



École Nationale Supérieure d'Informatique pour l'Industrie et l'Entreprise

# Rapport Stage 2A

Reconstruction de graphes à base de requêtes

---

*Élève : Maxime APARICIO  
Tuteur académique : (Mahdi Moeini)*

le cnam

*Entité : Conservatoire national des arts et métiers 292 Rue Saint-Martin 75003 Paris*

*Maîtres De Stage : Dimitri Watel et Cédric Bentz*

*Dates : 20 Juin, 2025 - 19 Septembre, 2025*

# Glossaire

Ce glossaire regroupe les définitions et précisions des notions qui seront fréquemment utilisées dans ce rapport. Il n'as pas vocation à être lu en premier il est conseillé de lire la suite et de s'y référer en cas de doute.

Partition : dans ce rapport, nous étudions des partitions de graphes d'ordre k non étiquetés. Ainsi, chaque partie de  $P$  contient un ou plusieurs graphes d'ordre  $n$ .

Raffinement : Étant données deux partitions  $P$  et  $R$ , on dit que  $P$  est un raffinement de  $R$  lorsque :

$$\forall A \in P, \exists B \in R | A \subset B$$

Ainsi, l'information contenue dans  $R$  est aussi contenue dans  $P$ . Par exemple, la partition  $E$ , donnant l'information sur le nombre d'arêtes, est un raffinement de la partition  $J$  donnant l'information sur la parité du nombre d'arêtes. Notons qu'être un raffinement est transitif. Notons aussi que toutes les conditions suffisantes permettant de reconstruire un graphe vis-à-vis d'une partition le sont aussi pour ses raffinements.

Requête : Une requête est le fait de récupérer une certaine information sur une partie du graphe. En général, une requête donne une information partielle, ne permettant pas directement de reconstruire le graphe en question.

requête vis-à-vis de  $P$  ( ou  $Q^R(.)$  ), avec  $P$  une partition : C'est une fonction qui à un sous-ensemble de sommet  $S$  de taille  $k$  un graphe  $G$  associe la partie de  $P$  contenant le (seul) graphe d'ordre  $n$  isomorphe à  $G[S]$  dans  $P$ . "vis-à-vis de  $P$ " permet de fixer la partition avec laquelle on travaille.

La classe d'un graphe  $G$  par rapport a une partition  $P$  : Étant donnée une partition  $P$ , la classe d'un graphe  $G$  est l'ensemble des graphes ayant le même résultat aux requêtes vis-à-vis de  $P$ , que le graphe  $G$ . Deux graphes sont dans la même classe pour une partition  $P$  lorsque chaque requête faite grâce à la partition  $P$  donne le même résultats pour chacun des graphes. Notons qu'une classe peut être définie à l'aide d'un graphe lui appartenant, ou grâce à un résultat de requête (auquel cas, la classe peut être vide).

Graphe reconstructible : Un graphe est reconstructible lorsque le résultat des requêtes faites sur ce graphe permet de reconstruire (c'est-à-dire retrouver les arêtes du graphes) le graphe de façon unique. En d'autres termes, le jeu de requêtes ne correspond qu'à un seul graphe. Un graphe reconstructible est seul dans sa classe.

$G \equiv^R H$  :  $G$  est équivalent vis-à-vis de la requête  $R$  à  $H$ , c'est-à-dire que  $G$  et  $H$  sont dans la même classe, vis-à-vis de la requête  $R$ .

$Q_G^R(S)$  : le résultats d'une requête sur le sous-ensemble de sommets  $S$  de  $G$ , vis-à-vis de la partition  $R$ . Le Q vient de Query.

$\mathcal{C}(Q_{I_5}^R)$  : la classe de  $I_5$  vis-à-vis de la partition  $R$ ; cet ensemble est composé de tous les graphes ayant les mêmes requêtes que  $I_5$  vis-à-vis de  $R$ .

Singleton : Ici ce terme fait référence à un des graphes d'ordre k comme ceux de la figure 1.1 pour l'ordre 4 (page 2).

$J_4$  : La requête de parité, qui indique si le nombre d'arêtes dans le sous graphe sélectionné est pair ou impair. ( voir figure 1.1)

$E_4$  : La requête du nombre d'arêtes qui renvoie le nombre d'arêtes dans le sous graphe sélectionné. ( voir figure 1.1)

$S_4$  : La requête singleton, qui indique à quel d'ordre k le sous graphe sélectionné est isomorphe. Cette requête est celle donnant l'information la plus fine. ( voir figure 1.1)

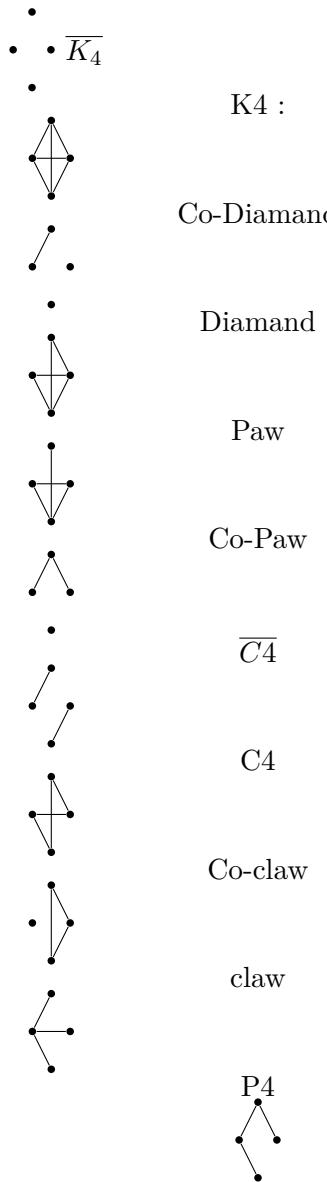


FIGURE 1 – Tout les graphes d'ordre 4, à isomorphisme près

# Table des figures

1	Tout les graphes d'ordre 4, à isomorphisme près . . . . .	ii
1.1	Exemple de partition des graphes d'ordre 4 . . . . .	2
1.2	Exemple d'un graphe et de ses Requêtes . . . . .	2
1.3	Illustration du fait que le rendu de la requête n'est pas étiqueté. . . . .	3
2.1	Graphes représentant la classe $\mathcal{C}(Q_{I_n}^{J_4})$ pour $n = 4$ (à isomorphisme près). . . . .	7
2.2	Configuration 1 et 2 évoquées durant la preuve de la proposition 2.2.3 . . . . .	8
2.3	Graphes représentant la classe $\mathcal{C}(Q_{I_n}^{J_4})$ pour $n = 4$ à isomorphisme près. . . . .	13
2.4	Différentes configurations . . . . .	14

# Table des matières

<b>Glossaire</b>	<b>i</b>
<b>Liste des figures</b>	<b>iii</b>
<b>1 Introduction : Environnement de travail et du stage</b>	<b>1</b>
1.1 Le laboratoire Cédric du Cnam . . . . .	1
1.1.1 L'équipe OC . . . . .	1
1.2 Exposition du Sujet . . . . .	1
1.3 Objectif . . . . .	3
<b>État de l'art</b>	<b>4</b>
<b>2 Contributions</b>	<b>5</b>
2.1 Graphe de raffinement et base de donnée . . . . .	5
2.2 Requête de parité : J4 . . . . .	6
2.2.1 Résultats sur <i>J4</i> . . . . .	6
2.3 Résultats sur <i>S4</i> . . . . .	10
<b>3 Conclusion et Prolongements</b>	<b>15</b>
3.1 Conclusion . . . . .	15
3.2 Prolongements . . . . .	15
<b>A DD&amp;RS -</b>	<b>16</b>
A.1 Entreprise . . . . .	16
A.2 Bilan carbone . . . . .	17
<b>B code</b>	<b>18</b>
<b>Bibliography</b>	<b>18</b>

# Chapitre 1

## Introduction : Environnement de travail et du stage

### 1.1 Le laboratoire Cédric du Cnam

Le Centre d'études et de recherche en informatique et communications (Cédric) regroupe les activités de recherche en sciences du numérique menées au Conservatoire national des arts et métiers (Cnam). Le laboratoire mène des recherches fondamentales et appliquées. Le laboratoire participe à des contrats de recherche fondamentale ou technologique (transfert et mise en œuvre de résultats de recherche, contrats industriels) ainsi qu'à des contrats de réseaux de recherche français et européens (ANR, IST, etc.).

Le laboratoire compte actuellement 176 membres dont 91 permanents et 5 agents administratifs affectés. Sa taille et sa maturité actuelles en font le laboratoire de recherche le plus important du Cnam et l'un des laboratoires d'informatique reconnus de la région parisienne.

#### 1.1.1 L'équipe OC

Plus précisément, on stage se déroulait au sein de l'équipe d'Optimisation Combinatoire (OC). Un de mes maîtres de stage, Dimitri Watel (Cnam, Cedric OC) a été amené à travailler sur la reconstruction d'un graphe à l'aide de ses requêtes lors d'un groupe de travail aux JGA (Journées Graphes et Algorithmes - journées annuelles de la communauté francophone en théorie et algorithmique des graphes) fin 2023, au côté de Hoang La (LISN), Florian Galliot (I2M), Matthieu Petiteau (Institut Fourier), Raphaëlle Maistre (LISN) – Rédaction d'un premier papier en cours (toujours en cours) –. Cédric Bentz (Cnam, Cedric OC) et Dimitri Watel m'ont ensuite proposé de sujet de stage pour poursuivre une des (nombreuses) perspectives des travaux démarrés.

### 1.2 Exposition du Sujet

(Identifiez les problèmes que le contexte actuel présente afin de justifier l'objectif de votre stage et les défis que vous avez dû affronter)

Dans les problèmes de reconstruction de graphe, nous avons un graphe étiqueté dont nous connaissons les sommets, mais pas les arêtes. L'objectif est de reconstruire le graphe, c'est-à-dire retrouver ses arêtes. Pour cela, il est possible d'effectuer des requêtes sur ce graphe pour obtenir des informations partielles qui mises bout à bout pourront aider à dévoiler ces arêtes.

Dans notre cas, chaque requête consiste à choisir un sous-ensemble de sommets  $S$  d'ordre fixé  $k$  du graphe qu'on souhaite reconstruire. Nous obtenons donc en sortie de requête une information sur ce sous-ensemble. L'information reçue dépend de la nature de la requête, et cette information peut être partielle, *i.e.*, elle ne nous dit pas nécessairement où sont les arêtes.

Dans ce stage, nous nous intéressons à un type bien particulier de requêtes. On se fixe une partition des graphes non orientés d'ordre  $k$  à isomorphisme près (voir par exemple la figure 1.1 pour des partitions des graphes d'ordre 4). La requête nous indique à quelle partie de cette partition appartient, à isomorphisme près, le sous-graphe induit par les sommets  $S$  sur lesquels on a fait la requête. Chaque partition nous donne un degré d'information plus ou moins élevé, c'est pourquoi il est important de préciser à chaque fois sur quelle partition nous travaillons.

$$\left\{ \begin{array}{c} \bullet \\ \cdot \\ \cdot \end{array}, \begin{array}{c} \bullet \\ \diagup \\ \cdot \end{array}, \begin{array}{c} \bullet \\ \diagdown \\ \cdot \end{array}, \begin{array}{c} \bullet \\ \diagup \\ \diagdown \end{array}, \begin{array}{c} \bullet \\ \diagdown \\ \diagup \end{array}, \begin{array}{c} \bullet \\ \diagup \\ \diagup \end{array}, \begin{array}{c} \bullet \\ \diagdown \\ \diagdown \end{array} \end{array} \right\}, \left\{ \begin{array}{c} \bullet \\ \cdot \\ \cdot \end{array}, \begin{array}{c} \bullet \\ \cdot \\ \cdot \end{array}, \begin{array}{c} \bullet \\ \cdot \\ \cdot \end{array}, \begin{array}{c} \bullet \\ \cdot \\ \cdot \end{array} \end{array} \right\}$$

Partition  $J_4$ , correspondant à la parité, contenant une partie avec tous les graphes d'ordre 4 avec un nombre pair d'arêtes, et une partie avec les graphes dont le nombre d'arêtes est impair.

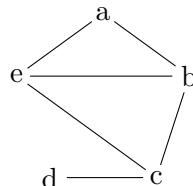
$$\left\{ \begin{array}{c} \bullet \\ \cdot \\ \cdot \end{array}, \begin{array}{c} \bullet \\ \cdot \\ \cdot \end{array} \end{array} \right\}$$

Partition  $E_4$ , correspondant au nombre d'arêtes, et où chaque partie contient tous les graphes ayant un même nombre d'arêtes.

$$\left\{ \begin{array}{c} \bullet \\ \cdot \\ \cdot \end{array}, \begin{array}{c} \bullet \\ \cdot \\ \cdot \end{array} \end{array} \right\} \cup \left\{ \begin{array}{c} \bullet \\ \bullet \\ \bullet \\ \bullet \end{array} \right\}$$

Partition  $S_4$ , ne contenant que des singltons.

FIGURE 1.1 – Exemple de partition des graphes d'ordre 4



Graphe d'ordre 5

Ensemble de sommets sélectionné	Résultat de la requête $E_4$
b,c,d,e	4
a,c,d,e	3
a,b,d,e	3
a,b,c,e	5
a,b,c,d	3

Les requêtes associées

FIGURE 1.2 – Exemple d'un graphe et de ses Requêtes

Nous insistons fortement sur le fait que les sommets du graphe sur lequel on fait les requêtes sont étiquetés mais que l'information reçue suite à la requête ne contient aucun étiquetage (cf 1.3). Ainsi,

même si on sait à quelle partie de Singleton appartient  $G[S]$  on n'est pas capable de placer les arêtes avec cette seule information. De plus le graphe d'origine étant étiqueté, on veut reconstruire le graphe étiqueté et non pas le graphe à isomorphisme près. Ainsi pour chaque paire de sommet, nous devons déterminer si cette paire est relié ou non. Ajoutons que si le graphe qu'on souhaite reconstruire n'était pas étiqueté, il ne serait pas possible de requêter précisément un sous-ensemble  $S$ .

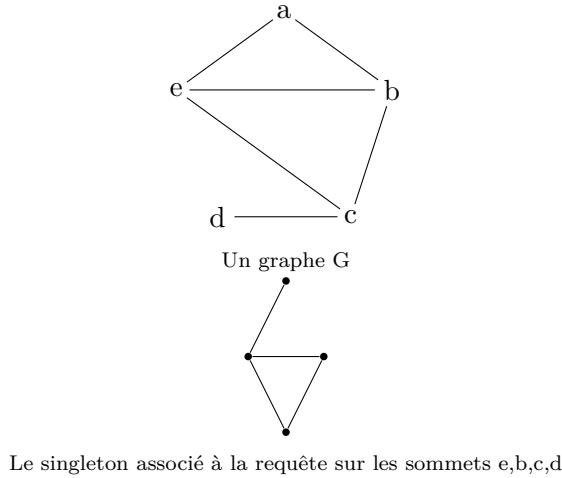


FIGURE 1.3 – Illustration du fait que le rendu de la requête n'est pas étiqueté.

### 1.3 Objectif

Étant donnée une partition (qui, on le rappelle, définit une requête), l'objectif est de déterminer les conditions nécessaires et suffisantes pour pouvoir reconstruire un graphe (retrouver ses arêtes). Toutes les requêtes d'ordre 3 ont déjà été traité dans [Galliot et al., 2025]), nous nous intéressons donc aux requêtes d'ordre 4.

Cependant, lorsque l'on s'intéresse au requêtes d'ordre 4, l'ensemble des partitions est vaste ; en effet, il y a 11 graphes d'ordre 4 différents (à isomorphisme près) et donc, il y a 678 570 partitions différentes. Ainsi, nous nous intéressons à deux partitions en particulier :

- Celle nous donnant l'information sur la parité. La partition contient deux parties, les graphes avec un nombre pair d'arêtes et ceux avec un nombre impair d'arêtes. De manière équivalente, on peut considérer que la requête nous renvoie 1 si le nombre d'arêtes dans le sous-graphe induit par  $S$  est pair, et 0 sinon ( voir figure 1.1 ).
  - Celle portant sur les singltons ; cette partition est donc de cardinal 11 et chacune de ses parties est un singleton. Notons que c'est la partition la plus fine, donnant le plus d'information.
- D'autres partitions proches seront étudiées ou mentionnées au cours du rapport.

De plus, au travers d'expérimentations, nous recherchons aussi des informations plus globales telles que des potentiels liens entre les partitions ou, pour chacune, combien de graphes d'un ordre donné sont reconstructibles.

# Etat de l'art

L'étude sur la reconstruction des graphes a été initiée par Kelly dans [Kelly, 1957]. En effet, il a montré que, en utilisant des requêtes de taille  $n - 1$ , avec  $n$  l'ordre du graphe sur des parties non étiqueté du graphe, , les arbres sont reconstructibles de manière unique. La différence avec ce que l'on fait, est que dans son papier, il a considéré un graphe non étiqueté, et donc reconstruit à isomorphisme près. Il a donc accès à tout les sous graphes d'ordre  $n-1$ , mais sans savoir quel a été le sommet qui a été retiré. Cette façon de procéder s'approche de notre requête singleton (  $S_{n-1}$  ) puisque l'on sait à quel graphe, le sous graphe de  $G$  est isomorphe, mais sans étiquettes ). Pour ce faire, il a considéré deux arbres  $A$  et  $B$  qui donnaient les mêmes réponses. Cela implique en particulier que tout sous-arbre d'ordre  $n - 1$  apparaissant dans  $A$  apparaît aussi dans  $B$  un nombre identique de fois. En considérant certains sous-arbres rattachés à des feuilles particulières, il a été capable d'en déduire l'égalité des graphes  $A$  et  $B$ .

Un autre papier, plus récent, [Galliot et al., 2025]) s'intéresse quant à lui aux requêtes d'ordre 3. Dans ce papier, deux questions ont été résolues :

- La caractérisation des graphes reconstructibles, c'est-à-dire, pour chaque partition  $P$  des graphes d'ordre 3, décrire des conditions nécessaires et suffisantes pour pouvoir reconstruire un graphe à partir des résultats des requêtes faites sur ce graphe.
- L'énumération : Étant donné un ensemble  $A$  de résultats de requêtes, existe-t-il un algorithme qui énumère tous les graphes dont les réponses aux requêtes sont les mêmes que celles données dans  $A$ , de telle sorte que le nombre d'opérations élémentaires effectuées entre deux graphes énumérés soit polynomial. Cet aspect porte sur une vision plus pratique de la reconstruction.

Mentionnons aussi que dans ces mêmes recherches, il a été montré qu'il est possible de reconstruire tous les graphes à  $n \geq k + 2$  sommets avec  $E_k$  où  $k$  est la taille de la requête. De plus, dans le cas  $n = k + 1$ , le graphe est reconstructible avec  $E_k$  si et seulement si il l'est à l'aide de la séquence des degrés.

# Chapitre 2

## Contributions

Afin d'avoir des pistes de recherche, une première partie consiste, pour une partition donnée, à énumérer toutes les classes afin d'en déduire potentiellement une propriété. Pour ce faire, il m'a été donné un programme *EnumerateClasses.jl*, qui énumère tous les graphes d'un ordre donné, fait les requêtes pour chacun d'entre eux, et classe chaque graphe dans une classe. Le programme affiche ensuite toutes les classes dans un document LaTeX. J'ai par la suite modifié ce programme afin qu'il puisse avoir le même fonctionnement, mais sur les graphes d'ordre 4. En l'occurrence, pour qu'il puisse identifier chaque singleton. Cette section fait régulièrement référence aux graphes d'ordre 4 par leur nom ; ils sont répertoriés dans le glossaire (voir figure 1).

### 2.1 Graphe de raffinement et base de donnée

Dans l'objectif d'avoir un regard plus large sur l'ensemble des partitions, j'ai construit un graphe des raffinements. Chaque sommet de ce graphe représente une partition, et un arc entre deux sommet signifie que l'un est un raffinement direct de l'autre (allant donc de la partition de cardinal 1, jusqu'à celle des singletons). Notons qu'il existe un programme permettant de créer ce graphe, qui s'appelle “écriture graphe de raffinement.jl”, et qui sauvegarde le graphe dans un fichier “raffinement.json”, ainsi qu'un programme permettant de lire et d'effectuer des requêtes sur ce graphe, qui s'appelle “lecture graphe de raffinement.jl”. Le programme “écriture graphe de raffinement.jl” permet également de créer le fichier “raffinement\_inverse.json”, qui décrit là encore un graphe de raffinement, mais où le sens des arcs est inversé, ce qui permet de remonter les raffinements jusqu'à la partition des singletons. Afin d'aller plus loin, j'ai aussi créé une base de données de l'ensemble des partitions et du pourcentage de graphes reconstruisables pour chacune d'entre elles. En effet, cette base de données contient toutes les partitions, et à une partition donnée et un ordre de graphe donnée est associé un nombre entre 0 et 1 représentant le pourcentage de graphes de cet ordre reconstruisables par la partition (1 signifiant que tous les graphes sont reconstruisables). La base de données est de la forme [partition => [n => pourcentage de graphe reconstruisible]]. Pour créer cette base de données, j'ai écrit un programme qui s'appelle “CreationRaffinement.jl”, et qui la remplit au fur et à mesure. Pour ce faire, il prend une partition de la base, un ordre  $n$ , et, à la manière de “EnumerateClasses.jl”, énumère tous les graphes, puis renvoie le nombre de classes de taille 1, qui correspondent donc aux graphes reconstruisables. J'ai également écrit un programme qui s'appelle “LectureRaffinement.jl”, qui lit cette base de données, et permet de faire des requêtes dessus.

Afin d'aller plus loin, j'ai fait les programmes *Connexite\_Raffinement.jl* et *composante\_connexes.jl*. L'objectif de ces programmes est de trouver des ensembles de partitions ayant des propriétés similaires et donnant un niveau d'information similaire et donc, en étudiant une partition, il sera facile d'étendre

les résultats trouvé aux autres partitions similaire. Pour ce faire, on utilise un graphe de reconstruction. Chaque noeud représente une partition et une arrête entre deux partitions signifie que l'une des partitions est un raffinement de l'autre et que les deux partitions ont le même nombre de graphe reconstructible. *Connexite\_Raffinement.jl* prend une partition *RAFFINEMENT\_DEPART*, et rend toutes les partitions étant des raffinement de *RAFFINEMENT\_DEPART*, qui sont dans la même composante connexe dans le Graphe de reconstruction. Notons que nous parcourrons ce graphe seulement en descendant, en direction des singletons ). *composante\_connexes.jl* quand a lui rend l'entièreté de la composante connexe de *RAFFINEMENT\_DEPART*. Le résultats est donc un ensemble de partition qui ont toutes le même nombre de graphe reconstructible. De plus, pour toute partition de cet ensemble, il existe un chemin de partition de cet ensemble, permettant d'aller à *RAFFINEMENT\_DEPART*. Notons que le graphe se parcours de façon non orienté, nous pouvons donc descendre et monter.

## 2.2 Requête de parité : J4

Intéressons-nous à la requête de parité. Cette requête est noté *J4*, *J* car, dans le papier [Galliot et al., 2025]), la requête de parité était notée *J3*, et le 4 signifie que l'on fait des requêtes sur des sous-ensembles d'ordre 4.

Après avoir utilisé l'outil informatique pour obtenir une intuition sur les résultats que l'on pouvait établir, nous les avons ensuite prouvés : voir la section qui suit.

### 2.2.1 Résultats sur *J4*

Contexte et intérêt :

Ici nous considérons des requêtes de taille 4. Les requêtes nous indiquent donc, pour chaque sous-graphe de taille 4, si son nombre d'arêtes est pair ou impair. Cette requête est intéressante car le cardinal de la partition associée est de deux (l'ensemble des graphes où le nombre d'arêtes  $m$  est pair, et celui où  $m$  est impair). Ainsi, de nombreuses partitions sont des raffinements de *J4*, ce qui permet de pouvoir réutiliser les résultats obtenus pour *J4*.

Requête sur la parité :

On sait que deux graphes sont dans la même classe si et seulement si leur différence symétrique est dans la même classe que  $I_n$ , c'est-à-dire le stable d'ordre  $n$  (cf le lemme 5.1 dans l'article [Galliot et al., 2025]) )

Ainsi, si l'on connaît  $\mathcal{C}(Q_{I_n}^{J_4})$  (la classe d'équivalence du stable de taille  $n$  pour la requête de parité sur 4 sommets) et que l'on se donne un graphe  $G$ , il est possible de reconstruire  $\mathcal{C}(Q_G^{J_4})$ . En effet, il suffit de faire la différence symétrique entre  $G$  et chacun des graphes de  $\mathcal{C}(Q_{I_n}^{J_4})$ . On voit donc que toutes les classes ont le même cardinal. Par conséquent, connaître tout  $\mathcal{C}(Q_{I_n}^{J_4})$  permet de connaître toutes les classes.

**Proposition 2.2.1.** *Pour  $n = 4$  les graphes étant dans la même classe que  $I_n$  sont tous les graphes ayant un nombre pair d'arêtes. C'est-à-dire que, à isomorphisme près, ce sont ceux de la figure 2.1 :*

**Proposition 2.2.2.** *Si  $G = (V, E)$  est un graphe d'ordre 5 tel que  $G \in \mathcal{C}(Q_{I_5}^{J_4})$ , alors,  $|E|$  est pair et tous ses sommets ont un degré pair.*

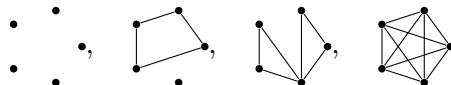
*Démonstration.* Montrons pour commencer que le nombre d'arêtes  $|E|$  de  $G$  est pair. Supposons par l'absurde que  $|E|$  est impair, et supposons de plus que  $G$  a un sommet de degré pair, noté  $v_i$ . Ainsi,

FIGURE 2.1 – Graphes représentant la classe  $\mathcal{C}(Q_{I_n}^{J_4})$  pour  $n = 4$  (à isomorphisme près).

en effectuant une requête sur  $G \setminus \{v_i\}$ , nous avons  $|E| - \deg(v_i)$  arêtes, ce qui fait donner un nombre impair d'arêtes (puisque  $|E|$  est impair et  $\deg(v_i)$  est pair par hypothèse). Or,  $G \in \mathcal{C}(Q_{I_5}^{J_4})$  et donc la requête renvoie la réponse “pair”, ce qui est impossible car le sous-graphe induit par ce sous-ensemble de sommets possède un nombre impair d'arêtes. Ainsi,  $G$  ne possède aucun sommet de degré pair. Or,  $G$  est de degré 5 donc la somme des degrés des sommets de  $G$  est impair, ce qui est impossible. Ainsi,  $G$  possède un nombre pair d'arêtes.

Supposons maintenant que  $G$  ait un sommet de degré impair, noté  $v_j$ . Ainsi, en faisant la requête  $G \setminus \{v_j\}$ , nous avons  $|E| - \deg(v_j)$  arêtes, ce qui fait un nombre impair d'arêtes (puisque  $|E|$  est pair et  $\deg(v_j)$  est impair par hypothèse), ce qui est impossible car  $G \in \mathcal{C}(Q_{I_5}^{J_4})$ . Ainsi, tous les sommets de  $G$  ont un degré pair.  $\square$

**Proposition 2.2.3.** Pour  $n = 5$ , les graphes étant dans la même classe que  $I_n$  sont (à isomorphisme près) le stable,  $C_4 + K_1$ , le papillon et la clique, représenté respectivement ci-après.



*Démonstration.* Soit  $G$  un graphe appartenant à  $\mathcal{C}(Q_{I_5}^{J_4})$ .

Nous savons que chaque sommets de  $G$  est de degré pair, donc égal à 0, 2 ou 4 (car  $n = 5$ ). Donc s'il y a 3 ou 5 sommets isolés (4 étant impossible),  $G$  est nécessairement le stable. S'il y a exactement deux sommets isolés, alors les trois restants ne peuvent être de degré 0 ou 4, ils sont donc de degré 2, ce qui impose qu'ils forment un triangle.  $G$  aurait ainsi trois arêtes, ce qui est impair et donc impossible. S'il y a exactement un sommets isolé, alors les quatre autres sommets forment un ensemble avec un nombre pair d'arêtes, sans sommets isolés (ce qui exclut  $P_3 + K_1$  et le stable), où chaque sommet est de degré pair (ce qui exclut  $K_{3,1}$ ,  $2K_2$ , Paw et la clique). Le seul graphe possible est donc  $C_4$ .

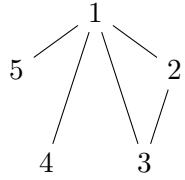
Supposons à présent qu'il n'y ait aucun sommet isolé. Soit tous les sommets sont de degré 4, ce qui nous donne la clique. Soit tous les sommets sont de degré 2, ce qui nous donne un  $C_5$  ce qui est exclu car un  $C_5$  possède 5 arêtes, ce qui est impair. Soit il y a au moins un sommet de degré 4 (noté sans perte de généralité  $v_1$ ) et au moins un sommet de degré 2 (noté  $v_2$ ). Ainsi,  $v_1$  est universel (*i.e.* relié à tous les autres sommets) et  $v_2$  est relié à  $v_1$  et un autre sommet, noté  $v_3$  ( configuration 1 2.2a ).

Il y a ensuite deux possibilités :

Soit  $v_3$  est de degré 4, et, dans ce cas, les deux derniers sommets ne peuvent être reliés entre eux (car ils sont déjà reliés à  $v_1$  et  $v_3$ , qu'ils ne peuvent pas être reliés à  $v_2$ , et que leur degré doit être pair) ; ils sont donc de degré 2. Ainsi nous avons 7 arêtes (configuration 2 ??) ce qui est impair donc exclu.

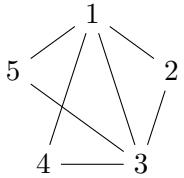
Soit  $v_3$  est de degré 2, ce qui implique que les deux derniers sommets sont reliés entre eux (puisque ils sont reliés à  $v_1$ , mais pas à  $v_2$  ni à  $v_3$ , et que leur degré doit être pair) ( configuration 3 ??). Nous obtenons ainsi le papillon.

Notons que, réciproquement, les configurations évoquées (le stable, la clique, le papillon et  $C_4 + K_1$ ) respectent bien les requêtes et font bien partie de  $\mathcal{C}(Q_{I_5}^{J_4})$ .  $\square$

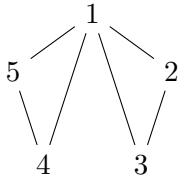


Configuration 1

Configuration 1



Configuration 2



Configuration 3

FIGURE 2.2 – Configuration 1 et 2 évoquées durant la preuve de la proposition 2.2.3

**Lemme 2.2.4.** Pour tout  $n \geq 6$ , la classe d'équivalence de  $I_n$  par la partition  $J_4$  est composée uniquement du stable d'ordre  $n$  et de la clique d'ordre  $n$ .

*Démonstration.* On procède par récurrence sur  $n$ .

Initialisation (cas  $n = 6$ ) :

Prenons un graphe  $G = (V, E)$  d'ordre 6 appartenant à la même classe que  $I_6$  pour la partition  $J_4$ , tel que  $G$  n'est pas le stable. Ainsi,  $G$  possède une arête, disons  $e = (v_1, v_2)$ . Considérons un sous-ensemble de 5 sommets contenant cette arête, par exemple  $S_1 = (v_1, v_2, v_3, v_4, v_5)$ . Toutes les requêtes faites sur tout quadruplet contenu dans  $V$ , en particulier dans  $S_1$ , renvoient pair et donc  $S_1 \in \mathcal{C}(Q_{I_5}^{J_4})$ .

Ainsi, par la proposition 2.2.3,  $G[S_1]$  est soit un papillon, soit un  $C_4 + K_1$ , soit un  $K_5$  (ce ne peut pas être le stable puisque  $G[S_1]$  contient l'arête  $e$ ).

Cas 1 :  $G[S_1] = C_4 + K_1$ .

Supposons que  $G[S_1]$  soit  $C_4 + K_1$  (nous savons donc qu'il y a exactement 4 arêtes dans  $G[S_1]$ ), et supposons de plus que le sommet isolé soit  $v_5$ . Ainsi,  $(v_1, v_2, v_3, v_4)$  induit un  $C_4$ . Sélectionnons l'ensemble de sommets  $S_2 = (v_1, v_2, v_3, v_5, v_6)$  : nous savons que cet ensemble appartient à  $Q_{I_5}^{J_4}$ . Cet ensemble ne peut pas être le stable, puisqu'il contient l'arête  $e = (v_1, v_2)$ , et il ne peut pas non plus être la clique, puisqu'il contient la non-arête  $(v_1, v_5)$  ( $v_5$  étant un sommet isolé du cycle  $(v_1, v_2, v_3, v_4)$ ). De plus, étant donné que le sommet  $v_5$  n'est pas relié aux sommets  $(v_1, v_2, v_3)$ , il possède 3 non-arêtes incidentes, or ce n'est le cas d'aucun sommet dans le papillon. Donc, le sous-ensemble  $S_2$  n'induit pas le papillon, et par conséquent il induit un  $C_4 + K_1$ , où  $v_5$  reste le sommet isolé.

Finalement, prenons le sous-ensemble  $S_3 = (v_1, v_2, v_3, v_4, v_6)$ . D'après ce qui a été prouvé précédemment, le sous-graphe induit par ce sous-ensemble contient 2  $C_4$  différents, induits respectivement par les ensembles de sommets  $(v_1, v_2, v_3, v_6)$  et  $(v_1, v_2, v_3, v_4)$ . Comme pour  $S_1$  et  $S_2$ , le graphe  $G[S_3]$  appartient à  $Q_{I_5}^{J_4}$ . Ce n'est pas la clique, puisque le  $C_4$   $(v_1, v_2, v_3, v_4)$  contient une non-arête. De plus, ni le papillon, ni le  $C_4 + K_1$ , ni le stable ne contiennent deux  $C_4$ . Ainsi, il y a contradiction avec la proposition 2.2.3. On en déduit que le cas 1 où  $G[S_1] = C_4 + K_1$  est impossible.

Cas 2 :  $G[S_1] = K_5$  ou le papillon.

Supposons que  $G[S_1]$  soit le papillon. Supposons aussi, sans perte de généralité, que  $v_1$  soit le sommet universel dans  $G[S_1]$ , et que les deux autres arêtes soient  $[v_2, v_3]$  et  $[v_4, v_5]$ . Considérons le sous-graphe induit par le sous-ensemble de sommets  $v_1, v_2, v_4, v_5, v_6$ . Ce sous-graphe possède un triangle, ainsi que l'arête  $[v_1, v_2]$ , et ne peut donc pas être le stable ou le  $C_4 + K_1$ . Par conséquent, c'est soit un papillon avec  $v_1$  comme sommet universel, soit une clique. Dans les deux cas, nous avons un triangle induit par  $v_1, v_2, v_6$ . Considérons maintenant le sous-graphe induit par le sous-ensemble de sommets  $v_1, v_2, v_3, v_5, v_6$ . Le sommet  $v_1$  y est de degré 4, et le sommet  $v_2$  y est de degré 3 (car il appartient aux deux triangles induits par  $v_1, v_2, v_6$  et par  $v_1, v_2, v_3$ , respectivement). Or, ce sous-graphe appartient à  $Q_{I_5}^{J_4}$ , et le seul graphe de  $Q_{I_5}^{J_4}$  qui contient deux sommets de degré 3 ou plus est la clique d'ordre 5. Ainsi, les sommets  $v_1, v_2, v_3, v_5, v_6$  induisent une clique. Considérons le sous-graphe induit par les sommets  $v_1, v_2, v_3, v_4, v_6$  : il contient au moins 7 arêtes (car les sommets  $v_1, v_2, v_3, v_6$  induisent une clique et que l'on a en plus l'arête  $[v_1, v_4]$ , puisque  $v_1$  est universel), or le seul graphe de  $Q_{I_5}^{J_4}$  contenant 7 arêtes ou plus est la clique. Ainsi, les sommets  $v_1, v_2, v_3, v_4, v_5$  induisent également une clique (les sommets  $v_4$  et  $v_5$  étant pour rappel reliés par une arête), ce qui nous donne que les sommets  $v_1, v_2, v_3, v_4, v_5, v_6$  induisent à leur tour une clique.

Ainsi, d'après les cas 1 et 2, si un graphe  $G$  appartient à  $Q_{I_6}^{J_4}$  et qu'il possède au moins une arête (*i.e.*,  $G$  n'est pas  $I_6$ ), alors c'est nécessairement une clique.

Réciproquement, la clique d'ordre 6 appartient bien à  $Q_{I_6}^{J_4}$ . En effet, les requêtes sont de taille 4, donc chaque requête renvoie un  $K_4$ , qui contient 6 arêtes, ce qui est bien pair. Ainsi,  $\mathcal{C}(Q_{I_6}^{J_4}) = \{K_6, I_6\}$ .

Hérité :

Supposons la propriété vraie pour un  $n \geq 6$ . Prenons un graphe  $G = (V, E)$  d'ordre  $n + 1$  appartenant à la même classe d'équivalence que  $I_n$  pour la partition  $J4$ , tel que  $G$  n'est pas le stable  $I_n$ . Ainsi, pour chaque requête faite sur  $G$ , la requête renvoie “pair”. De plus,  $G$  a au moins une arête  $e$  : sans perte de généralité, supposons que  $e = [v_1, v_2]$ .

Considérons le sous-graphe  $G \setminus v_{n+1}$ , qui est d'ordre  $n$ . Chaque requête de parité faite sur ce graphe est une requête de parité sur un sous-graphe de  $G$  et donc renvoie “pair”, ce qui signifie que ce sous-graphe est lui aussi dans la même classe d'équivalence que  $I_n$  pour la partition  $J4$ . Ainsi, par hypothèse de récurrence,  $G \setminus v_{n+1}$  est soit un stable, soit une clique. Or  $G \setminus v_{n+1}$  possède une arête ( $e$ ), et donc  $G \setminus v_{n+1} = K_n$ .

Considérons maintenant le sous-graphe  $G \setminus v_1$ , qui est lui aussi d'ordre  $n$ . Ce graphe est dans la même classe d'équivalence que  $I_n$  pour la partition  $J4$ , et donc, comme il possède au moins une arête (par exemple  $[v_2, v_3]$ , puisque  $G \setminus v_{n+1}$  est un graphe complet, d'après le paragraphe précédent), on a que  $G \setminus v_1$  est une clique, par hypothèse de récurrence. Ainsi, le sommet  $v_{n+1}$  est relié à tous les sommets de  $G$ , à l'exception (possiblement) de  $v_1$ . Pour finir, on considère le sous-graphe  $G \setminus v_2$ , qui est également d'ordre  $n$ , et, par les mêmes arguments, on a que  $G \setminus v_2$  est une clique. Cela implique que  $(v_1, v_{n+1}) \in E$ . Ainsi, comme  $G \setminus v_{n+1}$  est une clique, et que  $v_{n+1}$  est relié à tout les sommets (y compris  $v_1$ ),  $G$  est nécessairement une clique. Par conséquent, tout graphe d'ordre  $n + 1$  possédant

au moins une arête et étant dans la même classe d'équivalence que  $I_{n+1}$  pour la partition  $J4$  est une clique. En outre,  $K_{n+1}$  appartient bien à la même classe d'équivalence que  $I_{n+1}$  pour la partition  $J4$  (car  $K_4$  contient 6 arêtes, ce qui est pair), et donc  $\mathcal{C}(Q_{I_n}^{J_4})$  contient seulement  $I_{n+1}$  et  $K_{n+1}$ , ce qui conclut la preuve.  $\square$

On a ainsi répondu entièrement à la question de la reconstructibilité pour  $J4$ .

Pour aller plus loin, nous nous sommes aussi intéressés aux partitions d'ordre supérieur, mais  $\mathcal{C}(Q_{I_n}^{J_5})$  ne semble pas avoir de forme simple. Cependant, nous avons poussé plus loin en cherchant à comprendre ce qu'il en était pour  $J6$ .

J'ai donc écrit (en Julia) une variante plus spécialisée et optimisée pour cette tâche du programme *EnumerateClasses.jl*, que j'ai appelée *optimiseI8J6.jl*. La principale différence avec le programme initial est qu'il génère les graphes un à un, fait les requêtes directement, et, si ce graphe appartient à la classe de  $I_n$ , l'écrit directement dans le document LaTeX. Cette méthode permet de ne rien garder dans la RAM, et de conserver en mémoire seulement les graphes qui nous intéressent. Il en résulte que, pour  $n = 8$ ,  $\mathcal{C}(Q_{I_8}^{J_6})$  contient seulement le stable et la clique d'ordre 8. Cependant, nous ne l'avons pas encore prouvé formellement (*i.e.*, sans l'aide de ce programme). En effet, il faudrait utiliser une autre méthodologie, car la classe d'équivalence de  $I_7$  pour la partition  $J6$  est bien plus fournie, et donc il serait plus difficile et long d'utiliser les résultats liés à cette classe.

## 2.3 Résultats sur $S4$

Dans cette partie, nous nous intéressons à la partition associée aux singletons, notée  $S4$ . En d'autres termes, chaque partie de cette partition contient un unique graphe d'ordre 4, et chaque requête indique, pour chaque sous-graphe d'ordre 4, à quel graphe d'ordre 4 de cette partition il est isomorphe. Notons que cette partition est celle qui donne l'information la plus fine : en effet, elle est un raffinement de toutes les autres partitions, et aucune autre partition n'est un raffinement de celle-ci, donc nous pouvons utiliser les résultats précédents concernant  $J4$ .

Ainsi, tous les graphes non reconstructibles par cette partition ne le sont pour aucune autre. Notons que, à partir de l'ordre 6, cette partition permet de reconstruire tous les graphes. En effet, cette partition est un raffinement de  $E4$  (la partition qui donne l'information sur le nombre d'arêtes), qui elle peut reconstruire tous les graphes à partir de l'ordre 6.

Dans un premier temps, j'ai modifié le programme *EnumerateClasses.jl*, afin qu'il puisse comparer deux partitions  $P$  et  $R$ , où  $R$  est un raffinement de  $P$ . En effet, ce programme affiche chaque classe de  $P$ , puis les classes de  $R$  qui sont des sous-classes de cette classe de  $P$ . L'objectif est de déterminer quels graphes sont reconstructibles avec une partition et pas avec l'autre. Malheureusement, ce programme ne nous a pas donné d'information exploitable, car les différences entre  $S4$  et  $J4$  ne sont pas basées sur une seule et simple propriété.

**Proposition 2.3.1.** *Les graphes d'ordre 5 non reconstructibles par  $S4$  sont les  $C_5$ , ainsi que les échanges d'arêtes dont le cinquième sommet est de degré 0, 2 ou 4.*

Par échange d'arêtes nous entendons qu'il y a 4 sommets  $(v_1, v_2, v_3, v_4)$ , tel que  $[v_1, v_2]$  et  $[v_3, v_4]$  sont des arêtes du graphes et que  $[v_1, v_3]$  et  $[v_2, v_4]$  sont des non-arêtes pour ce graphes. Faire l'échange d'arête, c'est transformer  $[v_1, v_2]$  et  $[v_3, v_4]$  en non-arêtes et  $[v_1, v_3]$  et  $[v_2, v_4]$  en arêtes pour ce graphe.

*Démonstration.* Soit  $G$  et  $H$  deux graphes d'ordre 5 tels que  $G \equiv^{S4} H$ . Ainsi,  $S4$  étant un raffinement de  $J4$ , on a en particulier que  $G \equiv^{J_4} H$ . Par conséquent, on a  $G \triangle H \in \mathcal{C}(Q_{I_5}^{J_4})$ , ce qui signifie que leur différence symétrique  $G \triangle H$  est soit  $I_5$ , soit un papillon, soit un  $C_4 + K_1$ , soit un  $K_5$ .

Cas 1 :  $G \Delta H = I_5$ .

Dans ce cas,  $G$  et  $H$  ont les même arêtes, et donc  $G = H$ .

Cas 2 :  $G \Delta H = K_5$ .

Dans ce cas,  $G$  et  $H$  sont complémentaires. Ainsi, chaque sous-graphe étiqueté d'ordre 4 dans  $G$  est le complémentaire du sous-graphe étiqueté associé dans  $H$ . Mais comme les deux graphes appartiennent à la même classe, leur requêtes donnent les mêmes résultats, et donc les deux sous-graphes d'ordre 4, en plus d'être complémentaires, sont isomorphes au même singleton, et donc, isomorphes entre eux. Or,  $P_4$  est le seul singleton isomorphe à son complémentaire (cf figure ??). Donc, tous les sous-graphes d'ordre 4 de  $G$  sont des  $P_4$ . Montrons maintenant que  $G$  est un  $C_5$ . Pour ce faire, montrons que  $G$  a 5 arêtes, et que tous les sommets de  $G$  sont de degré 2. Étant donné que les requêtes sont de taille  $4 = n - 1$ , on a que le degré de tout sommet  $i$  est égal à  $|E|$  moins le nombre d'arêtes de la requête faite sur  $G \setminus i$ , qui est un  $P_4$  d'après ce qui a été établi précédemment. Par conséquent, en sommant sur tous les sommets de  $G$  on obtient que

$$\sum_{i \in V} (\deg(i)) = 5 \times |E| - 5 \times |E(P_4)|$$

D'où :

$$\sum_{i \in V} (\deg(i)) = 5 \times |E| - 5 \times 3$$

(Cela vient du fait qu'il y a 3 arêtes dans le sous-graphe résultant de chaque requête, puisque c'est un  $P_4$ , et qu'il y a 5 requêtes en tout, une pour chaque sommet de  $G$ .)

Ainsi, comme  $\sum_{i \in V} (\deg(i)) = 2 \times |E|$  dans tout graphe, on a :

$$2 \times |E| = 5 \times |E| - 5 \times 3$$

D'où  $|E| = 5$ , et le degré de chaque sommet est donc  $|E| - |E(P_4)| = 5 - 3 = 2$ . Ainsi, tous les sommets de  $G$  sont de degré 2, donc  $G$  est un ensemble de cycles, et comme  $G$  contient 5 arêtes,  $G$  est nécessairement un  $C_5$ .

Cas 3 :  $G \Delta H$  est un papillon.

Notons  $P = G \Delta H$  ce papillon, notons  $v_1$  le sommet universel dans  $P$ , et notons  $[v_2, v_3]$  et  $[v_4, v_5]$  les deux autres arêtes de  $P$ . En outre,  $G$  et  $H$  sont équivalents vis-à-vis de  $S4$ , et donc ils le sont également vis-à-vis de  $E4$  (requête portant sur le nombre d'arêtes, dont  $S4$  est un raffinement), ce qui implique que chaque sommet a le même degré dans  $H$  et dans  $G$ .

Supposons maintenant que l'arête  $[v_2, v_3]$  de  $P$  soit dans  $G$  (et donc qu'elle soit une non-arête dans  $H$ , par définition de  $P$ ). Il faut donc que la seconde arête incidente à  $v_2$  et à  $v_3$  soit dans  $H$  (et soit ainsi une non-arête dans  $G$ ), pour conserver l'égalité des degrés dans  $G$  et dans  $H$ . Ainsi, les arêtes  $[v_1, v_3]$  et  $[v_1, v_2]$  sont dans  $H$ . Le sommet  $v_1$  a donc deux arêtes incidentes dans  $P$  qui sont dans  $H$  (et qui sont par conséquent des non-arêtes dans  $G$ ). Il faut donc que ses deux autres arêtes incidentes dans  $P$  ( $[v_1, v_4]$  et  $[v_1, v_5]$ ) soient des arêtes dans  $G$ . Pour finir, la dernière arête  $[v_4, v_5]$  est dans  $H$ , pour les mêmes raisons. Nous avons donc (cf la configuration 0) :

Une fois que nous avons déterminé les arêtes qui sont différentes dans  $G$  et dans  $H$  (c'est-à-dire celles de  $P$ ), nous allons ensuite statuer sur les arêtes communes à  $G$  et  $H$ . En faisant une requête

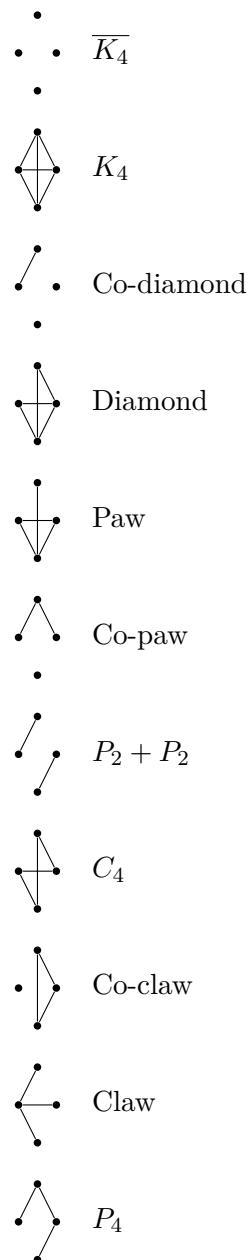
sur les 4 sommets  $v_1, v_2, v_3, v_4$ , on a, sans rajouter d'arête, un  $2K_2$  dans  $G$  et un  $P_3 + K_1$  dans  $H$  (cf la configuration 1). Ce ne sont pas les mêmes singletons, et il faut donc rajouter au moins une arête pour que cette requête renvoie le même singleton dans  $G$  et dans  $H$ . Notons aussi que rajouter deux arêtes créerait un  $C_4$  dans  $H$  mais pas dans  $G$  (cf la configuration 2), ce qui est donc impossible. Par conséquent, il faut rajouter une seule arête, incidente à  $v_4$  et, soit à  $v_3$ , soit à  $v_2$  (et, dans ce cas, nous avons un  $P_3$  dans  $G$  et dans  $H$ ; cf la configuration 3). Supposons, sans perte de généralité (les deux cas étant symétriques), que nous avons rajouté l'arête  $[v_2, v_4]$ . Ainsi, en faisant une requête sur les sommets  $v_1, v_2, v_3$ , et  $v_5$ , nous obtenons la même configuration, et nous avons donc deux possibilités :

- Soit nous rajoutons l'arête  $[v_2, v_5]$  dans  $G$  et dans  $H$ , mais, dans ce cas, en faisant la requête sur les sommets  $v_1, v_2, v_4$ , et  $v_5$ , nous nous retrouvons avec un  $C_4$  dans  $G$  et pas dans  $H$  (cf la configuration 4); cette possibilité est donc exclue.
- Soit nous rajoutons l'arête  $[v_3, v_5]$  dans  $G$  et dans  $H$ . Dans ce cas-là, nous avons un  $C_5$  dans  $G$  et dans  $H$ , donc toutes les requêtes renvoient  $P_3$ , et ainsi on a bien  $G \equiv^{S_4} H$ .

Cas 4 :  $G \Delta H = C_4 + K_1$ .

On a déjà établi que, pour chaque sommet, son degré dans  $G$  doit être égal à son degré dans  $H$ . Ainsi, dans ce cas, deux arêtes consécutives du  $C_4$  ne peuvent appartenir au même graphe, car sinon le sommet incident à ces deux arêtes aurait un degré supérieur dans l'un des graphes (cf la configuration 5). Isolons un des sous graphes d'ordre 4, contenant le sommet n'appartenant pas au  $C_4$ , et nommons le  $v_1$ . Ainsi, nous avons deux arêtes dont l'une est dans  $H$  et l'autre dans  $G$ , nommons  $v_2$ , le sommet incident à ces deux arêtes. Ainsi, nous avons les arêtes  $(v_2, v_3)$ ,  $(v_2, v_4)$  qui sont soit dans  $G$ , soit dans  $H$ . Rappelons que les autres arêtes appartiennent soit aux deux graphes  $H$  et  $G$ , soit à aucun des deux graphes. La question est donc de savoir si rajouter chaque arête change le singleton obtenu. Par symétrie, l'arête  $(v_1, v_2)$  ne change pas la configuration du graphe (en effet, les arêtes  $(v_2, v_3)$  et  $(v_2, v_4)$ , sont toutes deux incidente en un sommet). Il en est de même pour  $(v_3, v_4)$ . Pour ces deux arêtes, leur appartenance ou non au graphe ne change donc pas le singleton obtenu. Il ne reste donc qu'à statuer sur les arêtes  $(v_1, v_4)$  et  $(v_1, v_3)$ . Elles sont chacune incidente à une seule des arêtes appartenant soit à  $H$ , soit à  $G$ . Et donc, si une seule des deux appartient au sous-graphe, par exemple  $(v_1, v_4)$ , on aura d'un côté cette arête adjacente à  $(v_2, v_4)$  (dans  $H$ ) et d'un autre, cette arête non adjacente à  $(v_2, v_3)$  (dans  $G$ ). Ce qui ne créera pas le même singleton. Il faut donc que les deux arêtes, communément soit, ou ne soit pas dans le sous-graphe. Or ces deux arêtes relient deux sommets qui sont diagonaux dans le  $C_4$ . Ainsi, on en déduit que, pour les arêtes allant de  $v_1$  vers le  $C_4$ , les arêtes reliant deux sommets diagonaux doivent communément appartenir ou ne pas appartenir ou ne pas appartenir au graphe. On a donc, concernant les arêtes allant de  $v_1$  vers le  $C_4$ , 0 arête (aucune des arêtes), 2 arêtes (reliant seulement une seule des diagonales) ou 4 arêtes (reliant les deux diagonales). Réciproquement les échanges d'arêtes dont le cinquième sommet est de degré 0, 2 ou 4. En effet, comme précisé, les sous-graphes contenant  $v_1$  sont isomorphe au même singleton. Concernant le dernier sous-graphe, celui ne contenant pas  $v_1$ , mais contenant  $v_4$ , les deux arêtes diagonales ne change pas le singleton obtenu, nous avons donc un  $2K_2$  (aucune arêtes diagonales), un  $P_3$  (une seule arête diagonale) ou un  $C_4$  (les deux arêtes diagonales). Ainsi, tous les sous-graphes sont isomorphe.

□

FIGURE 2.3 – Graphes représentant la classe  $\mathcal{C}(Q_{I_n}^{J_4})$  pour  $n = 4$  à isomorphisme près.

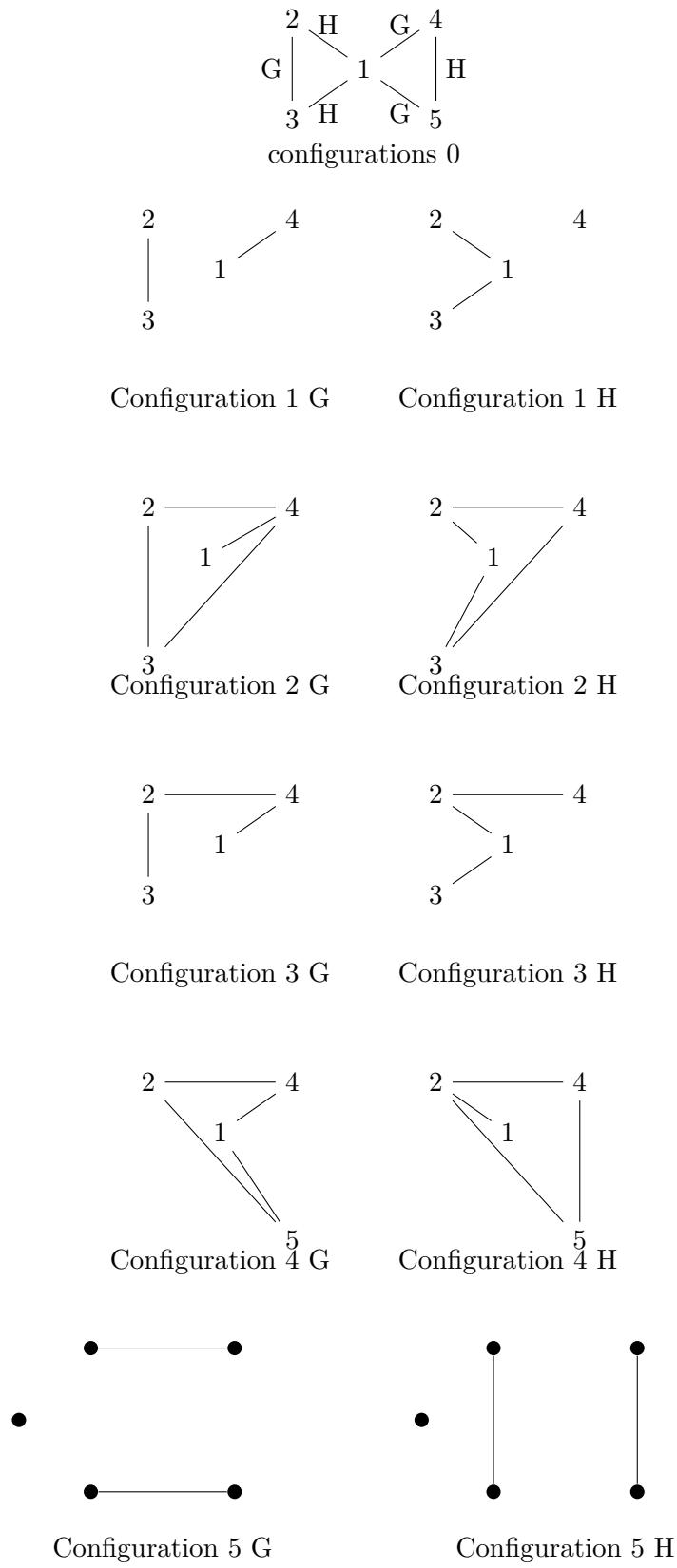


FIGURE 2.4 – Différentes configurations

# Chapitre 3

# Conclusion et Prolongements

## 3.1 Conclusion

Ce stage esquisse la problématique de la reconstruction de graphes à partir de requêtes, en particulier dans le cas des partitions d'ordre 4. Après avoir étudié et mis en œuvre des outils d'énumération permettant de regrouper les partitions donnant un niveau similaire d'information, j'ai pu mettre en évidence plusieurs résultats théoriques notables, notamment concernant la requête de parité  $J_4$  et la requête des singletons  $S_4$ . Ces travaux ont montré que certaines partitions permettent une reconstruction complète à partir d'un certain ordre, tandis que d'autres présentent des limites structurelles.

Au-delà des résultats obtenus, des outils logiciels (en Julia) ont également été développé, facilitant l'analyse et la manipulation de partitions, la génération de graphes, ainsi que la mise en place d'une base de données dédiée aux pourcentages de graphes reconstructibles. Ces contributions constituent une base solide pour de futures recherches, qu'il s'agisse de l'étude de nouvelles partition ou d'étendre les résultats obtenue à d'autres partitions.

## 3.2 Prolongements

Pour aller plus loin, deux possibilité sont envisagé :

- trouver les conditions nécessaire et suffisante pour la reconstruction des graphes, pour d'autres partitions non mentionné dans ce rapport
- trouver des algorithmes en délai polynomial permettant de reconstruire les graphes seulement à partir de leur requêtes.
- Continuer d'explorer le cas de la requête de singleton pour  $k = n - 1$  avec  $n$  l'ordre du graphe

# Annexe A

## DD&RS -

### A.1 Entreprise

#### 1. Contexte institutionnel

Le Cnam a engagé depuis 2025 un *schéma directeur DD&RS*, visant une labellisation **CIRSES en 2027**. Les thématiques prioritaires portent sur :

- la **décarbonation des activités**,
- la **sobriété énergétique**,
- la **gestion durable de la biodiversité**,
- l'intégration des compétences **TEDS** dans le premier cycle.

#### 2. Diagnostic

- **Bilan carbone 2023** : 8,7 tCO<sub>2</sub>/équivalent-temps plein.
- Principales sources d'émissions :
  - Achats (51%),
  - Actifs (21,9%),
  - Énergie (14,4%),
  - Déplacements (10,3%).

#### 3. Démarche participative

Au printemps 2025, des *ateliers d'intelligence collective* ont réuni près de 60 participants (administratifs, enseignants-chercheurs, directions pilotes) autour de trois axes :

- Politique sociale,
- Gestion environnementale,
- Formation et recherche.

Ces ateliers ont permis d'identifier des **actions prioritaires** et des **ambassadeurs DD&RS** chargés d'assurer la diffusion et l'animation de ces démarches.

#### 4. Focus « Formation et recherche »

Les recommandations issues de l'atelier thématique :

1. Intégrer le DD&RS dans les programmes et enseignements,
2. Développer les compétences DD&RS des personnels,

3. Promouvoir une recherche responsable et ouverte :
  - bilan carbone des projets (initiative Labo 1.5),
  - financement responsable,
  - science ouverte (HAL, diffusion des connaissances).
4. Renforcer la diffusion de la culture scientifique et le rôle international du Cnam dans le champ DD&RS.

## 5. Perspectives

- **2025–2026** : finalisation et validation du schéma, mise en place des ambassadeurs et du service DD&RS, communication et déploiement opérationnel.
- **Objectif 2027** : obtention du **label DD&RS CIRSES**.

## A.2 Bilan carbone

Ce stage étant principalement composé de preuve et de programme informatique tournant rapidement, la majeur partie des émission carbone proviennent du déplacement. Mon trajet consistait en 10.8km de voiture, puis une section de RER. Le tout émet donc  $10.8 * 99 + 134 = 1203.2g$  de Co2 par trajet, ce qui correspond à 96.256 kg de Co2 émit par ce stage. Cela correspond à 3.56 kg de boeuf ou 371 km en avion.

## **Annexe B**

### **code**

Tout le code produit pendant ce stage se trouve sur un git : <https://gitlab.com/mouton5000/graph-reconstruction-queries>.

# Bibliographie

- [Galliot et al., 2025] Galliot, F., La, H., Maistre, R., Petiteau, M., and Watel, D. (2025). Graph reconstruction from queries on triples. In *EUROCOMB 2025*. 3, 4, 6
- [Kelly, 1957] Kelly, P. J. (1957). A congruence theorem for trees. 4