavors hauria de tenir menys de $3\nu(K_5) - 6 = 9$ arestes i en té 10.

Com a conclusió directa en deduïm que $\mathrm{cr}(K_5) = 1$ ja que coneixem una representació 1.4 que assoleix aquesta cota. □

Lema 2.2.2. *Si G és un graf on totes les cares tenen quatre costats llavors $\alpha(G) = 2\nu(G) - 4$.*

Demostració. Com que totes les cares de G són quadrilàteres $4\zeta(G) = 2\alpha(G)$. Aplicant això a la formula del Teorema d'Euler tenim:

$$\nu(G) - \alpha(G) + \alpha(G)/2 = 2 \Leftrightarrow \alpha(G) = 2\nu(G) - 4.$$

□

Teorema 2.2.3. *$K_{3,3}$ no és planar, i en conseqüència* $\mathrm{cr}(K_{3,3}) = 1$.

Demostració. Com que totes les cares del $K_{3,3}$ són quadrilàters aquest hauria de tenir $2\nu(K_{3,3}) - 4 = 8$ arestes i en té 9.

Com a conclusió directa en deduïm que $\mathrm{cr}(K_{3,3}) = 1$ ja que coneixem una representació que assoleix aquesta cota que trobem en la Figura 2.1.

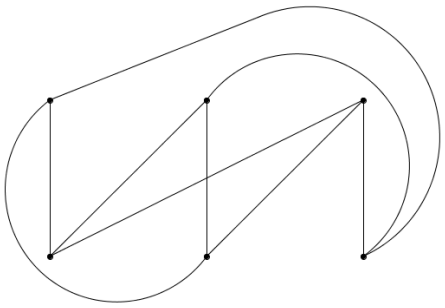

Figura 2.1: Representació satisfactòria de $K_{3,3}$.

□

Observació 2.2.4. La no planaritat dels grafs K_5 i $K_{3,3}$ no és únicament rellevant pel propi fet. Realment, està demostrat que cada tall d'un graf G qualsevol està inclòs en un subgraf K_5 o $K_{3,3}$ de G. Això ens ho mostra el següent teorema, el qual no demostrarem.

Teorema de Kuratowski 2.2.5. *Un graf és planar si i només si no conté cap K_5 ni $K_{3,3}$.*

Capítol 3

Nombres de talls dels grafs complets

Com ja s'ha comentat amb anterioritat en el Capítol 2, saber el nombre de talls d'un graf complet és un problema obert. Tanmateix, s'ha conjecturat que $\mathrm{cr}(K_n)$ és igual a

$$\mathrm{Z}(n) = \frac{1}{4}\left\lfloor\frac{n}{2}\right\rfloor\left\lfloor\frac{n-1}{2}\right\rfloor\left\lfloor\frac{n-2}{2}\right\rfloor\left\lfloor\frac{n-3}{2}\right\rfloor.$$

Veiem en una taula els valors de Z:

Vèrtexs	4	5	6	7	8	9	10	11	12	13
Z	0	1	3	9	18	36	60	100	150	225
Nombre de representacions no isomòrfiques	1	1	1	5	3	3080	5679	-	-	-

Referent a aquest tema en trobem una breu explicació en el lloc web de Wolfram, [15] on s'explica que aquesta conjectura ha estat provada per $n \leq 12$, i que tanmateix, no es coneix si és certa pels grafs complets amb més de 12 talls. Donaren tots els arguments, els quals es basen principalment en acotacions o en algoritmes, per demostrar els casos coneguts i, a més, mostrarem com a mínim una representació satisfactòria de cada un.

Per un altra banda, sabem és que hi ha cotes generals pel nombre de talls dels grafs complets. Aquests resultats els trobem com a combinació dels articles [9] i [12]. A partir d'ells, tal i com s'indica en [11] es pot provar la següent cota:

$$0,8594\,\mathrm{Z}(n) \leq \mathrm{cr}(K_n) \leq \mathrm{Z}(n).$$

Aquesta desigualtat és obtingut mitjançant computació i queda fora l'abast del projecte, tanmateix, es donarà una prova d'una cota menys ambiciosa a partit de resultats més senzills en el Teorema 3.7.1.

Per a explicar els coneixements actuals en aquesta àrea prendrem com a referència a Pan i Richter [11].

3.1 Representacions que assoleixen Z(n)

Teorema 3.1.1. *Per tot* $n \geq 3, \mathrm{cr}(K_n) \leq \mathrm{Z}(n)$.

Demostració. Donarem la construcció feta per Blazek i Koman que es pot trobar en l'article [1] i, a continuació, una manera diferent alternativa de construir els mateixos grafs feta per nosaltres.

Construcció 1

Per començar, considerem un polígon regular de n costats. La construcció consisteix en dibuixar la resta de vèrtexs dins el polígon o fora sense travessar la vora del polígon. Tanmateix, per facilitar la demostració, podem considerar que els vèrtexs que queden a l'exterior es dibuixen a l'interior d'un altre polígon igual. Això no genera un problema ja que hi ha un homeomorfisme entre tot compacte de $\mathbb{R}^2$ i la clausura de $\mathbb{R}^2$ menys l'interior del compacte, per tant, hi ha funció bijectiva que ens transportarà els vèrtex de l'interior a l'exterior.

Definim com han d'anar els vèrtexs en un gràfic, G^1, i en l'altre, G^2, hi quedaran la resta. Inicialment, numerem els vèrtexs de la forma que es mostra en la figura 3.1. A continuació afegim a G^1 les arestes que compleixin alguna de les següents condicions:

- Les rectes paral·leles a les arestes $\overline{v_i v_{i+1}}$ on $i = 1, 2, \ldots, \lfloor \frac{1}{4}(n+3) \rfloor$ i que vagin de vèrtex a vèrtex.
- Les rectes paral·leles al segment $\overline{v_{i-1} v_{i+1}}$ on $i = 1, 2, \ldots, \lfloor \frac{1}{4}(n+1) \rfloor$ i que vagin de vèrtex a vèrtex.

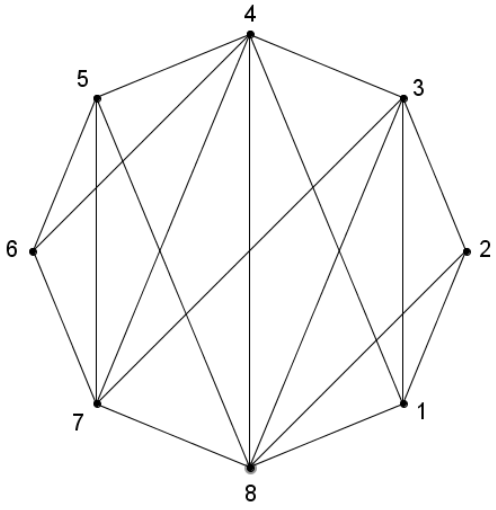

Figura 3.1: Representació del G^1 en el cas del K_8.

En la demostració de Blazek i Koman s'utilitzen certs conceptes tècnics per calcular el nombre de talls a cada part del graf i aquests no estan totalment detallats. Tanmateix, es pot veure que el nombre de talls en el gràfic G^1 és de:

- $2^{-7}n(n-2)^2(n-4)$ per $n \equiv 0 \pmod 4$
- $3^{-1}2^{-7}(n^2-1)(n-3)(3n-7)$ per $n \equiv \pm 1 \pmod 4$
- $2^{-7}(n-2)(n^3-6n^2+8n+16)$ per $n \equiv 2 \pmod 4$

I en el G^2 és de:

- $2^{-7}n(n-2)^2(n-4)$ per $n \equiv 0 \pmod 4$
- $3^{-1}2^{-7}(n-1)(n-3)(n-5)(3n-5)$ per $n \equiv \pm 1 \pmod 4$
- $2^{-7}(n-2)(n^3-6n^2+8n-16)$ per $n \equiv 2 \pmod 4$

Per tot valor de n superior o igual a 3 la suma d'aquests valors dóna la conjectura, per tant, $n \geq 3, \mathrm{cr}(K_n) \leq \mathrm{Z}(n)$.

Construcció 2

Com s'ha comentat a l'inici aquests mateixos grafs es poden generar d'un altre manera. En aquest cas, en podem veure algun exemple en les Figures 3.5, 3.6, 3.7 i 3.8. Com es pot observar, es començar a generar els grafs a partir d'una línia horitzontal que conté els vèrtexs de forma ordenada, de forma que el primer sigui el v_1 i l'últim el v_n. Es fan passat els vèrtexs per dalt o per sota aquesta en funció de la la suma dels índexs dels vèrtexs origen i final, que li direm $S_{o,f}$. Passarà per dalt en els casos en que:

- $S_{o,f} < \lfloor \frac{n}{2} \rfloor + 4$
- $n+3 < S_{o,f} < \lfloor \frac{3n}{2} \rfloor + 4$

La clau és evitar que aquestes arestes es tallin tret que no hi hagi més remei. O sigui dues arestes $\overline{v_o v_f}$ i $\overline{v_{o'} v_{f'}}$ es tallaran si i només si $o < o' < f < f'$ o $o' < o < f' < f$ i, a més, comparteixen costat. Calculant es pot veure que el nombre de talls en la part superior és de:

- $2^{-7}n(n-2)^2(n-4)$ per $n \equiv 0 \pmod 4$
- $3^{-1}2^{-7}(n-1)(n-3)(n-5)(3n-5)$ per $n \equiv \pm 1 \pmod 4$
- $2^{-7}(n-2)(n^3-6n^2+8n+16)$ per $n \equiv 2 \pmod 4$

I en la part inferior és de:

- $2^{-7}n(n-2)^2(n-4)$ per $n \equiv 0 \pmod 4$

- $3^{-1}2^{-7}(n^2-1)(n-3)(3n-7)$ per $n \equiv \pm 1 \pmod 4$
- $2^{-7}(n-2)(n^3-6n^2+8n-16)$ per $n \equiv 2 \pmod 4$

Com en l'altre construcció tenim que la suma d'ambdós ens dóna el resultat desitjat.

Aquesta última construcció no està comprovada per tot n, però tanmateix tenim un Algoritme(citar codi) que fins a $n = 250$ es compleixen els fets indicats. □

3.2 Paritat del nombre de talls

Definició 3.2.1. *Direm que un* ***tall és incident a un vèrtex*** *si una de les arestes que el formen és incident al vèrtex. Al nombre de talls incidents a un vèrtex v n'hi direm la seva* ***responsabilitat*** *i es notarà com r_v o r_i en el cas que els vèrtexs estiguin indexats.*

Observació 3.2.2. Per facilitar les següents demostracions és útil observar que cada tall d'un graf satisfactori està en dues arestes que no comparteixen cap vèrtex, per tant, cada tall és incident a quatre vèrtexs diferents.

Lema 3.2.3. *Tota representació satisfactòria del K_5 té un nombre senar de talls.*

Demostració. Suposem que tenim una representació R_5 amb un nombre parell de talls i arribarem a contradicció. Considerem la suma del nombre de talls incidents a cada vèrtex. Com que cada tall el comptem 4 cops, tenim la següent igualtat $4\,\mathrm{cr}(R_5) = \sum_{v \in V} r_v$, de la qual se'n dedueix que hi ha un vèrtex v' tal que $r_{v'}$ és parell. Si eliminem aquest vèrtex i les seves arestes adjacents de R_5 en resultarà una representació R_4 d'un K_4, la qual tindrà un nombre parell de talls ja que R_5 en tenia un nombre parell i li traiem un vèrtex amb responsabilitat parell. Construint totes les representacions satisfactòries d'un K_4 veiem que n'hi ha dues no isomòrfiques, una amb 1 tall i l'altre amb 0, per tant, com que el nombre de talls ha de ser parell sabem que $\mathrm{cr}(R_4) = 0$.

Un com tenim R_4, per reconstruir el R_5 hem d'afegir el vèrtex v'. Suposem que l'afegim a una cara qualsevol de R_4, C, per construcció es pot veure que les arestes que van dels vèrtexs que formen C a v' tindran un nombre parell del talls i l'altre, que anirà del vèrtex que no forma part de C a v', en tindrà un nombre senar. Aquests fets es donaran perquè la representació és satisfactòria. Per tant, $\mathrm{cr}(R_5)$ serà un nombre senar, contràriament al que havíem suposat. □

Teorema 3.2.4. *Tota representació satisfactòria d'un graf complet amb un nombre senar de costats $n > 4$, té la mateixa paritat que $\binom{n}{5}$.*

Demostració. El Lema 3.2.3 ens dóna el resultat per $n = 5$, el qual utilitzarem en aquesta demostració.

Considerem R una representació d'un K_n on $n > 4$ i senar. Dins R tenim $\binom{n}{5}$ diferents d'escollir de 5 vèrtexs dels n que té. Pel Lema 3.2.3 sabem que els subgrafs

generats pels 5 vèrtexs tindran un nombre senar de vèrtexs. La suma d'aquests, $\sum$, tindrà la mateixa paritat que $\binom{n}{5}$ ja que tots els nombres que sumem són senars. Cada tall apareixerà en els subgrafs que tinguin els 4 vèrtexs que li són adjacents més un cinquè diferent dins de les $n-4$ possibilitats que queden. Per tant, cada tall apareix repetit $n-4$ vegades i sorgeix la igualtat $(n-4)\,\mathrm{cr}(R) = \sum$. Com que $(n-4)$ és senar $\mathrm{cr}(R)$ tindrà la mateixa paritat que $\sum$ i en conseqüència que $\binom{n}{5}$. □

Observació 3.2.5. Tant la demostració del Lema 3.2.3 com la del Teorema 3.2.4 es troben en l'article [10] de McQuillan i Richter. La demostració original és de Kleitman [8], però és relativament confosa i per aquest motiu donem l'altre.

3.3 Acotacions de nombre de talls a partir d'anteriors i $\mathrm{cr}(K_6)$

Lema 3.3.1. *Per $n \geq 5$ es compleix la següent desigualtat:*

$$\mathrm{cr}(K_n) \geq \frac{n \cdot \mathrm{cr}(K_{n-1})}{n-4}.$$

Demostració. Si s'elimina un vèrtex d'una representació òptima de K_n i les seves arestes adjacents es genera una representació de K_{n-1}. Si això ho fem per separat amb cada vèrtex tenim n representacions de K_{n-1}, denotades com R_i. La Observació 3.2.2 ens permet veure que cada tall no hi serà en 4 representacions i sí que hi serà en $n-4$. D'això en deduïm:

$$\mathrm{cr}(K_n) = \frac{\sum_{i=1}^{n} \mathrm{cr}(R_i)}{n-4} \geq \frac{n \cdot \mathrm{cr}(K_{n-1})}{n-4}.$$

□

Teorema 3.3.2. *Si $\mathrm{cr}(K_n) = \mathrm{Z}(n)$ per un nombre n senar llavors $\mathrm{cr}(K_{n+1}) = \mathrm{Z}(n+1)$.*

Demostració. Com que $n+1$ és parell tenim:

$$\mathrm{Z}(n+1) = \frac{1}{4} \cdot \frac{n+1}{2} \cdot \frac{n-1}{2} \cdot \frac{n-1}{2} \cdot \frac{n-3}{2} = \frac{(n+1)\cdot(n-1)^2\cdot(n-3)}{4^3}.$$

Alhora com que n és senar tenim:

$$\mathrm{Z}(n) = \frac{1}{4} \cdot \frac{n-1}{2} \cdot \frac{n-1}{2} \cdot \frac{n-3}{2} \cdot \frac{n-3}{2} = \frac{(n-1)^2\cdot(n-3)^2}{4^3}.$$

Com a conseqüència tenim: $\mathrm{Z}(n+1) = \frac{n+1}{n-3} \cdot \mathrm{Z}(n)$ i acabem de demostrar en el Lema 3.3.1 que $\mathrm{cr}(K_{n+1}) \geq \frac{n+1}{n-3} \cdot \mathrm{cr}(K_n)$, per tant, si $\mathrm{Z}(n) = \mathrm{cr}(K_n)$ tenim que $\mathrm{cr}(K_{n+1}) \geq \mathrm{Z}(n+1)$. De 3.1.1 sabem que existeixen grafs que assoleixen la cota, llavors obtenim que $\mathrm{cr}(K_{n+1}) = \mathrm{Z}(n+1)$. □

Corol·lari 3.3.3. *Com que coneixem que $\mathrm{cr}(K_5) = 1$ immediatament podem dir que $\mathrm{cr}(K_6) = 3$.*

3.4 Nombre de talls de K_7 i K_8

Lema 3.4.1. *Sigui G un graf de n vèrtexs. Qualsevol representació R òptima d'aquest ha de contenir un vèrtex v amb $r_v \geq \lceil 4\,\mathrm{cr}(R)/n \rceil$.*

Demostració. En cas contrari si per tot vèrtex v de G $r_v < \lceil 4\,\mathrm{cr}(R)/n \rceil$ la suma de tots ells, $4\,\mathrm{cr}(R)$, seria més petita que $(\lceil 4\,\mathrm{cr}(R)/n \rceil) * n \leq 4\,\mathrm{cr}(r)$ i això és contradictori. □

Proposició 3.4.2. *Per cada $n \leq 6$ es compleix que cada representació òptima de K_n conté una representació òptima d'un K_{n-1}.*

Demostració. Pels casos en que $n < 5$ tots els grafs òptims són planars, per tant, és clar que es complirà.

Pels casos $n \in \{5, 6\}$ com que ja coneixem que $\mathrm{cr}(K_n) = \mathrm{Z}(n)$ i podem considerar que tindran un vèrtex amb responsabilitat superior o igual a $\lceil 4\,\mathrm{cr}(K_n)/n \rceil$. Resulta que $\lceil 4\,\mathrm{cr}(K_n)/n \rceil = \mathrm{cr}(K_n) - \mathrm{cr}(K_{n-1})$ i, per tant, podríem treure aquell vèrtex al K_n i trobar-nos un K_{n-1} òptim. □

Observació 3.4.3. Ara ja sabem que podem construirem totes les representacions òptimes d'aquests K_n a partir d'afegir un vèrtex i les arestes necessàries a les representacions òptimes d'un K_{n-1}.

Al construir K_5 a partir del K_4 ens trobem que és suficient considerar l'única representació òptima d'aquest llevat d'isomorfia que la tenim en la Figura 1.3. Afegint un vèrtex i les arestes corresponents a cada una de les cares d'aquesta figura veiem que l'única representació òptima de K_5 llevat d'isomorfia és la que tenim en la Figura 1.4.

Per construir el K_6 a partir del K_5, fem una construcció semblant a l'anterior i també acabem trobant una única representació òptima llevat d'isomorfisme Figura 3.2.

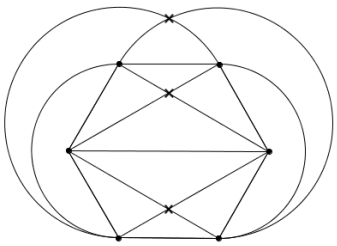

Figura 3.2: Única representació òptima de K_6 llevat d'isomorfisme.

Teorema 3.4.4. *Tota representació òptima d'un K_7 conté un K_6 i $\mathrm{cr}(K_7) = \mathrm{Z}(7) = 9$. A més, només tenim 5 representacions no isomòrfiques de K_7 amb $\mathrm{cr}(K_7) = 9$.*

Demostració. Considerem una representació R de K_7. Pel Teorema 3.2.4 sabem que $\mathrm{cr}(R)$ serà senar. A més, tenim que existeix una representació amb $\mathrm{Z}(7) = 9$ talls, per tant, $\mathrm{cr}(R)$ només pot ser igual a $3, 5, 7$ o 9. Veiem per quin motiu els casos $\mathrm{cr}(R) = 3$ i $\mathrm{cr}(R) = 5$ no són possibles:

- Si $\mathrm{cr}(R) = 3$ podrem crear una representació de K_6 amb menys de 3 talls traient a R un vèrtex amb responsabilitat diferent de 0 i els seves arestes incidents a R.
- Si $\mathrm{cr}(R) = 5$ pel Lema 3.4.1 hi haurà un vèrtex v tal que $r_v \geq \lceil 4\,\mathrm{cr}(R)/7 \rceil = 3$. En aquest cas, si li traiem aquest vèrtex i les arestes incidents a R ens quedem amb una representació de K_6 amb menys de 3 talls.

Veure que $\mathrm{cr}(R) \neq 7$ és més complex i s'utilitza que en una representació òptima de K_7 hi trobem una representació òptima d'un K_6, fet que ara demostrem.

Sabem que $\mathrm{cr}(R) \in \{7, 9\}$ i en els dos casos es compleix $\lceil 4\,\mathrm{cr}(R)/7 \rceil = \mathrm{cr}(R) - \mathrm{cr}(K_6)$. Tal i com en els casos anteriors, ara tenim que tota representació R d'algun K_7 existeix un vèrtex amb com a mínim $\mathrm{cr}(R) - \mathrm{cr}(K_6)$ talls incidents. Eliminem aquest vèrtex de R i obtenim una representació òptima de K_6.

Per utilitzar aquest fet, considerem la única representacions òptima de K_6 llevat d'isomorfisme i apliquem el mateix algoritme que en la Observació 3.4.3. Acabem veient que no hi ha cap manera distribuir les arestes de tal forma que el graf resultant tingui 7 talls i, a més, trobem 5 representacions no isomorfes de K_7 tals que $\mathrm{cr}(K_7) = 9$. Aquestes les podem trobar explicitades en l'article de R. K. Guy [4] i en la Figura 3.3. □

Corol·lari 3.4.5. *Com que coneixem que* $\mathrm{cr}(K_7) = \mathrm{Z}(7) = 9$ *immediatament pel Teorema 3.3.2 podem dir que* $\mathrm{cr}(K_8) = \mathrm{Z}(8) = 18$.

Proposició 3.4.6. *Cada representació òptima de* K_8 *conté una representació òptima d'un* K_7.

Demostració. Com que sabem que $\mathrm{cr}(K_8) = 18$ tenim que aquest tindrà un vèrtex amb responsabilitat superior o igual a $\lceil 4 \cdot 18/8 \rceil = 9 = \mathrm{cr}(K_8) - \mathrm{cr}(K_7)$. Podem treure aquell vèrtex al K_n i trobar-nos un K_{n-1} òptim. □

Observació 3.4.7. Si agafem els 5 grafs no isomorfs de K_7 i fem servir el mateix algoritme utilitzat en la Observació 3.4.3 per cada un d'ells es troben 3 representacions òptimes no isomorfes de K_8, les quals estan també recollides en [4] i en la Figura 3.4.

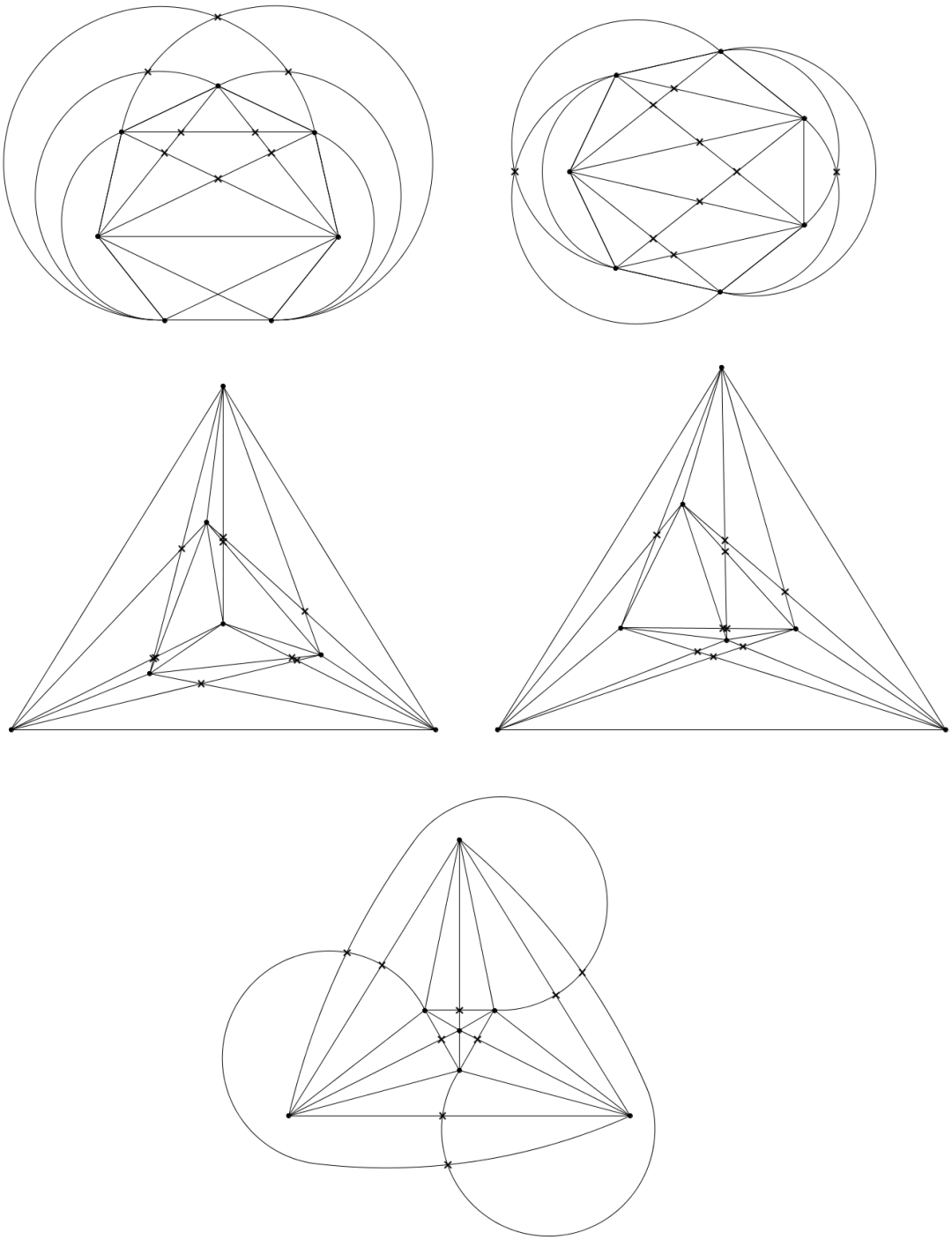

Figura 3.3: Les 5 representacions òptimes no isomòrfiques del K_7.

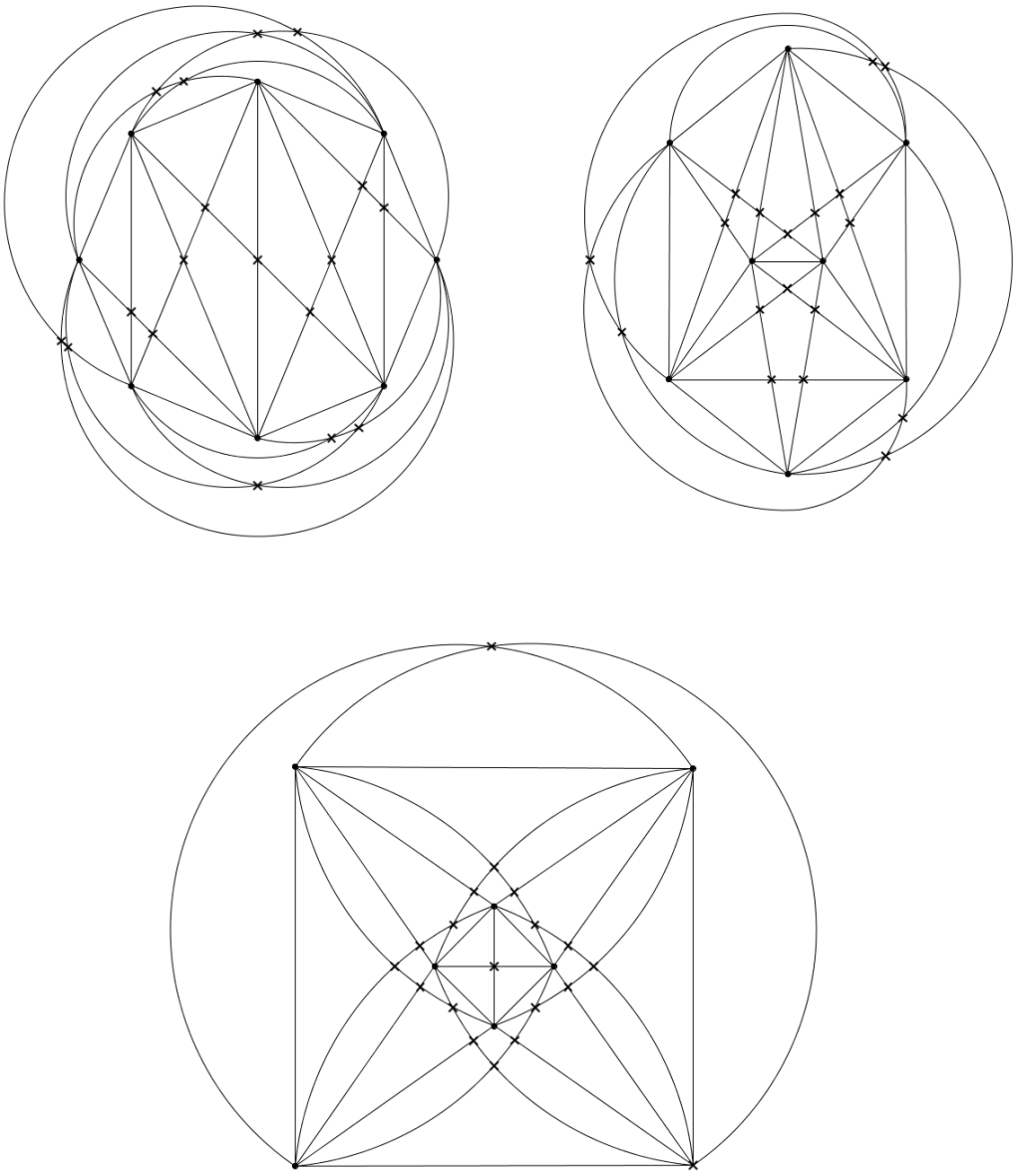

Figura 3.4: Les 3 representacions òptimes no isomòrfiques del K_8.

3.5 Nombre de talls de K_9 i K_{10}

Teorema 3.5.1. $\mathrm{cr}(K_9) = \mathrm{Z}(9) = 36$.

Demostració. Com que sabem que $\mathrm{cr}(K_8) = 18$ podem aplicar el Lema 3.3.1 i saber que $\mathrm{cr}(K_9) \geq \lceil 9 \cdot 18/5 \rceil = 33$. A més, pel Teorema 3.2.4 sabem que $\mathrm{cr}(K_9)$ ha de ser parell, doncs $\mathrm{cr}(K_9) = 34$ o $\mathrm{cr}(K_9) = 36$.

Pel Lema 3.4.1 tenim que hi ha un vèrtex amb com a mínim responsabilitat $\lceil 4 \cdot 34/9 \rceil = \lceil 4 \cdot 36/9 \rceil = 16$ en els dos casos, per tant, l'única opció que $\mathrm{cr}(K_9) = 34$ és que tingui una representació que en contingui una d'un K_8 amb 18 talls. Tanmateix, si intentem crear una representació d'un K_9 a partir d'un K_8 a partir de les 3 representacions òptimes no isomòrfiques ens trobem en què ens és impossible ja que sempre resulten representacions amb 36 talls, per tant, $\mathrm{cr}(K_9) = 36$. Tenim un exemple de representació òptima de K_9 en la Figura 3.5. □

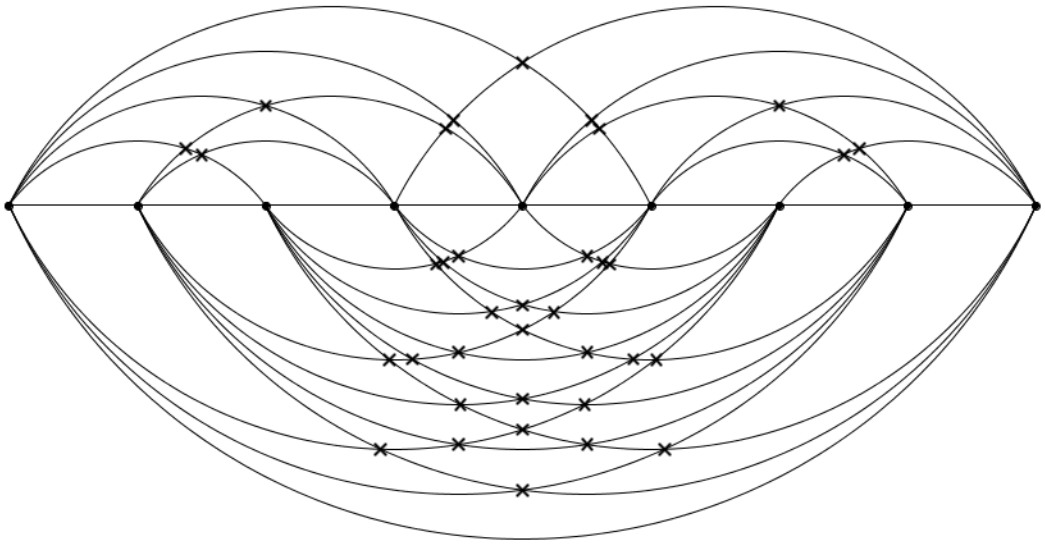

Figura 3.5: Una representació òptima del K_9.

Corol·lari 3.5.2. *Com que coneixem que* $\mathrm{cr}(K_9) = \mathrm{Z}(9) = 36$ *immediatament pel Teorema 3.3.2 podem dir que* $\mathrm{cr}(K_{10}) = \mathrm{Z}(10) = 60$. *Tenim un exemple de representació òptima de* K_{10} *en la Figura 3.6.*

Observació 3.5.3. En el que continua de projecte s'utilitzaran les desigualtats vistes en els Lemes 3.3.1 i 3.4.1, el Teorema 3.2.4 i fet que es coneguin els $\mathrm{cr}(K_n)$ determinats de forma recurrent i es donaran per certes sense prèvia notificació.

Proposició 3.5.4. *Tota representació òptima de* K_9 *conté una representació satisfactòria de* K_8 *amb* 20 *talls com a màxim i cada representació satisfactòria d'un* K_8 *amb com a molt* 20 *talls conté un* K_7 *òptim.*

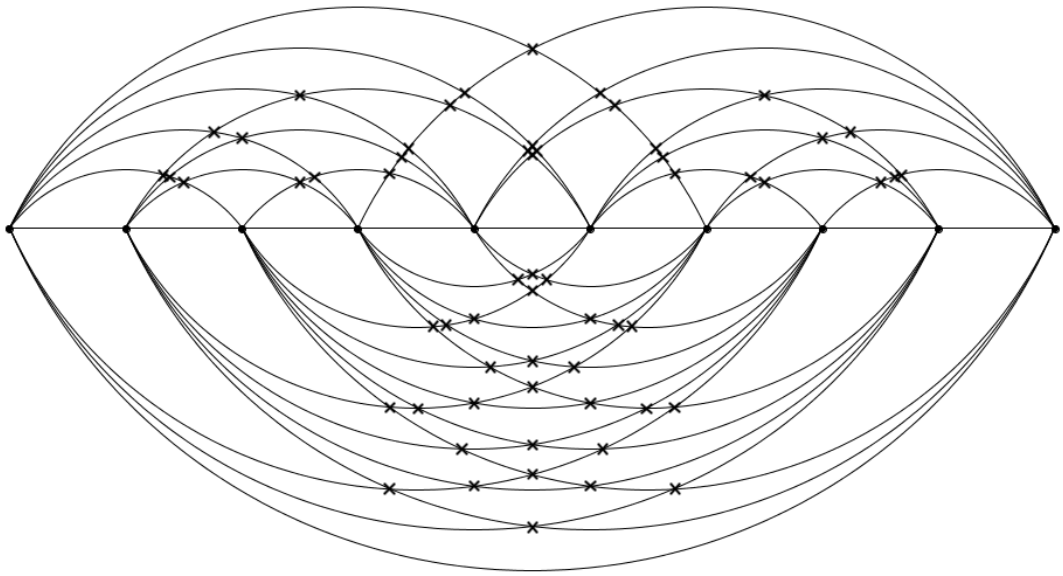

Figura 3.6: Una representació òptima del K_{10}.

Demostració. Per començar, considerem una representació R_9 òptima d'un K_9. Com que $\text{cr}(R_9) = 36$ tenim que $\lceil 4 \cdot 36/9 \rceil = 16 = \text{cr}(9) - 20$, per tant, tota representació òptima R_9 de K_9 conté un vèrtex amb com a mínim $\text{cr}(R) - 20$ talls incidents. Aquest fet implica que eliminant aquest vèrtex ens resulti una representació d'un K_8 amb 20 talls.

Per la segona part, només és necessari veure que pels casos en que el nombre de talls de la representació R sigui 19 o 20 ja que si és 18 ja ho hem demostrar amb anterioritat a la Proposició 3.4.2. Si $\text{cr}(R) = 19$ tenim que hi ha un vèrtex amb com a mínim responsabilitat $\lceil 4 \cdot 19/8 \rceil = 10$, per tant, si li trèiem aquest vèrtex a R ens queda una representació d'un K_7 amb 9 talls. Si $\text{cr}(R) = 20$ també tenim que hi ha un vèrtex amb com a mínim responsabilitat $\lceil 4 \cdot 19/8 \rceil = 10$. Al treure-li vèrtex a R ens queda una representació d'un K_8 amb com a molt 10 talls. Pel Teorema 3.2.4 sabem que les representacions de K_7 tenen nombre senat de talls, doncs la representació resultant tindrà 9 talls. □

3.6 Nombre de talls de K_{11} i K_{12}

Proposició 3.6.1. *Tota representació òptima de K_{11} amb menys de 100 talls conté una representació satisfactòria de K_{10} amb 62 talls com a màxim i cada representació satisfactòria d'un K_{10} amb com a molt 62 talls conté un K_9 òptim.*

Demostració. Per començar tenim que $\text{cr}(K_{11}) \geq \lceil 11 \cdot 60/7 \rceil) = 95$, de la qual cosa en deduïm que si $\text{cr}(K_{11}) < 100$ només hem de considerar dos casos:

- Si $\text{cr}(K_{11}) = 96$, sabem que existirà un vèrtex del graf amb responsabilitat

$\lceil 4 \cdot 96/11 \rceil = 35$ talls com a mínim. Eliminant-lo obtenim una representació de K_{10} amb com a molt 61 talls.

- Si $\mathrm{cr}(K_{11}) = 98$, sabem que existirà un vèrtex del graf amb responsabilitat $\lceil 4 \cdot 98/11 \rceil = 36$ talls com a mínim. Eliminant-lo obtenim una representació de K_{10} amb com a molt 62 talls.

En els dos casos ens resulta que el K_{11} conté un K_{10} amb com a molt 62 talls.

Per la segona part considerem les tres opcions:

- Si $\mathrm{cr}(K_{10}) = 60$, sabem que existirà un vèrtex del graf amb responsabilitat $\lceil 4 \cdot 60/10 \rceil = 24$ talls com a mínim. Eliminant-lo obtenim una representació de K_9 amb 36 talls.

- Si $\mathrm{cr}(K_{10}) = 61$, sabem que existirà un vèrtex del graf amb responsabilitat $\lceil 4 \cdot 61/10 \rceil = 25$ talls com a mínim. Eliminant-lo obtenim una representació de K_9 amb 36 talls.

- Si $\mathrm{cr}(K_{10}) = 62$, sabem que existirà un vèrtex del graf amb responsabilitat $\lceil 4 \cdot 62/10 \rceil = 25$ talls com a mínim. Eliminant-lo obtenim una representació de K_9 amb com a molt 37 talls. Com que les representacions de K_9 no poden tenir nombre de senar talls, sabem que en té 36.

En els tres casos em trobat una representació òptima d'un K_9 dins el K_{10}. □

Teorema 3.6.2. *Hi ha* 3080 *representacions òptimes no isomòrfiques del* K_9, 5679 *del* K_{10} *i a partir d'elles obtenim* $\mathrm{cr}(K_{11}) = \mathrm{Z}(11) = 100$.

Demostració. Donarem una idea de la demostració, la qual es troba àmpliament explicada en l'article [11]. A més en aquest document hi trobem una referència a un algoritme dels propis autors que construeix totes les representacions de K_n a partir d'afegir un vèrtex i les arestes necessàries a una representació de K_{n-1}.

Utilitzant la Proposició 3.5.4 i l'algoritme es demostra que hi ha 3080 representacions òptimes no isomòrfiques del K_9 i 5679 del K_{10}. Això es fa construint totes les representacions d'un K_8 amb menys de 21 talls a partir de les 5 del K_7 no isomorfes entre elles i, a continuació, construint les del K_9. A partir de les del òptimes del K_9 i gràcies a la Proposició 3.6.1 s'obté el nombre de òptimes del K_{10}.

A continuació, per veure que $\mathrm{cr}(K_{10}) = 100$ es considera la Proposició 3.6.1. Es construeixen primer totes les representacions de K_{10} amb menys de 62 talls i, seguidament, aplicant l'algoritme a aquestes s'observa que només es generen K_{11} amb com a mínim 100 talls. Tenim un exemple de representació òptima de K_{12} en la Figura 3.7. □

Corol·lari 3.6.3. *Com que coneixem que* $\mathrm{cr}(K_{11}) = \mathrm{Z}(11) = 100$ *immediatament pel Teorema 3.3.2 podem dir que* $\mathrm{cr}(K_{12}) = \mathrm{Z}(12) = 150$. *Tenim un exemple de representació òptima de* K_{12} *en la Figura 3.8.*

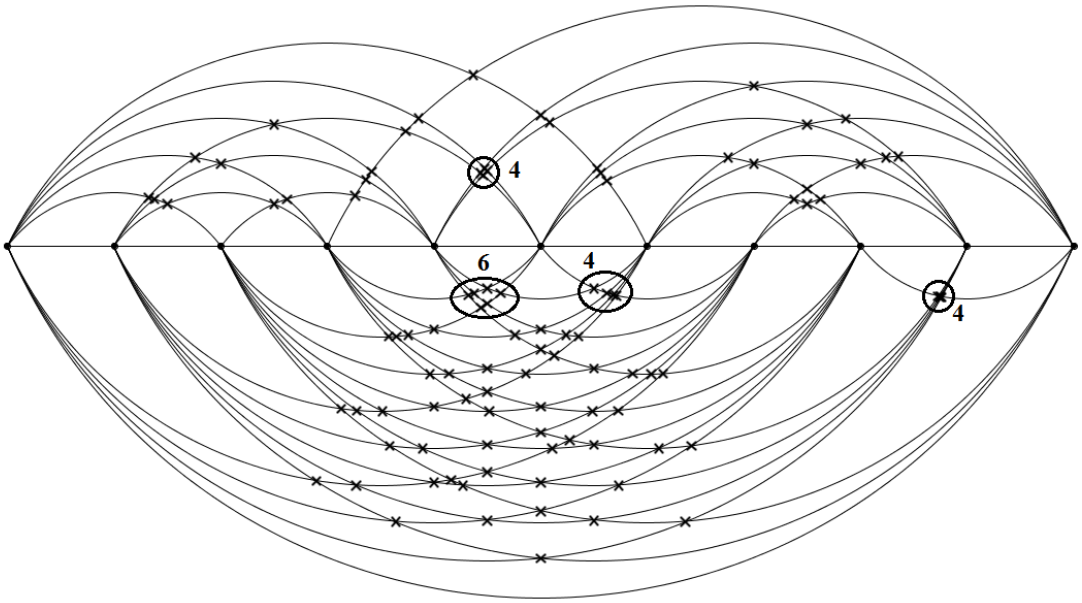

Figura 3.7: Una representació òptima del K_{11}.

3.7 Cota mínima pels grafs complets

Teorema 3.7.1. *Per tot n natural* $0,\widehat{80}\,\mathrm{Z}(n) \leq \mathrm{cr}(K_n)$.

Demostració. Per començar veurem que per tot $n \geq 5$:

$$\frac{\mathrm{cr}(K_{n+1})}{\binom{n+1}{4}} \geq \frac{\mathrm{cr}(K_n)}{\binom{n}{4}}.$$

Per fer-ho utilitzem el Lema 3.3.1, gràcies al qual podrem fer el pas de la desigualtat:

$$\frac{\mathrm{cr}(K_{n+1})}{\binom{n+1}{4}} \geq \frac{\frac{(n+1)\cdot \mathrm{cr}(K_n)}{n-3}}{\binom{n+1}{4}} = \frac{\mathrm{cr}(K_n)}{\frac{(n+1)!\cdot(n-3)}{4\cdot(n-3)!\cdot(n+1)}} = \frac{\mathrm{cr}(K_n)}{\binom{n}{4}}.$$

A continuació demostrem que per tot $n \geq 4$ tenim que $\mathrm{Z}(n) < \binom{n}{4}$:

$$\mathrm{Z}(n) = \frac{1}{4}\left\lfloor\frac{n}{2}\right\rfloor\left\lfloor\frac{n-1}{2}\right\rfloor\left\lfloor\frac{n-2}{2}\right\rfloor\left\lfloor\frac{n-3}{2}\right\rfloor < \frac{1}{4}\cdot\frac{n}{2}\cdot\frac{n-1}{2}\cdot\frac{n-2}{2}\cdot\frac{n-3}{2}$$

$$= \frac{n!}{2^6\cdot(n-4)!} = \frac{3}{8}\frac{n!}{4!\cdot(n-4)!} = \frac{3}{8}\binom{n}{4}.$$

Juntament amb les desigualtats anteriors i el Teorema 3.6.2 que ens diu que $\mathrm{cr}(K_{11}) = 100$, veiem que per $n \geq 11$:

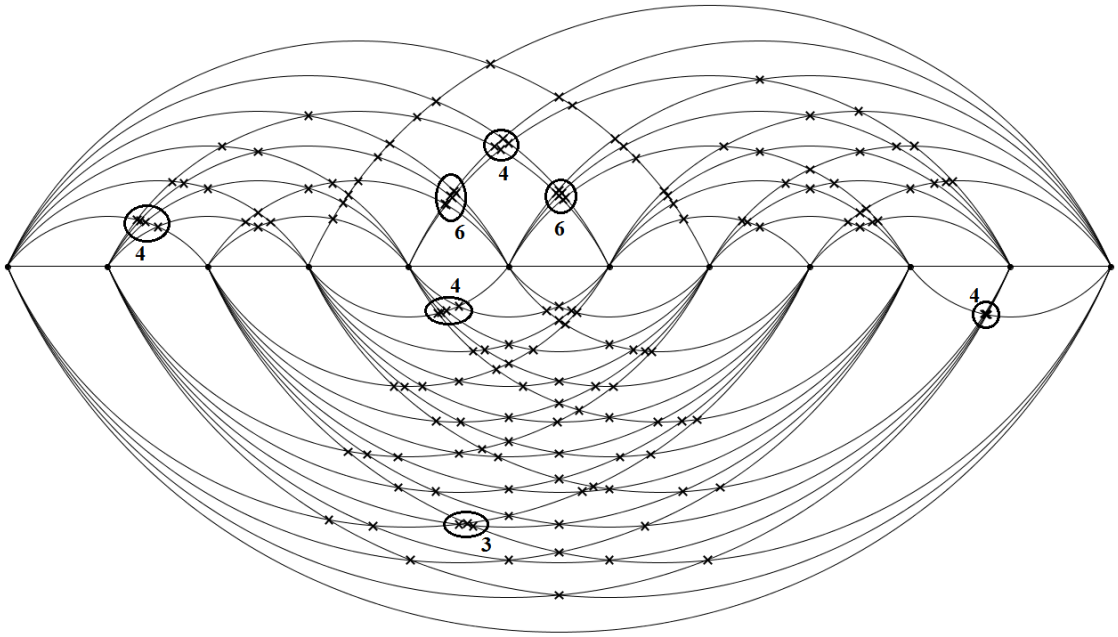

Figura 3.8: Una representació òptima del K_{12}.

$$\frac{\mathrm{cr}(K_n)}{Z(n)} > \frac{\mathrm{cr}(K_n)}{\binom{n}{4}}\frac{8}{3} \geq \frac{\mathrm{cr}(K_{11})}{\binom{11}{4}}\frac{8}{3} = \frac{80}{99} = 0,\widehat{80}.$$

Per completar la demostració falten els casos on $n < 11$, però sabem que en aquests casos $\mathrm{cr}(K_n) = Z(n)$, per tant, ja hem acabat. □

Capítol 4

Nombres de talls dels grafs bipartits complets

En el cas dels grafs bipartits també es té una conjectura, la conjectura de Zarankiewicz, la qual ens diu que $\mathrm{cr}(K_{m,n})$ és igual a la següent funció

$$\mathrm{Y}(m,n) = \left\lfloor \frac{m}{2} \right\rfloor \left\lfloor \frac{m-1}{2} \right\rfloor \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor.$$

En la següent taula tenim els valors de Y en funció de m i n:

m, n	2	3	4	5	6	7	8	9	10
2	0	0	0	0	0	0	0	0	0
3	0	1	2	4	6	9	12	16	20
4	0	2	4	8	12	18	24	32	40
5	0	4	8	16	24	36	48	64	80
6	0	6	12	24	36	54	72	96	120
7	0	9	18	36	54	81	108	144	180
8	0	12	24	48	72	108	144	192	240
9	0	16	32	64	96	144	192	256	360
10	0	20	40	80	120	180	240	360	400

En el lloc web de Wolfram [16] hi trobem una breu referència a aquest tema on s'explica que aquesta conjectura ha estat provada per únicament els valors (m,n) tals que $m \leq 6$, $m = 7, 8$ i $n = 7, 8, 9, 10$, i, òbviament, els seus simètrics. Tanmateix, també es coneix una demostració pel casos $m = 9, 10$ i $n = 9, 10$ la qual ha estat publicada aquest any. Donarem les demostracions per tots els casos, o si més no, la idea d'aquestes.

Com en el cas dels grafs complets el que sí que hi ha són unes cotes generals pel nombre de talls dels grafs, les quals estan recollides en [9] on es dóna la següent estimació:

$$\lim_{n\to\infty} \frac{\mathrm{cr}(K_{m,n})}{\mathrm{Y}(m,n)} \geq 0,8594 \frac{m}{m-1}.$$

Els primers resultats mostrats estan recollits en l'article de Kleitman [7] i les demostracions per $K_{7,7}$ i $K_{7,9}$ les trobem en el de Woodall [17]. En el cas que algun fet no es trobi en cap d'aquests dos casos s'indica on es poden trobar les demostracions originals.

4.1 Representacions que assoleixen $\mathbb{Z}(m,n)$.

Observació 4.1.1. Tots els fets demostrats per $K_{m,n}$ a partir d'una condició imposada sobre m són aplicables per $K_{n,m}$ i viceversa, ja que $K_{m,n} = K_{n,m}$.

Teorema 4.1.2. *Per tot* $m, n \geq 2$, $\mathrm{cr}(K_{m,n}) \leq \mathrm{Y}(m,n)$.

Demostració. La demostració la tenim gràcies a Zarankiewich i es troba publicada en [18].

L'objectiu és generar grafs bipartits que assoleixin la cota, i així es fa. Es considera el pla euclidi i es situen els vèrtexs del graf $K_{m,n}$ en l'eix abscissa si pertanyen U o en l'ordenat si pertanyen W. La clau de la col·locació es posar-ne $\lfloor \frac{m}{2} \rfloor$ de U sobre la recta $y = 0, x > 0$, la resta de U sobre $y = 0, x < 0$, $\lfloor \frac{n}{2} \rfloor$ de W sobre $x = 0, y > 0$ i la resta de W sobre $x = 0, y < 0$. Per fer-ho d'una manera més ordenada considerem que col·loquem els vèrtexs $u_i \in U$ en les posicions $(i(-1)^i, 0)$ i els vèrtexs $w_j \in W$ en les posicions $(0, j(-1)^j)$. A continuació, tracem segments rectes entre els vèrtexs de U i W, els quals ens generaran una representació R d'un graf $K_{m,n}$, com es pot veure en la Figura 4.1. Veiem que la quantitat de talls és $\mathrm{Y}(m,n)$.

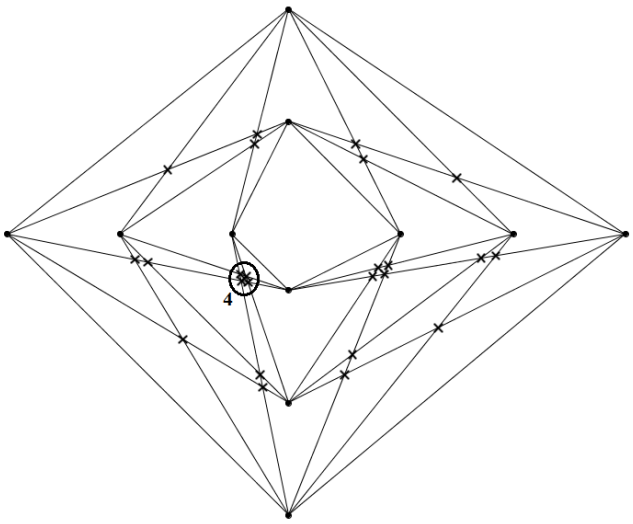

Figura 4.1: Una representació òptima del $K_{5,6}$.

Per començar, és interessant observar que els talls quedaran dividits ens els 4 quadrants, Q_1, Q_3, Q_3, Q_4. Es fàcil comprovar que la quantitat de talls existents en cada quadrant on s'entrellacin les arestes sortints de k vèrtexs per un eix i l per

l'altre és de $\frac{(k-1)\cdot k}{2}\frac{(l-1)\cdot l}{2}$. Tindrem quatre quadrants, en el primer $(k,l) = (\lfloor \frac{m}{2} \rfloor, \lfloor \frac{n}{2} \rfloor)$, en el segon $(\lfloor \frac{m+1}{2} \rfloor, \lfloor \frac{n}{2} \rfloor)$, en el tercer $(\lfloor \frac{m+1}{2} \rfloor, \lfloor \frac{n+1}{2} \rfloor)$ i en el quart $(\lfloor \frac{m}{2} \rfloor, \lfloor \frac{n+1}{2} \rfloor)$. Comprovem la fórmula per casos:

- Si m, n són parells tenim que en cada quadrant hi haurà la mateixa quantitat de talls, per tant:

$$\begin{aligned}\operatorname{cr}(R) &= 4\operatorname{cr}(Q_1) = 4\frac{(\frac{m}{2}-1)(\frac{m}{2})}{2}\frac{(\frac{n}{2}-1)(\frac{n}{2})}{2} \\ &= \left(\frac{m-2}{2}\right)\left(\frac{m}{2}\right)\left(\frac{n-2}{2}\right)\left(\frac{n}{2}\right) = \mathrm{Y}(m,n).\end{aligned}$$

- Si m és parell i n és senar tenim que en el Q_1 hi haurà els mateixos talls que en el Q_2 i en el Q_3 que en el Q_4. Per tant:

$$\begin{aligned}\operatorname{cr}(R) &= 2\operatorname{cr}(Q_1) + 2\operatorname{cr}(Q_3) \\ &= 2\left(\frac{(\frac{m}{2}-1)(\frac{m}{2})}{2}\frac{(\frac{n-1}{2}-1)(\frac{n-1}{2})}{2} + \frac{(\frac{m}{2}-1)(\frac{m}{2})}{2}\frac{(\frac{n+1}{2}-1)(\frac{n+1}{2})}{2}\right) \\ &= \frac{1}{2}\left(\frac{m}{2}-1\right)\left(\frac{m}{2}\right)\left(\frac{n-1}{2}\right)\left(\frac{n+1}{2}+\frac{n-3}{2}\right) \\ &= \left(\frac{m-2}{2}\right)\left(\frac{m}{2}\right)\left(\frac{n-1}{2}\right)^2 = \mathrm{Y}(m,n).\end{aligned}$$

- Si m és senar i n és parell ens trobem un cas equivalent a l'anterior canviant m per n.

- Si m, n són senars tots els quadrants són diferents. Aquest el càlcul és llarg i similar els anteriors només en denotem una part:

$$\begin{aligned}\operatorname{cr}(R) &= \operatorname{cr}(Q_1) + \operatorname{cr}(Q_2) + \operatorname{cr}(Q_3) + \operatorname{cr}(Q_4) \\ &= \ldots = \left(\frac{m-1}{2}\right)^2\left(\frac{n-1}{2}\right)^2 = \mathrm{Y}(m,n).\end{aligned}$$

□

4.2 Paritat del nombre de talls

Lema 4.2.1. *Tota representació satisfactòria del $K_{3,3}$ té un nombre senar de talls.*

Demostració. Considerem D una representació d'un $K_{3,3}$ i suposem que $\operatorname{cr}(D)$ és parell. Diem A i B als dos subconjunts de vèrtexs del graf tals que A conté als vèrtexs d'una part de la bipartició i B als altres. Com que totes les arestes van d'un vèrtex de A a un de B tenim que cada tall serà incident a un parell de A i a un parell de B, per tant, $\sum_{v\in A} r_v = 2\operatorname{cr}(D)$. Aquest fet que ens indica que algun

vèrtex de A haurà de tenir responsabilitat parella i doncs, podem considerar aquest vèrtex, li direm v, i l'eliminem de D.

Ara ens queda una representació D' de $K_{2,3}$ amb un nombre parell de talls. Cada un d'aquests talls és incident a 4 vèrtexs, els dos que queden de A i dos de B. Doncs, sabem que cada tall està en un sol $K_{2,2}$. Per un altre banda, sabem que tenim 3 $K_{2,2}$ dins una representació d'un $K_{2,3}$ ja que hem d'eliminar un vèrtex de B, i només n'hi ha 3. Com que cada $K_{2,2} = C_4$ satisfactori només pot tenir un tall $\operatorname{cr}(D') \leq 3$, per tant, sabent que $\operatorname{cr}(D')$ és parell $\operatorname{cr}(D') \in \{0, 2\}$, i en conseqüència algun $K_{2,2}$ de D' no té talls.

Considerant la representació de $K_{2,2} = C_4$ que no té talls, observem que té dues cares i que el vèrtex que hem tret de D' ha d'estar en una d'elles. Fent totes les combinacions i imposant que $\operatorname{cr}(D') \in \{0, 2\}$ només hi ha 3 opcions no isomòrfiques possibles de col·locar el vèrtex, dues de les quals amb dos talls i una amb cap.

Ara ens disposem a tornar a posar v en el $K_{2,3}$. En cada una de les 3 opcions anteriors succeeix el mateix, si considerem que v ha d'anar a una cara determinada llavors la paritat del nombre de talls de les arestes $\overline{vw}$ on $w \in B$ serà sempre la mateixa, independentment de per on fem passar l'aresta $\overline{vw}$. Això és cert ja que les arestes de $K_{2,3}$ no incidents amb w generen un cicle de 4 elements. De forma clar, es veu que només una de les arestes tindrà paritat senar, per tant, $\operatorname{cr}(D) = \operatorname{cr}(D') + \operatorname{cr}(\overline{vw_1}) + \operatorname{cr}(\overline{vw_2}) + \operatorname{cr}(\overline{vw_3})$ serà senar i arribem a contradicció. □

Teorema 4.2.2. *Tota representació satisfactòria d'un $K_{m,n}$ on m i n senars té la mateixa paritat que $\binom{m}{3}\binom{n}{3}$.*

Demostració. Dins un $K_{m,n}$ trobem $\binom{m}{3}\binom{n}{3}$ maneres diferents de seleccionar tres vèrtexs de cada un dels costats de $K_{m,n}$ i cada tria genera un $K_{3,3}$ diferent. Sumant la quantitat de talls de cada $K_{3,3}$ obtenim el nombre de talls de $K_{m,n}$ multiplicat per $(m-2)(n-2)$ ja que cada tall apareix en $(m-2)(n-2)$ $K_{3,3}$.

Com que $(m-2)$ i $(n-2)$ són senars tenim que la paritat de $\operatorname{cr}(K_{m,n})$ és la mateixa que de $(m-2)(n-2)\operatorname{cr}(K_{m,n})$ i, alhora, aquesta és la mateixa que la de la suma dels $\binom{m}{3}\binom{n}{3}$ $K_{3,3}$. A més, el fet que $\operatorname{cr}(K_{3,3}) \equiv 1 \pmod 2$ ens indica que és igual a la de $\binom{m}{3}\binom{n}{3}$. Doncs, acabem tenim $\operatorname{cr}(K_{m,n}) \equiv \binom{m}{3}\binom{n}{3} \pmod 2$. □

Observació 4.2.3. Com en el cas dels grafs complets, el Lema 4.2.1 i el Teorema 4.2.2 es troben en l'article [10] de McQuillan i Richter.

4.3 Acotacions de talls a partir dels anteriors

Lema 4.3.1. *Per $n \geq 3$ es compleix la següent desigualtat:*

$$\operatorname{cr}(K_{m,n}) \geq \frac{n \cdot \operatorname{cr}(K_{m,n-1})}{n-2}.$$

Demostració. La demostració és semblant a la del Lema 3.3.1. Considerem una representació òptima de $K_{m,n}$ i n'eliminem un vèrtex de W i les seves arestes adjacents, obtenim una representació de $K_{m,n-1}$. Si això ho fem per separat amb cada

vèrtex tenim n representacions de $K_{m,n-1}$. A més, com que cada tall és incident a dos vèrtexs de W tenim que cada tall el trobarem a $n-2$ grafs dels anteriors. Donat que les representacions de $K_{m,n-1}$ tindran com a mínim $\mathrm{cr}(K_{m,n-1})$ talls, en deduïm:

$$(n-2)\cdot \mathrm{cr}(K_{m,n}) = n\cdot \mathrm{cr}(K_{m,n-1}).$$

□

Teorema 4.3.2. *Si* $\mathrm{cr}(K_{m,n}) = \mathrm{Y}(m,n)$ *per un nombre* n *senar llavors* $\mathrm{cr}(K_{m,n+1}) = \mathrm{Y}(m,n+1)$.

Demostració. Pel Lema 4.3.1 tenim:

$$\begin{aligned}\mathrm{cr}(K_{m,n+1}) &\geq \frac{(n+1)\cdot \mathrm{cr}(K_{m,n})}{n-1} = \frac{n+1}{n-1}\left\lfloor\frac{m}{2}\right\rfloor\left\lfloor\frac{m-1}{2}\right\rfloor\frac{n-1}{2}\frac{n-1}{2}\\ &= \left\lfloor\frac{m}{2}\right\rfloor\left\lfloor\frac{m-1}{2}\right\rfloor\frac{n+1}{2}\frac{n-1}{2} = \mathrm{Y}(m,n+1).\end{aligned}$$

Del Teorema 4.1.2 sabem que existeixen grafs que assoleixen la cota, llavors obtenim que $\mathrm{cr}(K_{m,n+1}) = \mathrm{Y}(m,n+1)$. □

Teorema 4.3.3. *Si* $\mathrm{cr}(K_{m,n}) = \mathrm{Y}(m,n)$ *per un nombre* n *parell llavors:*

$$\mathrm{cr}(K_{m,n+1}) \geq \mathrm{Y}(m,n+1) - \left\lfloor\left\lfloor\frac{m}{2}\right\rfloor\left\lfloor\frac{m-1}{2}\right\rfloor\frac{n}{2(n-1)}\right\rfloor.$$

Demostració. Pel Lema 4.3.1 tenim:

$$\begin{aligned}\mathrm{cr}(K_{m,n+1}) &\geq \frac{(n+1)\cdot \mathrm{cr}(K_{m,n})}{n-1} = \frac{n+1}{n-1}\left\lfloor\frac{m}{2}\right\rfloor\left\lfloor\frac{m-1}{2}\right\rfloor\frac{n}{2}\frac{n-2}{2}\\ &= \frac{(n+1)\cdot(n-2)}{n\cdot(n-1)}\cdot \mathrm{Y}(m,n+1)\\ &= \left(1+\frac{(n+1)\cdot(n-2)-n\cdot(n-1)}{n\cdot(n-1)}\right)\cdot \mathrm{Y}(m,n+1)\\ &= \mathrm{Y}(m,n+1) + \left\lfloor\frac{m}{2}\right\rfloor\left\lfloor\frac{m-1}{2}\right\rfloor\frac{n}{2}\frac{n}{2}\frac{(n+1)\cdot(n-2)-n\cdot(n-1)}{n\cdot(n-1)}\\ &= \mathrm{Y}(m,n+1) - \left\lfloor\frac{m}{2}\right\rfloor\left\lfloor\frac{m-1}{2}\right\rfloor\frac{n}{2(n-1)}.\end{aligned}$$

Com que $\mathrm{cr}(K_{m,n})$ és enter podem considerar que $\left\lfloor\frac{m}{2}\right\rfloor\left\lfloor\frac{m-1}{2}\right\rfloor\frac{n}{2(n-1)}$ també ho és, per tant, ja hem acabat. □

4.4 Nombre de talls de $K_{3,n}$ i $K_{4,n}$

Notació 4.4.1. Donat una representació i dos vèrtexs d'ella, v_1, v_2, notarem el nombre de talls que són incidents als dos alhora com $\mathrm{cr}(v_1, v_2)$. És clar observar que donada una representació tenim la següent igualtat:

$$\mathrm{cr}(R) = \sum_{i=1}^{m} \sum_{j=i+1}^{m} \mathrm{cr}(u_i, u_j).$$

I equivalentment, pels vèrtexs de W.

Lema 4.4.2. *Sigui R una representació de $K_{m,n}$ on $m, n \geq 3$. Si hi ha dos vèrtexs en W que no tenen cap tall incident compartit es compleix la següent desigualtat:*

$$\mathrm{cr}(D) \geq \mathrm{cr}(K_{m,n-2}) + (n-2)\,\mathrm{cr}(K_{m,3}).$$

Demostració. Considerem una representació R òptima de $K_{m,n}$ i n'eliminem dos vèrtexs de $w_1, w_2 \in W$. Com a mínim haurem eliminat $(n-2)\,\mathrm{cr}(K_{m,3})$ talls ja que tenim $(n-2)$ vèrtexs restants en W, que si els unim a w_1, w_2 i a tots els de U conformen $(n-2)$ $K_{m,3}$. A més, $\mathrm{cr}(w_1, w_2) = 0$ per hipòtesi, per tant, els $\mathrm{cr}(K_{m,3})$ no compartiran talls. Per aquest motiu i pel fet que com a mínim les representacions dels $K_{m,3}$ tindran $\mathrm{cr}(K_{m,3})$ talls, obtenim la desigualtat buscada:

$$\mathrm{cr}(R) \geq \mathrm{cr}(K_{m,n-2}) + (n-2)\,\mathrm{cr}(K_{m,3}).$$

□

Teorema 4.4.3. $\mathrm{cr}(K_{3,n}) = \mathrm{Y}(3, n)$.

Demostració. Per començar veiem que $\mathrm{cr}(K_{3,m}) = \mathrm{Y}(3, m)$.

A partir del Teorema 4.3.2 i del fet que coneguem que $\mathrm{cr}(K_{3,3}) = 1 = \mathrm{Y}(3, 3)$ directament tenim que $\mathrm{cr}(K_{3,4}) = 2 = \mathrm{Y}(3, 4)$.

Prèviament, veiem que per tota n, si R és una representació òptima de $K_{3,n}$, aquesta té dos vèrtexs en W que no comparteixen talls incidents a ells. En cas contrari, tindríem que les arestes de cada parella de vèrtexs en W formaria com a mínim un tall. Com que tenim $\binom{n}{2}$ combinacions possibles de dos talls i $\binom{n}{2} > \mathrm{Y}(3, n)$, llavors R no seria òptim i arribaríem a contradicció. Aquest fet ens permet utilitzar el Lema 4.4.2 si $m = 3$ per totes les representacions.

Ara usem inducció per acabar la demostració. Els casos inicials són $K_{3,3}$ i $K_{3,4}$, que sabem que són certs. El pas d'inducció que utilitzem per a demostrar que $\mathrm{cr}(K_{3,n}) = \mathrm{Y}(3, n)$ és que coneixem que $\mathrm{cr}(K_{3,n-2}) = \mathrm{Y}(3, n-2)$. Treballant per casos tenim:

- Si n és senar obtenim la següent desigualtat:

$$\begin{aligned}\mathrm{cr}(R) &\geq \mathrm{cr}(K_{3,n-2}) + (n-2)\,\mathrm{cr}(K_{3,3}) = \frac{n-3}{2}\,\frac{n-3}{2} + (n-2) \\ &= \frac{n^2-6n+9}{4}\,\frac{4n-8}{4} = \frac{n^2-2n+1}{4} = \frac{(n-1)^2}{4} = \mathrm{Y}(3, n).\end{aligned}$$

- Si n és parell obtenim la següent desigualtat:

$$\begin{aligned}\operatorname{cr}(R) &\geq \operatorname{cr}(K_{3,n-2}) + (n-2)\operatorname{cr}(K_{3,3}) = \frac{n-2}{2}\,\frac{n-4}{2} + (n-2)\\ &= \frac{n^2-6n+8}{4}\,\frac{4n-8}{4} = \frac{n^2-2n}{4} = \frac{n(n-1)}{4} = \mathrm{Y}(3,n).\end{aligned}$$

Ja hem vist que $\operatorname{cr}(K_{3,n}) \geq \mathrm{Y}(3,n)$ i sabem que $\mathrm{Y}(3,n) \geq \operatorname{cr}(K_{3,n})$, per tant, $\mathrm{Y}(3,n) = \operatorname{cr}(K_{3,n})$. □

Corol·lari 4.4.4. *Com que coneixem que* $\forall n \operatorname{cr}(K_{3,n}) = \mathrm{Z}(3,n)$ *immediatament pel Teorema 4.3.2 podem dir que* $\forall n \operatorname{cr}(K_{4,n}) = \mathrm{Z}(4,n)$.

4.5 Nombre de talls de $K_{5,n}$ i $K_{6,n}$

Lema 4.5.1. *Sigui D una representació de $K_{2r+1,2s+1}$. Donats $w_1, w_2 \in W$, $\operatorname{cr}(w_1)+\operatorname{cr}(w_2)-\operatorname{cr}(w_1,w_2)$ serà parell si r és parell i serà senar si r és senar.*

Demostració. Per veure aquest fet utilitzem el Teorema 4.2.2. Aquest ens diu que si tenim una representació optima de $K_{4t+1,2s+1}$ tindrà un nombre parell de talls i si és de la forma $K_{4t+3,4u+3}$ tindrà un nombre senar de talls.

Per altra banda tenim que $\operatorname{cr}(w_1)+\operatorname{cr}(w_2)-\operatorname{cr}(w_1,w_2)$ és la diferència entre el nombre de talls de la representació D de $K_{2r+1,2s+1}$ i aquesta mateixa havent eliminat w_1, w_2. Pel que tenim del paràgraf anterior podem concloure que $\operatorname{cr}(w_1)+\operatorname{cr}(w_2)-\operatorname{cr}(w_1,w_2)$ serà parell si $2r+1 = 4t+1$ o senar si $2r+1 = 4t+3$. □

Observació 4.5.2. Donada una representació d'un graf $K_{m,n}$ podem construir un $K_{m,n+1}$ de la següent manera. Triem un vèrtexs de $w_i \in W$ i li afegim el nou vèrtex w_{n+1} dins un cercle c_i de radi ε, tant petit com sigui necessari de tal manera que les úniques arestes que hi hagi siguin les sortints de w_i. A continuació, tracem les rectes del vèrtex w_{n+1} cap els u_j per $j = 1,\ldots,m$ de tal manera que cada $\overline{w_{n+1}u_j}$ vagi totalment enganxada a $\overline{w_iu_j}$ i acabi resultant que $\operatorname{cr}(w_{n+1},w_k) = \operatorname{cr}(w_i,w_k)$ per $k = 1,\ldots,\hat{i},\ldots,n$. A més, és fàcil observar que es pot complir la següent acotació $\operatorname{cr}(w_{n+1},w_i) \geq \mathrm{Y}(m,3)$. Donem la Figura 4.2 perquè es pugui veure com s'assoleix la cota per $m = 6$ i, per simetria, com es pot fer en els demés m.

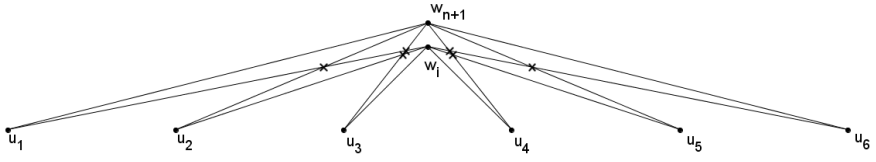

Figura 4.2: S'assoleix l'acotació per $m = 6$.

Definició 4.5.3. *Sigui R una representació d'un $K_{m,n}$ i $w_i \in W$. Considerem un cercle c_i de radi ε, tant petit com sigui necessari de tal manera que les úniques arestes que hi hagi siguin les sortints de w_i i només es creuïn un cop amb $\overline{w_i u_j}$. Definim el punt $P_{i,j} = j$ com la intercessió de c_i amb $\overline{w_i u_j}$. Començant en un punt arbitrari de c_i i fem un gir en el sentit de les agulles del rellotge obtenim una successió de valors que corresponen a una permutació de $(1, \ldots, m)$ que definim com π_i. Definim $\overline{\pi_i}$ com la successió que trobem girant cap l'altre sentit.*

Observació 4.5.4. Donat que comencem en un lloc arbitrari, dues permutacions diferents poden definir la mateixa col·locació de les arestes. Això ho fem així perquè potser més útil utilitzar una permutació o un altre en funció del moment. Veiem un exemple de dues permutacions que són la mateixa a partir de la Figura 4.3:

$$(6, 5, 3, 4, 1, 2) = (1, 2, 6, 5, 3, 4).$$

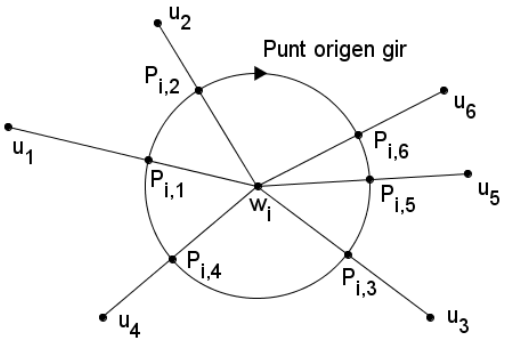

Figura 4.3: Exemple de π_i.

Definició 4.5.5. *Siguin π_i, π_j dues permutacions diferents. Definim $\chi_{i,j}$ com el mínim nombre d'intercanvis d'elements adjacents que s'han d'aplicar a π_i per obtenir $\overline{\pi_j}$. Hem de tenir en compte π_i i π_j en totes les seves formes variants.*

Lema 4.5.6. *Sigui R una representació d'un $K_{m,n}$ i $w_i, w_j \in W$. Llavors* $\mathrm{cr}(w_i, w_j) \geq \chi_{i,j}$, *i si $\pi_i = \pi_j$ llavors* $\mathrm{cr}(w_i, w_j) \geq \mathrm{Y}(m, 3)$.

Demostració. Donem una idea de la demostració. Per començar, veiem el primer cas de $\mathrm{cr}(w_i, w_j) \geq \chi_{i,j}$, si $\mathrm{cr}(w_i, w_j) = 0$ aleshores $\pi_i = \overline{\pi_j}$. Aquest fet és clar i es pot observar imposant la condició sobre un dibuix. La desigualtat general es demostra fent inducció sobre el nombre de talls.

Per l'altre part del lema utilitzem la Figura 4.2. Podem veure que considerar que si $\pi_i = \pi_j$ els vèrtexs es trobaran d'aquella manera distribuïts o semblant. Llavors és clara la desigualtat. □

Teorema 4.5.7. $\mathrm{cr}(K_{5,n}) = \mathrm{Y}(5, n)$.

Demostració. Per començar veiem que $\mathrm{cr}(K_{5,m}) = \mathrm{Y}(5,m)$.

A partir dels Teoremes 4.1.2, 4.3.2 i 4.3.3 podem dir que si coneixem $\mathrm{cr}(K_{5,2s-1}) = \mathrm{Y}(5,2s-1)$ per un s determinat, sabem que $\mathrm{cr}(K_{5,2s}) = \mathrm{Y}(5,2s)$ i $\mathrm{Y}(5,2s+1) \geq \mathrm{cr}(K_{5,2s+1}) \geq \mathrm{Y}(5,2s+1)-2$. A més, gràcies al Teorema 4.2.2 sabem que $\mathrm{cr}(K_{5,2s+1})$ ha de ser parell, per tant, no pot ser $\mathrm{Y}(5,2s+1)-1$.

Si veiem que tampoc pot ser $\mathrm{cr}(K_{5,2s+1}) = \mathrm{Y}(5,2s+1)-2$ ja haurem acabat, i per fer-ho aplicarem inducció. Coneixem que el resultat és cert per $s=1$, suposarem que és cert per $s-1$ i veurem que ho és per s. Acabarem concloent, doncs, que $\mathrm{cr}(K_{5,2s+1}) = \mathrm{Y}(5,2s+1)$ per tot s.

Per començar, considerant n senar coneixem que $\mathrm{cr}(K_{5,n-1}) = \mathrm{Y}(5,n-1)$ i $\mathrm{cr}(K_{5,n-2}) = \mathrm{Y}(5,n-2)$ per hipòtesi d'inducció. Suposem que existeix una representació R de $K_{5,n}$ amb $\mathrm{cr}(R) = \mathrm{Y}(5,n)-2$. Si eliminem un vèrtex de $W \subset R$ es queda una representació de $K_{5,n-1}$ amb com a molt $\mathrm{Y}(5,n-1)$ talls. Aquest fet prova la següent desigualtat:

$$\forall w_i \in W, \mathrm{cr}(w_i) \geq \mathrm{Y}(5,n) - 2 - \mathrm{Y}(5,n-1) = 2n-4.$$

Alhora, sabem que si sumem tots els nombres de talls de tots els vèrtexs de W obtenim el doble dels talls de R ja que els sumem tots els talls dos cops:

$$\sum_{i=1}^{n} \mathrm{cr}(w_i) = 2\,\mathrm{cr}(R) = n(2n-4) - 2.$$

Aquests dos fets donen poc marge amb el nombre de talls dels vèrtexs ja que si tots valguessin el màxim, $\forall w_i \in W, \mathrm{cr}(w_i) = 2n-4$ obtindrien que $\sum_{i=1}^{n} = n(2n-4)$. Això no pot ser ja que ens passem de dos talls, però ens indica que tenim dues opcions, llevat l'ordre dels índexs:

1. $\mathrm{cr}(w_1) = 2n-6$ i $\mathrm{cr}(w_i) = 2n-4$ per $i \in 2,\ldots,n$.

2. $\mathrm{cr}(w_1) = \mathrm{cr}(w_2) = 2n-5$ i $\mathrm{cr}(w_i) = 2n-4$ per $i \in 2,\ldots,n$.

Possibilitat 1

Veiem que $\mathrm{cr}(w_1) = 2n-6$ i $\mathrm{cr}(w_i) = 2n-4$ per $i \in 2,\ldots,n$ no és possible. Per fer-ho utilitzem el Lema 4.5.1 que ens diu que en aquest cas per tot i,j $\mathrm{cr}(w_i)+\mathrm{cr}(w_j)-\mathrm{cr}(w_i,w_j)$ serà parell. Com que tots els $\mathrm{cr}(w_i)$ són parells tenim que $\mathrm{cr}(w_i,w_j)$ serà sempre parell.

Veiem que $\mathrm{cr}(w_i,w_j) \neq 0$ per tot i,j. Si hi hagués algun cas que no fos així, podríem aplicar el Lema 4.4.2 per $m=5$. Com que n és senar obtindríem la següent desigualtat:

$$\begin{aligned}\mathrm{cr}(R) &\geq \mathrm{cr}(K_{5,n-2}) + (n-2)\,\mathrm{cr}(K_{5,3}) = 4 \cdot \frac{n-3}{2}\,\frac{n-3}{2} + 4(n-2) \\ &= n^2 - 6n + 9 + 4n - 8 = 4 \cdot \frac{n^2-2n+1}{4} = 4 \cdot \frac{(n-1)^2}{4} = \mathrm{Y}(5,n).\end{aligned}$$

Això contradiria al fet que $\mathrm{cr}(D) = \mathrm{Y}(5,n) - 2$.

Hem vist que no hi ha cap $\mathrm{cr}(w_i, w_j) = 0$, per tant, tots serien com mínim 2 ja que són parells i tindríem que:

$$\mathrm{cr}(D) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathrm{cr}(u_i, u_j) \geq 2\binom{n}{2} > \mathrm{Y}(5, n) - 2.$$

Possibilitat 2

Que $\mathrm{cr}(w_1) = \mathrm{cr}(w_2) = 2n - 5$ i $\mathrm{cr}(w_i) = 2n - 4$ per $i \in 2, \ldots, n$ tampoc és possible, ho veiem a continuació. Per començar utilitzem el Lema 4.5.1 d'on trèiem que $\mathrm{cr}(w_i, w_j)$ ha de ser parell per $i = 1; j = 2$ i per $3 \geq i < j \geq n$ i senar per $i = 1, 2; 3 \leq j \leq n$. Imposem les igualtats següents:

$$\mathrm{cr}(D) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathrm{cr}(u_i, u_j).$$
$$\mathrm{cr}(w_i) = \sum_{1 \geq j \geq n, j=i} \mathrm{cr}(u_i, u_j).$$

Trobem que els valors de $\mathrm{cr}(w_i, w_j)$ han de valer el següent:

$$\begin{aligned} \mathrm{cr}(w_i, w_j) &= 2 \qquad \text{si } 3 \geq i < j \geq n. \\ \mathrm{cr}(w_i, w_j) &= 2 \qquad \text{si } i = 1, 2;\ 3 \geq j \geq n. \\ \mathrm{cr}(w_1, w_2) &= n - 3. \end{aligned}$$

Per veure que aquestes condicions no són possibles eliminarem el vèrtex w_1 de R, obtenint una representació R' d'un $K_{5,n-1}$. A continuació generarem un altre cop una representació R'' d'un $K_{5,n}$ mitjançant la construcció donada en la Observació 4.5.2 i veurem que $\mathrm{cr}(R) > \mathrm{cr}(R'')$ en els casos $n > 7$. Utilitzant aquella construcció posem el $w_{1'}$ en un entorn del w_2 i afegirem les arestes de tal manera que en R'' acabarem tenint que:

$$\begin{aligned} \mathrm{cr}(w_i, w_j) &= 2 \qquad \text{si } 3 \geq i < j \geq n. \\ \mathrm{cr}(w_i, w_j) &= 2 \qquad \text{si } i = 1', 2;\ 3 \geq j \geq n. \\ \mathrm{cr}(w_{1'}, w_2) &= 4. \end{aligned}$$

En els casos en que $n > 7$ les dues primeres igualtats dels valors de $\mathrm{cr}(w_i, w_j)$ de R són iguals que les de R'', però $\mathrm{cr}(w_1, w_2) > \mathrm{cr}(w_{1'}, w_2)$, per tant, $\mathrm{cr}(R) > \mathrm{cr}(R'')$. Els casos en que $n = 5, 7$ s'han de treballar més.

Per $n = 5$ utilitzem el Lema 4.5.6. Considerant el valors de $\mathrm{cr}(w_i, w_j)$ tenim que $\chi_{i,j} \leq 2$ si $i, j \in \{3, 4, 5\}$, $\chi_{i,j} \leq 1$ si $i \in \{1, 2\}$ $j \in \{3, 4, 5\}$ i $\chi_{1,2} \leq 2$. Si per alguna $j \in \{3, 4, 5\}$ $\chi_{1,j} = 0$ llavors la primera condició no es podria satisfer, per tant, $\forall j$ $\chi_{1,j} = 1$ i per simetria $\chi_{1,j} = 1$.

Sense pèrdua de generalitat podem considerar que $\pi_1 = (12345)$, per tant, tot $\overline{\pi_j}$ per $j \in \{3, 4, 5\}$ ha d'estar entre els següents:

$$(21345), (13245), (12435), (12354), (52341).$$

Imposant que hi hagi un π_2 tal que $\chi_{2,j} = 1$ ens resulta que $\pi_2 = \pi_1$. Aquest fet implica $\text{cr}(w_1, w_2) = 4$ i és dos, per tant, hem arribat a contradicció en tots els casos per $n = 5$.

Per $n = 7$ s'acaba obtenint que $\chi_{i,j} = 2$ si $i, j \in \{3, 4, 5, 6, 7\}$, $\chi_{i,j} = 1$ si $i \in \{1, 2\}$ $j \in \{3, 4, 5\}$ i $\chi_{1,2} = 4$. Més en particular, i considerant una ordre particular les vèrtexs que no ens fa perdre generalitat, tenim que $\pi_1 = \pi_2 = (12345), \overline{\pi_3} = (21345), \overline{\pi_4} = (13245), \overline{\pi_5} = (12435), \overline{\pi_6} = (12354), \overline{\pi_7} = (52341)$.

Eliminant w_1 veurem que no podem construir un $K_{5,6}$ amb aquests requisits. Mostrem dues parts de la representació del $K_{5,6}$ que exemplifiquen la resta. El primer exemple en la Figura 4.4, on els únics vèrtexs de W que hi ha són els w_2, w_3 i, el segon en la Figura 4.5, on els únics vèrtexs de W són els w_2, w_6. Sabem que seran d'alguna d'aquestes formes llevat possibles moviments dels punts sobre el pla, o sigui que considerant les dues opcions de cada una no perdem generalitat. Veurem que considerant aquestes dues i les 3 que falten es generen incompatibilitats.

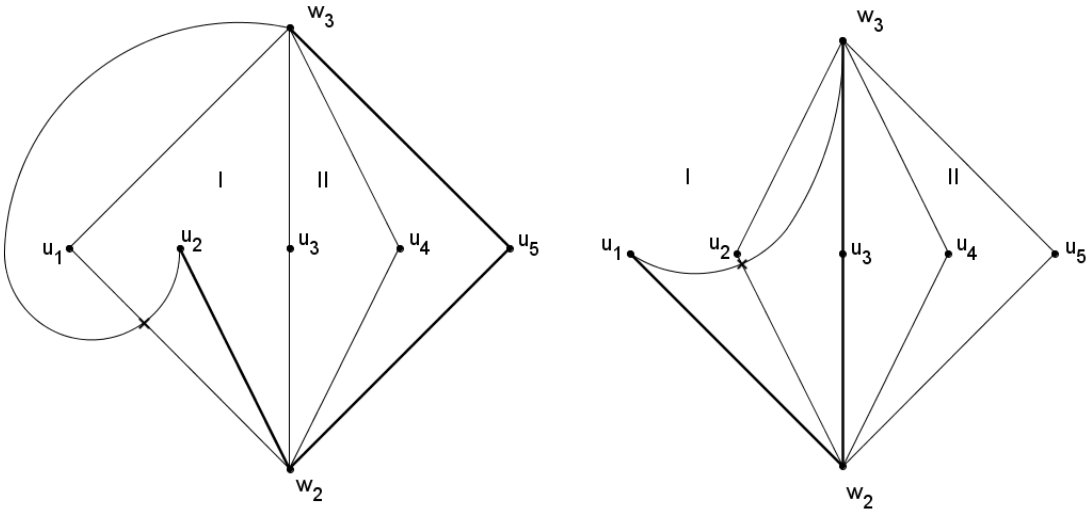

Figura 4.4: Part de la representació del $K_{5,6}$ on hi figuren els vèrtexs w_2, w_3.

En els gràfics hi ha més elements que simples vèrtexs i arestes. Per començar es mostren en quines cares poden anar els altres vèrtexs, marcades amb I, II. Aquest fet és evident, ja que si no anessin en aquestes cares llavors no és podria complir que $\text{cr}(w_2, w_j) = 1, \text{cr}(w_i, w_j) = 2$ per $i = 3$ i $j = 4, 5, 6, 7$ en el primer gràfic i $i = 6$ i $j = 3, 4, 5, 7$ en el segon per cap j. A més, en els dos gràfics hi ha dos tipus d'arestes i s'indiquen si podran ser tallades per d'altres sortints dels w_j restants o no. Les arestes que poden ser tallades són fines, i les que no, gruixudes. Aquest últim fet es veu també imposant els valors de $\text{cr}(w_i, w_j)$ i considerant els possibles dibuixos d'un w_j sobre els gràfics.

Indiquem en cada un dels 10 casos, 2 per w_j, quina és l'aresta $\overline{w_2 u_i}$ que tallen i quines dues resten no podran ser tallades per els altres w_k $k \geq 3k \neq j$.

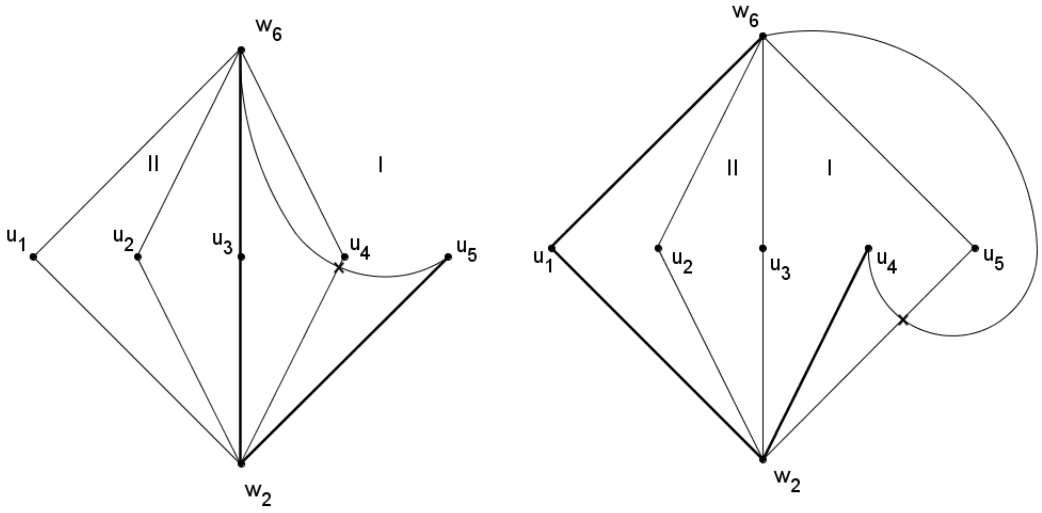

Figura 4.5: Part de la representació del $K_{5,6}$ on hi figuren els vèrtexs w_2, w_6.

Vèrtexs	Cas	Es talla amb	La resta no es tallaran amb
w_3	1	$\overline{w_2u_1}$	$\overline{w_2u_5}, \overline{w_2u_2}$
	2	$\overline{w_2u_2}$	$\overline{w_2u_1}, \overline{w_2u_3}$
w_4	1	$\overline{w_2u_2}$	$\overline{w_2u_1}, \overline{w_2u_3}$
	2	$\overline{w_2u_3}$	$\overline{w_2u_2}, \overline{w_2u_4}$
w_5	1	$\overline{w_2u_3}$	$\overline{w_2u_2}, \overline{w_2u_4}$
	2	$\overline{w_2u_4}$	$\overline{w_2u_3}, \overline{w_2u_5}$
w_6	1	$\overline{w_2u_4}$	$\overline{w_2u_3}, \overline{w_2u_5}$
	2	$\overline{w_2u_5}$	$\overline{w_2u_4}, \overline{w_2u_1}$
w_7	1	$\overline{w_2u_5}$	$\overline{w_2u_4}, \overline{w_2u_1}$
	2	$\overline{w_2u_1}$	$\overline{w_2u_5}, \overline{w_2u_2}$

Ara veiem la contradicció. Considerant el cas 1 de w_3 ens veiem obligats a escollir el cas 2 de w_4, el cas 1 de w_5 i en el w_6 tots dos casos són incompatibles amb els anteriors. Considerant el cas 2 de w_3 ens veiem obligats a escollir el cas 1 de w_4, el cas 2 de w_5, el cas 1 de w_6 i en el w_7 tots dos casos són incompatibles amb els anteriors.

Per tant, hem acabat la demostració per tots els n senars i tenim que $\mathrm{cr}(K_{5,n}) = \mathrm{Y}(5,n)$.

Veure que $\mathrm{cr}(K_{m,n}) = \mathrm{Y}(m,n)$ per $m = 6$ és directe a partir del fet que coneguem que es cert per $m = 5$ i del Teorema 4.3.2. □

Corol·lari 4.5.8. *Com que coneixem que* $\forall n\, \mathrm{cr}(K_{5,n}) = \mathrm{Z}(5,n)$ *immediatament pel Teorema 4.3.2 podem dir que* $\forall n\, \mathrm{cr}(K_{6,n}) = \mathrm{Z}(6,n)$.

4.6 Nombre de talls de $K_{7,n}$ i $K_{8,n}$ on $n \in \{7, 8, 9, 10\}$

Definició 4.6.1. *Definim l'**enèsim graf d'ordre cíclic**, CO_n, com un graf amb $(n-1)!$ vèrtexs. Aquests els anomenem amb una de les permutacions dels elements de $(0, 1, 2, \ldots, n-1)$ llevat de la següent equivalència $(i, j, \cdots, k, l, m) = (m, i, j, \cdots, k, l)$ $\forall i, j, \cdots, k, l, m \in \{0, 1, 2, \ldots, n-1\}$ diferents. En general, escriurem el 0 com a primer valor del nom del vèrtex, però realment no hi ha un nombre que sigui el primer perquè el nom és cíclic.*

Les arestes les definim a partir del nom dels vèrtexs. Dos vèrtexs tindran una aresta que els uneix si substituint un element del nom d'un dels vèrtexs s'obté el nom de l'altre. Veiem un exemple del CO_4 on s'indica el nom dels vèrtexs i per cada aresta els valors que s'han d'intercanviar en els noms dels vèrtexs adjacents per tal de que el vèrtex s'hagi de generar.

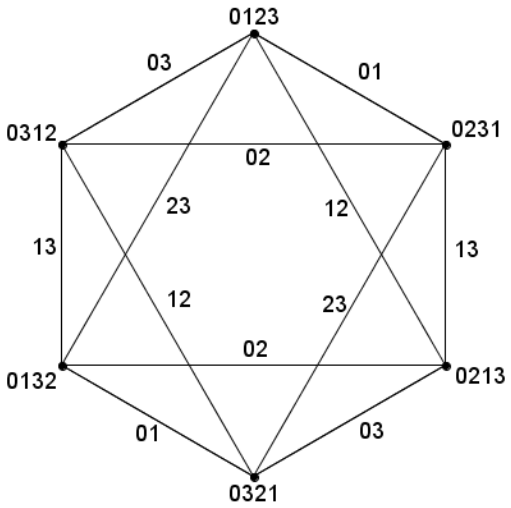

Figura 4.6: Representació del CO_4.

Notació 4.6.2. Donada un vèrtex v de CO_n denotem $\overline{v}$ com l'expressió amb ordre invers del nom de v. $\overline{v}$ també és la representació d'un altre vèrtex w i quan sigui necessari ens hi referim pròpiament.

Definició 4.6.3. *Siguin v, w dos vèrtexs de CO_n diferents. Definim $\phi_{i,j}$ com el mínim nombre d'intercanvis d'elements adjacents que s'han d'aplicar a v per obtenir $\overline{w}$. Hem de tenir en compte v i w en totes les seves formes variants. Aquesta definició és igual a la Definició 4.5.5 ja que la manera de notar en aquell cas les permutacions d'elements és la mateixa que notem ara els vèrtexs de CO_4, per tant, certs resultats ja vists en aquell cas els podrem aplicar.*

Proposició 4.6.4. *Pels n senars CO_n és un graf bipartit.*

Demostració. Per començar, observem que els $n!$ possibles noms dels vèrtexs tenen permutacions parelles o senars respecte elles i $(0,\ldots,n-1)$. És a dir, que o bé necessiten parells intercanvis entre elles i $(0,\ldots,n-1)$ o senars, respectivament.

Sabem que hi ha n permutacions que representen el mateix vèrtex. Per passar d'una a l'altra es necessitaran $n-1$ intercanvis i com que n és senar, tindrem parells intercanvis, doncs serà una permutació parella. Per veure-ho posem un exemple, suposem que volem desplaçar tota una permutació dos elements enrere. Ho aconseguim intercanviant amb l'ordre indicat el primer element amb el tercer, el tercer amb el cinquè i, així successivament, fins fer-ne $n-1$. Veiem-ho per $n=5$:

$$(1,2,3,4,5) \to (3,2,1,4,5) \to (3,2,5,4,1) \to (3,1,5,4,2) \to (3,4,5,1,2)$$

D'aquesta manera es veu que per passar d'una permutació parella a una senar i viceversa, sempre necessitaràs un nombre senar d'intercanvis i per passar entre dues de la mateixa paritat, en caldran parells. Per tant, tindrem dos subconjunts de vèrtexs un amb les permutacions parelles i un altre amb les senars. □

Lema 4.6.5. *Tota representació R de $K_{m,n}$ conté una representació de $K_{m-1,n}$ amb com a molt* $\mathrm{cr}(R)(m-2)/m$ *talls.*

Demostració. Aquest resultat és una generalització del Lema 4.3.1 i per aquest motiu la demostració és similar. Com en aquell cas, l'únic que hem de considerar és que en $K_{m,n}$ hi ha m representacions de $K_{m-1,n}$ i cada tall de $K_{m,n}$ es troba $m-2$ cops en les representacions de tots els $K_{m-1,n}$. □

Definició 4.6.6. *Donat un graf CO_n, k vèrtexs $u_1,\ldots,u_k$ de la component U i l vèrtexs $w_1,\ldots,w_l$ de la component W prefixats, notarem com* $\mathrm{cr}(u_1\ldots u_k)$ *al nombre de talls que hi ha al subgraf de CO_n amb les arestes u_i i w_j.*

Teorema 4.6.7. $\mathrm{cr}(K_{7,7}) = \mathrm{Z}(7,7) = 81$.

Demostració. Aquesta demostració es basa en un algoritme informàtic que ens dóna totes les opcions. Perquè el temps d'execució sigui raonable s'han d'aplicar certes condicions.

Per començar, considerem que existeix una representació de $K_{7,7}$ amb menys de 81 talls a CO_7. Diem u_i $i \in 1,\ldots,7$ als vèrtexs de U de $K_{7,7}$ els quals es trobaran tots en U_{CO_7} o en W_{CO_7}, però no poden estar repartits. A continuació, aplicant el Lema 4.6.5, el Teorema 4.2.2 i considerant que els u_i els podem escollir en l'ordre que vulguem s'obtenen les següents desigualtats:

$$\begin{array}{lll} \mathrm{cr}(u_1\ldots u_7) \le 79 & \mathrm{cr}(u_1\ldots u_6) \le 56 & \mathrm{cr}(u_1\ldots u_5) \le 36 \\ \mathrm{cr}(u_1\ldots u_4) \le 21 & \mathrm{cr}(u_1u_2u_3) \le 9 & \mathrm{cr}(u_1u_2) \le 3. \end{array}$$

L'algoritme intentarà construir el $K_{7,7}$ desitjat imposant aquests fets. Per començar, utilitzarà que $\mathrm{cr}(u_1u_2) \le 3$, fet que eliminarà moltes possibilitats, a continuació, amb les que restin que $\mathrm{cr}(u_1u_2u_3) \le 9$, i així successivament fins veure que cap representació pot assolir que $\mathrm{cr}(u_1\ldots u_7) \le 79$. □

Corol·lari 4.6.8. *Directament a partir del fet que* $\mathrm{cr}(K_{7,7}) = \mathrm{Z}(7,7) = 81$ *i el Teorema 4.3.2 obtenim que* $\mathrm{cr}(K_{7,8}) = \mathrm{Z}(7,8) = 108$ *i* $\mathrm{cr}(K_{8,8}) = \mathrm{Z}(8,8) = 144$. *Veiem una representació del* $K_{7,8}$ *en la Figura 4.7.*

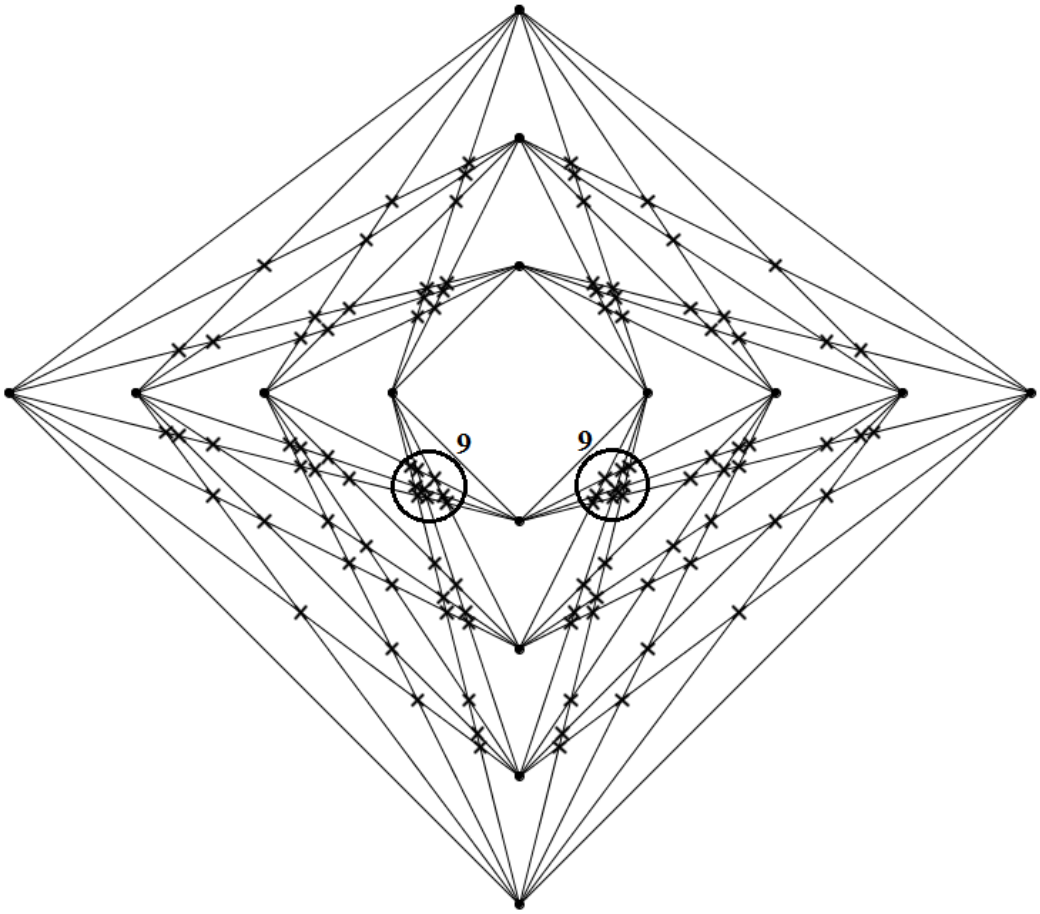

Figura 4.7: Una representació òptima del $K_{7,8}$.

Observació 4.6.9. A partir d'algoritmes semblants al utilitzat també es poden trobar el nombre de valors de representacions no isomòrfiques de $K_{5,5}$ i $K_{5,7}$ que assoleixen $\mathrm{Z}(m,n)$ que són 32 i 96821, respectivament. És clar que també les deuen haver obtingut d'altres grafs amb inferior nombre de vèrtexs, però tanmateix no ho denoten.

Teorema 4.6.10. $\mathrm{cr}(K_{7,9}) = \mathrm{Z}(7,9) = 144$.

Demostració. Aquesta demostració és també conseqüència d'aplicar certs algoritmes i resultats sobre el $K_{7,9}$ semblants als del Teorema 4.6.10. Com s'ha comentat amb anterioritat els podem trobar en l'article [17]. □

Corol·lari 4.6.11. *Directament a partir del fet que* $\mathrm{cr}(K_{7,9}) = \mathrm{Z}(7,9) = 144$ *i el Teorema 4.3.2 obtenim que* $\mathrm{cr}(K_{7,10}) = \mathrm{Z}(7,10) = 180$*,* $\mathrm{cr}(K_{7,10}) = \mathrm{Z}(8,9) = 192$ *i* $\mathrm{cr}(K_{7,10}) = \mathrm{Z}(8,10) = 240$*.*

4.7 Nombre de talls de $K_{9,9}$ i $K_{9,10}$

Teorema 4.7.1. $\mathrm{cr}(K_{9,9}) = \mathrm{Z}(9,9) = 256$*.*

Observació 4.7.2. Podem trobar les demostracions d'aquest fet en dos articles diferents.

El primer, de Vassilevska, data del 2002 [14] i es basa en una adaptació dels programes de Woodall [17]. Veure que no hi havia cap cas que millorava la cota va requerir 123 dies d'execució en 30 CPU. Aquest article no ha estat publicat en cap revista que nosaltres hàgim trobat.

El segon, el devem a Farahani i s'ha publicat aquest mateix 2013 [2]. Es basa en construir els grafs $K_{j,j}$ a partir de matrius que s'hi corresponen. Finalment, es veu que $\mathrm{cr}(K_{9,9}) = 256$.

Corol·lari 4.7.3. *Directament a partir del fet que* $\mathrm{cr}(K_{9,9}) = \mathrm{Z}(9,9) = 256$ *i el Teorema 4.3.2 obtenim que* $\mathrm{cr}(K_{9,10}) = \mathrm{Z}(7,10) = 360$ *i* $\mathrm{cr}(K_{10,10}) = \mathrm{Z}(8,9) = 400$*.*

Capítol 5

Algoritmes

Com s'ha pogut observar en els capítols anteriors en les demostracions pels casos més complexes intervenen algoritmes informàtics. Nosaltres hem creat uns algoritmes propis que es troben recollits en l'Annex 7 els quals busquen solucionar el problema dels nombre de talls dels grafs complets. Per fer-ho, en centrem en els casos en què els grafs complets K_n contenen un subgraf cíclic C_n dins seu, les arestes del qual no es tallen amb cap tall del graf complet.

Presentem dos programes, el primer comprova que una construcció determinada dels K_n tinguin el nombre de talls igual que $\mathrm{Z}(n)$ i el segon ens dóna totes les representacions d'un K_n que compleixen la premissa anteriorment dita i assoleixin $\mathrm{Z}(n)$. Malauradament, aquest segon programa només comprova les opcions amb un temps raonable fins a $n = 11$.

Els dos els trobem en l'Annex 7.

5.1 Idea original i evolució dels programes

La idea original pel programa la devem a Alejandro de Miquel i era disposar els vèrtexs en una línia recta, ens els podem imaginar sobre l'eix $y = 0$ del pla. Llavors, les arestes fer-les passar de tal forma que si els vèrtexs que unia eren contigus aquesta passava per l'eix $y = 0$ i si no ho eren aquesta podia passar per les quatre opcions que veiem en la Figura 5.1, la opció 0 per dalt, la opció 1 per baix, la opció 2 sobrepassant tot el graf passant per l'inici o la opció 3 passant pel final. Aquest fet feia que no acabés de quedar un C_n sense talls però finalment veurem que s'anul·laran les opcions 2 i 3 i ja queda el C_n.

Imposant aquestes premisses obtenim que cada K_n té $(n-1) * (n-2)/2$ arestes amb 4 opcions. Les diferents opcions les agrupem en una matriu de nombres. Aquesta matriu tindrà $n-2$ files que les anomenarem com $2, 3, \ldots, n-1$; i $i-1$ columnes $0, 1, \ldots, i-2$ per cada fila i. En veiem una exemple expandit de matriu d'un graf amb 8 vèrtexs en la Figura 5.2. En aquesta matriu els únics valors que realment ens interessen són els números ja que els ens llocs on hi trobem una - corresponen a les arestes que estan a l'eix $y = 0$, les X corresponen a teòriques arestes que van de cada vèrtex a si mateixos les quals no existeixen i els valors dels * ja els

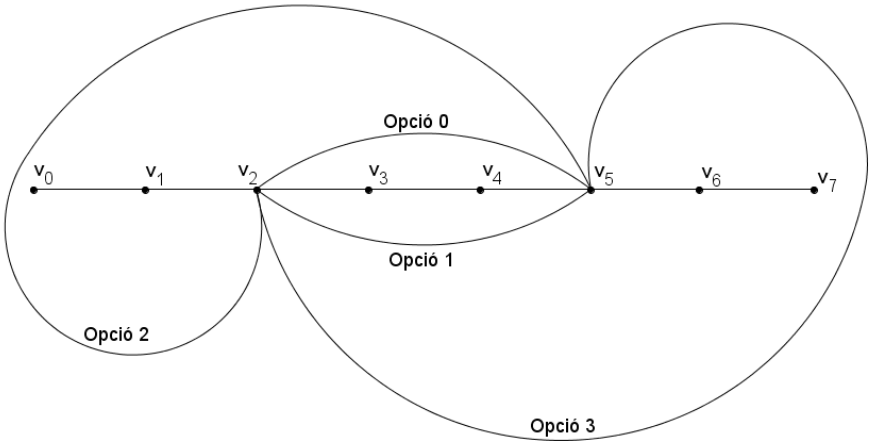

Figura 5.1: Exemple de les quatre opcions d'una aresta.

vèrtex destí	vèrtex origen							
	0	1	2	3	4	5	6	7
0	X	-	*	*	*	*	*	*
1	-	X	-	*	*	*	*	*
2	1	-	X	-	*	*	*	*
3	3	0	-	X	-	*	*	*
4	3	3	2	-	X	-	*	*
5	0	1	0	3	-	X	-	*
6	2	0	0	1	0	-	X	-
7	3	0	1	1	1	1	-	X

Figura 5.2: Exemple de taula en el cas de 8 vèrtexs.

tenim perquè el graf ha de ser simètric. Per aquest motiu en el programa només ens quedem amb la matriu triangular de $(n-1)*(n-2)/2$ valors anteriorment dita.

Veient que era impossible avaluar d'una forma eficient totes les opcions si augmentàvem el nombre de vèrtexs vam fer un programa que ens busqués les que menys talls tenien d'una forma ràpida. Aquest algoritme li donaves un graf i avaluava totes les opcions que es diferenciaven d'aquest primer per una aresta i es quedava amb la opció que menys vèrtexs tenia. Iterant aquest fet vam obtenir uns resultats sorprenents ja que tots els grafs que trobava que assolien la cota conjecturada únicament tenien les opcions 0 i 1. Es van aplicar per $n \leq 40$, punt on el programa començava a alentir-se un mica, tanmateix, si hagués sigut necessari més resultats se'n podien trobar més.

En aquests resultats s'observava que l'estructura dels grafs seguia una certa simetria, fet va fer pensar que podríem trobar resultats satisfactoris per tot n. D'aquí neix l'Algoritme 7.1 que pot avaluar per tot n un graf determinat.

Per altra banda, en els programes es va eliminar la possibilitat de les opcions 2 i 3, fet que va reduir el nombre de casos a l'arrel dels inicials i ha acabat amb el programa l'Algoritme 7.3.

5.2 Parts comuns dels algoritmes

Com es pot observar en els algoritmes, ambdós basen la seva estructura en la *class edge*. Definim una matriu d'arestes en que cada un dels seus elements té una sèrie de component. Les comuns són:

- Identificadors que ens determinen de quin vèrtex a quin altre va l'aresta.
- Una variable booleana que ens informa de quina opció, la 0 o la 1, agafa l'aresta.
- El nombre de talls de cada aresta i uns punters que es dirigeixen a les altres arestes que es tallen amb la que estem treballant. Aquests punters fan una feina molt important en el segon algoritme reduint el temps d'execució del programa.

A més, en el segon programa també tenim uns altres punters que ordenen les arestes per files o per columnes, fet que ens proporciona un augment de la eficiència del d'aquest ja que no perdem temps al canviar d'aresta. En el primer també es podria implementar, però no és necessari ja que l'algoritme ja és prou eficient i tampoc milloraria en excés.

Altres parts comuns dels algoritmes són les funcions *creategraph*, *ncutall*, *adjcut* del primer amb *adjecencygraph2* del segon i *printgraph*. Que siguin semblants no vol dir que siguin intercanviables ja que cada una està adaptada al seu programa, però sí que l'escènica del seu treball és el mateix.

- *creategraph* és una funció que inicialitza el graf, donant els valors que pertoquen a les components de cada aresta en funció de les necessitats.
- *ncutall* conta el nombre de talls de cada aresta i els posa en la variable *e_nc* respectivament.
- *adjcut* i *adjecencygraph2* són dues funcions que fan el mateix servei cadascuna en el seu algoritme, unir les arestes que es tallen mitjançant punters.
- Finalment, *printgraph* imprimeix els resultats en un fitxer en funció de les necessitats.

5.3 Particularitats de l'algoritme I

Bona part el programa ja està explicada i només falta veure com es relacionen tots conceptes. Per fer-ho es còmode seguir la funció *main*, la qual és la directora

del programa. Aquesta comença interactuant amb l'usuari, reservant la memòria necessària i creant el graf en *creategraph*. Es pot veure en la funció el criteri que es fa servir per associar la opció 0 o 1 a la aresta el qual l'expliquem a continuació. Dient n al nombre de vèrtexs del graf i i i j als vèrtexs de les arestes tenim que aquesta escollirà la opció 0 si $(i+j) < \lfloor \frac{n}{2} \rfloor + 2$ o $n+1 < (i+j) < \lfloor \frac{3n}{2} \rfloor + 2$ i la opció 1 altrament. A continuació, el programa calcula el nombre de talls, i les adjacències si l'usuari així ho vol i finalment dóna els resultats. S'ha provat el programa fins a $n = 250$ i mai ha fallat. Tanmateix, donada l'eficiència del programa, el cas $n = 250$ s'obté amb menys de 10 segons, i el fet que el programa augmenti la seva complexitat linealment indica que es podria provar per molts més valors, però creiem que ja és un resultat suficient.

Podem veure com queda la matriu i les adjacències per casos petits en la Secció 7.2. Si es volen més resultats sempre es pot utilitzar el programa que està en la Secció 7.1 tenint en compte que no es recomana demanar la informació d'adjacències pels casos $n > 50$ ja que hi ha molta informació i es generarà un fitxer de text molt gran.

Temps després d'obtenir els resultats vam veure que aquests grafs eren els mateixos que els que es definien en [1]. Com s'ha demostrat en el Teorema 3.1.1 aquestes representacions obtenen que per tot n $\mathrm{Z}(n) = \mathrm{cr}(R)$.

5.4 Particularitats de l'algoritme II

Aquest segon algoritme és considerablement més complex que l'anterior degut a que calcula totes les possibles configuracions matricials. Exactament pel cas de n vèrtexs tenim que s'avaluen $2^{(n-2)(n-1)/2-2}$ casos ja que la matriu té $(n-2)(n-1)/2$ arestes, una de les quals mai té talls, ja que és la que tanca el C_n, i un altre la podem considerar que pren la opció 1 sense pèrdua de generalitat. Per començar, explicarem les funcions que encara no s'han explicat.

- La funció *adjencygraph1* ens relaciona cada vèrtex amb el seu següent, tant sigui per files com per columnes. D'aquesta manera el programa resulta més eficient.

- La funció *ncut* rep un vèrtex i ens retorna el nombre de talls que hi ha en el graf si aquell vèrtex va amb la opció 1 menys el nombre de talls si va amb opció 0. Aquest resultat no s'obté d'aquest càlcul explicit sinó del fet que sabem que la suma dels dos valors anteriors és una constant coneguda que no varia en funció de les opcions de les altres arestes.

- L'última funció especial és *recursive*. Aquesta funció recursiva és la que ens ajuda a passar per totes les opcions de forma ordenada. A més quan detecta que un graf assoleix la cota *printfile* perquè el posi en les solucions.

Un cop tenim totes les funcions externes només falta el *main*. Aquest torna a distribuir les tasques d'una forma semblant. Tanmateix, cal destacar que donat que

aquest algoritme és més costós és interessant saber quan tarda a recórrer tots els casos i per això donem el temps que tarda. En referència en aquest fet s'ha avaluat el cas $n = 10$ en un PC amb processador a 3.1 Ghz i s'ha obtingut un resultat final en 651 segons. Donat que el nombre de casos es multiplica per $2^{(n-2)}$ del pas n al cas $n + 1$ extrapolem que el cas $n = 11$ tardaria uns 4 dies i que el cas $n = 12$ surt fora de les nostres possibilitats. No hem calculat el cas per $n = 11$ ja que no considerem que ens pugi donar uns resultats rellevants.

5.5 Resultats de l'algoritme II

Trobem els resultats executar el programa per diversos valors de n en la Secció 7.4. Mitjançant aquest algoritme es troben representacions isomorfes múltiples vegades. Tanmateix tret del en que $n = 9$ podrem determinar quines representacions són no isomòrfiques entre ells.

Per començar es pot observar clarament que el cas de 7 vèrtexs té 3 representacions no isomòrfiques com a mínim de les 35 trobades. Aquesta conclusió es pot treure del fet que el graf 1 té 3 talls en un costat del C_7 i 6 en l'altre, el graf 2 en té 2 i 7 i el graf 3 en té 4 i 5. Ara bé, pel Teorema 3.4.4 sabíem hi havia 5 representacions no isomòrfiques de les quals només 3 contenen un C_7 sense talls, per tant podem determinar que exactament n'hi ha 3.

Per $n = 9$ passa el mateix i en aquest cas pel mateix motiu podem veure que com a mínim n'hi haurà 4. En aquest cas com que no tenim les 3080 no sabem si en tindrem només 4 o en podem trobar més.

Pels casos parells coneguts $n = 4, 6, 8, 10$ la cosa canvia. Per 4 i 6 ja sabíem que només hi hauria una opció no isomòrfica ja que hem vist que únicament n'hi ha una en general. Alhora, pels caos 8 i 10, que a priori no sabíem si succeiria el mateix que en els que precedeixen, veiem que realment tornem a tenir només una opció. El fet es pot observar dibuixant els 4 resultats per $n = 8$ i els 5 per $n = 10$ i veient que són representacions isomòrfiques. O sigui que concloem que de les 3 representacions no isomòrfiques del K_8 només una conté un C_8 que no es talla dins seu i de les 5679 representacions no isomòrfiques del K_{10} també només una conté un C_{10}.

Aquest fet ens obre una pregunta, succeirà el mateix per $n = 12$? Jo personalment, pels resultats que he obtingut mitjançant els programes de busca eficient de solucions crec que sí i espero millorar aquest algoritme per poder-ho demostrar.

Capítol 6

Conclusions

Un cop finalitzat el projecte es pot concloure que conèixer el nombre de talls d'un graf és un problema molt complex i que no té una manera sistemàtica de ser resolt eficientment. Com es pot veure en el treball les demostracions dels casos més grans es basen en algoritmes que calculen totes les possibles distribucions i això és un fre en els casos molt grans. Aquest fet fa que sembli impossible trobar a curt termini una solució general pel problema ja que els algoritmes avancen molt lentament.

A més, durant el treball s'ha pogut comprovar que dels nostres algoritmes en resulten grafs que s'han construït en anterioritat per altres matemàtics, fet que denota que podem estar en el camí correcte. Centrant-nos en els nostres algoritmes crec que encara tenen opcions d'evolucionar ja que actualment s'hi consideren tots els casos i no és necessari.

Més en general, tot i que no podria assegurar com continuaran els avenços ja que hi ha diverses investigacions ara mateix crec que és necessari trobar una manera d'eliminar nombres molt grans de casos ja que en cas contrari em sembla impensable poder millorar en facilitat els resultats actuals.

Capítol 7

Annex

En els annexos trobem dos programes explicats en el Capítol 5 i part dels seus resultats.

7.1 Algoritme I

Aquest algoritme és generador de grafs complets amb nombre de talls Z(n).

```
#include <iostream>
#include <time.h>
#include <fstream>
#include <sstream>

using namespace std;

class edge{
  public:
    //identificador ME*original vertex + final vertex
    int e_id;
    //original vertex
    int e_ov;
    //final vertex
    int e_fv;
    //side of the vertex
    bool e_side;
    //edges cuts
    edge **e_ecu;
    //number of cuts
    int e_nc;
    //number of cuts couter
    int e_ncc;
};
```

```
void creategraph(edge**);
void ncutall(edge**);
void adjcut(edge**);
void printfile(edge**, int);

int V, A, ME=1000;
bool flag;

int main(){
  edge **graph1;
  int i, j, end, cut;
  cout << "Number of vertixes: ";
  cin >> V;
  cout << "If you want the adjency information (more computing time)
return 1." << endl;
  cout << "If you only want our graph and to check it return 0." << e
ndl;
  cin >> flag;
  A=(int)(V/2)*(int)((V-1)/2)*(int)((V-2)/2)*(int)((V-3)/2)/4;
  graph1=new edge*[V];
  for(i=2;i<V;i++)
    graph1[i]=new edge[i-1];
  creategraph(graph1);
  ncutall(graph1);
  cut=0;
  for(i=2; i<V; i++){
    for(j=0; j<i-1; j++){
      cut+=graph1[i][j].e_nc;
      graph1[i][j].e_ecu=new edge*[graph1[i][j].e_nc];
    }
  }
  cut/=2;
  if (cut != A)
    cout << "Sorry, not fouted the right solution for " << V << " ver
texes" << endl;
  if (flag==1){
    adjcut(graph1);
  }
  printfile(graph1, cut);
  cout << "Written the solution in \"Solucion_of_" << V << "_vertices
_" << flag << ".txt\"" << endl;
  for(i=2;i<V;i++)
    free (graph1[i]);
```

```
  free (graph1);
  system("Pause");
  return 0;
}

void creategraph(edge** graph){
  for(int i=2; i<V; i++){
    for(int j=0; j<i-1; j++){
      graph[i][j].e_ov=i;
      graph[i][j].e_fv=j;
      graph[i][j].e_id=ME*i+j;
      graph[i][j].e_nc=0;
      graph[i][j].e_ncc=0;
      if( ((i+j))<(int)(V/2)+2 || (((i+j))>V+1 && ((i+j))<(int)(3*V/2
)+2))
        graph[i][j].e_side=0;
      else
        graph[i][j].e_side=1;
    }
  }
  return;
}

void ncutall(edge **graph){
  int cut=0;
  for(int i1=0; i1<V-2; i1++){
    for(int j1=i1+2; j1<V; j1++){
      for(int i2=i1+1; i2<V-2; i2++)
        for(int j2=i2+2; j2<V; j2++)
          if(graph[j1][i1].e_side==graph[j2][i2].e_side && j1>i2 && j
1<j2){
            graph[j1][i1].e_nc++;
            graph[j2][i2].e_nc++;
          }
    }
  }
  return;
}

void adjcut(edge **graph){
  for(int i1=0; i1<V-2; i1++){
    for(int j1=i1+2; j1<V; j1++){
      int k=0;
      for(int i2=i1+1; i2<V-2; i2++)
```

```
        for(int j2=i2+2; j2<V; j2++){
          if(graph[j1][i1].e_side==graph[j2][i2].e_side && j1>i2 && j
1<j2){
            graph[j1][i1].e_ecu[graph[j1][i1].e_ncc++]=&graph[j2][i2]
;
            graph[j2][i2].e_ecu[graph[j2][i2].e_ncc++]=&graph[j1][i1]
;
          }
          if(graph[j1][i1].e_ncc==graph[j1][i1].e_nc){
            j2=V;
            i2=V-2;
          }
        }
    }
  }
  return;
}

void printfile(edge** graph, int cut){
  int numcut0=0;
  ostringstream name;
  name.flush( );
  name << "Solucion_of_" << V << "_vertices_"<< flag <<".txt";
  ofstream a_file (name.str().c_str());
  if ( !a_file.is_open() ){
    cout << "error printing file";
    exit (-1);
  }
  a_file << "Graph cuts: " << cut << endl;
  a_file << "Conj. cuts: " << A << endl << endl;
  for(int i=2; i<V; i++){
    for(int j=0; j<i-1; j++){
      a_file << graph[i][j].e_side;
      if(graph[i][j].e_side==0)
        numcut0+=graph[i][j].e_nc;
    }
    a_file << endl;
  }
  numcut0/=2;
  a_file << endl << endl;
  a_file << "Number of cuts in side 0: " << numcut0 << endl << "Numbe
r of cuts in side 1: " << A-numcut0 << endl;
  a_file << endl << endl;
  if (flag==1){
```

```
    for(int i=2; i<V; i++){
      for(int j=0; j<i-1; j++){
        a_file << "edge: " << graph[i][j].e_id << "  side: " << graph
[i][j].e_side << "  cuts: " << graph[i][j].e_nc;
        if(graph[i][j].e_nc!=0){
          a_file<< "  cuts with: ";
          for(int k=0; k<graph[i][j].e_nc; k++){
            a_file << " " << graph[i][j].e_ecu[k]->e_id;
          }
        }
        a_file << endl;
      }
    }
  }
  a_file.close();
  return;
}
```

7.2 Resultats algoritme I

Donem el resultat del programa si li demanem una representació pels grafs complets amb nombre de talls n tal que $4 \leq n \leq 12$. Aquests són els casos en que hem demostrat que $\mathrm{Z}(n) = \mathrm{cr}(K_n)$ en el Capítol 3.

$\underline{n = 4}$

```
Graph cuts: 0
Conj. cuts: 0

0
01

Number of cuts in side 0: 0
Number of cuts in side 1: 0

edge: 2000  side: 0  cuts: 0
edge: 3000  side: 0  cuts: 0
edge: 3001  side: 1  cuts: 0
```

$\underline{n = 5}$

```
Graph cuts: 1
Conj. cuts: 1
```

```
0
01
111

Number of cuts in side 0: 0
Number of cuts in side 1: 1

edge: 2000  side: 0  cuts: 0
edge: 3000  side: 0  cuts: 0
edge: 3001  side: 1  cuts: 1  cuts with:  4002
edge: 4000  side: 1  cuts: 0
edge: 4001  side: 1  cuts: 0
edge: 4002  side: 1  cuts: 1  cuts with:  3001
```

$\underline{n = 6}$

```
Graph cuts: 3
Conj. cuts: 3

0
00
011
1110

Number of cuts in side 0: 2
Number of cuts in side 1: 1

edge: 2000  side: 0  cuts: 1  cuts with:  3001
edge: 3000  side: 0  cuts: 0
edge: 3001  side: 0  cuts: 1  cuts with:  2000
edge: 4000  side: 0  cuts: 1  cuts with:  5003
edge: 4001  side: 1  cuts: 1  cuts with:  5002
edge: 4002  side: 1  cuts: 0
edge: 5000  side: 1  cuts: 0
edge: 5001  side: 1  cuts: 0
edge: 5002  side: 1  cuts: 1  cuts with:  4001
edge: 5003  side: 0  cuts: 1  cuts with:  4000
```

$\underline{n = 7}$

```
Graph cuts: 9
Conj. cuts: 9

0
00
```

```
011
1111
11100

Number of cuts in side 0: 2
Number of cuts in side 1: 7

edge: 2000  side: 0  cuts: 1  cuts with:  3001
edge: 3000  side: 0  cuts: 0
edge: 3001  side: 0  cuts: 1  cuts with:  2000
edge: 4000  side: 0  cuts: 1  cuts with:  6003
edge: 4001  side: 1  cuts: 3  cuts with:  5002 6002 5003
edge: 4002  side: 1  cuts: 1  cuts with:  5003
edge: 5000  side: 1  cuts: 2  cuts with:  6001 6002
edge: 5001  side: 1  cuts: 1  cuts with:  6002
edge: 5002  side: 1  cuts: 1  cuts with:  4001
edge: 5003  side: 1  cuts: 2  cuts with:  4001 4002
edge: 6000  side: 1  cuts: 0
edge: 6001  side: 1  cuts: 1  cuts with:  5000
edge: 6002  side: 1  cuts: 3  cuts with:  5000 4001 5001
edge: 6003  side: 0  cuts: 1  cuts with:  4000
edge: 6004  side: 0  cuts: 0
```

$\underline{n = 8}$

```
Graph cuts: 18
Conj. cuts: 18

0
00
001
0111
11110
111000

Number of cuts in side 0: 9
Number of cuts in side 1: 9

edge: 2000  side: 0  cuts: 2  cuts with:  3001 4001
edge: 3000  side: 0  cuts: 1  cuts with:  4001
edge: 3001  side: 0  cuts: 1  cuts with:  2000
edge: 4000  side: 0  cuts: 1  cuts with:  7003
edge: 4001  side: 0  cuts: 3  cuts with:  2000 3000 7003
edge: 4002  side: 1  cuts: 2  cuts with:  5003 6003
edge: 5000  side: 0  cuts: 3  cuts with:  7003 6004 7004
```

```
edge: 5001  side: 1  cuts: 3  cuts with:  6002 7002 6003
edge: 5002  side: 1  cuts: 1  cuts with:  6003
edge: 5003  side: 1  cuts: 1  cuts with:  4002
edge: 6000  side: 1  cuts: 2  cuts with:  7001 7002
edge: 6001  side: 1  cuts: 1  cuts with:  7002
edge: 6002  side: 1  cuts: 1  cuts with:  5001
edge: 6003  side: 1  cuts: 3  cuts with:  5001 4002 5002
edge: 6004  side: 0  cuts: 2  cuts with:  5000 7005
edge: 7000  side: 1  cuts: 0
edge: 7001  side: 1  cuts: 1  cuts with:  6000
edge: 7002  side: 1  cuts: 3  cuts with:  6000 5001 6001
edge: 7003  side: 0  cuts: 3  cuts with:  4000 5000 4001
edge: 7004  side: 0  cuts: 1  cuts with:  5000
edge: 7005  side: 0  cuts: 1  cuts with:  6004
```

$\underline{n = 9}$

```
Graph cuts: 36
Conj. cuts: 36

0
00
001
0111
11111
111100
1110000

Number of cuts in side 0: 11
Number of cuts in side 1: 25

edge: 2000  side: 0  cuts: 2  cuts with:  3001 4001
edge: 3000  side: 0  cuts: 1  cuts with:  4001
edge: 3001  side: 0  cuts: 1  cuts with:  2000
edge: 4000  side: 0  cuts: 1  cuts with:  8003
edge: 4001  side: 0  cuts: 3  cuts with:  2000 3000 8003
edge: 4002  side: 1  cuts: 3  cuts with:  5003 6003 7003
edge: 5000  side: 0  cuts: 3  cuts with:  8003 7004 8004
edge: 5001  side: 1  cuts: 6  cuts with:  6002 7002 8002 6003 7003 60
04
edge: 5002  side: 1  cuts: 3  cuts with:  6003 7003 6004
edge: 5003  side: 1  cuts: 2  cuts with:  4002 6004
edge: 6000  side: 1  cuts: 5  cuts with:  7001 8001 7002 8002 7003
edge: 6001  side: 1  cuts: 3  cuts with:  7002 8002 7003
edge: 6002  side: 1  cuts: 2  cuts with:  5001 7003
```

```
edge: 6003  side: 1  cuts: 3  cuts with:  5001 4002 5002
edge: 6004  side: 1  cuts: 3  cuts with:  5001 5002 5003
edge: 7000  side: 1  cuts: 2  cuts with:  8001 8002
edge: 7001  side: 1  cuts: 2  cuts with:  6000 8002
edge: 7002  side: 1  cuts: 3  cuts with:  6000 5001 6001
edge: 7003  side: 1  cuts: 6  cuts with:  6000 5001 6001 4002 5002 60
02
edge: 7004  side: 0  cuts: 3  cuts with:  5000 8005 8006
edge: 7005  side: 0  cuts: 1  cuts with:  8006
edge: 8000  side: 1  cuts: 0
edge: 8001  side: 1  cuts: 2  cuts with:  6000 7000
edge: 8002  side: 1  cuts: 5  cuts with:  6000 7000 5001 6001 7001
edge: 8003  side: 0  cuts: 3  cuts with:  4000 5000 4001
edge: 8004  side: 0  cuts: 1  cuts with:  5000
edge: 8005  side: 0  cuts: 1  cuts with:  7004
edge: 8006  side: 0  cuts: 2  cuts with:  7004 7005
```

$\underline{n = 10}$

```
Graph cuts: 60
Conj. cuts: 60

0
00
000
0011
01111
111110
1111000
11100000

Number of cuts in side 0: 31
Number of cuts in side 1: 29

edge: 2000  side: 0  cuts: 3  cuts with:  3001 4001 5001
edge: 3000  side: 0  cuts: 3  cuts with:  4001 5001 4002
edge: 3001  side: 0  cuts: 2  cuts with:  2000 4002
edge: 4000  side: 0  cuts: 2  cuts with:  5001 9003
edge: 4001  side: 0  cuts: 3  cuts with:  2000 3000 9003
edge: 4002  side: 0  cuts: 3  cuts with:  3000 3001 9003
edge: 5000  side: 0  cuts: 3  cuts with:  9003 8004 9004
edge: 5001  side: 0  cuts: 6  cuts with:  2000 3000 4000 9003 8004 90
04
edge: 5002  side: 1  cuts: 5  cuts with:  6003 7003 8003 6004 7004
edge: 5003  side: 1  cuts: 2  cuts with:  6004 7004
```

```
edge: 6000  side: 0  cuts: 6  cuts with:  9003 8004 9004 7005 8005 90
05
edge: 6001  side: 1  cuts: 6  cuts with:  7002 8002 9002 7003 8003 70
04
edge: 6002  side: 1  cuts: 3  cuts with:  7003 8003 7004
edge: 6003  side: 1  cuts: 2  cuts with:  5002 7004
edge: 6004  side: 1  cuts: 2  cuts with:  5002 5003
edge: 7000  side: 1  cuts: 5  cuts with:  8001 9001 8002 9002 8003
edge: 7001  side: 1  cuts: 3  cuts with:  8002 9002 8003
edge: 7002  side: 1  cuts: 2  cuts with:  6001 8003
edge: 7003  side: 1  cuts: 3  cuts with:  6001 5002 6002
edge: 7004  side: 1  cuts: 5  cuts with:  6001 5002 6002 5003 6003
edge: 7005  side: 0  cuts: 3  cuts with:  6000 8006 9006
edge: 8000  side: 1  cuts: 2  cuts with:  9001 9002
edge: 8001  side: 1  cuts: 2  cuts with:  7000 9002
edge: 8002  side: 1  cuts: 3  cuts with:  7000 6001 7001
edge: 8003  side: 1  cuts: 6  cuts with:  7000 6001 7001 5002 6002 70
02
edge: 8004  side: 0  cuts: 6  cuts with:  5000 6000 5001 9005 9006 90
07
edge: 8005  side: 0  cuts: 3  cuts with:  6000 9006 9007
edge: 8006  side: 0  cuts: 2  cuts with:  7005 9007
edge: 9000  side: 1  cuts: 0
edge: 9001  side: 1  cuts: 2  cuts with:  7000 8000
edge: 9002  side: 1  cuts: 5  cuts with:  7000 8000 6001 7001 8001
edge: 9003  side: 0  cuts: 6  cuts with:  4000 5000 6000 4001 5001 40
02
edge: 9004  side: 0  cuts: 3  cuts with:  5000 6000 5001
edge: 9005  side: 0  cuts: 2  cuts with:  6000 8004
edge: 9006  side: 0  cuts: 3  cuts with:  8004 7005 8005
edge: 9007  side: 0  cuts: 3  cuts with:  8004 8005 8006
```

$\underline{n = 11}$

```
Graph cuts: 100
Conj. cuts: 100

0
00
000
0011
01111
111111
1111100
11110000
```

```
111000001

Number of cuts in side 0: 35
Number of cuts in side 1: 65

edge: 2000  side: 0  cuts: 3  cuts with:  3001 4001 5001
edge: 3000  side: 0  cuts: 3  cuts with:  4001 5001 4002
edge: 3001  side: 0  cuts: 2  cuts with:  2000 4002
edge: 4000  side: 0  cuts: 2  cuts with:  5001 10003
edge: 4001  side: 0  cuts: 3  cuts with:  2000 3000 10003
edge: 4002  side: 0  cuts: 3  cuts with:  3000 3001 10003
edge: 5000  side: 0  cuts: 3  cuts with:  10003 9004 10004
edge: 5001  side: 0  cuts: 6  cuts with:  2000 3000 4000 10003 9004 1
0004
edge: 5002  side: 1  cuts: 7  cuts with:  6003 7003 8003 9003 6004 70
04 8004
edge: 5003  side: 1  cuts: 3  cuts with:  6004 7004 8004
edge: 6000  side: 0  cuts: 6  cuts with:  10003 9004 10004 8005 9005
10005
edge: 6001  side: 1  cuts: 10  cuts with:  7002 8002 9002 10002 7003
8003 9003 7004 8004 7005
edge: 6002  side: 1  cuts: 6  cuts with:  7003 8003 9003 7004 8004 70
05
edge: 6003  side: 1  cuts: 4  cuts with:  5002 7004 8004 7005
edge: 6004  side: 1  cuts: 3  cuts with:  5002 5003 7005
edge: 7000  side: 1  cuts: 9  cuts with:  8001 9001 10001 8002 9002 1
0002 8003 9003 8004
edge: 7001  side: 1  cuts: 6  cuts with:  8002 9002 10002 8003 9003 8
004
edge: 7002  side: 1  cuts: 4  cuts with:  6001 8003 9003 8004
edge: 7003  side: 1  cuts: 4  cuts with:  6001 5002 6002 8004
edge: 7004  side: 1  cuts: 5  cuts with:  6001 5002 6002 5003 6003
edge: 7005  side: 1  cuts: 4  cuts with:  6001 6002 6003 6004
edge: 8000  side: 1  cuts: 5  cuts with:  9001 10001 9002 10002 9003
edge: 8001  side: 1  cuts: 4  cuts with:  7000 9002 10002 9003
edge: 8002  side: 1  cuts: 4  cuts with:  7000 6001 7001 9003
edge: 8003  side: 1  cuts: 6  cuts with:  7000 6001 7001 5002 6002 70
02
edge: 8004  side: 1  cuts: 9  cuts with:  7000 6001 7001 5002 6002 70
02 5003 6003 7003
edge: 8005  side: 0  cuts: 5  cuts with:  6000 9006 10006 9007 10007
edge: 8006  side: 0  cuts: 2  cuts with:  9007 10007
edge: 9000  side: 1  cuts: 3  cuts with:  10001 10002 10008
edge: 9001  side: 1  cuts: 4  cuts with:  7000 8000 10002 10008
```

```
edge: 9002  side: 1  cuts: 6  cuts with:  7000 8000 6001 7001 8001 10
008
edge: 9003  side: 1  cuts: 10  cuts with:  7000 8000 6001 7001 8001 5
002 6002 7002 8002 10008
edge: 9004  side: 0  cuts: 6  cuts with:  5000 6000 5001 10005 10006
10007
edge: 9005  side: 0  cuts: 3  cuts with:  6000 10006 10007
edge: 9006  side: 0  cuts: 2  cuts with:  8005 10007
edge: 9007  side: 0  cuts: 2  cuts with:  8005 8006
edge: 10000  side: 1  cuts: 0
edge: 10001  side: 1  cuts: 3  cuts with:  7000 8000 9000
edge: 10002  side: 1  cuts: 7  cuts with:  7000 8000 9000 6001 7001 8
001 9001
edge: 10003  side: 0  cuts: 6  cuts with:  4000 5000 6000 4001 5001 4
002
edge: 10004  side: 0  cuts: 3  cuts with:  5000 6000 5001
edge: 10005  side: 0  cuts: 2  cuts with:  6000 9004
edge: 10006  side: 0  cuts: 3  cuts with:  9004 8005 9005
edge: 10007  side: 0  cuts: 5  cuts with:  9004 8005 9005 8006 9006
edge: 10008  side: 1  cuts: 4  cuts with:  9000 9001 9002 9003
```

$\underline{n = 12}$

```
Graph cuts: 150
Conj. cuts: 150

0
00
000
0001
00111
011111
1111110
11111000
111100000
1110000001

Number of cuts in side 0: 75
Number of cuts in side 1: 75

edge: 2000  side: 0  cuts: 4  cuts with:  3001 4001 5001 6001
edge: 3000  side: 0  cuts: 5  cuts with:  4001 5001 6001 4002 5002
edge: 3001  side: 0  cuts: 3  cuts with:  2000 4002 5002
edge: 4000  side: 0  cuts: 4  cuts with:  5001 6001 5002 11003
edge: 4001  side: 0  cuts: 4  cuts with:  2000 3000 5002 11003
```

```
edge: 4002  side: 0  cuts: 3  cuts with:  3000 3001 11003
edge: 5000  side: 0  cuts: 4  cuts with:  6001 11003 10004 11004
edge: 5001  side: 0  cuts: 6  cuts with:  2000 3000 4000 11003 10004
11004
edge: 5002  side: 0  cuts: 7  cuts with:  3000 4000 3001 4001 11003 1
0004 11004
edge: 5003  side: 1  cuts: 4  cuts with:  6004 7004 8004 9004
edge: 6000  side: 0  cuts: 6  cuts with:  11003 10004 11004 9005 1000
5 11005
edge: 6001  side: 0  cuts: 10  cuts with:  2000 3000 4000 5000 11003
10004 11004 9005 10005 11005
edge: 6002  side: 1  cuts: 9  cuts with:  7003 8003 9003 10003 7004 8
004 9004 7005 8005
edge: 6003  side: 1  cuts: 5  cuts with:  7004 8004 9004 7005 8005
edge: 6004  side: 1  cuts: 3  cuts with:  5003 7005 8005
edge: 7000  side: 0  cuts: 10  cuts with:  11003 10004 11004 9005 100
05 11005 8006 9006 10006 11006
edge: 7001  side: 1  cuts: 10  cuts with:  8002 9002 10002 11002 8003
 9003 10003 8004 9004 8005
edge: 7002  side: 1  cuts: 6  cuts with:  8003 9003 10003 8004 9004 8
005
edge: 7003  side: 1  cuts: 4  cuts with:  6002 8004 9004 8005
edge: 7004  side: 1  cuts: 4  cuts with:  6002 5003 6003 8005
edge: 7005  side: 1  cuts: 3  cuts with:  6002 6003 6004
edge: 8000  side: 1  cuts: 9  cuts with:  9001 10001 11001 9002 10002
 11002 9003 10003 9004
edge: 8001  side: 1  cuts: 6  cuts with:  9002 10002 11002 9003 10003
 9004
edge: 8002  side: 1  cuts: 4  cuts with:  7001 9003 10003 9004
edge: 8003  side: 1  cuts: 4  cuts with:  7001 6002 7002 9004
edge: 8004  side: 1  cuts: 6  cuts with:  7001 6002 7002 5003 6003 70
03
edge: 8005  side: 1  cuts: 7  cuts with:  7001 6002 7002 6003 7003 60
04 7004
edge: 8006  side: 0  cuts: 4  cuts with:  7000 9007 10007 11007
edge: 9000  side: 1  cuts: 5  cuts with:  10001 11001 10002 11002 100
03
edge: 9001  side: 1  cuts: 4  cuts with:  8000 10002 11002 10003
edge: 9002  side: 1  cuts: 4  cuts with:  8000 7001 8001 10003
edge: 9003  side: 1  cuts: 6  cuts with:  8000 7001 8001 6002 7002 80
02
edge: 9004  side: 1  cuts: 10  cuts with:  8000 7001 8001 6002 7002 8
002 5003 6003 7003 8003
edge: 9005  side: 0  cuts: 9  cuts with:  6000 7000 6001 10006 11006
```

```
10007 11007 10008 11008
edge: 9006  side: 0  cuts: 5  cuts with:  7000 10007 11007 10008 1100
8
edge: 9007  side: 0  cuts: 3  cuts with:  8006 10008 11008
edge: 10000  side: 1  cuts: 3  cuts with:  11001 11002 11009
edge: 10001  side: 1  cuts: 4  cuts with:  8000 9000 11002 11009
edge: 10002  side: 1  cuts: 6  cuts with:  8000 9000 7001 8001 9001 1
1009
edge: 10003  side: 1  cuts: 10  cuts with:  8000 9000 7001 8001 9001
6002 7002 8002 9002 11009
edge: 10004  side: 0  cuts: 10  cuts with:  5000 6000 7000 5001 6001
5002 11005 11006 11007 11008
edge: 10005  side: 0  cuts: 6  cuts with:  6000 7000 6001 11006 11007
 11008
edge: 10006  side: 0  cuts: 4  cuts with:  7000 9005 11007 11008
edge: 10007  side: 0  cuts: 4  cuts with:  9005 8006 9006 11008
edge: 10008  side: 0  cuts: 3  cuts with:  9005 9006 9007
edge: 11000  side: 1  cuts: 0
edge: 11001  side: 1  cuts: 3  cuts with:  8000 9000 10000
edge: 11002  side: 1  cuts: 7  cuts with:  8000 9000 10000 7001 8001
9001 10001
edge: 11003  side: 0  cuts: 9  cuts with:  4000 5000 6000 7000 4001 5
001 6001 4002 5002
edge: 11004  side: 0  cuts: 6  cuts with:  5000 6000 7000 5001 6001 5
002
edge: 11005  side: 0  cuts: 4  cuts with:  6000 7000 6001 10004
edge: 11006  side: 0  cuts: 4  cuts with:  7000 10004 9005 10005
edge: 11007  side: 0  cuts: 6  cuts with:  10004 9005 10005 8006 9006
 10006
edge: 11008  side: 0  cuts: 7  cuts with:  10004 9005 10005 9006 1000
6 9007 10007
edge: 11009  side: 1  cuts: 4  cuts with:  10000 10001 10002 10003
```

7.3 Algoritme II

Aquest algoritme avalua tots els grafs complets que contenen un subgraf cíclic C_n dins seu amb la propietat que les arestes d'aquest no es tallen amb cap altra aresta del graf complet.

```
#include <iostream>
#include <time.h>
#include <fstream>
#include <sstream>
```

```
using namespace std;

class edge{
  public:
    //identificador ME*original vertex + final vertex
    int e_id;
    //original vertex
    int e_ov;
    //final vertex
    int e_fv;
    //side of the vertex
    int e_side;
    //number of cuts
    int e_nc;
    //maxim number of cuts of our edge
    int e_ncmax;
    //posible edges that cuts the one with we work
    edge **e_pcut;
    //number of cuts counter
    int e_ncc;
    //previous edge by rows
    edge *e_prevr;
    //next edge by rows
    edge *e_nextr;
    //previous edge by columns
    edge *e_prevc;
    //next edge by columns
    edge *e_nextc;
};

void creategraph(edge**);
void adjencygraph1(edge**);
void adjencygraph2();
int ncutall();
int ncut(edge*);
void recursive(edge*, int);
void printfile();

int V=9, A, ME=100;//Maximum egdes
long long counter1=0, counter2=0;
bool flag;
edge *origin;

int main(){
```

```
  edge **graph1;
  int grcut, i;
  cout << "Number of vertixes (maximum 10): ";
  cin >> V;
  cout << "If you want the adjency information of each graph return 1
." << endl;
  cout << "If you only want graphs which ones achieve the minimum con
jectured return 0." << endl;
  cin >> flag;
  A=(int)(V/2)*(int)((V-1)/2)*(int)((V-2)/2)*(int)((V-3)/2)/4;
  graph1=new edge*[V];
  for(i=2;i<V;i++)
    graph1[i]=new edge[i-1];
  origin=&graph1[2][0];
  creategraph(graph1);
  adjencygraph1(graph1);
  grcut=ncutall();
  for(; origin!=NULL; origin=origin->e_nextr)
    origin->e_ncmax=origin->e_nc;
  origin=&graph1[2][0];
  adjencygraph2();
  long t1;
  t1=time(NULL);
  recursive(&graph1[2][0], grcut);
  cout << "Written the solutions in \"Solucions_of_" << V << "_vertic
es_" << flag << ".txt\"" << endl;
  cout << "Number of graphs analised: " << counter1 << endl;
  cout << "Number of graphs which ones achieve the minimum conjecture
d: " << counter2 << endl;
  cout << "Final time: "<< time(NULL)-t1 << endl;
  system("pause");
  return 0;
}

void creategraph(edge** graph){
  for(int i=2; i<V; i++){
    for(int j=0; j<i-1; j++){
      graph[i][j].e_ov=j;
      graph[i][j].e_fv=i;
      graph[i][j].e_id=ME*i+j;
      graph[i][j].e_nc=0;
      graph[i][j].e_ncc=0;
  graph[i][j].e_ncmax=0;
      graph[i][j].e_side=0;
```

```
    }
  }
  graph[V-1][0].e_side=2;
  return;
}

void adjencygraph1(edge** graph){
  int i, j;
  //adjency by rows
  graph[2][0].e_prevr=NULL;//First case
  for(i=2; i<V-1; i++){
    for(j=0; j<i-2; j++){//if j==i-2 in another case
      graph[i][j].e_nextr=&graph[i][j+1];
      graph[i][j+1].e_prevr=&graph[i][j];
    }
    if(i!=V-2){
      graph[i][j].e_nextr=&graph[i+1][0];
      graph[i+1][0].e_prevr=&graph[i][j];
    }
    else{
      graph[i][j].e_nextr=&graph[i+1][1];
      graph[i+1][1].e_prevr=&graph[i][j];
    }
  }
  graph[V-1][0].e_nextr=NULL;//Special case
  graph[V-1][0].e_prevr=NULL;
  for(j=1; j<V-3; j++){//Last row
    graph[V-1][j].e_nextr=&graph[V-1][j+1];
    graph[V-1][j+1].e_prevr=&graph[V-1][j];
  }
  graph[V-1][V-3].e_nextr=NULL;//Last case
  //adjency by columns
  graph[2][0].e_prevc=NULL;//First case
  for(i=2; i<V-1; i++){
    for(j=0; j<i-1; j++){
      graph[i][j].e_nextc=&graph[i+1][j];
      graph[i+1][j].e_prevc=&graph[i][j];
    }
  }
  graph[V-2][0].e_nextc=&graph[3][1];//Special case
  graph[3][1].e_prevc=&graph[V-2][0];
  graph[V-1][0].e_nextc=NULL;
  graph[V-1][0].e_prevc=NULL;
  for(j=1; j<V-3; j++){//Last line
```

```
    graph[V-1][j].e_nextc=&graph[j+3][j+1];
    graph[j+3][j+1].e_prevc=&graph[V-1][j];
  }
  graph[V-1][V-3].e_nextc=NULL;//Last case
}

void adjencygraph2(){
  edge *graph1, *graph2;
  //Posible cuts adjency
 for(graph1=origin; graph1!=NULL; graph1=graph1->e_nextr)
    graph1->e_pcut=new edge*[graph1->e_ncmax];
  //We suppose that all the side are the same
 for(graph1=origin; graph1->e_nextc!=NULL; graph1=graph1->e_nextc)
   for(graph2=graph1->e_nextc; graph2!=NULL; graph2=graph2->e_nextc)
      if(graph1->e_ov!=graph2->e_ov && graph2->e_ov<graph1->e_fv && g
raph1->e_fv<graph2->e_fv){
        graph1->e_pcut[graph1->e_ncc]=graph2;
        graph2->e_pcut[graph2->e_ncc]=graph1;
        graph1->e_ncc++;
        graph2->e_ncc++;
      }
  for(graph1=origin; graph1!=NULL; graph1=graph1->e_nextr)
    graph1->e_ncc=0;
  return;
}

int ncutall(){
  edge *graph1, *graph2;
  for(graph1=origin; graph1!=NULL; graph1=graph1->e_nextr)
    graph1->e_nc=0;
  int grcut=0;
  for(graph1=origin; graph1->e_nextc!=NULL; graph1=graph1->e_nextc)
    for(graph2=graph1->e_nextc; graph2!=NULL; graph2=graph2->e_nextc)
  if(graph1->e_side==graph2->e_side && graph1->e_ov!=graph2->
e_ov && graph2->e_ov<graph1->e_fv && graph1->e_fv<graph2->e_fv){
    graph1->e_nc++;
        graph2->e_nc++;
        grcut++;
  }
  return grcut;
}

int ncut(edge *pedge){
  int dif=0;
```

```
  for(int i=0;i<pedge->e_ncmax;i++)
    dif+=pedge->e_pcut[i]->e_side;
  return (2*dif-pedge->e_ncmax);
}

void recursive(edge* pedge, int grcut){
  counter1++;
  edge *pedgeaux=pedge;
  grcut=grcut+ncut(pedge);
  pedge->e_side=1;
  if(grcut==A){
    printfile();
  }
  while(pedge->e_nextr!=NULL){
    pedge=pedge->e_nextr;
    recursive(pedge, grcut);
  }
  pedge=pedgeaux;
  pedge->e_side=0;
  return;
}

void printfile(){
  edge *graph;
  int numcut=0;
  counter2++;
  ostringstream name;
  name.flush( );
  ncutall();
  for(graph=origin; graph!=NULL; graph=graph->e_nextr)
    if(graph->e_side==0)
  numcut+=graph->e_nc;
  numcut/=2;
  name << "Solucions_of_" << V << "_vertices_"<< flag <<".txt";
  ofstream a_file (name.str().c_str(),ios_base::app);
  if ( !a_file.is_open() ){
    cout << "error printfile";
    exit (-1);
  }
  a_file << "Graph " << counter2 << endl << endl;
  graph=origin;
  a_file << graph->e_side;
  while(graph->e_nextr!=NULL){
    graph=graph->e_nextr;
```

```
    if(graph->e_fv!=graph->e_prevr->e_fv){
      a_file << endl;
      if(graph->e_fv==V-1)
        a_file << " ";
    }
    a_file << graph->e_side;
  }
  a_file << endl << endl;
  a_file << "number of cuts in edges in side 0: " << numcut << endl <
< "number of cuts in edges in side 1: " << A-numcut << endl;
  a_file << endl << endl;
  if (flag==1)
    for(graph=origin; graph!=NULL; graph=graph->e_nextr)
      a_file << "edge: " << graph->e_id << "  side: " << graph->e_sid
e << "  cuts: " << graph->e_nc << endl;
  a_file << endl << endl;
  a_file.close();
  return;
}
```

7.4 Resultats algoritme II

Donem tots els resultat del programa pels casos amb 6, 8, 10 vèrtexs, els 3 representants no isomorfs dels 35 trobats en el cas de 7 vèrtexs i 4 representants no isomorfs dels 44 trobats en el cas de 9.

$\underline{n = 6}$

```
Graph 1

1
11
100
 001

number of cuts in edges in side 0: 1
number of cuts in edges in side 1: 2


Graph 2

1
10
000
```

```
 011

number of cuts in edges in side 0: 2
number of cuts in edges in side 1: 1


Graph 3

1
00
001
 111

number of cuts in edges in side 0: 1
number of cuts in edges in side 1: 2
```

$\underline{n = 7}$

```
Graph 1

1
11
111
1000
 0001

number of cuts in edges in side 0: 3
number of cuts in edges in side 1: 6


Graph 2

1
11
110
1000
 0011

number of cuts in edges in side 0: 2
number of cuts in edges in side 1: 7


Graph 3

1
```

```
11
110
1000
 0001

number of cuts in edges in side 0: 5
number of cuts in edges in side 1: 4
```

$\underline{n = 8}$

```
Graph 1

1
11
110
1000
00001
 00111

number of cuts in edges in side 0: 9
number of cuts in edges in side 1: 9


Graph 2

1
11
100
0000
00011
 01111

number of cuts in edges in side 0: 9
number of cuts in edges in side 1: 9


Graph 3

1
10
000
0001
00111
 11110
```

```
number of cuts in edges in side 0: 9
number of cuts in edges in side 1: 9


Graph 4

1
00
000
0011
01111
 11100

number of cuts in edges in side 0: 9
number of cuts in edges in side 1: 9
```

$\underline{n = 9}$

```
Graph 1

1
11
111
1110
10000
100001
 000111

number of cuts in edges in side 0: 14
number of cuts in edges in side 1: 22


Graph 2

1
11
111
1110
10000
000001
 000111

number of cuts in edges in side 0: 17
number of cuts in edges in side 1: 19
```

```
Graph 6

1
11
111
1100
10000
000011
 001111

number of cuts in edges in side 0: 11
number of cuts in edges in side 1: 25

Graph 8

1
11
111
1100
10000
000001
 001111

number of cuts in edges in side 0: 16
number of cuts in edges in side 1: 20
```

$\underline{n = 10}$

```
Graph 1

1
11
111
1100
10000
000001
0000111
 0011111

number of edge cut in side 0: 29
number of edge cut in side 1: 31


Graph 2
```

```
1
11
110
1000
00000
000011
0001111
 0111110

number of edge cut in side 0: 31
number of edge cut in side 1: 29


Graph 3

1
11
100
0000
00001
000111
0011111
 1111100

number of edge cut in side 0: 29
number of edge cut in side 1: 31


Graph 4

1
10
000
0000
00011
001111
0111110
 1111000

number of edge cut in side 0: 31
number of edge cut in side 1: 29


Graph 5
```

```
1
00
000
0001
00111
011111
1111100
 1110000

number of edge cut in side 0: 29
number of edge cut in side 1: 31
```

Bibliografia

[1] J. Blazek i M. Koman, *A minimal problem concerning complete plane graphs,* Theory of Graphs and Its Applications, Czechoslovak Academy of Sciences (June 1963), 113-117.

[2] M. R. Farahani, *A good drawing of complete bipartite graph $K_{9,9}$, whose crossing number holds Zarankiewich conjectures*, Studia Universitatis Babes-Bolyai Series Informatica, Vol. **58**, No.1 (2013), 21–28.

[3] M. R. Garey i D. S. Johnson, *Crossing number is NP-complete*, SIAM Journal Algebraic Discrete Methods, Vol. **4**, No. 3 (September 1983), 312–316.

[4] R. K. Guy, *Latest results on crossing numbers,* Recent Trends in Graph Theory, Lecture Notes in Mathematics, Vol. **186**, Springer-Verlag, Heidelberg, Berlin, New York (1970), 143–156.

[5] R. K. Guy, *Crossing numbers of graphs,* In Graph Theory and Applications, Lecture Notes in Mathematics, Vol. **303**, Springer-Verlag, Heidelberg, Berlin, New York (1972), 111–124.

[6] F. Harary, *Graph Theory,* Addison-Wesley, Reading, MA (1969).

[7] D. J. Kleitman, *The crossing number of $K_{5,n}$*, Journal of Combinatorial Theory, Vol. **9** (1971), 315–323.

[8] D. J. Kleitman, *A note on the parity of the numbers of crossings of a graph*, Journal of Combinatorial Theory B, Vol. **21** (1976), 88–89.

[9] E. de Klerk, D. V. Pasechnik i A. Schrijver, *Reduction of symmetric semidefinite programs using the regular *-representation*, Mathematical Programming, Series B, Vol. **109**, (2007), 613–624.

[10] D. McQuillan i R. B. Richter, *A parity theorem for drawings of complete and complete bipartite graphs*, The American Mathematical Monthly, Vol. **117**, No. 3 (March 2010), 267–273.

[11] S. Pan i R. B. Richter, *The crossing number of K_{11} is 100*, J. Graph Theory, Vol. **56** (2007), 128–134.

[12] R. B. Richter i C. Thomassen, *Relations Between Crossing Numbers of Complete and Complete Bipartite Graphs*, The American Mathematical Monthly, Vol. **104**, No. 2 (February 1997), 131–137.

[13] P. Turán, *A Note of Welcome*, J. Graph Theory, Vol. **1**, (1977), 7–9.

[14] V. Vassilevska, *On the crossing number of* $K_{9,9}$, http://www.cs.cmu.edu/~virgi/final_report02.pdf

[15] E. Weisstein, *Graph Crossing Number*, From MathWorld–A Wolfram Web Resource, http://mathworld.wolfram.com/GraphCrossingNumber.html

[16] E. Weisstein, *Zarankiewicz's Conjecture*, From MathWorld–A Wolfram Web Resource, http://mathworld.wolfram.com/ZarankiewiczsConjecture.html

[17] D.R. Woodall, *Cyclic-Order Graphs and Zarankiewicz's Crossing-Number Conjecture*, J. Graph Theory, Vol. **16** (1993), 657–671.

[18] K. Zarankiewicz, *On a Problem of P. Turán Concerning Graphs*, Fund. Math, Vol. **41** (1954), 137–145.