

Les graphes 2-intervallaires

Philippe Gambette

ENS Cachan

Stage de Master MPRI

Sous la direction de Michel Habib

LIAFA, Paris VII

Printemps-été 2006

Résumé :

Les graphes 2-intervallaires, généralisation des classiques graphes d'intervalles, ont récemment fait l'objet de l'attention des bioinformaticiens, pour l'étude et la comparaison de structures secondaires de l'ARN.

Nous synthétisons tout d'abord les connaissances sur cette classe de graphes, puis les manières d'utiliser la théorie des graphes afin de résoudre le problème de recherche de plus long motif de 2-intervalles respectant certaines contraintes, en améliorant au passage la complexité de quelques algorithmes. Nous prouvons de plus l'inclusion stricte de la classe des graphes 2-intervallaires équilibrés dans celle des graphes 2-intervallaires. Enfin, nous étudions un autre objet, restriction des ensembles de 2-intervalles, et qui a aussi bénéficié d'une attention récente de la part des bioinformaticiens, les séquences arc-annotées, afin de présenter un algorithme de génération exhaustive et des formules de dénombrement.

Mots-clés : théorie des graphes, bioinformatique, ordonnancement, 2-intervalles, séquences arc-annotées, recherche de motifs, stable maximum.



LIAFA



Université Paris VII

Table des matières

1	Introduction	7
2	Graphes 2-intervallaires	9
2.1	Définitions	9
2.1.1	Définition des 2-intervalles	9
2.1.2	Relations entre 2-intervalles	9
2.1.3	Définition des graphes 2-intervallaires	10
2.2	Motivations	11
2.2.1	2-intervalles et ordonnancement	11
2.2.2	2-intervalles et ADN	12
2.2.3	2-intervalles et ARN	12
2.2.4	2-intervalles et analyse musicale	13
2.3	2-intervallaires et classes apparentées	14
2.3.1	Une sous-classe connue, les graphes d'intervalles	14
2.3.2	Les graphes R -2-intervallaires	14
2.3.3	Eclatement de séquences arc-annotées	16
2.3.4	Classe des 2-intervallaires	17
2.3.5	Classe des $\{\curvearrowright, \sqsubset\}$ -(1,1)-intervallaires	18
2.3.6	Classe des $\{\emptyset\}$ -(1,1)-intervallaires et des $\{\curvearrowright, \emptyset\}$ -(1,1)-intervallaires	21
2.3.7	Classe des $\{\curvearrowright, \emptyset\}$ -2-intervallaires	21
2.3.8	Classe des $\{\emptyset\}$ -2-intervallaires	21
2.3.9	Classe des $\{\curvearrowright, <, \emptyset\}$ -2-intervallaires	21
2.3.10	Classe des $\{\sqsubset\}$ -(1,1)-intervallaires et des $\{\curvearrowright, <, \emptyset\}$ -(1,1)-intervallaires	22
3	Stable max et recherche de motifs sur les 2-intervalles	25
3.1	2-IP, la recherche de plus long motif respectant certaines contraintes	25
3.2	2-IP(GENERAL, $\{<, \sqsubset, \emptyset\}$)	26
3.3	2-IP(DISJOINT, $\{<, \sqsubset, \emptyset\}$)	27
3.4	2-IP(DISJOINT, $\{<, \emptyset\}$)	27
3.4.1	Approche par la théorie des graphes	27
3.4.2	Approche par programmation dynamique	27
3.4.3	Perspectives	29
3.5	2-IP(DISJOINT, $\{\sqsubset, \emptyset\}$)	29
3.6	2-IP(GENERAL, $\{<, \sqsubset\}$)	30
3.7	2-IP(GENERAL, $\{\emptyset\}$)	30
3.8	2-IP(DISJOINT, $\{\emptyset\}$)	30
3.9	2-IP(GENERAL, $\{<\}$)	30
3.10	2-IP(GENERAL, $\{\sqsubset\}$)	30

4	2-intervalles équilibrés	31
4.1	Inclusion stricte de la classe des graphes 2-intervallaires équilibrés	31
4.1.1	Etape 1 : huit 2-intervalles pour simuler un intervalle	31
4.1.2	Etape 2 : huit 2-intervalles pour contraindre l'ordre de placement de 7 intervalles	33
4.1.3	Etape 3 : 7 intervalles dans un ordre contraint pour empêcher l'équilibrage de trois 2-intervalles	34
4.2	Bilan sur les inclusions de classes 2-intervallaires équilibrées	36
5	Annexe : théorie des graphes	37
5.1	Quelques définitions	37
5.1.1	Des définitions de base	37
5.1.2	Des classes de graphes	37
5.1.3	Des transformations de graphes	39
5.2	Quelques problèmes sur les graphes	40
5.2.1	Couverture de sommets	40
5.2.2	Ensemble indépendant	40
5.2.3	Clique	41
5.2.4	Coloration	41
5.2.5	Domination	41
5.2.6	Couplage maximum	41
5.2.7	Coupe maximale	42
5.2.8	Cycle hamiltonien	42
5.2.9	Cycle eulérien	42
5.2.10	Dénombrement	42
6	Annexe : dénombrement des séquences arc-annotées	43
6.1	Majoration du nombre de séquences arc-annotées	43
6.2	Génération	43
6.3	Dénombrement à nombre fixé d'arcs	44
6.4	Dénombrement à nombre fixé d'extrémités	48
7	Conclusion	49
7.1	Remerciements	49
	Index	50
	Bibliographie	53

Table des figures

2.1	Diverses restrictions sur le support d'un ensemble de 2-intervalles	10
2.2	Positions possibles de 2-intervalles disjoints	10
2.3	Récapitulatif des 4 types de positions possibles pour deux 2-intervalles	10
2.4	Graphe de 2-unions	11
2.5	ARN et 2-intervalles	13
2.6	Représentation cyclique de l'échelle chromatique et relations d'inversion et de transposition	14
2.7	Classes de graphes R -...-intervallaires	15
2.8	Nombre de graphes non étiquetés et de séquences arc-annotées aux arcs non étiquetés	16
2.9	Eclatement d'une extrémité partagée dans une séquence arc-annotée	17
2.10	Un graphe biparti complet et sa réalisation $\{\curvearrowright, \square\}$ -(1,1)-intervallaire	19
2.11	Des cycles et leur réalisation $\{\curvearrowright, \square\}$ -(1,1)-intervallaire	19
2.12	Contrexemple de graphe non- $\{\curvearrowright, \square\}$ -(1,1)-intervallaire	19
2.13	Séquence de génération et de filtrage de séquences arc-annotées pour un contreexemple de graphe non- $\{\curvearrowright, \square\}$ -(1,1)-intervallaire	20
2.14	Graphes de chevauchement et (1,1)-intervalles entrelacés	21
2.15	2-intervalles entrelacés et trapézoïdes circulaires	22
2.16	Trapèzes et 2-intervalles emboîtés	22
2.17	Graphes de permutation et (1,1)-intervalles emboîtés	23
3.1	Complexité des diverses variantes du problème 2-IP	26
3.2	Illustration de la réduction de 3-SAT à MAXIMUMINDEPENDENTSET sur les graphes de 2-unions	26
3.3	Divers graphes à triplet astéroïde et leur représentation $\{\curvearrowright, \square\}$ -(1,1)-intervallaire	27
3.4	Décomposition d'un ensemble de 2-intervalles se précédant ou entrelacés	28
3.5	Contrexemple à l'utilisation de la programmation dynamique	29
3.6	(1,1)-intervalles sans relation de précédence et cliques des intervalles couvrants	29
4.1	Structure clé du contreexemple de graphe 2-intervallaire non équilibré	31
4.2	Graphe 2-intervallaire regroupant huit 2-intervalles pour simuler un intervalle	32
4.3	Parcours eulériens et ordres possibles des intervalles dans une réalisation 2-intervallaire	34
4.4	Graphe 2-intervallaire avec contrainte sur l'ordre des intervalles dans sa réalisation	35
4.5	Contrexemple de graphe 2-intervallaire non équilibré	36
4.6	Contrexemple de graphes 2-intervallaires équilibré mais non unitaire d'une part, unitaire mais non (1,1)-intervallaire d'autre part	36
5.1	Clique, stable, chemin, cycle, et biparti complet	37
5.2	Permutation et graphe de permutation associé	38
5.3	Graphe de cordes et chevauchement d'intervalles	39
5.4	Graphes contenant un triplet astéroïde	39
5.5	Hierarchies de quelques classes de graphes	39

5.6	Couverture de sommets d'un graphe	40
5.7	Ensemble indépendant et couverture de sommets	40
6.1	Recollements dans une séquence arc-annotée à arcs disjoints	44
6.2	Génération récursive, selon le nombre d'arcs, des séquences arc-annotées	44
6.3	Génération récursive, selon le nombre d'extrémités, des séquences arc-annotées	46
6.4	Le code Mathematica de calcul de $AA(n)$	46
6.5	Les vingt premiers termes de $AA(n)$	47

Chapitre 1

Introduction

De nombreux problèmes peuvent être modélisés à l'aide de graphes, et résolus en utilisant des outils ou propriétés classiques de théorie des graphes. En particulier, la classe des graphes d'intervalles, où chaque graphe correspond à un ensemble d'intervalles de la droite des réels (chaque noeud représentant un intervalle et chaque arête reliant deux noeuds dont les intervalles associés s'intersectent), a été bien étudiée [Gol80, Fis85] : les nombreux algorithmes efficaces disponibles sur cette classe permettent de résoudre des problèmes d'ordonnement [Gar05] ou d'archéologie [Ken69], voire des énigmes policières [Ber94].

La bioinformatique (prédiction de structure secondaire ou recherche de motifs dans un ARN) et l'ordonnement (de couples de tâches) ont motivé l'étude d'une classe plus générale que celle des graphes d'intervalles : les graphes 2-intervallaires, dont chaque noeud est associé à un 2-intervalle (c'est à dire une union de 2 intervalles "support" disjoints) relié par une arête à un autre ssi ces deux unions d'intervalles ont une intersection non vide.

Toutefois, les algorithmes sur cette classe de graphes sont plus compliqués, la reconnaissance des graphes 2-intervallaires étant par exemple NP-complète. La distinction des 3 cas possibles quand l'intersection de deux 2-intervalles est vide, puis le choix d'ajouter ou non une arête dans le graphe pour chacun de ces cas, permet de refléter des contraintes biologiques et de définir des classes voisines parfois plus faciles à traiter. La restriction de support disjoint, c'est à dire imposer que l'intersection de deux 2-intervalles soit seulement l'ensemble vide ou l'un des intervalles support des 2-intervalles tout entier, permet aussi de se rapporter à des classes de graphes connues et d'utiliser des algorithmes polynomiaux.

Les informations sur ces diverses classes voisines des graphes 2-intervallaires, exhibées de façon plus ou moins explicite dans divers articles et thèses [Via01, BFV04, Via04, Bli05], sont synthétisées au chapitre 2. Elles permettent d'utiliser les algorithmes de résolution du problème de l'ensemble indépendant maximal sur ces diverses classes, afin de répondre à diverses variantes d'un problème de pattern-matching sur les 2 intervalles au chapitre 3. L'utilisation des graphes de cordes permet notamment d'améliorer la meilleure complexité connue pour l'algorithme de recherche du plus grand sous-ensemble de 2-intervalles entrelacés parmi un ensemble de 2-intervalles à support disjoint. Dans les cas où l'approche par la théorie des graphes n'aboutit pas, l'utilisation de la programmation dynamique permet de résoudre certaines de ces variantes.

La prise en compte de la restriction de support disjoint permet de faire le lien avec un autre objet connu des bioinformaticiens, les séquences arc-annotées [Eva99], dont le dénombrement selon le nombre d'arcs ou de lettres est effectué au chapitre 6.

Une autre restriction introduite dans [CHLV05], moins forte que celle du support disjoint, impose que les intervalles d'un 2-intervalle aient même longueur. Elle est particulièrement réaliste dans le contexte de l'étude de l'ARN où les intervalles support d'un 2-intervalle correspondent à des portions complémentaires, et donc de même longueur. Les graphes 2-intervallaires respectant cette restriction sont dits équilibrés, et le chapitre 4 prouve que l'inclusion de cette classe dans celle des graphes 2-intervallaires est stricte.

Ce mémoire faisant appel à de nombreux outils de théorie des graphes, les prérequis sur ce domaine sont rappelés au chapitre 5. L'index et ses liens cliquables dans la version électronique de ce document¹ devraient permettre une navigation facilitée entre cette annexe et le reste du texte.

¹ disponible, ainsi que sa traduction en anglais, sur <http://philippe.gambette.free.fr/LIAFA/>.

Chapitre 2

Graphes 2-intervallaires

Les graphes 2-intervallaires permettent de modéliser des problèmes de bioinformatique ou d'ordonnancement. En effet, comme détaillé ci-dessous, un graphe 2-intervallaire permet de représenter les intersections de couples d'intervalles de réels. Or, un couple d'intervalles permet de modéliser des tâches coupées en 2 dans un problème d'ordonnancement, des portions similaires d'un brin d'ADN, des portions complémentaires d'un brin d'ARN ou encore des extraits de partitions musicales pouvant être mis en relation. Après quelques définitions de base, ces applications sont présentées en section 2.2. Puis plusieurs classes de graphes voisines des 2-intervallaires sont définies (en introduisant des contraintes naturelles sur les placements relatifs des 2-intervalles) et mises en relation avec des classes de graphes connues.

2.1 Définitions

2.1.1 Définition des 2-intervalles

Soient deux intervalles de \mathbb{R} , $I_1 = [d_1, f_1]$ et $I_2 = [d_2, f_2]$. On définit deux relations de comparabilité sur les intervalles, la *précédence* et l'*inclusion*.

- Si $d_1 < f_1 < d_2 < f_2$, on dit que I_1 précède I_2 , noté $I_1 <_{[]} I_2$.
- Si $d_1 \leq d_2 \leq f_2 \leq f_1$, on dit que I_2 est inclus dans I_1 , noté $I_2 \subset I_1$.
- S'il n'y a ni inclusion ni précédence, on dit que I_1 et I_2 sont *chevauchants*.

Un *t-intervalle* est un ensemble de t intervalles disjoints I_1, \dots, I_t de \mathbb{R} noté $I = (I_1, \dots, I_t)$, avec $I_1 <_{[]} \dots <_{[]} I_t$. En particulier, un *2-intervalle* est l'union de deux intervalles disjoints d'une ligne, comme illustré en figure 2.1. Les I_k sont appelés *intervalles support* du *t-intervalle*. Pour un ensemble de 2-intervalles $\{(I_1^k, I_2^k), 1 \leq k \leq n\}$, son *support* est l'ensemble des intervalles support de ses 2-intervalles, c'est à dire $\bigcup_{k=1}^n \{I_1^k, I_2^k\}$. On dit qu'un ensemble de 2-intervalles est à *support unitaire* si son support est un ensemble d'intervalles de longueur 1. Il est à *support disjoint* s'il est à support unitaire et que de plus son support est un ensemble d'intervalles disjoints : on peut alors considérer les intervalles comme de simples points. On se retrouve dans le domaine des *séquences arc-annotées*, et on peut alors appeler les 2-intervalles des *arcs* et leurs intervalles supports des *extrémités*. Ces variantes sur le support sont illustrées en figure 2.1. L'intervalle $[d_1, f_2]$ est appelé l'*intervalle couvrant* du 2-intervalle $([d_1, f_1], [d_2, f_2])$. Par extension, on dira que $[\min\{d_i\}, \max\{f'_i\}]$ est l'*intervalle couvrant* de la famille de 2-intervalles $\{[d_i, f_i], [d'_i, f'_i]\}$.

2.1.2 Relations entre 2-intervalles

Deux 2-intervalles $I = (I_1, I_2)$ et $J = (J_1, J_2)$ sont *disjoints* s'ils ne s'intersectent en aucun point : $(I_1 \cup I_2) \cap (J_1 \cup J_2) = \emptyset$. Dans ce cas ils peuvent être [BFV04] :

- en ordre de précédence : $I < J$ si $I_1 <_{[]} I_2 <_{[]} J_1 <_{[]} J_2$.

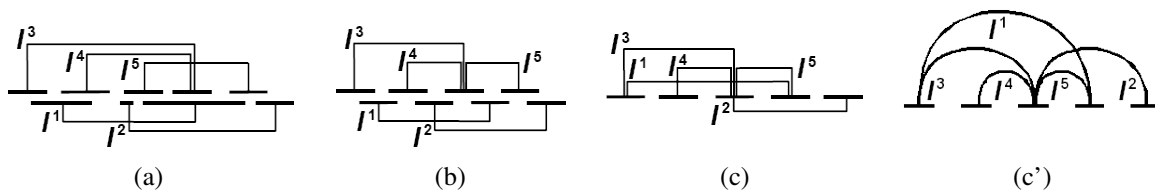


FIG. 2.1 – Le support d’un ensemble de 2-intervalles peut être sans restriction (a), unitaire (b), ou disjoint (c). Dans ce dernier cas, les intervalles du support peuvent être considérés comme des points (des entiers), reliés par des arcs (des couples d’entiers), on parle alors de *séquences arc-annotées* (c’).

- emboîtés : $I \sqsubset J$ si $I_1 <_{[]} J_1 <_{[]} J_2 <_{[]} I_2$.
- entrelacés : $I \bowtie J$ (parfois aussi noté \bowtie [Bli05], \times [Kar05]) si I et J ne sont ni en ordre de précedence ni emboîtés (c’est à dire si $I_1 <_{[]} J_1 <_{[]} I_2 <_{[]} J_2$ ou $J_1 <_{[]} I_1 <_{[]} J_2 <_{[]} I_2$).

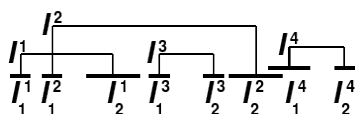


FIG. 2.2 – Un ensemble de 2-intervalles. I^1, I^2 et I^3 sont 2 à 2 disjoints, et $I^1 < I^3, I^3 \sqsubset I^2, I^1 \bowtie I^2$. Les intervalles I_2^2 et I_4^1 sont chevauchants.

Si au contraire leur intersection est non vide, on peut noter $I \frown J$.

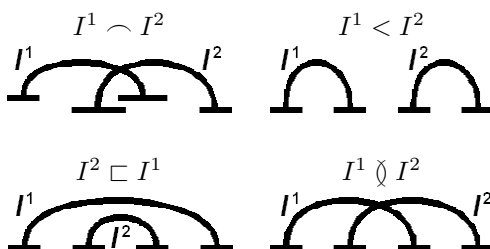


FIG. 2.3 – Récapitulatif des 4 types de positions possibles pour deux 2-intervalles.

2.1.3 Définition des graphes 2-intervallaires

A partir d’un ensemble de t -intervalles $\{I^1, \dots, I^n\}$ on peut définir le *graphe t -intervallaire* suivant :

- les noeuds représentent les t -intervalles I^i .
- (I^i, I^j) est une arête du graphe ssi $\exists k, l \in \{1, \dots, t\} / I_k^i \cap I_l^j \neq \emptyset$, c’est à dire que $I^i \frown I^j$, les deux t -intervalles I^i et I^j ont un de leurs intervalles qui se chevauchent ou sont inclus l’un dans l’autre.

Les graphes 1-intervallaires sont les classiques *graphes d’intervalles*. On dit qu’un ensemble de t -intervalles correspondant à un graphe t -intervallaire en est une *réalisation*. Un exemple de *graphe 2-intervallaire* est donné en figure 4.2(a), avec deux réalisations possibles en figure 4.2(b) et (c).

Les classes de graphes t -intervallaires sont des classes de *graphes d’intersection*, c’est à dire des graphes définis en associant des ensembles aux noeuds, et où les arêtes signifient que les ensembles ont une intersection non vide. Nous rencontrerons tout au long de ce rapport d’autres exemples de tels graphes : les graphes de permutations, de cordes, trapézoïdaux, de trapézoïdes circulaires, d’arcs circulaires...

Le *nombre intervallaire* d’un graphe G est le plus petit entier positif t tel que G est un graphe t -intervallaire. Ce nombre est bien défini, puisque tout graphe G de degré maximum d est un graphe

d -intervallaire : chaque arête (v, v') correspond à l'intersection d'un intervalle support du d -intervalle associé à v intersectant un intervalle support du d -intervalle associé à v' , et n'intersectant aucun autre intervalle.

De même, on peut définir un *graphe de t -unions*, ou de *t -pistes* :

- les noeuds représentent les t -intervalles I^i .
- (I^i, I^j) est une arête du graphe ssi $\exists k \in \{1, \dots, t\} / I_k^i \cap I_k^j \neq \emptyset$.

Un graphe de t -unions est donc une union de t graphes d'intervalles, et l'ensemble des graphes de t -unions est une sous-classe de celle des graphes t -intervallaires. On peut considérer ces graphes comme les graphes d'intersection de projections sur les axes de parallélépipèdes aux côtés parallèles aux axes, comme illustré dans la figure 2.4 pour le cas $t = 2$.

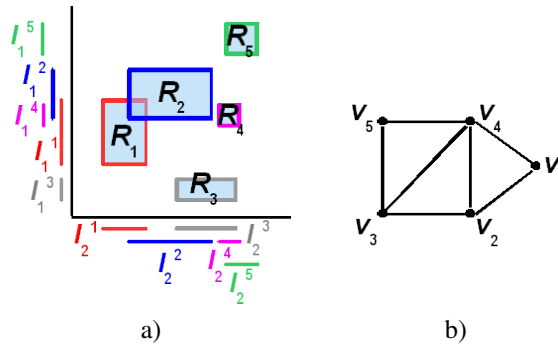


FIG. 2.4 – Un ensemble de rectangles et l'ensemble des intervalles associés aux côtés (a), et le graphe de 2-unions correspondant (b) : chaque sommet v_i correspond au 2-intervalle (I_1^i, I_2^i) , c'est à dire au rectangle R_i .

Si de plus on impose que pour tout 2-intervalle, le premier ait pour longueur $x \in \mathbb{N}$ et le second $y \in \mathbb{N}$, et que les deux aient des bornes entières, on parle pour les graphes associés de graphes (x, y) -intervallaires et de graphes de (x, y) -unions.

On peut par exemple remarquer que les graphes 2-intervallaires d'ensembles de 2-intervalles à support disjoint sont exactement les graphes $(1, 1)$ -intervallaires. Ce sont aussi exactement les graphes adjoints (voir la propriété 6).

2.2 Motivations

2.2.1 2-intervalles et ordonnancement

L'*ordonnancement* consiste à affecter des dates de début et de fin d'exécution de tâches sur une ou plusieurs machines, qui ne peuvent chacune traiter qu'une tâche simultanément. En particulier, quand les tâches sont coupées en deux (pour une raison quelconque, en particulier dans le domaine de radars, où une tâche liée à l'émission d'un signal précède toujours une tâche liée à sa réception), on peut considérer ces *bitâches* constituée d'une tâche 1 et d'une tâche 2 comme un couple d'intervalles : $([d_1, f_1][d_2, f_2])$, où d_i est la date de début de la tâche i , et f_i sa date de fin. Un ordonnancement de bitâches sur une seule machine est un *stable* dans le graphe 2-intervallaire correspondant [BYHN⁺02]. Nous verrons en section 3, comment trouver le stable maximum dans un graphe 2-intervallaire, c'est à dire optimiser le nombre de bitâches sur une machine. De plus, trouver le nombre minimal de machines pour exécuter un ensemble de bitâches correspond à un problème de coloration sur le graphe 2-intervallaire.

2.2.2 2-intervalles et ADN

L'ADN est le support de l'information génétique, situé dans le noyau des cellules. Elle se présente sous forme d'une double chaîne de *nucléotides*, c'est à dire des constituants moléculaires de base, de 4 types différents : A, C, G et T [WC53]. Les deux chaînes contiennent la même information puisqu'elles sont complémentaires, c'est à dire que les nucléotides A, C, G et T d'un brin sont liées respectivement aux nucléotides T, G, C et A du second brin, et c'est à partir de cette information et du code génétique que sont construites les protéines. Ainsi, on peut évaluer la similarité de deux ADN en comparant simplement la chaîne de caractères constituée par la suite de nucléotides d'un brin du premier aux chaînes de caractères constituées par la suite de nucléotides des deux brins du second, et en choisissant la similarité maximale parmi les deux trouvées.

Dans le cas où les séquences sont assez semblables, un algorithme classique de programmation dynamique permet de les aligner. Il faut être plus subtil dans le cas d'ADN ayant subi d'importantes restructurations (*transposition*, quand un bloc est déplacé, éventuellement suivie d'une *inversion*, quand un bloc subit un changement de sens, et de brin d'ADN, lors du recollement). En représentant un ADN de c nucléotides par l'intervalle $[0, c]$, le i -ième nucléotide étant codé par l'intervalle $[i-1, i]$, on peut modéliser chaque portion d'ADN par un intervalle.

Ainsi, pour toute portion assez longue d'ADN similaire (ou presque complémentaire, et de sens contraire) dans les deux chaînes comparées, on considérera les deux intervalles correspondants comme un 2-intervalle [BNR96]. La recherche de telles séquences répétées peut s'effectuer en $O(t \log(t))$, t étant le nombre de nucléotides de l'ADN, à l'aide de l'algorithme de Karp-Miller-Rosenberg [KMR72]. Une fois que l'on a obtenu un ensemble assez grand de tels 2-intervalles, on en recherche un sous-ensemble, de taille maximale, constitué uniquement de 2-intervalles sans point commun (afin qu'un bloc d'un ADN corresponde bien à un et un seul bloc dans l'autre ADN). Ceci se rapporte à une recherche de stable max dans un graphe de 2-unions, une sous-classe de graphes 2-intervallaires définie en section 2.1.3.

2.2.3 2-intervalles et ARN

L'ARN est une molécule permettant de passer l'information génétique du noyau des cellules, où elle est enregistrée dans l'ADN, vers leur cytoplasme, où elle dirige la création des protéines. C'est le dogme central de la biologie moléculaire : il y a *transcription* de l'ADN en ARN, puis *traduction* de l'ARN en protéine.

L'ARN est une chaîne de nucléotides comme l'ADN, sauf que le nucléotide T est remplacé par U (aussi complémentaire de A), et que la chaîne est constituée d'un seul brin (l'ARN, qui sert de messenger, n'a pas besoin de se dupliquer comme l'ADN). Des portions complémentaires du brin ont tendance à se plier l'une vers l'autre afin de se lier.

Ces rapprochements de portions complémentaires d'ARN peuvent être visualisés sur sa *structure secondaire*, qui les représente de façon planaire, comme sur l'exemple de la figure 2.5. Deux ARN sont similaires si leur structure secondaire est similaire. On peut, tout comme avec l'ADN, modéliser chaque rapprochement de blocs d'ARN complémentaires par un couple d'intervalles [Via04, Via01]. Ainsi, la position respective des 2-intervalles correspond à la structure secondaire de l'ARN. Le problème de déterminer les rapprochements valides parmi une liste de portions complémentaires pour obtenir une structure secondaire planaire se rapporte donc à une recherche de stable sur un graphe 2-intervallaire, comme détaillé en section 3. De plus, si l'ARN ne présente pas de *pseudo-noeud*¹, on peut remarquer que les 2-intervalles correspondants sont tous en relation de précédence ou d'emboîtement.

¹ Un pseudo-noeud correspond à l'appariement entrelacé de quatre nucléotides, c'est à dire qu'il existe quatre positions $i < j < k < l$ dans la chaîne d'ARN, telles que le i -ième nucléotide est apparié avec le k -ième et le j -ième avec le l -ième [RPP⁺82, SB05].

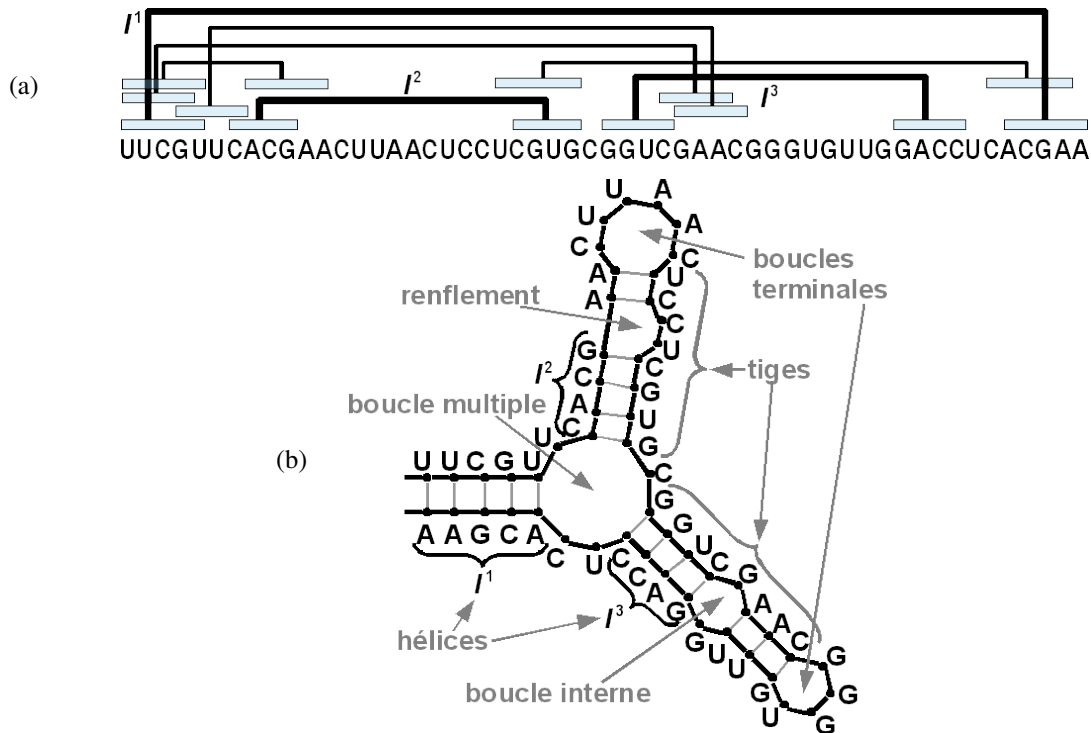


FIG. 2.5 – La structure primaire de l’ARN est une chaîne de nucléotides (a). Des portions complémentaires et retournées définissent des 2-intervalles. Certains 2-intervalles, I^1 , I^2 et I^3 , correspondent à des blocs effectivement face à face au sein d’une même hélice de la structure secondaire. La structure secondaire de l’ARN (b) permet de le représenter de façon lisible tout en donnant de bonnes informations sur sa structure géométrique. Elle est décomposable en plusieurs éléments : boucles terminales, boucles multiples et tiges, elles-mêmes composées d’hélices (les portions appariées qui correspondent aux 2-intervalles), renflements, et boucles internes.

2.2.4 2-intervalles et analyse musicale

En introduisant une nouvelle conception de l’analyse musicale appelée *analyse transformationnelle*, David Lewin [Lew87] définit en particulier deux relations possibles entre des ensembles de notes consécutives : la *transposition* et l’*inversion*. Ces relations se fondent uniquement sur les hauteurs de notes (qu’on peut considérer comme des éléments de $\mathbb{Z}/12\mathbb{Z}$, pour chacun des 11 demi-tons différents, modulo les octaves). La première met en relation deux ensembles de notes consécutives ssi le second est constitué de la transposition d’un nombre constant de demi-tons (compris entre 0 et 12) des notes du premier. Une représentation géométrique permet de définir plus aisément la seconde relation : en dessinant les 12 demi-tons sur un cercle, deux ensembles de notes sont en relation d’inversion ssi l’ensemble des hauteurs de l’un est symétrique à celui des hauteurs de l’autre par rapport à un des diamètres du cercle.

En exhibant de telles relations dans une partition de musique, on définit donc des 2-intervalles dont les intervalles support sont constitués des ensembles de notes en relation de transposition ou d’inversion. Une collaboration avec Yun-Kang Ahn, qui réalise en stage de master à l’IRCAM un logiciel pour la détection assistée et automatique de ces 2-intervalles, permettra de déterminer si la recherche du plus grand sous-ensemble de 2-intervalles disjoints (qui correspond à la recherche du stable maximum détaillée en section 3), en respectant éventuellement certaines contraintes usuelles pour les 2-intervalles disjoints, permet d’exhiber une certaine structure de l’oeuvre en sélectionnant uniquement des relations significatives entre ensembles de notes consécutives.

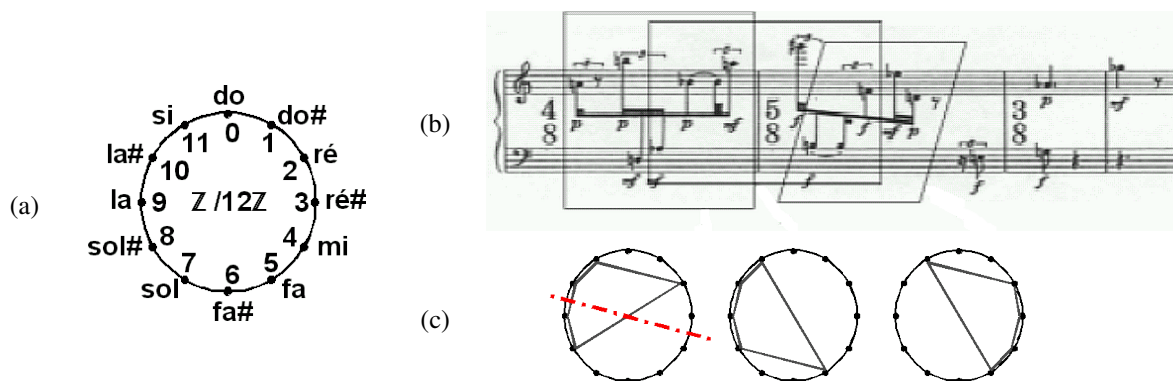


FIG. 2.6 – Représentation cyclique de l'échelle chromatique (a) afin de faire apparaître une relation d'inversion et de transposition (c) dans le *Klavierstück III* de Karlheinz Stockhausen (b).

2.3 2-intervallaires et classes apparentées

2.3.1 Une sous-classe connue, les graphes d'intervalles

Pour un état de l'art sur le sujet, on pourra se référer au chapitre 8 du livre de Martin Golumbic [Gol80], au cours de Thérèse Biedl [Bie05], ou à la synthèse d'András Gyárfas [Gyá].

2.3.2 Les graphes R -2-intervallaires

Certains des problèmes pratiques cités en section 2.2 imposent des conditions sur les 2-intervalles disjoints solutions. Par exemple, en ce qui concerne la structure secondaire de l'ARN, les 2-intervalles associés aux hélices dans la chaîne de nucléotides, qui sont disjoints, sont le plus généralement uniquement en relation $<$ ou \sqsubset . En effet, ceci est dû à la structure secondaire de l'ARN qui fait alterner des boucles et des hélices, ces dernières se trouvant donc soit en relation de précédence, soit emboîtées. Pour ramener tout de même ce problème à la recherche d'un stable maximum sur un graphe, on peut définir une nouvelle classe de graphe proche des 2-intervallaires où l'on place un arête entre deux 2-intervalles non disjoints, mais aussi entre deux 2-intervalles placés en relation \checkmark .

Pour prendre en compte toutes les combinaisons de contraintes possibles, on définit donc les graphes R -2-intervallaires (et on définit de façon similaire les R -(1,1)-intervallaires, on parlera parfois de graphes R -...-intervallaires pour évoquer simultanément les 2 types de classes) :

- les noeuds représentent les 2-intervalles I^i .
- (I^i, I^j) est une arête du graphe R -2-intervallaire ssi il existe une relation $\mathcal{R} \in R \subset \{\frown, <, \sqsubset, \checkmark\}$ telle que $I^i \mathcal{R} I^j$ ou $I^j \mathcal{R} I^i$.

Remarquons que si un graphe est un graphe R -...-intervallaire, alors son complémentaire est un graphe $(\{\frown, <, \sqsubset, \checkmark\} \setminus R)$ -...-intervallaire.

Certaines des classes de graphes R -2-intervallaires, et surtout R -(1,1)-intervallaires sont en fait des classes bien connues. Ces relations sont indiquées dans le tableau de la figure 2.7, et détaillées ci-dessous. Elles seront utiles notamment pour la résolution du problème 2-IP, de recherche du plus grand sous-ensemble de 2-intervalles vérifiant certaines propriétés, abordé dans la section 3.

Remarquons qu'aucune classe de graphes R -(1,1)-intervallaires ne contient l'ensemble de tous les graphes. En effet, l'opération qui consiste à déduire pour un certain R le graphe R -(1,1)-intervallaire non étiqueté à partir d'une séquence arc-annotée aux arcs non étiquetés est surjective, c'est à dire :

$$|\{\text{graphes } R\text{-(1,1)-intervallaires à } n \text{ sommets non étiquetés}\}| \leq AA(n),$$

où $AA(n)$ est le nombre de séquences arc-annotées à n arcs non étiquetés. Or, pour n au voisinage

R	R -2-intervallaires	Reconnaissance	$R - (1,1)$ -intervallaires	Reconnaissance
$\{\curvearrowright, \sqsubset, <, \emptyset\}$	= cliques	linéaire	= cliques	linéaire
\emptyset	= stables	linéaire	= stables	linéaire
$\{\curvearrowright\}$	= 2-intervallaires, cf. section 2.3.4 \supset arbres, adjoints, intervallaires	NP-complet	= (1-1)-intervallaires = adjoints	linéaire [Leh74]
$\{\sqsubset, <, \emptyset\}$		NP-complet		polynomial (complémentaire)
$\{\curvearrowright, \sqsubset\}$			\supset cycles, bipartis complets, de permutation, cf. section 2.3.5	?
$\{<, \emptyset\}$?	?
$\{\curvearrowright, <\}$?	?
$\{\sqsubset, \emptyset\}$?	?
$\{\curvearrowright, \emptyset\}$	\subset graphes de trapézoïdes circulaires, cf. section 2.3.7	?	= graphes de cordes, cf. section 2.3.6	polynomial, [Spi94]
$\{<, \sqsubset\}$				polynomial (complémentaire)
$\{\curvearrowright, \sqsubset, <\}$				polynomial (complémentaire)
$\{\emptyset\}$	\subset graphes de croisement de trapé- zoïdes circulaires, cf. section 2.3.8	?	= graphes de cordes, cf. section 2.3.6	polynomial, [Spi94]
$\{\curvearrowright, \sqsubset, \emptyset\}$	= graphes d'intervalles (des intervalles couvrants)	linéaire [BL76, KMMS03]	= graphes d'intervalles (des intervalles couvrants)	linéaire [BL76, KMMS03]
$\{<\}$	\subset graphes de comparabilité (de la relation $<$)	polynomial (complémentaire)	\subset graphes de comparabilité (de la relation $<$)	polynomial (complémentaire)
$\{\curvearrowright, <, \emptyset\}$	= graphes trapézoïdaux, cf. section 2.3.9	polynomial [MS94]	= graphes de permutation, cf. section 2.3.10	linéaire [KMMS03]
$\{\sqsubset\}$	\subset graphes de comparabilité (de la relation \sqsubset)	polynomial (complémentaire)	= graphes de permutation, cf. section 2.3.10	linéaire [KMMS03]

FIG. 2.7 – Les classes de graphes R -...-intervallaires. Remarquons que la $2i$ -ième classe (selon l'ordre des lignes) correspond à la classe complémentaire de la $(2i - 1)$ -ième. Les diverses variantes du problème 2-IP au chapitre suivant se ramèneront ainsi chacune à un problème de recherche de la clique max sur la $2i$ -ième classe, ou du stable max sur la $(2i - 1)$ -ième.

de $+\infty$, en notant $g(n)$ le nombre de graphes non étiquetés à n sommets, on a :

$$\begin{cases} AA(n) < \frac{(2n)!2^{n-1}}{n!} & , \text{ d'après la section 6.1,} \\ g(n) \sim \frac{2^{\frac{n(n-1)}{2}}}{n!} & , \text{ d'après la section 5.2.10.} \end{cases}$$

Or $\frac{(2n)!2^{n-1}}{n!} = \frac{(2n)!}{2^{\frac{(n-1)(n-2)}{2}}} \rightarrow 0$, quand $n \rightarrow +\infty$, donc $AA(n) = o(g(n))$.

Ainsi, pour n assez grand (on peut même préciser pour $n \geq 16$ grâce à la figure 2.8), il existe un graphe à n noeuds qui n'est pas R -(1,1)-intervallaire.

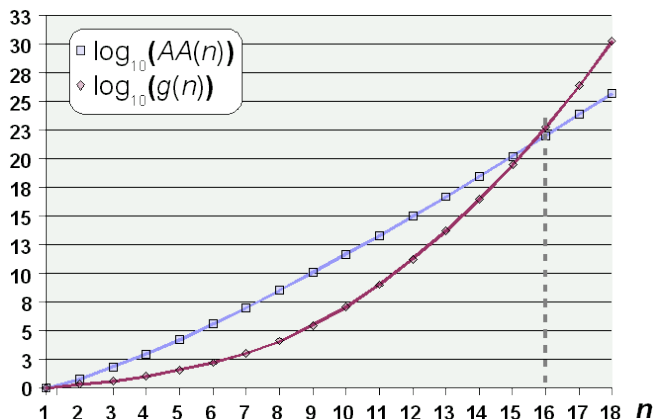


FIG. 2.8 – Croissance comparée du nombre $g(n)$ de graphes non étiquetés à n noeuds avec $AA(n)$, le nombre de séquences arc-annotées à n arcs non étiquetés : on en déduit que pour tout $n \geq 16$, il existe un graphe à n noeuds qui n'est pas R -(1,1)-intervallaire.

2.3.3 Éclatement de séquences arc-annotées

Considérons maintenant une opération sur les séquences arc-annotées utile ci-dessous dans l'analyse des classes de graphes R -(1,1)-intervallaires : l'éclatement des extrémités communes à plusieurs arcs. Cette opération, appliquée sur toutes les extrémités communes à plusieurs arcs de la séquence, permet de passer d'une séquence arc-annotée quelconque à une séquence arc-annotée à arcs disjoints :

- soit en transformant la relation \frown en \emptyset : *éclatement*($\frown \mapsto \emptyset$).
- soit en transformant la relation \frown en \sqsubset ou $<$: *éclatement*($\frown \mapsto \sqsubset, <$).

Détaillons ces opérations, illustrées en figure 2.9. Soit la séquence arc-annotée définie comme un ensemble de 2-intervalles unitaires à support disjoint $\mathcal{I} = \{I^k = ([i_k, i_k + 1], [i'_k, i'_k + 1]), 1 \leq k \leq n\}$. Sans perte de généralité, on considère que le support de \mathcal{I} est de la forme $\{[0, 1], [2, 3], \dots, [2p, 2p+1]\}$. Soit $[i, i+1]$, une extrémité commune à un ensemble de m arcs, ensemble qu'on appelle \mathcal{I}_i . On peut distinguer ces arcs selon le fait que $[i, i+1]$ en est une extrémité droite ou gauche : $\mathcal{I}_i = \{([i_k, i_k + 1], [i, i + 1]), 1 \leq k \leq l\} \cup \{([i, i + 1], [i_k, i_k + 1]), l + 1 \leq k \leq m\}$, avec les i_k rangés par ordre croissant : $k < l \Leftrightarrow i_k < i_l$.

L'éclatement($\frown \mapsto \emptyset$) de $[i, i+1]$ consiste à transformer \mathcal{I}_i en $\{([i_k, i_k + 1], [i + 2(m-l-1+k), i + 2(m-l-1+k) + 1]), 1 \leq k \leq l\} \cup \{([i + 2(k-l-1), i + 2(k-l-1) + 1], [i_k, i_k + 1]), l + 1 \leq k \leq m\}$.

L'éclatement($\frown \mapsto \sqsubset, <$) de $[i, i+1]$ consiste à transformer \mathcal{I}_i en $\{([i_k, i_k + 1], [i + 2(l-k), i + 2(l-k) + 1]), 1 \leq k \leq l\} \cup \{([i + 2(l+m-k), i + 2(l+m-k) + 1], [i_k, i_k + 1]), l + 1 \leq k \leq m\}$.

On obtient donc les lemmes suivants :

Lemme 1. Une séquence arc-annotée et le résultat de l'éclatement($\frown \mapsto \emptyset$) de l'ensemble de toutes ses extrémités ont le même graphe $\{\frown, \emptyset\}$ -(1,1)-intervallaire, $\{<, \sqsubset\}$ -(1,1)-intervallaire, $\{\frown, <, \emptyset\}$ -(1,1)-intervallaire, $\{\sqsubset\}$ -(1,1)-intervallaire, $\{\frown, \sqsubset, \emptyset\}$ -(1,1)-intervallaire, et $\{<\}$ -(1,1)-intervallaire.

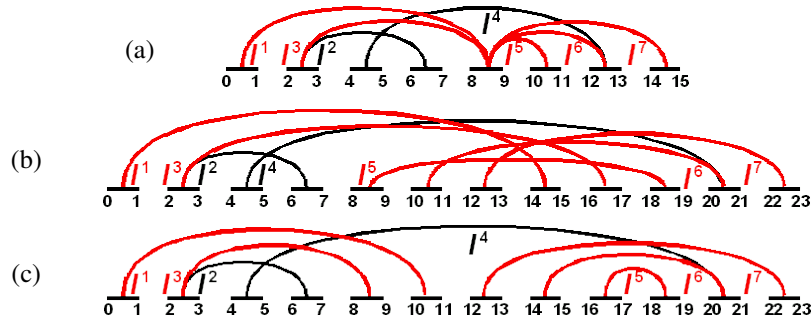


FIG. 2.9 – Une séquence arc-annotée (a), le résultat de l'éclatement $(\frown \mapsto \checkmark)$ de l'extrémité $[8, 9]$ (b), et le résultat de l'éclatement $(\frown \mapsto \sqsubset, \prec)$ de l'extrémité $[8, 9]$ (c).

Lemme 2. Une séquence arc-annotée et le résultat de l'éclatement $(\frown \mapsto \prec, \sqsubset)$ de l'ensemble de toutes ses extrémités ont le même graphe $\{\checkmark\}$ - $(1,1)$ -intervallaire et le même graphe $\{\frown, \prec, \sqsubset\}$ - $(1,1)$ -intervallaire.

Ces lemmes permettent donc de travailler seulement sur des séquences arc-annotées à arcs disjoints lors de l'étude de diverses classes de graphes R - $(1,1)$ -intervallaire. Le fait que les arcs soient disjoints permet de nombreuses simplifications (on peut le voir en particulier en section 6.1 à propos du dénombrement de ces objets) ainsi que des liens plus visibles avec d'autres objets comme les permutations (en section 2.3.10) ou les intervalles chevauchants (en section 2.3.6).

2.3.4 Classe des 2-intervallaires

Théorème 1. [WS84] Pour $t \geq 2$, reconnaître qu'un graphe est t -intervallaire est NP complet.

Théorème 2. [GW79] Un graphe G est un graphe t -intervallaire, pour $t \geq \lceil \frac{\Delta(G)+1}{2} \rceil$, où $\Delta(G)$ est le degré maximum de G .

Propriété 1. [TH79] Les arbres sont des graphes 2-intervallaires.

Démonstration. On peut en effet construire pour tout arbre une réalisation en 2-intervalles en utilisant l'idée suivante : pour chaque noeud-père dans l'arbre, l'intervalle support droit du 2-intervalle correspondant intersecte tous les intervalles supports gauche, disjoints deux à deux, des 2-intervalles correspondant à ses fils. Trotter et Harary précisent qu'un arbre est un graphe d'intervalles ssi c'est une chenille (c'est à dire qu'en supprimant toutes les feuilles, on obtient un chemin). \square

Propriété 2. [BLS⁺] Les graphes d'arcs circulaires, c'est à dire les graphes d'intersection d'arcs du cercle unité, sont des graphes 2-intervallaires.

Démonstration. On peut obtenir directement les 2-intervalles à partir des arcs circulaires en suivant l'idée de construction suivante : on choisit un point du cercle où on coupe tous les arcs contenant ce point, puis on projette depuis ce point tous les arcs circulaires sur une droite afin d'obtenir des 2-intervalles (les arcs déjà coupés) ou des intervalles, enfin on coupe en deux les intervalles pas encore coupés. \square

Propriété 3. [KW99] Les graphes planaires extérieurs, c'est à dire les graphes planaires ayant tous leurs sommets autour d'une même face, sont des graphes 2-intervallaires.

Démonstration. Tout graphe planaire extérieur est l'union de deux graphes d'intervalles, c'est à dire un graphe de 2-unions, donc un graphe 2-intervallaire. \square

Propriété 4. [BOP04] Les graphes planaires de degré maximum inférieur ou égal à 4 sont des graphes 2-intervallaires.

Propriété 5. [BLS⁺] Les 2-subdivisions de graphes sont des graphes 2-intervallaires.

Démonstration. Soit G un graphe à n sommets étiquetés $v_i, 1 \leq i \leq n$ et G' sa 2-subdivision, obtenue en ajoutant les nouveaux sommets étiquetés $v_i, n < i \leq n + 2|E|$. Construisons une réalisation 2-intervallaire de G' :

- à tout sommet v_i de $G \cap G'$ on associe le 2-intervalle $([4i, 4i + 1], [4i + 2, 4i + 3])$.
- tout sommet v_j de $G' \setminus G$ étant relié à un sommet $v_k \in G \cap G'$ et $v_l \in G' \setminus G$, on lui associe le 2-intervalle $(I_j, [4 \min(j, l), 4 \min(j, l) + 3])$, où I_j est un intervalle intersectant seulement l'intervalle $[4k, 4k + 1]$, et de longueur $\frac{1}{2 \text{degré}(v_k)}$ par exemple. □

Propriété 6. [BLS⁺] Les graphes adjoints sont des graphes 2-intervallaires.

Démonstration. Soit $G = (V, E)$ un graphe, avec $V = (v_1, \dots, v_n)$ et $G' = (E, E')$ son graphe adjoint. On peut construire une réalisation 2-intervallaire de G' directement en "associant un intervalle à chaque sommet de G' " : à toute arête $e = (v_i, v_j)$, avec $i < j$, on associe le 2-intervalle $([2i, 2i + 1], [2j, 2j + 1])$. □

Propriété 7. [TH79] Le nombre intervallaire du graphe biparti $K_{m,n}$ est $\lceil \frac{mn+1}{m+n} \rceil$.

Ainsi, $K_{3,5}$ est un graphe 2-intervallaire, mais pas $K_{3,6}$.

Coloration sur les graphes 2-intervallaires

Ce problème est utile pour les ordonnancements de tâches coupées en 2 : le nombre chromatique du graphe 2-intervallaire correspond au nombre de machines nécessaire pour traiter les tâches.

Théorème 3. [Via01] Trouver le nombre chromatique d'un graphe 2-intervallaire est NP-complet.

En effet, ce problème est déjà NP-complet sur la classe des graphes adjoints, par réduction immédiate du problème de l'index chromatique.

Stable maximal sur les graphes 2-intervallaires

Ce problème est étudié de façon détaillée en section 3, il est NP-complet.

Autres problèmes sur les graphes 2-intervallaires

On trouve en annexe de la thèse de Stéphane Vialette [Via01] une preuve de NP-complétude de la coupe maximum et de la partition en cliques sur les graphes 2-intervallaires. La domination γ est NP-complète d'après [BLS⁺]. La complexité du problème de la clique max n'est toujours pas déterminée, bien qu'on le sache polynomial sur les graphes de 2-unions et NP-complet sur les graphes 3-intervallaires.

2.3.5 Classe des $\{\curvearrowright, \square\}$ -(1,1)-intervallaires

Les cliques (ensemble de 2-intervalles emboîtés), les bipartis complets (figure 2.10). sont des graphes $\{\curvearrowright, \square\}$ -(1,1)-intervallaires, ainsi que les cycles (figure 2.11).

De plus, d'après la section 2.3.10, tout graphe de permutation est un graphe $\{\square\}$ -(1,1)-intervallaire, d'une certaine séquence arc-annotée, et aussi du résultat de son éclatement $(\curvearrowright \mapsto \emptyset)$ d'après le lemme 1, qui est une séquence arc-annotée à arcs disjoints, c'est à dire que ses graphes $\{\square\}$ -(1,1)-intervallaire et $\{\curvearrowright, \square\}$ -(1,1)-intervallaire sont isomorphes. Ainsi, tout graphe de permutation est aussi un graphe $\{\curvearrowright, \square\}$ -(1,1)-intervallaire.

Cherchons maintenant à mieux caractériser cette classe en exhibant de petits graphes interdits. La hiérarchie des classes de graphes nous indique que la classe des cographe, c'est à dire des

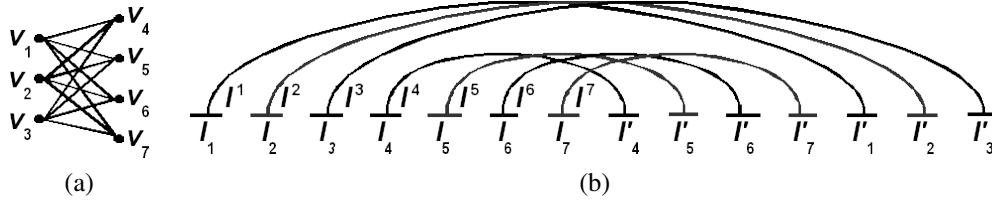


FIG. 2.10 – Un graphe biparti complet (a), et l'ensemble de 2-intervalles dont il est le graphe $\{\frown, \sqsupset\}$ -(1,1)-intervallaire (b). Pour le construire, on considère qu'il est composé du stable $V_1 = \{v_1, \dots, v_m\}$ et du stable $V_2 = \{v_{m+1}, \dots, v_n\}$. A chaque noeud v_i on associe alors le 2-intervalle $I^i = (I_i, I'_i)$, où $I_i = [2i, 2i + 1]$ et $I'_i = [4n - 2m + 2i, 4n - 2m + 2i + 1]$ si $v_i \in V_1$, et $I_i = [2i, 2i + 1]$ et $I'_i = [2n + 2i - 2m, 2n + 2i - 2m + 1]$ si $v_i \in V_2$.

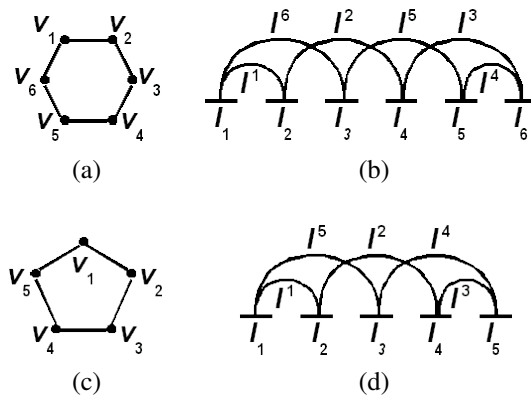


FIG. 2.11 – Un cycle pair (a), et l'ensemble de 2-intervalles dont il est le graphe $\{\frown, \sqsupset\}$ -(1,1)-intervallaire (b). Un cycle impair (c), et l'ensemble de 2-intervalles dont il est le graphe $\{\frown, \sqsupset\}$ -(1,1)-intervallaire (d).

graphes sans P_4 , est incluse dans celle des graphes de permutation, donc un graphe non $\{\frown, \sqsupset\}$ -(1,1)-intervallaire ne peut être un cografe, il a donc P_4 comme sous-graphe induit. De plus, on a vu que les arbres, et donc P_4 , sont $\{\frown, \sqsupset\}$ -(1,1)-intervallaires, donc tous les graphes à 4 sommets ou moins sont $\{\frown, \sqsupset\}$ -(1,1)-intervallaires.

Il est possible de trouver un graphe à 9 sommets qui ne soit pas $\{\frown, \sqsupset\}$ -(1,1)-intervallaire, celui indiqué en figure 2.12. La construction de ce graphe est fondée sur le fait que les graphes bipartis complets ont une réalisation assez contrainte : peu de séquences arc-annotées à 6 sommets permettent d'obtenir $K_{3,3}$ comme graphe $\{\frown, \sqsupset\}$ -(1,1)-intervallaire (40 "seulement" sur 384 668). La

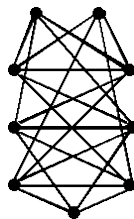


FIG. 2.12 – Contreexemple de graphe non- $\{\frown, \sqsupset\}$ -(1,1)-intervallaire.

suite d'opérations de génération et de filtrage de séquences arc-annotées, permettant de démontrer par un programme assez rapide qu'aucune n'est associée à ce graphe, est donnée en figure 2.13.

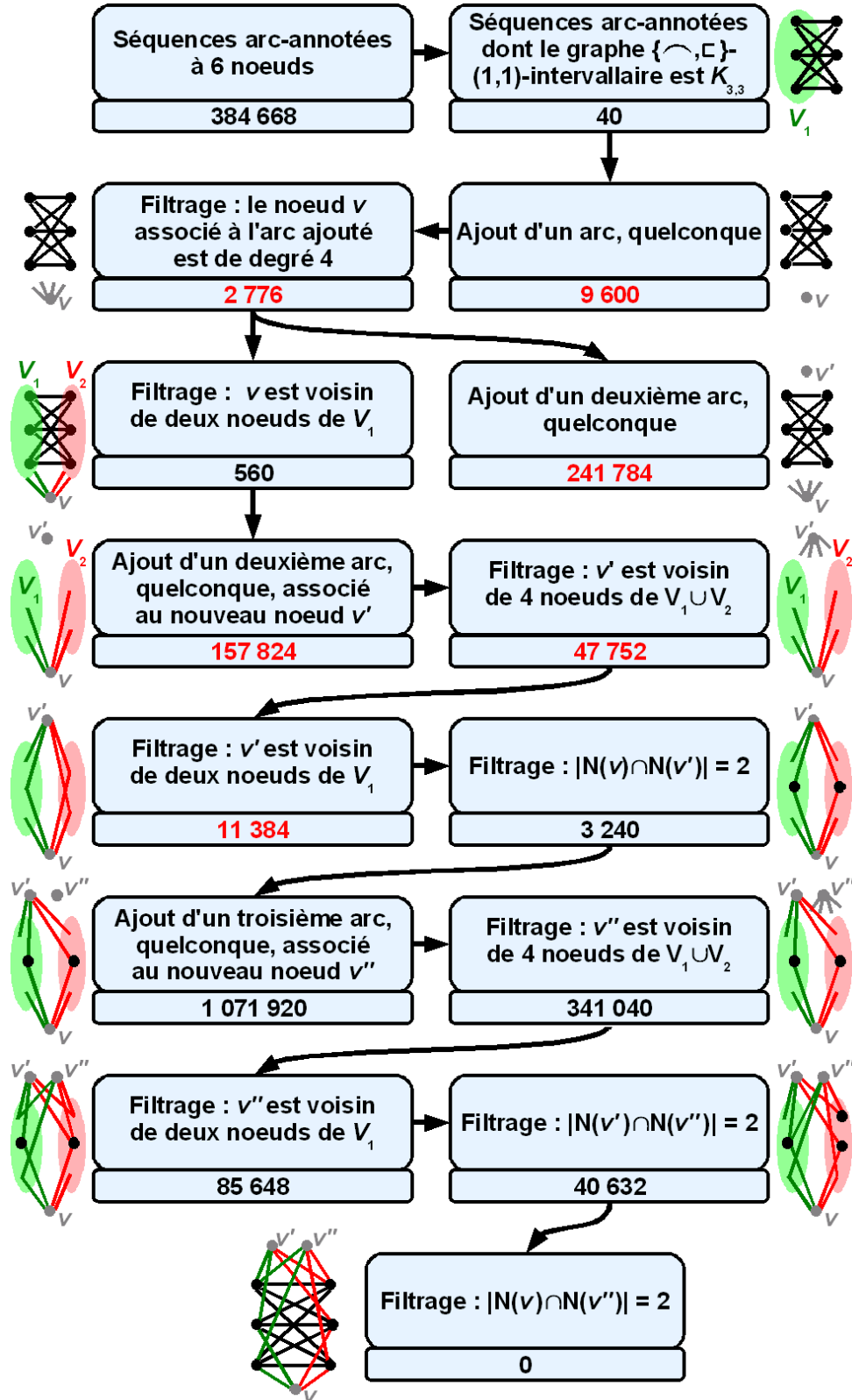


FIG. 2.13 – Séquence de génération et de filtrage de séquences arc-annotées pour un contreexemple de graphe non- $\{\curvearrowright, \sqsubset\}$ - $(1,1)$ -intervallaire. Les phases de filtrage servent à réduire le nombre d'objets à traiter de manière à éviter l'explosion combinatoire et obtenir un temps de traitement final assez court (moins d'une minute).

2.3.6 Classe des $\{\emptyset\}$ -(1,1)-intervallaires et des $\{\curvearrowright, \emptyset\}$ -(1,1)-intervallaires

Ces classes sont équivalentes à celle des *graphes de cordes*, ou, pour se rapporter à des intervalles, à celle des *graphes de chevauchement* (l'équivalence étant illustrée en figure 5.3).

En effet, soit G , un graphe $\{\emptyset\}$ -(1,1)-intervallaire à n noeuds. Considérons une réalisation en séquence arc-annotée et effectuons tous les éclatements $(\curvearrowright \mapsto \square, <)$ possibles. On obtient une séquence arc-annotée à arcs disjoints $\{([d_i, d_i + 1][f_i, f_i + 1])\}$. Si on considère les intervalles couvrants de ses arcs, leur graphe de chevauchement est exactement G (figure 2.14).

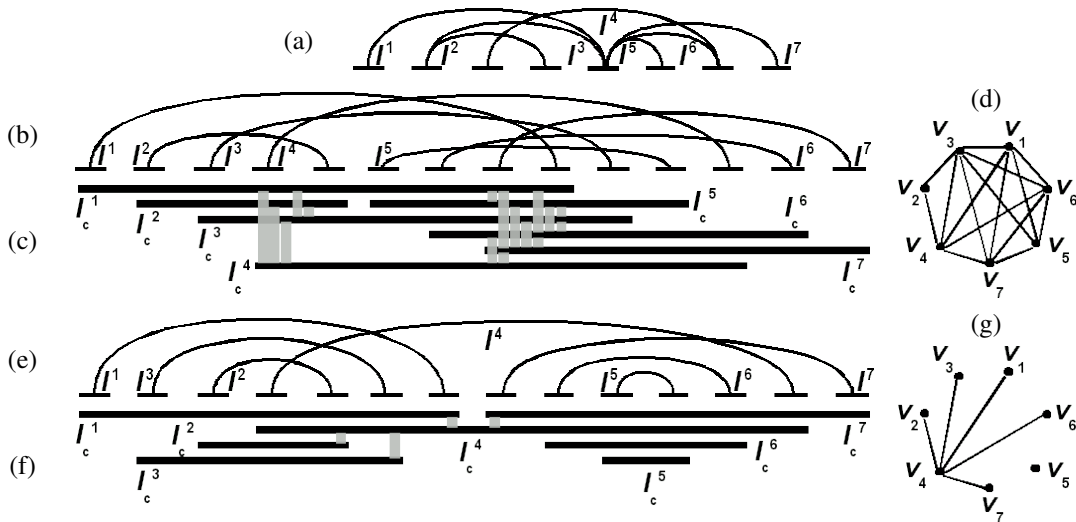


FIG. 2.14 – Une séquence arc-annotée (a), dont le graphe $\{\curvearrowright, \emptyset\}$ -(1,1)-intervallaire (d) est isomorphe au graphe de chevauchement de l'ensemble des intervalles couvrants (c) des (1,1)-intervalles associés à l'éclatement $(\curvearrowright \mapsto \emptyset)$ de cette séquence (b). D'autre part, son graphe $\{\emptyset\}$ -(1,1)-intervallaire (g) est isomorphe au graphe de chevauchement de l'ensemble des intervalles couvrants (f) des (1,1)-intervalles associés à son éclatement $(\curvearrowright \mapsto \square, <)$ (e).

De même, le graphe de chevauchement des intervalles couvrants du résultat de l'application de tous les éclatements $(\curvearrowright \mapsto \emptyset)$ de la réalisation d'un graphe $\{\curvearrowright, \emptyset\}$ -(1,1)-intervallaire G est exactement G .

2.3.7 Classe des $\{\curvearrowright, \emptyset\}$ -2-intervallaires

Cette classe est incluse dans celle des graphes de trapézoïdes circulaires, comme indiqué par la figure 2.15(b).

2.3.8 Classe des $\{\emptyset\}$ -2-intervallaires

La classe des graphes de croisement de trapézoïdes circulaires a été introduite dans [Via01], justement pour inclure la classe des graphes $\{\emptyset\}$ -2-intervallaires. C'est simplement une variante des graphes de trapézoïdes circulaires, dans laquelle on met en relation dans le graphe deux trapézoïdes circulaires T_1 et T_2 s'ils s'intersectent et que deux côtés adjacents de T_1 n'intersectent jamais un même côté de T_2 , comme illustré par la figure 2.15(c).

2.3.9 Classe des $\{\curvearrowright, <, \emptyset\}$ -2-intervallaires

Soit un ensemble de 2-intervalles $\{([j_i, k_i], [l_i, m_i])/i \in \{1, \dots, n\}\}$ et le graphe associé. Ce graphe est un graphe trapézoïdal. En effet on peut décomposer tout 2-intervalle en deux intervalles : $[j_i, k_i]$ qu'on place sur une première ligne, et $[-m_i, -l_i]$ qu'on place sur la deuxième ligne. On

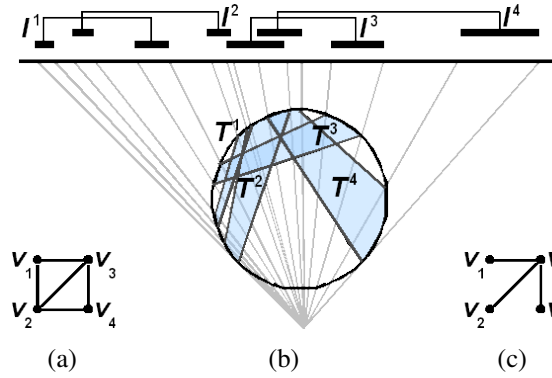


FIG. 2.15 – Un ensemble de 2-intervalles I^i et les trapézoïdes circulaires T^i associés (b). T^i intersecte T^j si et seulement si $I^i \cap I^j$ ou $I^i \not\supseteq I^j$, ce qui fournit le graphe $\{\cap, \not\supseteq\}$ -2-intervallaire associé (a). T^i croise T^j (seules des cordes ont une intersection non vide, pas les arcs des trapézoïdes) si et seulement si $I^i \not\supseteq I^j$, ce qui fournit le graphe $\{\not\supseteq\}$ -2-intervallaire associé (c).

remarque alors que deux 2-intervalles sont en relation d’emboîtement si et seulement si les trapèzes correspondants ne s’intersectent pas.

Inversement, on peut montrer que tout graphe trapézoïdal est $\{\cap, <, \not\supseteq\}$ -2-intervallaire, par une construction illustrée en figure 2.16. Soit l’ensemble de trapèzes $\mathcal{T} = \{([d_i, f_i], [d'_i, f'_i])/d_i, f_i, d'_i, f'_i \in \mathbb{R}\}$, et $m = \max_i f_i + \max_i f'_i$. Alors le graphe $\{\cap, <, \not\supseteq\}$ -2-intervallaire associé à l’ensemble de 2-intervalles $\{([d_i, f_i], [m + 1 - d'_i, m + 1 - f'_i])\}$ est exactement le graphe trapézoïdal associé à \mathcal{T} .

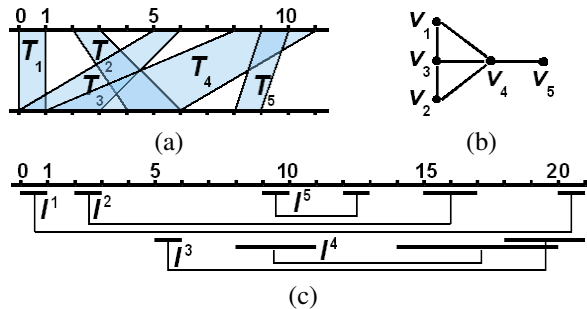


FIG. 2.16 – Un ensemble de trapèzes $\mathcal{T} = \{([0, 1], [0, 1]), ([2, 3], [4, 6]), ([5, 6], [0, 3]), ([8, 11], [1, 7]), ([9, 10], [8, 9])\}$ (a), le graphe trapézoïdal associé (b), ainsi qu’un ensemble de 2-intervalles ayant ce graphe comme graphe $\{\cap, <, \not\supseteq\}$ -2-intervallaire (c).

2.3.10 Classe des $\{\square\}$ -(1,1)-intervallaires et des $\{\cap, <, \not\supseteq\}$ -(1,1)-intervallaires

Cette classe est exactement celle des graphes de permutation. En effet, soit G , un graphe $\{\square\}$ -(1,1)-intervallaire.

Considérons une représentation de G , à laquelle on applique tous les éclatements ($\cap \mapsto \square, <$), comme montré en figures 2.17(a) et (b). A partir de la séquence arc-annotée aux arcs disjoints obtenue, on étiquette les arcs selon l’ordre de leur extrémité gauche. Soit $\sigma(i)$, l’ordre de l’extrémité droite de l’arc i parmi toutes les extrémités droites. La permutation σ ainsi définie a G pour graphe de permutation. Inversement, à partir de toute permutation σ de n entiers, on peut définir une séquence arc-annotée à arcs disjoints dont le graphe $\{\square\}$ -(1,1)-intervallaire sera isomorphe au graphe de permutation de σ : celle constituée des arcs $(i, n + \sigma(i))$, pour $i \in \llbracket 1, n \rrbracket$.

Comme énoncé en section 5.1.2, un graphe est un graphe de permutation si et seulement si c’est un graphe de comparabilité et de co-comparabilité. Ainsi, les complémentaires des graphes $\{\square\}$ -

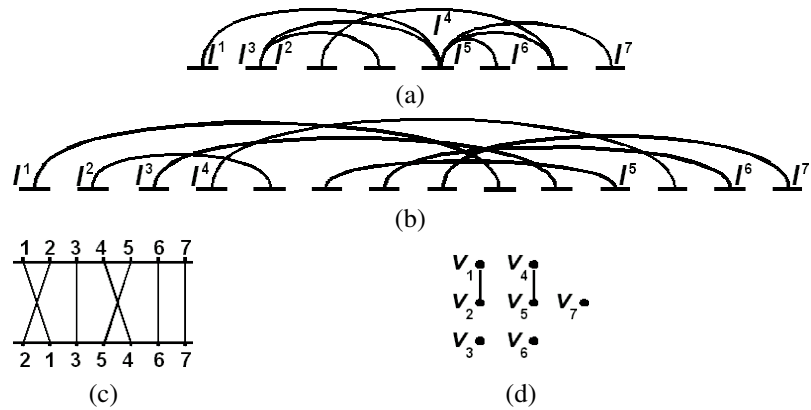


FIG. 2.17 – Une séquence arc-annotée (a) et le résultat de tous les éclatements ($\frown \mapsto \sqsubset, \prec$) possibles sur cette séquence (b). En étiquetant tous les arcs par l'ordre de leurs arcs gauches, la permutation définie par l'ordre des arcs droits $\begin{pmatrix} 1234567 \\ 2135467 \end{pmatrix}$ (c) a un graphe de permutation isomorphe au graphe $\{\sqsubset\}$ -(1,1)-intervallaire de la séquence arc-annotée de départ (d).

(1,1)-intervallaires, c'est à dire les graphes $\{\frown, \prec, \checkmark\}$ -(1,1)-intervallaires, sont aussi des graphes de permutation.

Chapitre 3

Stable max et recherche de motifs sur les 2-intervalles

Ce chapitre synthétise les divers algorithmes permettant de trouver au sein d'un ensemble de 2-intervalles le plus grand sous-ensemble de 2-intervalles disjoints, en autorisant seulement, sur ces 2-intervalles, certaines relations parmi $\{<, \boxtimes, \sqsubset\}$.

Le point d'interrogation de l'article [BFV04] sur la complexité de la recherche parmi un ensemble de 2-intervalles à support disjoint, du plus grand sous-ensemble de 2-intervalles qui sont uniquement en relation $<$ ou \boxtimes est toujours d'actualité, malgré diverses approches infructueuses résumées en section 3.4.

3.1 2-IP, la recherche de plus long motif respectant certaines contraintes

Description du problème :

Entrée : un ensemble de 2-intervalles $\mathcal{I} = \{I^1, \dots, I^n\}$, un entier positif k et un modèle $R \subseteq \{\curvearrowright, <, \sqsubset, \boxtimes\}$.

Question : existe-t-il un sous-ensemble $\mathcal{I}' \subset \mathcal{I}$, de taille k , qui soit R -comparable (c'est à dire que $\forall 1 \leq i < j \leq n, \exists \mathcal{R} \in R / I^i \mathcal{R} I^j$ ou $I^j \mathcal{R} I^i$) ?

La complexité varie selon le modèle utilisé (autoriser ou non que les 2-intervalles soient en relation $<$, \boxtimes ou \sqsubset). Dans les cas où le problème est NP-complet, Stéphane Vialette propose d'introduire les restrictions sur le support définies en section 2.1 afin d'obtenir une complexité moins élevée [Via01] :

- GENERAL : support sans restriction.
- UNITAIRE : support unitaire.
- DISJOINT : support disjoint.

Notons que le problème 2-IP(GENERAL, R) correspond à la recherche d'un stable maximum sur un graphe $(\{\curvearrowright, <, \sqsubset, \boxtimes\} \setminus R)$ -2-intervallaire, et le problème 2-IP(DISJOINT, R) à la recherche d'un stable maximum sur un graphe $(\{\curvearrowright, <, \sqsubset, \boxtimes\} \setminus R)$ -(1,1)-intervallaire.

Précisons cette correspondance dans le premier cas (le second se traitant de façon identique). Si $\curvearrowright \notin R$, pour résoudre le problème du stable maximum sur un graphe $(\{\curvearrowright, <, \sqsubset, \boxtimes\} \setminus R)$ -2-intervallaire G :

- trouver une réalisation en 2-intervalles de G .
- résoudre le problème 2-IP(GENERAL, R) sur cette réalisation.

Pour résoudre le problème 2-IP(GENERAL, R) sur un ensemble \mathcal{I} de 2-intervalles :

- construire G , le graphe $(\{\curvearrowright, <, \sqsubset, \boxtimes\} \setminus R)$ -2-intervallaire associé à \mathcal{I} .
- résoudre le problème du stable maximum sur G .

La complexité des deux problèmes n'est donc pas identique. On verra par exemple en section 3.9 qu'on atteint une complexité meilleure en évitant de construire le graphe pour résoudre 2-IP (GENERAL, $\{<\}$). Inversement, il est possible que pour un certain R (les cases non remplies du tableau 2.7),

Problème	2-IP GENERAL	2-IP UNITAIRE	2-IP DISJOINT
$\{<, \sqsubset, \emptyset\}$	NP-complet, cf 3.2		$O(n\sqrt{n})$, cf 3.3 (stable max de graphe adjoint)
$\{<, \emptyset\}$	NP-complet		? (cf 3.4)
$\{\sqsubset, \emptyset\}$	NP-complet		$O(n^2\sqrt{n})$, cf 3.5 (clique des intervalles couvrants)
$\{<, \sqsubset\}$	$O(n^2)$, cf 3.6 (programmation dynamique)		
$\{\emptyset\}$	$O(n^2 \log n)$, cf 3.7 (clique max de graphe de croisement de trapézoïdes circulaires)		$O(n\sqrt{n})$, cf 3.8 (clique max de graphe de cordes)
$\{<\}$	$O(n \log n)$, cf 3.9 (stable max de graphe d'intervalles)		
$\{\sqsubset\}$	$O(n \log n)$, cf 3.10 (stable max de graphe trapézoïdal)		

FIG. 3.1 – La complexité des diverses variantes du problème 2-IP. En particulier, l'utilisation des graphes de cordes permet d'obtenir une meilleure complexité que celle connue pour le problème 2-IP(DISJOINT, $\{\emptyset\}$). Celle de 2-IP(DISJOINT, $\{<, \emptyset\}$), reste en revanche inconnue.

trouver une réalisation d'un graphe $(\{\curvearrowright, <, \sqsubset, \emptyset\} \setminus R)$ -2-intervallaire soit NP-complet, alors que le problème 2-IP(GENERAL, R) serait polynomial...

Le tableau de la figure 3.1 indique les diverses complexités des variantes du problème 2-IP. On peut noter que la restriction de support unitaire semble n'apporter aucune amélioration : elle est plus intéressante pour les algorithmes d'approximation pour ce problème [CHLV05].

3.2 2-IP(GENERAL, $\{<, \sqsubset, \emptyset\})$

Trouver le stable maximum sur les graphes de 2-unions est NP-complet par réduction de 3-SAT [BNR96]. Donnons l'idée de la réduction : on se place dans le cadre de rectangles aux côtés parallèles aux axes s'"intersectant" (ou plus précisément, dont les projections sur les axes s'intersectent) pour illustrer un graphe 2-unions. Dans ce contexte, on cherche à représenter une formule à m clauses et n variables. Pour chaque variable x , on construit, comme illustré en figure 3.2, un "cycle" de $2m$ rectangles pour ses diverses valuations dans toutes les clauses $\{x_1, \bar{x}_1, \dots, x_m, \bar{x}_m\}$ de telle manière que la projection de chaque rectangle du cycle s'intersecte avec celle de ses 2 voisins (et seulement avec elles). Ainsi, choisir le plus grand ensemble indépendant sur ces cycles reviendra à choisir soit tous les x_i , soit tous les \bar{x}_i . Construire la i -ième clause contenant x_i (respectivement \bar{x}_i) reviendra à ajouter un rectangle dont la projection intersecte seulement celle du rectangle représentant \bar{x}_i (respectivement x_i), ainsi que celle des 2 rectangles correspondant aux 2 autres littéraux de la i -ième clause. De plus, on peut voir que cette construction peut être réalisée en utilisant seulement des carrés de côté 1, donc le problème 2-IP(UNITAIRE, $\{<, \sqsubset, \emptyset\})$ est aussi NP-complet.

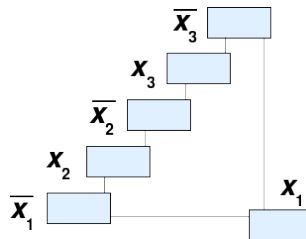


FIG. 3.2 – La réduction de 3-SAT à MAXIMUM INDEPENDENT SET, le problème de recherche du stable maximum, sur les graphes de 2-unions : des cycles de rectangles pour représenter les variables et leur valuation dans chaque clause. Les arêtes grises représentent la présence d'intersection des projections des rectangles.

On a même un résultat sur l'approximabilité de la recherche du stable pondéré maximum :

Théorème 4. [BYHN⁺02] MWIS est APX-dur¹ sur les graphes t -intervallaires, pour $t \geq 2$.

3.3 2-IP(DISJOINT, {<, □, ∅})

Le problème correspond à la recherche de stable maximum sur un graphe (1,1)-intervallaire à n sommets, qui est un graphe adjoint, d'un certain graphe G [Via01]. Il se rapporte donc à une recherche de couplage maximum sur G , qui a n arêtes, qu'on peut effectuer en temps $O(n\sqrt{n})$ [MV80].

3.4 2-IP(DISJOINT, {<, ∅})

Ce problème n'est toujours pas résolu. Résumons tout de même les approches tentées afin de le traiter.

3.4.1 Approche par la théorie des graphes

2-IP(DISJOINT, {<, ∅}) est équivalent à la recherche d'un stable maximum d'un graphe $\{\curvearrowright, \square\}$ -(1,1)-intervallaire. Ce problème est polynomial sur les graphes parfaits [GLS81], et sans triplet-astéroïde [BKKM99], mais les graphes $\{\curvearrowright, \square\}$ -(1,1)-intervallaires ne font pas partie de ces classes puisqu'ils contiennent les cycles (voir section 2.3.5) et donc respectivement C_5 et C_6 (ainsi que d'autres petits graphes à triplets astéroïdes comme illustré en figure 3.3).

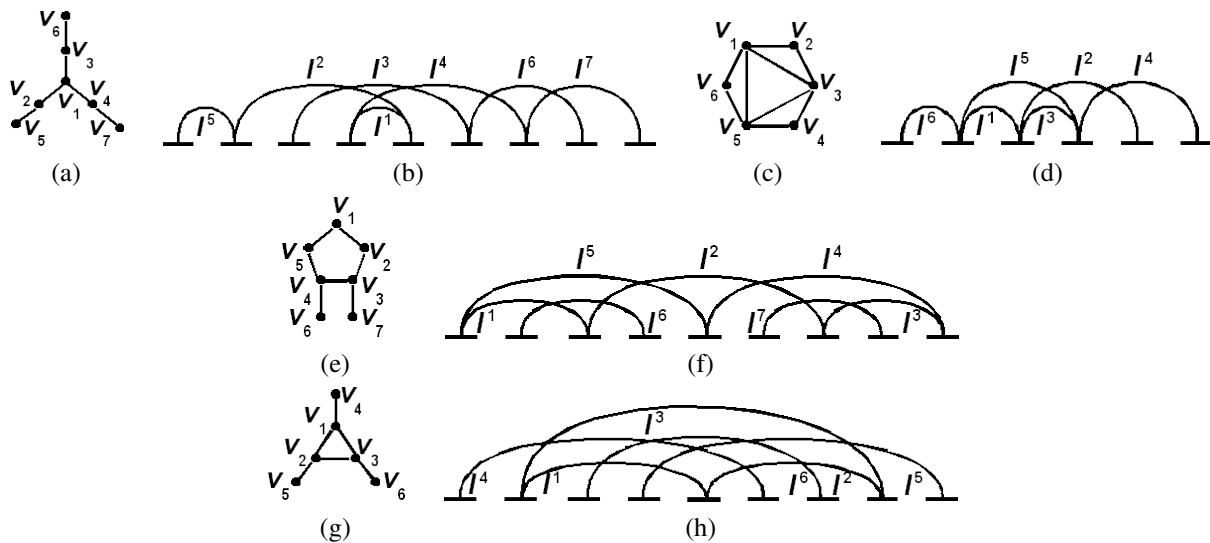


FIG. 3.3 – T_2 (a), graphe contenant un triplet astéroïde (les trois feuilles) est un graphe $\{\curvearrowright, \square\}$ -(1,1)-intervallaire (c). " S_3 " (c), graphe contenant un triplet astéroïde ($\{v_2, v_4, v_6\}$ par exemple), a aussi une réalisation $\{\curvearrowright, \square\}$ -(1,1)-intervallaire (d), tout comme le graphe (e) contenant un triplet astéroïde ($\{v_1, v_6, v_7\}$) (f), et le graphe (g) contenant un triplet astéroïde ($\{v_4, v_5, v_6\}$) (h).

3.4.2 Approche par programmation dynamique

Rappelons qu'on recherche dans un ensemble de 2-intervalles à support disjoint le plus grand sous-ensemble de 2-intervalles qui soient uniquement en relation de précédence ou d'entrelacement.

¹ dans le domaine des algorithmes d'approximation, APX-dur désigne un problème tel que s'il existe un algorithme $(1 + \epsilon)$ -approché pour le résoudre en temps polynomial alors il en existe un pour résoudre en temps polynomial tous les problèmes de la classe APX (ceux pour lesquels il existe un algorithme en temps polynomial, mais seulement ρ -approché avec $\rho > 1$).

Donnons tout d'abord une décomposition récursive de tout ensemble de 2-intervalles placés seulement en relation de précédence ou d'entrelacement, et montrons que cette décomposition ne suffit pas pour obtenir un algorithme polynomial de programmation dynamique.

Lemme 3. Soit $\{I^1, \dots, I^n\}$, un ensemble de 2-intervalles disjoints tels que $\forall i < j \in \llbracket 1, n \rrbracket, I^i < I^j$ ou $I^i \bowtie I^j$, et $I_1^i < I_1^j$ (les I^i sont uniquement en relation de précédence et d'entrelacement, et rangés dans le même ordre que leurs intervalles gauches). Alors $\exists \{k_1, \dots, k_m\} \subset \llbracket 1, n \rrbracket$ appelé ensemble de précédence tel que :

- $I^{k_1} < I^{k_2} < \dots < I^{k_m}$,
- $\forall i \in \llbracket 1, m-1 \rrbracket, \forall j \neq k \in \llbracket k_i, k_{i+1} - 1 \rrbracket, I^j \bowtie I^k$,
- et $\forall j \neq k \in \llbracket k_m, n \rrbracket, I^j \bowtie I^k$.

En d'autres termes, on peut extraire un ensemble de m 2-intervalles qui sont uniquement en relation de précédence les uns les autres (les trois en gras dans la figure 3.4), et tels que les 2-intervalles restants placés entre deux 2-intervalles consécutifs parmi ces m sont tous en relation d'entrelacement.

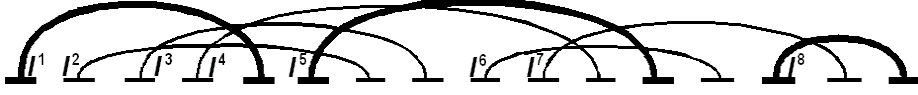


FIG. 3.4 – Décomposition d'un ensemble de 2-intervalles se précédant ou entrelacés : on extrait tout d'abord le 2-intervalle le plus à gauche, puis celui le plus à gauche qui est précédé par le premier, puis celui le plus à gauche précédé par le deuxième... : on obtient les 3 intervalles en gras : I^1 , I^5 et I^8 . Les intervalles restants peuvent être regroupés en ensembles de 2-intervalles "consécutifs" (selon l'ordre des intervalles gauche) tous en relation d'entrelacement : $\{I^1, I^2, I^3, I^4\}$, $\{I^5, I^6, I^7\}$ et $\{I^8\}$.

Démonstration. On procède par récurrence sur n .

Initialisation : trivial pour $n = 1, k_1 = 1$.

Hérédité : supposons que la propriété est vraie pour tout $k < n$ et montrons qu'elle l'est au rang $n + 1$. Considérons le 2-intervalle le plus à gauche $I^1 = (I_1^1, I_2^1)$, et l'ensemble des indices des 2-intervalles précédés par I^1 : $\{k \in \llbracket 2, n+1 \rrbracket / I^1 < I^k\}$

- soit cet ensemble est vide. Or les seules relations autorisées entre les I^k sont la précédence et l'entrelacement, donc $\forall k \in \llbracket 2, n+1 \rrbracket, I^k \bowtie I^1$. Supposons par l'absurde qu'il existe j et k dans $\llbracket 2, n+1 \rrbracket$ tels que $I^j < I^k$. Alors $I_2^j <_{\llbracket \]} I_1^k$. Or $I^1 \bowtie I^j$, et $I_1^1 <_{\llbracket \]} I_1^j$, donc $I_2^1 <_{\llbracket \]} I_2^j <_{\llbracket \]} I_1^k$. Ainsi, $I^1 < I^k$: absurde puisque $I^1 \bowtie I^k$! Donc $\exists \{k_1\} = \{1\}$ tel que $\forall j \neq k \in \llbracket 2, n+1 \rrbracket, I^j \bowtie I^k$, qui est la propriété recherchée.
- soit il n'est pas vide, on appelle k_2 son minimum. On applique alors l'hypothèse de récurrence sur $\{I^k / k \in \llbracket k_2, n+1 \rrbracket\}$ qui a une taille inférieure ou égale à n pour obtenir un ensemble $\{k_2, \dots, k_m\}$ qui vérifie les propriétés voulues. Il reste à prouver que $\forall j \neq k \in \llbracket 1, k_2 - 1 \rrbracket, I^j \bowtie I^k$. Déjà, d'après la définition de k_2 , pour $k \in \llbracket 1, k_2 - 1 \rrbracket, I^1$ ne précède aucun des autres I^k , donc $I^1 \bowtie I^k$. De même qu'au cas précédent, supposons par l'absurde qu'il existe j et k dans $\llbracket 2, k_2 - 1 \rrbracket$ tels que $I^j < I^k$. Alors $I_2^j <_{\llbracket \]} I_1^k$. Or $I^1 \bowtie I^j$, et $I_1^1 <_{\llbracket \]} I_1^j$, donc $I_2^1 <_{\llbracket \]} I_2^j <_{\llbracket \]} I_1^k$. Ainsi, $I^1 < I^k$: absurde puisque $I^1 \bowtie I^k$! Ainsi $\forall j \neq k \in \llbracket 1, k_2 - 1 \rrbracket, I^j \bowtie I^k$. En posant $k_1 = 1$, on a donc bien trouvé un ensemble $\{k_1, \dots, k_m\}$ vérifiant les propriétés voulues.

Etant initialisée et héréditaire, la propriété est bien vraie pour tout n . \square

Cette décomposition ne permet pas toutefois d'obtenir un algorithme polynomial de programmation dynamique. Même en ayant le plus grand motif \mathcal{I}_{k_2} dont l'ensemble de précédence a pour 2-intervalle le plus à gauche I^{k_2} , on ne peut en déduire directement le plus grand motif \mathcal{I}_{k_1, k_2} dont l'ensemble de précédence a pour 2-intervalles les plus à gauche $I^{k_1} < I^{k_2}$. En effet, il est possible que certains 2-intervalles entrelacés avec I^{k_1} aient leur extrémité droite entre $I_1^{k_2}$ et $I_2^{k_2}$, empêchant ainsi de sélectionner certains 2-intervalles de \mathcal{I}_{k_2} entrelacés avec I^{k_2} , comme montré en figure 3.5.

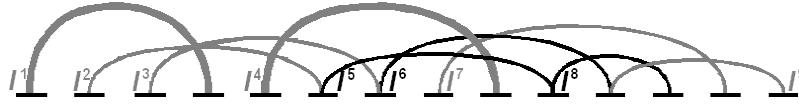


FIG. 3.5 – Exemple problématique pour l'utilisation récursive du lemme 3 : on ne peut trouver un stable maximum dont l'ensemble de précédence est $\{I^1, I^4\}$ à partir d'un stable maximum dont l'ensemble de précédence est $\{I^4\}$, puisque sélectionner dans le stable maximum les arcs I^2 et I^3 , empêche de sélectionner les arcs I^5 et I^6 . Cette sélection de I^2 et I^3 permet toutefois de trouver le stable maximum, constitué des 6 arcs en gris.

3.4.3 Perspectives

Les algorithmes sur les séquences arc-annotées avec des restrictions d'arcs uniquement en relation d'entrelacement ou de précédence sont rares. Le problème du bicoloriage² sur de telles séquences peut être réalisé de façon simple (par l'algorithme glouton naturel) en temps polynomial [Bou06], mais fait apparaître la décomposition exhibée par le lemme 3.

En outre, l'algorithme de programmation dynamique proposé dans [Gra04], afin de déterminer si un certain motif de (1,1)-intervalles uniquement en relation de précédence ou d'entrelacement peut être trouvé au sein d'un ensemble, plus grand, de (1,1)-intervalles, s'avère finalement buggé, Shuai-Cheng Li et Ming Li fournissant un contreexemple et finalement une preuve de NP-complétude pour le problème [LL06].

Mes tentatives de modification du graphe de la preuve "classique" de NP-complétude de la recherche du stable maximum (par une réduction de 3-SAT) pour le rendre $\{\curvearrowright, \square\}$ -(1,1)-intervallaire, ou de recherche d'une réduction plus directe depuis 3-SAT n'ont toutefois pas fonctionné.

3.5 2-IP(DISJOINT, $\{\square, \emptyset\}$)

L'utilisation des intervalles couvrants permet d'obtenir un algorithme polynomial, en $O(n^2\sqrt{n})$, illustré en figure 3.6.

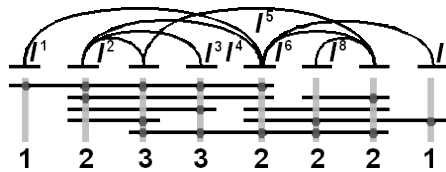


FIG. 3.6 – Exemple de résolution du problème 2-IP(DISJOINT, $\{\square, \emptyset\}$) : toutes les cliques du graphe d'intervalles des intervalles couvrants sont trouvées (matérialisées par les barres grises verticales), puis on extrait pour chacune le sous-ensemble maximum de (1,1)-intervalles disjoints (indiqués par une pastille foncée, la taille de l'ensemble étant mentionnée sur la dernière ligne). $\{I^1, I^3, I^5\}$ est donc par exemple un ensemble maximal de (1,1)-intervalles disjoints et sans relation de précédence.

En effet, les (1,1)-intervalles du motif solution feront partie d'une clique du graphe d'intervalles des intervalles couvrants, puisque les 2-intervalles du motif ne peuvent être en relation de précédence. Or la recherche de toutes les cliques se fait en temps linéaire. Il suffit de parcourir l'ensemble des intervalles couvrants, de gauche à droite : chacun des e intervalles du support correspond à un point de fin ou de début d'un intervalle couvrant, donc les intervalles couvrants intersectant cet intervalle de longueur 1 constituent une clique.

² Ce problème consiste à colorier les extrémités des arcs de deux couleurs différentes, de telle sorte que la couleur du début et la fin de chaque arc soit différente, en minimisant le nombre de changements de couleurs entre deux extrémités consécutives.

Pour chacune des $e = O(n)$ cliques ainsi trouvées, on extrait avec l'algorithme de résolution de 2-IP(DISJOINT, $\{<, \sqsubset, \emptyset\}$) (en $O(n\sqrt{n})$, voir section 3.3), parmi les (1,1)-intervalles associés à ces intervalles couvrants, le sous-ensemble maximal de (1,1)-intervalles disjoints.

3.6 2-IP(GENERAL, $\{<, \sqsubset\}$)

Cette restriction du problème 2-IP est particulièrement utile aux biologistes puisqu'elle correspond à un ARN sans pseudo-noeud. Il existe un algorithme quadratique de programmation dynamique [Via04] qui recherche récursivement des motifs I_1XI_2Y , où X est le plus grand ensemble de 2-intervalles en relation $<$ ou \sqsubset , et tous emboîtés dans le 2-intervalle (I_1, I_2) , et Y le plus grand ensemble de 2-intervalles en relation $<$ ou \sqsubset , tous précédés par (I_1, I_2) .

3.7 2-IP(GENERAL, $\{\emptyset\}$)

Ce problème revient à rechercher une clique maximale dans un graphe de croisement de trapézoïdes circulaires G [Via01], ce qui revient à chercher, pour tout sommet v du graphe, une clique maximale dans le sous-graphe de G induit par v et son voisinage : $G[\mathcal{N}[v]]$. Or il existe une clique de taille k dans un tel graphe si et seulement si il existe un stable de taille k dans le graphe $\overline{G[\mathcal{N}[v]]}$, qui s'avère être un graphe trapézoïdal. Comme mentionné ci-dessus en section 3.10, c'est faisable en $O(n \log n)$ [FMW97]. La complexité finale du problème est donc $O(n^2 \log n)$.

3.8 2-IP(DISJOINT, $\{\emptyset\}$)

Comme on l'a vu en section 2.3.6, ce problème est équivalent à la recherche de clique maximale dans le graphe $\{\emptyset\}$ -(1,1)-intervallaire, qui est un graphe de cordes. Or, il existe un algorithme en $O(n^2)$ [AAH92], et même depuis très récemment en $O(n\sqrt{n})$ [Tis06] pour ce problème sur un graphe à n sommets, sous réserve de fournir en entrée de l'algorithme une représentation associée au graphe. La complexité connue pour ce problème est donc améliorée, de $O(n^2 \log n)$ à $O(n\sqrt{n})$.

3.9 2-IP(GENERAL, $\{<\}$)

Le problème revient à trouver un stable maximum dans le graphe d'intervalles des intervalles couvrants des 2-intervalles [Via04]. Ceci peut être fait sans construire le graphe d'intervalles, seulement à partir de l'ensemble des intervalles, en $O(n \log n)$ [GLL82]. Si l'on fournit en entrée de l'algorithme la liste ordonnée des extrémités des intervalles, on peut même utiliser un algorithme en $O(n)$ [HTC92].

3.10 2-IP(GENERAL, $\{\sqsubset\}$)

Comme on l'a vu en section 2.3.9, ce problème est équivalent à la recherche de stable maximum dans un graphe trapézoïdal [BFV04]. Or il est possible d'éviter la construction de ce graphe pour trouver parmi une famille de trapèzes le sous-ensemble, de taille maximale, de trapèzes disjoints en $O(n \log n)$ [FMW97].

Mentionnons aussi que pour le cas 2-IP(DISJOINT, $\{\sqsubset\}$), le problème est équivalent à la recherche de stable maximum dans un graphe de permutation, qui peut s'effectuer en $O(n \log \log n)$ [Mäk99]. Encore faut-il fournir la permutation à l'algorithme à partir des (1,1)-intervalles à notre disposition, ce qui peut être fait en temps linéaire par la construction vue en section 2.3.10 si l'on a la liste des (1,1)-intervalles classés selon l'ordre de leurs intervalles gauches.

Chapitre 4

2-intervalles équilibrés

Un 2-intervalle (I_1, I_2) est dit *équilibré* si les intervalles I_1 et I_2 ont même longueur. On note $l(I_1)$ la longueur de I_1 .

Cette restriction est naturelle dans le contexte de l'ARN, où les 2-intervalles représentant des blocs complémentaires constituent une hélice, et sont donc de même taille. Elle permet de plus d'obtenir une amélioration du ratio d'approximation pour le problème 2-IP($\{<, \emptyset\}$) [CHLV05]. Nous répondons avec le théorème suivant à une question de Stéphane Vialette.

4.1 Inclusion stricte de la classe des graphes 2-intervallaires équilibrés

Théorème 5. *La classe des graphes 2-intervallaires équilibrés est strictement incluse dans celle des graphes 2-intervallaires.*

Démonstration. Le graphe de la figure 4.5 est un graphe 2-intervallaires qui n'admet aucune réalisation en 2-intervalles équilibrés.

On détaille la construction de ce contreexemple en trois étapes : un ensemble de huit 2-intervalles qui permet de simuler un intervalle (présenté en figure 4.2), un ensemble de huit 2-intervalles I^0 à I^7 qui force l'ordre de placement de 7 intervalles ainsi simulés J^1 à J^7 (en figure 4.4) pour réaliser une structure qui empêche d'équilibrer trois 2-intervalles I^1 , I^2 et I^3 (en figure 4.1). \square

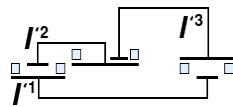


FIG. 4.1 – Si l'on arrive à imposer une telle structure dans la réalisation d'un graphe 2-intervallaire, comme il est impossible d'y équilibrer les intervalles I^1 , I^2 et I^3 , alors le graphe est un graphe 2-intervallaire non équilibré.

4.1.1 Etape 1 : huit 2-intervalles pour simuler un intervalle

Le graphe G_I de la figure 4.2(a) permet de simuler un intervalle, c'est à dire que si dans un graphe G ayant G_I comme sous-graphe induit, un noeud v_k n'est relié à aucun des 8 noeuds de G_I , alors dans toute réalisation de G , les intervalles supports du 2-intervalle I^k correspondant à v_k n'intersecteront pas l'intervalle couvrant de l'ensemble des huit 2-intervalles associés aux noeuds de G_I . On peut l'exprimer autrement : l'union de tous les intervalles du support de toute réalisation de G_I est un intervalle.

Etudions les réalisations possibles de G_I . Pour cela, étudions toutes les réalisations possibles de $G_I[V \setminus \{v_8\}]$. Les 2-intervalles I^1 , I^2 , I^3 et I^4 étant disjoints, leur support $\mathcal{I}_{\text{fixe}} = \{[d_i, f_i], 1 \leq i \leq 8,$

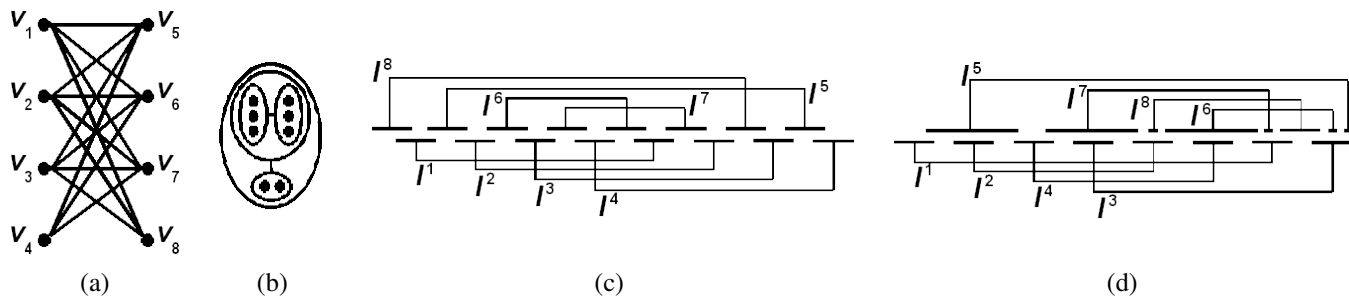


FIG. 4.2 – Toute réalisation du graphe (a) (représenté plus lisiblement en (b) grâce à la décomposition modulaire [dm03]), "regroupe" les 2-intervalles $I^1, I^2, I^3, I^4, I^5, I^6, I^7$ et I^8 en un bloc. Ce graphe a une réalisation où chaque intervalle intersecte au maximum deux intervalles (c), et une où certains intervalles (I_1^5, I_1^7 et I_2^2 par exemple) intersectent chacun trois intervalles (d).

$f_i < d_{i+1}$ est un ensemble de huit intervalles disjoints. De même, $\mathcal{I}_{\text{mobile}}$, le support de I^5, I^6 et I^7 , est un ensemble de six intervalles disjoints. Soit x_i , le nombre d'intervalles de $\mathcal{I}_{\text{mobile}}$ qui intersectent $i \leq 8$ intervalles de $\mathcal{I}_{\text{fixe}}$. On en déduit directement :

$$x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 = |\mathcal{I}_{\text{mobile}}| = 6. \quad (4.1)$$

De plus, comme il y a 12 arêtes dans $G_I[V \setminus \{v_8\}]$ qui est biparti, alors :

$$x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 + 6x_6 + 7x_7 + 8x_8 = 12. \quad (4.2)$$

Supposons par l'absurde que quatre intervalles de $\mathcal{I}_{\text{mobile}}$ intersectent chacun moins de deux intervalles de $\mathcal{I}_{\text{fixe}}$. Ces quatre intervalles appartiennent au support de trois 2-intervalles disjoints (I^5, I^6 et I^7), donc nécessairement deux de ces quatre intervalles constituent le support d'un de ces trois 2-intervalles. Ce dernier intersecte donc moins de quatre 2-intervalles, ce qui est impossible, donc :

$$x_0 + x_1 < 4. \quad (4.3)$$

On peut aussi préciser que pour chaque intervalle support de I^5, I^6 ou I^7 , intersectant moins de deux intervalles de $\mathcal{I}_{\text{fixe}}$, il faut que l'autre intervalle support intersecte strictement plus de deux intervalles de $\mathcal{I}_{\text{fixe}}$ donc :

$$x_0 + x_1 \leq x_3 + x_4 + x_5 + x_6 + x_7 + x_8. \quad (4.4)$$

Enfin, pour que l'on puisse construire à partir d'une réalisation de $G_I[V \setminus \{v_8\}]$ une réalisation de G_I , on doit pouvoir placer I^8 de telle sorte qu'il intersecte trois intervalles disjoints de $\mathcal{I}_{\text{fixe}}$. Ainsi, un des intervalles supports de I^8 intersecte au moins deux intervalles $[d_k, f_k]$ et $[d_l, f_l]$ ($k < l$) de $\mathcal{I}_{\text{fixe}}$. Donc il existe un "trou entre ces deux intervalles", par exemple $[f_k, d_{k+1}]$, qui est inclus dans un des intervalles supports de I^8 . Ainsi, on remarque que I^8 va devoir boucher un des sept trous de $\mathcal{I}_{\text{fixe}}$. Les intervalles de $\mathcal{I}_{\text{mobile}}$ ne peuvent donc boucher plus de 6 trous, et en observant qu'un intervalle qui intersecte i intervalles consécutifs, pour $i \geq 1$, bouche $i - 1$ trous, on obtient :

$$x_2 + 2x_3 + 3x_4 + 4x_5 + 5x_6 + 6x_7 + 7x_8 \leq 6. \quad (4.5)$$

Ainsi, les équations 4.1, 4.2, 4.3, 4.4 et 4.5 sont nécessairement vraies pour toute réalisation de $G_I[V \setminus \{v_8\}]$ permettant une réalisation valide de G_I .

Supposons par l'absurde que l'union de tous les intervalles du support de toute réalisation de G_I n'est pas un intervalle. Alors il existe un trou, c'est à dire un intervalle inclus dans l'intervalle couvrant de $\{I^1, \dots, I^8\}$, n'intersectant aucun I^i . Cela implique qu'un nouveau trou ne pourra être bouché par les intervalles de $\mathcal{I}_{\text{mobile}}$, donc l'équation 4.5 devient :

$$x_2 + 2x_3 + 3x_4 + 4x_5 + 5x_6 + 6x_7 + 7x_8 \leq 5. \quad (4.6)$$

En ajoutant 4.1 et 4.6, et en soustrayant 4.2, on obtient $x_0 \leq -1$: absurde. Donc l'union de tous les intervalles du support de toute réalisation de G_I est bien un intervalle.

Remarque 1 : on peut continuer les calculs pour trouver des conditions nécessaires sur la réalisation de G_I en commençant par ajouter 4.1 et 4.5, et en soustrayant 4.2, pour obtenir $x_0 \leq 0$, c'est à dire $x_0 = 0$. Ceci fournit le système :

$$\begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 = 6 \\ x_2 + 2x_3 + 3x_4 + 4x_5 + 5x_6 + 6x_7 + 7x_8 = 6 \\ x_1 < 4 \\ x_1 < x_3 + x_4 + x_5 + x_6 + x_7 + x_8 \end{cases}$$

On remarque alors qu'il n'y a pas de solution s'il existe $i > 3$ tel que $x_i > 0$. Ainsi les intervalles de $\mathcal{I}_{\text{mobile}}$ intersectent chacun au plus 3 intervalles de $\mathcal{I}_{\text{fixe}}$. Deux réalisations possibles sont indiquées en figure 4.2c) et d).

Remarque 2 : on observe que cet ensemble de huit 2-intervalles ne simule pas exactement un intervalle. Pour un intervalle I , on peut construire un nombre aussi grand que l'on veut d'intervalles deux à deux disjoints pouvant intersecter I , alors que seuls **deux** intervalles disjoints peuvent avoir une intersection non vide avec tous les 2-intervalles I^1, \dots, I^8 (dans la réalisation de la figure 4.2c), intersecter le premier intervalle avec seulement tous les intervalles gauches et le second avec seulement tous les intervalles droits).

4.1.2 Etape 2 : huit 2-intervalles pour contraindre l'ordre de placement de 7 intervalles

Le graphe de la figure 4.4(a) contient sept sous-graphes induits isomorphes à G_I , dont l'ensemble des 2-intervalles dans la réalisation est appelée J^i , $1 \leq i \leq 7$. Montrons que dans toute réalisation de ce graphe, l'ordre des J^i est imposé, à symétrie près. Plus formellement, en appelant $I_{\text{couvrant}}(J^k)$ l'intervalle couvrant de J^k , montrons que

- soit $I_{\text{couvrant}}(J^1) <_{[]} I_{\text{couvrant}}(J^2) <_{[]} \dots <_{[]} I_{\text{couvrant}}(J^7)$,
- soit $I_{\text{couvrant}}(J^7) <_{[]} I_{\text{couvrant}}(J^6) <_{[]} \dots <_{[]} I_{\text{couvrant}}(J^1)$.

Montrons tout d'abord que si J^i et J^j sont disjoints (tous les 2-intervalles de J^i sont disjoints des 2-intervalles de J^j), alors les intervalles couvrants $I_{\text{couvrant}}(J^i)$ et $I_{\text{couvrant}}(J^j)$ sont disjoints. Supposons par l'absurde que ce ne soit pas le cas. Alors il existe un 2-intervalle $I^{\text{intersectant}}$ dont un des intervalles support intersecte $I_{\text{couvrant}}(J^j)$. Or d'après la propriété démontrée à la section précédente, $I^{\text{intersectant}}$ n'intersectant aucun 2-intervalle de J^j , il n'intersecte pas $I_{\text{couvrant}}(J^j)$. Contradiction ! Ainsi, les J^i étant deux à deux disjoints, leurs intervalles couvrants $I_{\text{couvrant}}(J^i)$ sont aussi deux à deux disjoints, donc on peut définir σ , la permutation associée à l'ordre des $I_{\text{couvrant}}(J^i)$, c'est à dire telle que $I_{\text{couvrant}}(J^{\sigma(1)}) <_{[]} I_{\text{couvrant}}(J^{\sigma(2)}) <_{[]} \dots <_{[]} I_{\text{couvrant}}(J^{\sigma(7)})$.

On cherche donc à montrer que pour toute réalisation du graphe de la figure 4.4, seules deux permutations sont possibles pour σ (l'identité et son renversement).

$I^0 = (I_1, I_2)$, $I^1 = (I_3, I_4)$, $I^2 = (I_5, I_6)$ et $I^3 = (I_7, I_8)$ sont des 2-intervalles disjoints, leur support $\mathcal{I}_{\text{disjoints}}$ est donc constitué de huit intervalles disjoints, séparés par sept intervalles "trous". On peut donc définir σ' , la permutation associée à l'ordre des I_k c'est à dire telle que $I_{\sigma'(1)} <_{[]} I_{\sigma'(2)} <_{[]} \dots <_{[]} I_{\sigma'(8)}$.

Sept intervalles disjoints, les $I_{\text{couvrant}}(J^i)$ doivent intersecter chacun deux intervalles disjoints parmi huit (les I_k), ils doivent donc chacun "boucher" un intervalle trou différent.

En notant $I_{\text{couvrant}}(J^i) = [d_i, f_i]$, et $I_k = [d'_k, f'_k]$, on a donc :

$$d'_{\sigma'(1)} < d_{\sigma(1)} < f'_{\sigma'(1)} < d'_{\sigma'(2)} < f_{\sigma(1)} < d_{\sigma(2)} < f'_{\sigma'(2)} < \dots < d_{\sigma(7)} < f'_{\sigma'(7)} < d'_{\sigma'(8)} < f_{\sigma(7)} < f'_{\sigma'(8)}$$

Ainsi, en suivant la ligne des réels dans l'ordre croissant à partir de $d'_{\sigma'(1)}$ on rencontre successivement un intervalle support de I^l ($1 < l < 4$), un $I_{\text{couvrant}}(J^i)$, un intervalle support de I^l , un $I_{\text{couvrant}}(J^i)$, \dots , un $I_{\text{couvrant}}(J^i)$, un intervalle support de I^l . Chaque fois que l'on rencontre un

$I_{\text{couvrant}}(J^i)$, comme il n'intersecte que deux 2-intervalles I^l et qu'on a déjà rencontré un des deux I^l qu'il intersecte, alors le prochain intervalle rencontré sera intervalle support de l'autre 2-intervalle qu'il intersecte. Ainsi, à chaque ordre possible sur les I_k et les $I_{\text{couvrant}}(J^i)$, on peut associer un chemin eulérien dans le multigraphe de la figure 4.3, dont les noeuds correspondent aux 2-intervalles I^l et dont les arêtes correspondent aux intervalles $I_{\text{couvrant}}(J^i)$. σ est l'ordre des arêtes dans le parcours, et $\lfloor \frac{\sigma'-1}{2} \rfloor$ est l'ordre des noeuds dans le parcours (la fonction $k \mapsto \lfloor \frac{k-1}{2} \rfloor$ donne le numéro du 2-intervalle dont I_k est intervalle support).

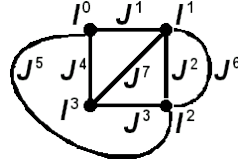


FIG. 4.3 – L'ensemble des parcours eulériens de ce multigraphe est en bijection avec l'ensemble des ordres possibles pour les I^k et les $I_{\text{couvrant}}(J^i)$ du graphe de la figure 4.4(a) dont on exclut les sommets I^k , $k > 3$. Par exemple, le parcours eulérien $(I^0, J^1, I^1, J^2, I^2, J^3, I^3, J^4, I^0, J^5, I^2, J^6, I^1, J^7, I^3)$ correspond à l'ordre visible dans la réalisation de la figure 4.4(b).

On génère donc tous les ordres possibles grâce à un algorithme qui énumère tous les parcours eulériens sur le multigraphe de la figure 4.3 (procédure récursive qui pour tout noeud du graphe explore en profondeur le graphe jusqu'à ce qu'il atteigne arête déjà visitée) : il y en a 88.

Prenons maintenant en compte les contraintes venant des 2-intervalles I^4, I^5, I^6 et I^7 , intersectant respectivement les cinq intervalles $I^1 J^1 J^2 J^3$ et $I^3, I^2 J^2 J^3 J^4$ et $I^0, I^3 J^3 J^4 J^5$ et $I^2, I^0 J^4 J^5 J^6$ et I^1 . Ainsi, chaque 2-intervalle I^l , $4 \leq l \leq 7$, intersecte 3 intervalles couvrants que l'on renomme $J_1(l), J_2(l)$ et $J_3(l)$, et deux 2-intervalles que l'on renomme $I_1(l)$ et $I_2(l)$. En considérant $\mathcal{P}_l = \{J_1(l), J_2(l), J_3(l), I_1(l), I_2(l)\}$ comme un ensemble de points de passage du parcours eulérien, il faut que pour tout $4 < l < 7$, le parcours contienne 5 points de \mathcal{P}_l présents consécutivement ou 4 points de \mathcal{P}_l présents consécutivement et un autre quelconque, ou encore 3 points de \mathcal{P}_l présents consécutivement et 2 autres points \mathcal{P}_l présents consécutivement (précisons que ces conditions sont nécessaires et pas suffisantes, elles servent juste à éliminer les parcours eulériens invalides). Or, on vérifie que seuls quatre parcours eulériens parmi les 88 contiennent 3 points de \mathcal{P}_l placés consécutivement pour tous les \mathcal{P}_l :

- le parcours voulu $(I^0, J^1, I^1, J^2, I^2, J^3, I^3, J^4, I^0, J^5, I^2, J^6, I^1, J^7, I^3)$,
- son symétrique,
- $C = (I^0, J^1, I^1, J^2, I^2, J^3, I^3, J^7, I^1, J^6, I^2, J^5, I^0, J^4, I^3)$,
- son symétrique.

Or sur le parcours C , considérons $\mathcal{P}_6 = \{J^3, J^4, J^5, I^2, I^3\}$: J^2, I^2 et J^3 sont bien consécutifs dans C , mais pour atteindre J^5 , on est obligé de passer par $J^6 \notin \mathcal{P}_6$, et pour atteindre J^4 , même depuis J^5 , on doit passer par $I^0 \notin \mathcal{P}_6$. Donc ce parcours est invalide, et son symétrique l'est aussi.

Finalement, il existe bien un seul ordre possible (à symétrie près) sur les J^i . Par la suite, on considère seulement les réalisations respectant l'ordre $I_{\text{couvrant}}(J^1) <_{[]} I_{\text{couvrant}}(J^2) <_{[]} \dots <_{[]} I_{\text{couvrant}}(J^7)$, l'autre cas se traitant par symétrie.

4.1.3 Etape 3 : 7 intervalles dans un ordre contraint pour empêcher l'équilibrage de trois 2-intervalles

On ajoute au graphe obtenu à l'étape 2 trois noeuds, I^1, I^2 et I^3 et les intersections indiquées en figure 4.5. Etudions toutes les réalisations possibles de ce graphe en fonction des contraintes impliquées par ses arêtes :

- I^3 intersecte J^5, J^6 et I^3 , et n'intersecte ni I^0 placé entre J^5 et l'intervalle gauche de I^3 , ni I^1 placé entre J^6 et l'intervalle droit de I^3 . Donc un de ses intervalles supports intersecte à la fois

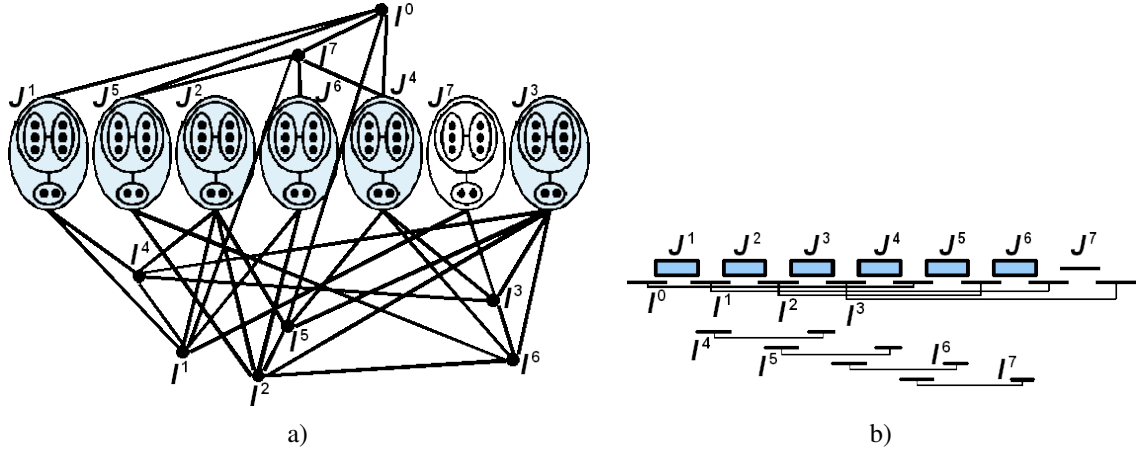


FIG. 4.4 – Toute réalisation du graphe (a) contraint l'ordre de $J^1, J^2, J^3, J^4, J^5, J^6$ et J^7 , en particulier la réalisation (b).

J^5 et J^6 . Il peut alors intersecter aussi I^2 et I^6 , donc le second intervalle de I^3 doit intersecter I^3, I^1 et I^2 .

- De même, I^2 intersecte J^3, J^4 et I^1 , et n'intersecte ni I^2 placé entre J^3 et l'intervalle gauche de I^1 , ni I^0 placé entre J^4 et l'intervalle droit de I^1 . Donc un de ses intervalles supports intersecte à la fois J_3 et J_4 . Il peut alors intersecter aussi I^3, I^4 , et I^6 donc le second intervalle de I^3 doit intersecter I^1, I^1 et I^3 .
- On a donc placé un intervalle de I^1 et un intervalle de I^2 , de façon disjointe. Cherchons où placer le second intervalle de I^3 . Pour cela, étudions toutes les possibilités pour placer un point d'intersection de I^2 et I^3 , qu'on appelle $x \in \mathbb{R}$. Soit x se trouve en dehors des intervalles déjà placés, c'est à dire à gauche de l'intervalle gauche de I^0 ou à droite de l'intervalle droit de I^3 : dans le premier cas, comme I^2 n'intersecte pas I^0 , il ne pourra pas intersecter I^1 ; dans le second, comme I^2 n'intersecte pas J^7 , il ne pourra pas intersecter I^1 . x appartient donc à un des intervalles supports des I^k ou un des J^k . Seuls I^3 et I^6 intersectent à la fois I^2 et I^3 . Supposons que x appartienne à l'intervalle droit de I^6 , alors x appartient aussi à J^5 ou I^2 : impossible car I^2 n'intersecte pas J^5 ni I^2 . Supposons alors que x appartient à l'intervalle droit de I^3 : impossible, car I^2 qui n'intersecte pas J^7 ne pourra alors pas intersecter I^1 . Il reste donc le dernier cas : x appartient à l'intervalle gauche de I^3 et l'intervalle gauche de I^6 . Donc le second intervalle de I^3 intersecte l'intervalle déjà placé de I^2 . **Il est même strictement inclus dedans** puisqu'il n'intersecte ni J^3 ni J^4 .
- Il reste à placer le second intervalle de I^2 ainsi que les deux de I^1 . I^1 doit intersecter J^2 et I^3 . Or il n'intersecte pas J^3 , placé entre J^2 et les deux intervalles de I^3 . Ainsi, l'intervalle gauche de I^1 intersecte J^2 , et son intervalle droit intersecte I^3 . Comme I^1 n'intersecte pas I^3 , alors son intervalle droit I_2^1 intersecte en fait l'intervalle droit I_2^3 de I^3 . Or il n'intersecte pas J^5 ni J^6 , donc I_2^1 **est strictement inclus dans** I_2^3 , il intersecte alors bien I^2 . L'intervalle gauche de I^1 doit intersecter J^1, J^2, I^1, I^4 .
- Considérons un point d'intersection y de I^2 et I^1 . S'il appartient à l'intervalle droit de I^1 , alors il est aussi strictement entre J^5 et J^6 , donc il ne peut intersecter I^1 ! y appartient donc à l'intervalle gauche de I^1 , qui intersecte lui-même J^1, J^2, I^1, I^4 et le second intervalle I_2^2 de I^2 intersecte l'intervalle gauche de I^1 . Comme il n'intersecte pas J^1 et J^2 , alors il est même **strictement inclus dans l'intervalle gauche** I_1^1 de I^1 .

Ainsi, pour toute réalisation du graphe on a les relations $l(I_1^3) < l(I_2^2), l(I_2^1) < l(I_2^3),$ et $l(I_1^2) < l(I_1^1)$. Supposons par l'absurde que le graphe soit 2-intervallaire équilibré, alors $l(I_1^1) = l(I_2^1)$ et $l(I_1^3) = l(I_2^3)$ donc $l(I_2^3) > l(I_2^1) = l(I_1^1) > l(I_1^2) = l(I_2^2) > l(I_1^3),$ donc $l(I_2^3) \neq l(I_1^3)$: impossible !

Le graphe 2-intervallaire de la figure 4.5(a) n'est donc pas 2-intervallaire équilibré.

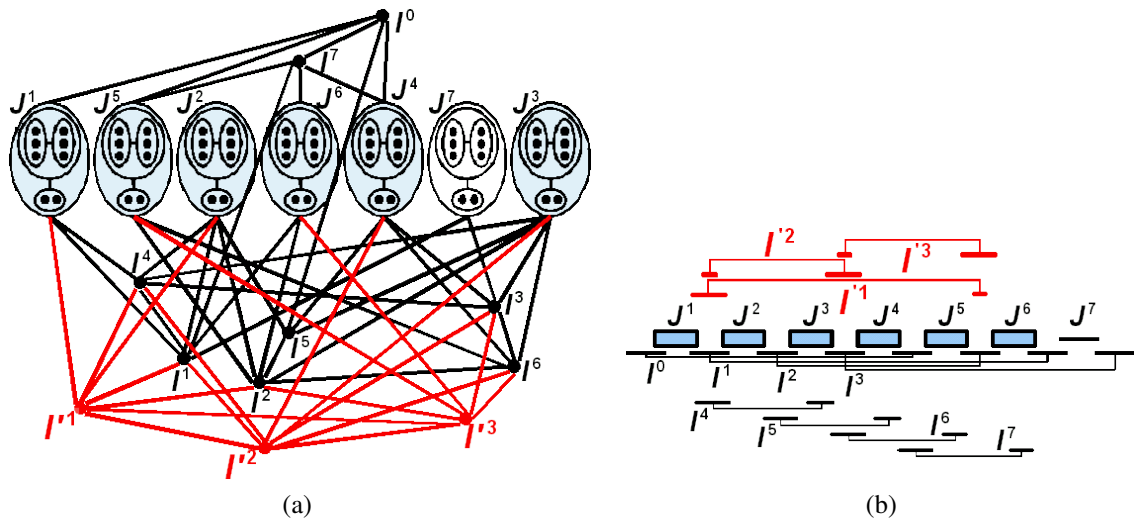


FIG. 4.5 – Un graphe n'admettant pas de réalisation par des 2-intervalles équilibrés (a), et une des ses réalisations 2-intervallaires (b).

4.2 Bilan sur les inclusions de classes 2-intervallaires équilibrées

Théorème 6. On a les inclusions suivantes :

$$\{(1,1)\text{-intervallaires}\} \subsetneq \{2\text{-intervallaires unitaires}\} \subsetneq \{2\text{-intervallaires équilibrés}\} \subsetneq \{2\text{-intervallaires}\}$$

Démonstration. Les inclusions sont évidentes, et on a déjà vu l'inclusion stricte pour la classe des 2-intervallaires équilibrés. Le graphe "étoile" $K_{1,6}$, dont la réalisation en 2-intervalles est visible en figure 4.6(a), est un contreexemple de graphe 2-intervallaire équilibré qui n'est pas 2-intervallaire unitaire, et $K_{1,4}$, à la réalisation en figure 4.6(b), est un contreexemple de graphe 2-intervallaire unitaire qui n'est pas (1,1)-intervallaire. \square

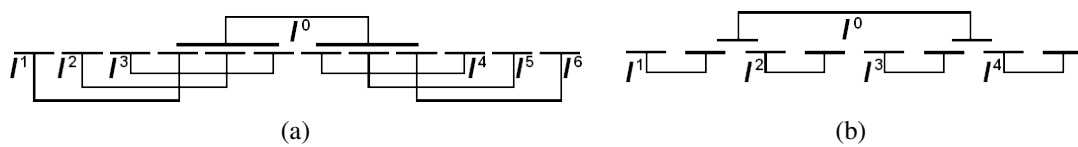


FIG. 4.6 – Une réalisation du graphe 2-intervallaire équilibré mais pas 2-intervallaire unitaire $K_{1,6}$ (a), et une réalisation du graphe 2-intervallaire unitaire mais pas (1,1)-intervallaire $K_{1,4}$ (b).

Chapitre 5

Annexe : théorie des graphes

5.1 Quelques définitions

5.1.1 Des définitions de base

Un graphe G est défini comme un ensemble V de n sommets ou noeuds, et un ensemble $E \subset V^2$ de m arêtes reliant ces sommets. Si deux sommets peuvent être reliés par plusieurs arêtes, on parle de *multigraphes*. Une arête reliant deux sommets v et v' est notée (v, v') . Dans le cas de graphes non orientés, $(v, v') = (v', v)$.

Deux sommets reliés par une arête sont dits *adjacents* ou *voisins*. Le *voisinage* ou voisinage ouvert d'un sommet v , soit l'ensemble de ses sommets adjacents, est noté $\mathcal{N}(v)$. Sa taille est appelée *degré* du sommet v et un sommet de degré nul est dit *isolé*. Le *voisinage clos* est défini comme $\mathcal{N}[v] = \mathcal{N}(v) \cup \{v\}$. Un graphe est *connexe* si pour tous sommets x et y , il existe un chemin reliant x à y , c'est à dire une suite de noeuds $(v_0 = x, v_1, \dots, v_{k+1} = y)$ telle que v_i voisin de v_{i+1} pour tout $i \in \llbracket 0, k \rrbracket$.

Le *complémentaire* d'un graphe non orienté G , noté \overline{G} , est le graphe de sommets V et d'arêtes $V^2 \setminus E$ (si deux sommets v et v' sont adjacents dans G , ils ne le seront pas dans \overline{G} et inversement).

Le sous-graphe de G induit par un sous-ensemble de sommets $S \subseteq V$, noté $G[S]$ est le graphe de sommets S et d'arêtes $E \cap S^2$, c'est à dire le graphe obtenu en gardant de G ses sommets dans S et toutes les arêtes les reliant. Soit G' un graphe. On dit qu'un graphe G est *sans* G' si aucun des sous-graphes de G n'est *isomorphe* (identique à réétiquetage des noeuds près) à G' .

5.1.2 Des classes de graphes

K_n est la *clique*, c'est à dire le graphe *complet*, à n sommets : $\forall v, v' \in V, (v, v') \in E$. Son complémentaire (c'est à dire le graphe sans arêtes à n éléments) est noté S_n et appelé le *stable* à n éléments. On note P_n le *chemin*, ou la *chaîne* à n sommets, c'est à dire le graphe à n sommets v_1, \dots, v_n et $n - 1$ arêtes (v_i, v_{i+1}) , pour $1 \leq i < n$. Si l'on ajoute une n -ième arête (v_n, v_1) , on obtient le *cycle* à n sommets C_n .

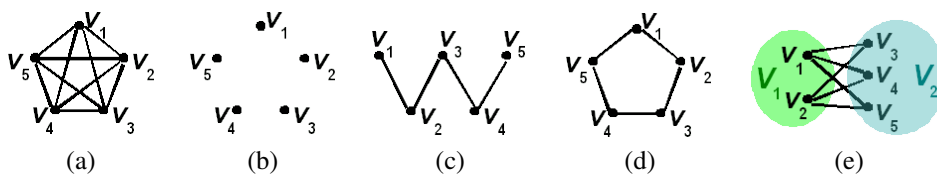


FIG. 5.1 – La clique K_5 (a), le stable S_5 (b), le chemin P_5 (c), le cycle C_5 (d), le graphe bipartite complet $K_{2,3}$ (e).

Un graphe (V, E) dont l'ensemble de sommets V peut être partitionné en $V_1 \cup V_2$ de telle sorte

que V_1 et V_2 sont stables est un graphe *biparti*. Si de plus $E = V_1 \times V_2$, c'est à dire que toutes les arêtes entre V_1 et V_2 sont dans G , alors le graphe est *biparti complet*, et noté $K_{|V_1|,|V_2|}$.

Un *arbre* est un graphe connexe sans cycle. Un graphe *triangulé* est un graphe sans C_k , pour $k \geq 4$. Un *cographe* est un graphe sans P_4 .

Un *graphe de comparabilité* est tel qu'il existe une orientation transitive de ses arêtes. Un *graphe de co-comparabilité* est tel que son complémentaire est un graphe de comparabilité.

Soit $\pi = \begin{pmatrix} 1 & \dots & n \\ \pi(1) & \dots & \pi(n) \end{pmatrix}$ une permutation. Le graphe formé des sommets $v_i, i \in \{1, \dots, n\}$ reliés ssi $(i - j)(\pi^{-1}(i) - \pi^{-1}(j)) < 0$ (i et j sont inversés par la permutation) est un *graphe de permutation*. On peut visualiser ces inversions d'entiers dans la permutation comme montré en figure 5.2 : tracer deux lignes parallèles, et y placer n points étiquetés par $\{1, \dots, n\}$ dans l'ordre croissant, puis relier tout entier i de la première ligne à $\pi(i)$ de la deuxième, les intersections de ces liens correspondent alors aux arêtes du graphe. On remarque que le complémentaire d'un graphe de permutation en est

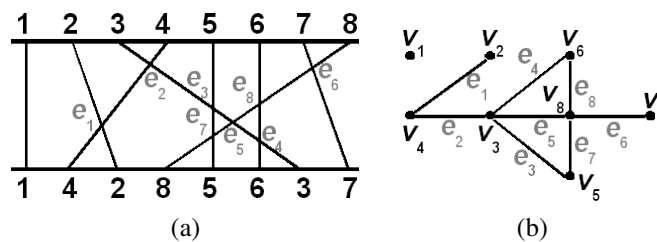


FIG. 5.2 – La représentation de la permutation $\begin{pmatrix} 12345678 \\ 4285637 \end{pmatrix}$ par des liens entre deux lignes parallèles (a), et le graphe de permutation associé (b).

un aussi (de la permutation $\pi' : i \mapsto \pi(n + 1 - i)$; "renverser" la permutation permet en effet d'inverser les présences ou absences d'intersections de liens). D'autre part, le complémentaire d'un graphe de permutation est un graphe de comparabilité (si v_i et v_j ne sont pas reliés, alors les liens correspondants entre les deux lignes parallèles ne s'intersectent pas, on peut donc ordonner v_i et v_j : $v_i \prec v_j$ si $i < j$, et la relation \prec ainsi obtenue est bien transitive). Ces deux remarques montrent que si un graphe est de permutation, alors c'est un graphe de comparabilité et de co-comparabilité. Il y a en fait équivalence, d'après le théorème de Dushnik et Miller [DM41].

Une extension des graphes de permutation, en considérant la construction de ces graphes en tant que graphes d'intersection illustrée en figure 5.2(a), est la classe des *graphes trapézoïdaux*. Pour définir cette classe, considérons deux lignes horizontales. Un *trapèze* entre ces deux lignes est défini comme une paire de segments, un sur celle du dessus et l'autre sur celle du dessous. Un graphe est trapézoïdal s'il existe un ensemble de trapèzes correspondant aux sommets du graphe tel que deux sommets sont joints par une arête si et seulement si les trapèzes correspondants s'intersectent. Les figures 2.16(a) et (b) illustrent cette définition.

Une autre classe de graphes d'intersection est celle des *graphes de cordes*, qui sont les graphes d'intersection des cordes d'un cercle. On peut démontrer l'égalité entre cette classe de graphes et celle des *graphes de chevauchement* d'intervalles (chaque noeud correspond à un intervalle, on place une arête si les deux intervalles se chevauchent), par une construction réalisant l'équivalence entre intersection de deux cordes et chevauchement de deux intervalles, illustrée en figure 5.3. De même que pour les graphes de permutations, on peut définir une surclasse de celle des graphes de cordes, la classe des graphes de trapézoïdes circulaires [Lin06], qui sont bien sûr les graphes d'intersections des trapézoïdes circulaires (constitués par l'union de deux cordes d'un cercle ainsi que les deux arcs de cercle reliant ces cordes, comme illustré en figure 2.15).

Un *triplet astéroïde* est un ensemble indépendant de trois sommets tels que tout couple de sommets parmi ces trois est relié par un chemin qui évite le voisinage du troisième. Par exemple, les graphes de la figure 5.4 contiennent des triplets astéroïdes. Un graphe est dit *sans triplet astéroïde* (ou *AT-free*) [COS97] si aucun ensemble de trois de ses sommets forme un triplet astéroïde.

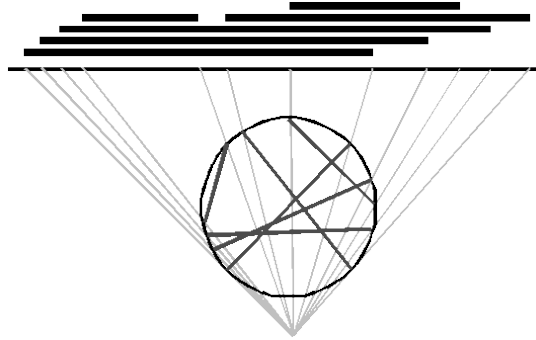


FIG. 5.3 – Illustration de l'équivalence entre les intersections de cordes d'un cercle et les chevauchements d'intervalles.

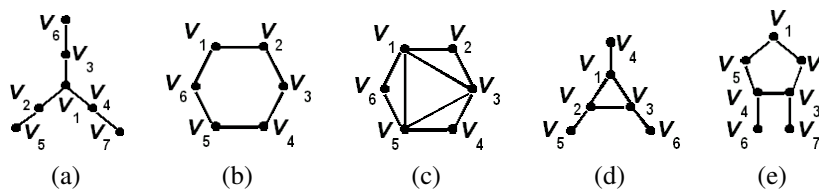


FIG. 5.4 – Quelques petits graphes contenant un triplet astéroïde (respectivement : les feuilles pour (a), $\{v_1, v_3, v_5\}$ pour (b), $\{v_2, v_4, v_6\}$ pour (c), $\{v_4, v_5, v_6\}$ pour (d), $\{v_1, v_6, v_7\}$ pour (e)).

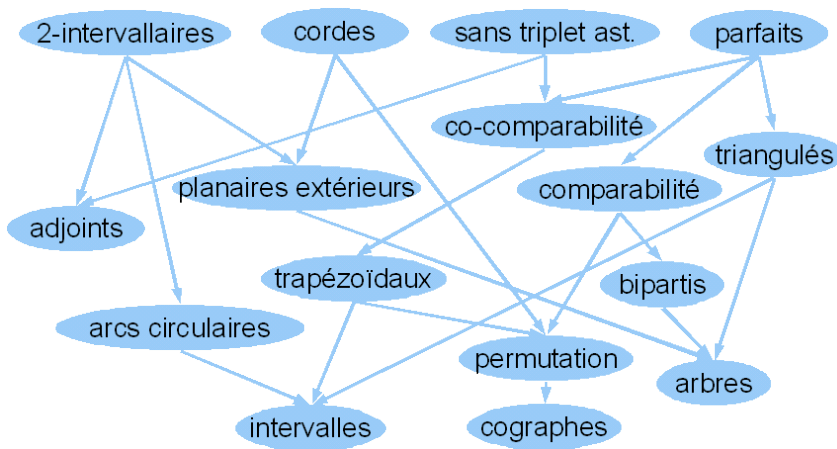


FIG. 5.5 – La hiérarchie de quelques classes de graphes [BLS99] : les arêtes orientées ($C_1 \rightarrow C_2$) de ce graphe indiquent que la classe C_2 est incluse dans la classe C_1 .

5.1.3 Des transformations de graphes

La *2-subdivision* du graphe $G = (V, E)$ consiste à remplacer dans ce graphe chaque arête (v, v') de E par un $P_4 : v - v_1 - v_2 - v'$. Ainsi, le graphe G' résultant de la 2-subdivision de G a $|V| + 2|E|$ sommets et $3|E|$ arêtes.

L'*adjoint* du graphe $G = (V, E)$ est $G' = (E, E')$, où $(e, e') \in E' \Leftrightarrow e$ et e' , sommets de G' mais aussi arêtes de G , ont une extrémité commune dans G en anglais) de G a les arêtes de G comme sommets et les relie si les arêtes correspondantes dans G avaient une extrémité commune.

5.2 Quelques problèmes sur les graphes

5.2.1 Couverture de sommets

Une *couverture de sommets* est un sous-ensemble V' de sommets de V tel que toute arête de E relie un sommet de cet ensemble V' à un autre sommet de V , comme montré en figure 5.6. Plus

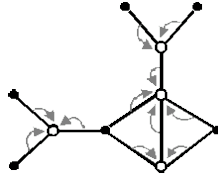


FIG. 5.6 – L'ensemble des sommets blancs constitue une couverture de sommets (minimale) du graphe.

formellement, $V' \subset V$ est une couverture de sommets de G si $\forall e = (v, v') \in E, v \in V'$ ou $v' \in V'$.

Le problème `MINIMUMVERTEXCOVER` consiste à trouver une couverture de sommets de taille minimale (cette taille est alors appelée *nombre transversal* du graphe G , notée $\tau(G)$).

`MINIMUMVERTEXCOVER` est NP-complet [GJ79] par réduction de 3-SAT, en construisant pour toute instance de 3-SAT avec n variables et c clauses¹ un graphe bien choisi avec $2n + 3c$ noeuds (un noeud étiqueté par chaque variable et sa négation, ainsi qu'un noeud étiqueté pour chacun des trois littéraux d'une clause) et $n + 6c$ arêtes (les trois littéraux de chaque clause reliés entre eux, ainsi qu'à celui des $2n$ noeuds dont l'étiquette lui correspond).

5.2.2 Ensemble indépendant

Un *ensemble indépendant*, ou *stable*, est un sous-ensemble de sommets de V non adjacents. Le problème `MAXIMUMWEIGHTEDINDEPENDENTSET`, abrégé en *MWIS* ou *WIS*, consiste, étant donné des scores à chaque noeud de V , à trouver l'ensemble indépendant dont la somme des scores des noeuds est maximale. Soit plus formellement, en affectant à chaque noeud v le score $s(v) \in \mathbb{N}$, et en se fixant k dans \mathbb{N} , savoir s'il existe $V' \subset V$ tel que $\bigcup_{v \in V'} s(v) \geq k$ et $\forall v, v' \in V', (v, v') \notin E$.

Ce problème est NP-complet. En effet, en posant $s(v) = 1$ pour tout noeud v , on tombe sur le problème `MAXIMUMINDEPENDENTSET` (*MIS*) qui consiste à trouver le plus grand ensemble indépendant d'un graphe, et qui est NP-complet, par réduction de `MINIMUMVERTEXCOVER`. En effet, V' est une couverture de sommets si et seulement si $V \setminus V'$ est un ensemble indépendant, comme montré en figure 5.7. Ainsi, le complémentaire d'une couverture de sommets de taille minimale est un ensemble indépendant de taille maximale.

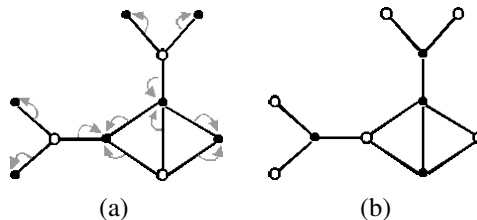


FIG. 5.7 – L'ensemble des 3 sommets blancs de (a) est un ensemble indépendant, et les sommets noirs constituent une couverture de sommets. L'ensemble des 5 sommets blancs de (b) est l'ensemble indépendant maximum, ou stable maximum, du graphe et les sommets blancs forment la couverture de sommets minimum déjà rencontrée en figure 5.6.

¹ clause : disjonction de *littéraux*, c 'est à dire de variables ou de négations de variables.

5.2.3 Clique

Trouver la clique maximale d'un graphe G consiste à trouver le sous-ensemble de sommets de taille maximale tel que le sous-graphe induit par lui est une clique. Cette taille maximale est appelée *nombre de clique* de G et noté $\omega(G)$.

Ce problème, noté MAXCLIQUE est NP-complet par réduction de MAXIMUMINDEPENDENTSET. En effet, V est un une clique maximale de G si et seulement si c'est un stable maximum de \overline{G} , le complémentaire de G .

5.2.4 Coloration

Le *nombre chromatique* d'un graphe G , noté $\chi(G)$, est le nombre minimum de couleurs nécessaires pour colorier ses sommets de telle sorte que deux sommets adjacents n'aient pas la même couleur. On peut remarquer que dans une clique, tous les sommets doivent avoir une couleur différente, donc $\omega(G) \leq \chi(G)$.

Le problème du K -COLORIAGE consiste à savoir si un graphe G donné en entrée est K -coloriable, c'est à dire tel que $\chi(G) \leq K$. Ce problème est NP-complet pour G quelconque et $K \geq 3$.

Un graphe est dit *parfait* si son nombre chromatique est égal à son nombre de clique. Cette classe de graphes est importante [Gol80], puisqu'elle contient de nombreuses autres classes utiles, comme on peut le voir sur la figure 5.5.

L'*index chromatique* d'un graphe G , noté $\chi'(G)$, est le nombre de couleurs nécessaires pour colorier ses arêtes de telle sorte que deux arêtes adjacentes n'aient pas la même couleur. C'est donc le nombre chromatique du graphe adjoint de G . Le déterminer pour un graphe quelconque est NP-complet [Hol81].

5.2.5 Domination

On dit qu'un sommet *domine* son voisinage clos. L'ensemble des sommets dominés par un sous-ensemble V' de V est donc $\bigcup_{v \in V'} \mathcal{N}[v]$. Si cet ensemble dominé est égal à V , on dit que V' est un *ensemble dominant*.

Trouver l'ensemble dominant d'un graphe arbitraire de taille minimale est NP-complet, par réduction de VERTEX COVER [GJ79] : une couverture de sommets sur un graphe correspond à un ensemble dominant sur le graphe où chaque arête (v, v') a été remplacée par un $P_3 : v - u - v'$.

5.2.6 Couplage maximum

Un *couplage* d'un graphe G est un ensemble d'arêtes non adjacentes de G . Quand cet ensemble est maximal, c'est à dire qu'on ne peut y ajouter d'arête sans perdre la propriété de non-adjacence, on parle de *couplage maximal*. Si l'ensemble est de taille maximale parmi tous les couplages de G , on le qualifie de *couplage maximum*. La taille de cet ensemble est alors appelée *nombre de couplage* du graphe, et notée $\nu(G)$. Enfin, un couplage est dit *parfait* si tout sommet du graphe est relié à une arête du couplage.

Si l'on connaît un couplage maximum d'un graphe G , alors la couverture de sommets de G doit contenir au moins un sommet de chaque arête de ce couplage, donc elle a une taille supérieure à ou égale à $\nu(G)$. On a donc pour tout graphe $\nu(G) \leq \tau(G)$. Il y a égalité pour les graphes bipartis.

Déterminer le couplage maximum d'un graphe se fait en temps polynomial, à l'aide de recherches de chemins augmentants (c'est à dire de chemins qui commencent et finissent par une arête n'appartenant pas au couplage, ces deux arêtes n'ayant pas de sommet commun, et font alterner successivement une arête du couplage, et une arête n'y appartenant pas). Pour des graphes bipartis $(V_1 \uplus V_2, E)$, en ajoutant une source reliée à tous les sommets de V_1 et un puits relié à tous ceux de V_2 , les chemins augmentants peuvent être vus comme des chemins augmentant un flot dans ce nouveau graphe [CLR90]

et donc le problème peut être résolu par l'algorithme de Ford Fulkerson [FF62]. Pour les graphes non-bipartis, Edmonds propose un algorithme qui consiste à gérer efficacement les cycles [Edm65], et Micali et Vazirani ont un algorithme plus compliqué en $O(m\sqrt{n})$ [MV80]. Ces approches sont regroupées dans une synthèse de Zviv Galil [Gal86].

5.2.7 Coupe maximale

Une *coupe* est une partition des sommets d'un graphe en 2 ensembles. Trouver la *coupe maximum*, ou *coupe de poids maximum* (problème MAXCUT), consiste à trouver la partition $V = S \uplus T$ telle que la taille de l'ensemble d'arêtes joignant les sommets de S à ceux de T , soit $\{(v_1, v_2)/v_1 \in S \text{ et } v_2 \in T\}$, est maximale. Ce problème est NP-complet [GJ79].

5.2.8 Cycle hamiltonien

Un cycle hamiltonien est un cycle qui contient tous les sommets de V . Vérifier si un graphe quelconque contient un cycle hamiltonien est NP-complet [GJ79].

5.2.9 Cycle eulérien

Un cycle eulérien est un cycle qui contient toutes les arêtes de E . Un graphe non orienté connexe possède un cycle eulérien si et seulement si tous ses sommets sont pairs. Un graphe non orienté connexe possède un parcours eulérien si et seulement si tous ses sommets sont de degré pair à l'exception d'au plus 2 d'entre eux.

On peut construire un cycle eulérien en $O(m)$ par l'algorithme récursif suivant : commencer par construire un cycle partant d'un sommet arbitraire, supprimer les arêtes du cycle, puis pour les sommets restants ayant un degré non nul, construire un cycle.

5.2.10 Dénombrement

Il y a $2^{\frac{n(n-1)}{2}}$ graphes ayant n sommets étiquetés (pour chacune des $\frac{n(n-1)}{2}$ arêtes non orientées, on choisit si elle appartient au graphe ou pas). Trouver le nombre $g(n)$ de graphes à n sommets non étiquetés, c'est à dire le nombre de graphes non isomorphes à n sommets, est un peu plus compliqué. On est amené à faire appel au théorème d'énumération de Pólya [Pó137], ce qui est détaillé dans [Har69], ou dans la Wikipedia², et qui permet de trouver la formule asymptotique pour n au voisinage de $+\infty$:

$$g(n) \sim \frac{2^{\frac{n(n-1)}{2}}}{n!}.$$

Les premiers termes sont les suivants³ : 1, 1, 2, 4, 11, 34, 156, 1 044, 12 346, 274 668, 12 005 168, 1 018 997 864...

² <http://en.wikipedia.org/wiki/Pólyaenumerationtheorem>

³ <http://www.research.att.com/~njas/sequences/A000088> [Slo]

Chapitre 6

Annexe : dénombrement des séquences arc-annotées

Dans ce chapitre, on présente diverses méthodes de dénombrement des séquences arc-annotées, en fonction du nombre d'arcs ou du nombre d'extrémités.

6.1 Majoration du nombre de séquences arc-annotées

Cherchons un majorant de $AA(n)$, le nombre de séquences arc-annotées à n arcs non étiquetés. Commençons par dénombrer celles à arcs disjoints, étiquetés, et où les extrémités des arcs sont étiquetés (ainsi l'arc $(1, 2)$ est différent de l'arc $(2, 1)$) : cela correspond au nombre de façons de placer les $2n$ sommets des arcs, soit $(2n)!$. Pour obtenir les séquences aux extrémités non étiquetées, gardons uniquement celles où les arcs sont de la forme $(i, i + 1)$ (et pas $(i + 1, i)$) : il y en a $\frac{(2n)!}{2^n}$. Pour obtenir les séquences aux arcs non étiquetés, on divise par le nombre d'étiquetages possibles pour les arcs, ce qui donne $\frac{(2n)!}{2^n n!}$. Pour passer outre la restriction des arcs disjoints, on doit autoriser les arcs à avoir une extrémité commune, c'est à dire que l'on "recolle" certains ensembles d'extrémités consécutives (voir figure 6.1(a)). On va donc majorer $AA(n)$ par le nombre de tous les "recolllements" possibles de toutes les $\frac{(2n)!}{2^n n!}$ séquences arc-annotées non étiquetées à arcs disjoints (la majoration est stricte, puisque les recolllements de la figure 6.1(b) et (c) sont interdits) :

$$AA(n) < \frac{(2n)!2^{2n-1}}{2^n n!} = \frac{(2n)!2^{n-1}}{n!}.$$

6.2 Génération

On peut calculer les premières valeurs de $AA(n)$ en générant sur ordinateur toutes les séquences arc-annotées à n arcs, considérées dans cette section, pour plus de clarté, non plus comme des ensembles de couples d'intervalles unitaires mais comme des ensembles de couples d'entiers.

Pour cela, on remarque qu'une séquence arc-annotée \mathcal{I} à $n + 1$ arcs peut-être générée de façon non ambiguë à partir d'une séquence arc-annotée \mathcal{I}' à n arcs, en ajoutant un arc supérieur, pour l'ordre lexicographique, à tous ceux de \mathcal{I} . Ainsi, on peut utiliser l'algorithme 1 de la page 45 illustré en figure 6.2.

On trouve les premières valeurs suivantes :

$$AA(1) = 1, AA(2) = 6, AA(3) = 62, AA(4) = 900, AA(5) = 16824, AA(6) = 384668.$$

Comme l'algorithme 1 génère toutes les séquences à n arcs, il est en $O(AA(n))$, et calculer des valeurs plus grandes de $AA(n)$ avec cet algorithme prend plusieurs minutes sur un PC de bureau.

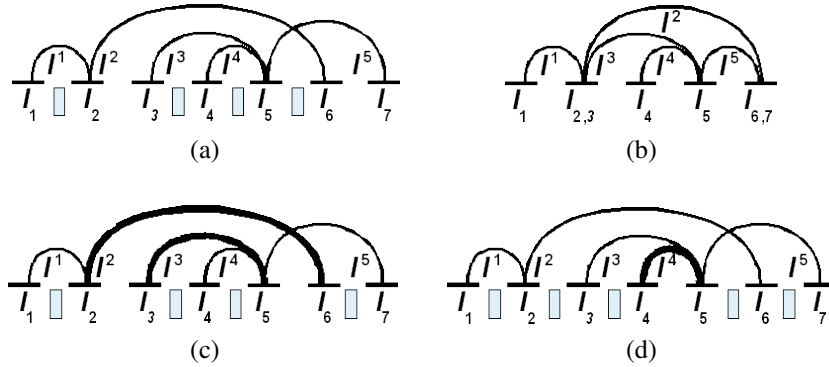


FIG. 6.1 – Pour effectuer un recollement, on va placer des boîtes bleues de séparation entre les extrémités d’arcs (a) : toutes les extrémités placées entre deux boîtes bleues consécutives (I_2 et I_3 d’une part, I_6 et I_7 d’autre part) seront "recolées" (b). On ne peut effectuer les recollements des extrémités I_2 et I_3 ainsi que I_5 et I_6 (c), puisque les arcs en gras I^2 et I^3 sont alors confondus, ce qui revient à supprimer un arc. On ne peut pas non plus recoller les extrémités I_4 et I_5 (d), puisque cela supprime l’arc I^4 .

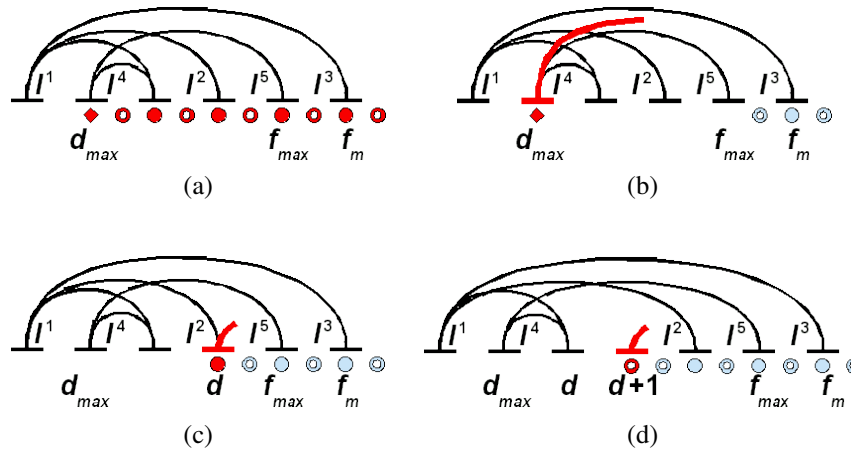


FIG. 6.2 – Illustration de l’algorithme 1 de génération récursive des séquences arc-annotées par rapport au nombre d’arcs. On souhaite ajouter un arc lexicographiquement plus grand que tous les arcs présents d’une séquence (a). L’extrémité gauche de ce nouvel arc peut donc se placer sur la dernière extrémité gauche déjà placée d_{max} , c’est à dire au niveau du losange, ou après d_{max} , c’est à dire au niveau des disques et couronnes. Trois cas se présentent alors. Si on place l’extrémité gauche du nouvel arc en d_{max} (b), on doit placer son extrémité droite après f_{max} , sur une extrémité existante (disques) ou en créant une nouvelle extrémité (couronnes). Si on place l’extrémité gauche du nouvel arc après d_{max} sur une extrémité existante d (c), on doit placer son extrémité droite après d , au niveau des disques ou des couronnes. Si on place l’extrémité gauche du nouvel arc après d_{max} sur une nouvelle extrémité $d + 1$ (d), on doit décaler les extrémités après d puis placer son extrémité droite après $d + 1$, au niveau des disques ou des couronnes.

Toutefois, ces 6 valeurs ont suffi à vérifier, au moment de la réalisation de ce travail, que cette suite n’était pas référencée dans l’encyclopédie en ligne des entiers [Slo].

6.3 Dénombrement à nombre fixé d’arcs

Après avoir présenté un algorithme pour générer les séquences à n arcs, récursif du point de vue du nombre d’arcs, on présente une formule utilisant une récurrence par rapport au nombre d’extrémités afin de calculer $AA(n, e)$, le nombre de séquences arc-annotées à n arcs et e extrémités.

Algorithme 1 (Génération des séquences annotées de taille n , illustré en figure 6.2).

GENERATION(entier n) \rightarrow ensemble d'ensembles de n couples d'entiers

$\mathcal{I} = \emptyset$;

si $n = 2$:

– alors $\mathcal{I}' = \{\{(0, 1)\}\}$;

– sinon $\mathcal{I}' = \text{GENERATION}(n - 1)$;

Pour tout $S \in \mathcal{I}'$ (S est un ensemble de $n - 1$ couples d'entiers, c'est à dire une séquence arc-annotée à $n - 1$ arcs) faire :

– Ajout d'un arc à S :

$d_{max} = 0$; $f_{max} = 0$; $f_m = 0$;

Pour tout $(d, f) \in S$:

– si $d > d_{max}$ alors $d_{max} = d$; $f_{max} = f$;

– si $f_m > f$ alors $f_m = f$;

– Cas 1, ajout d'un arc sur la dernière extrémité gauche d_{max} :

– Cas 1.1, extrémité droite sur une extrémité droite :

Pour tout f de $f_{max} + 1$ à f_m :

– $\mathcal{I}' = \mathcal{I}' \cup \{S \cup [d_{max}, f]\}$;

– Cas 1.2, extrémité droite juste après une extrémité droite :

Pour tout f de f_{max} à f_m :

– $\mathcal{I}' = \mathcal{I}' \cup \{\text{DECALE}(f, S) \cup [d_{max}, f + 1]\}$;

– Cas 2, ajout d'un arc après la dernière extrémité gauche d_{max} :

– Cas 2.1, extrémité gauche sur une extrémité droite :

Pour tout d de $d_{max} + 1$ à f_m :

– Cas 2.1.1, extrémité droite sur une extrémité droite :

Pour tout f de $d + 1$ à f_m :

– $\mathcal{I}' = \mathcal{I}' \cup \{S \cup [d, f]\}$;

– Cas 2.1.2, extrémité droite juste après une extrémité droite :

Pour tout f de d à f_m :

– $\mathcal{I}' = \mathcal{I}' \cup \{\text{DECALE}(f, S) \cup [d, f + 1]\}$;

– Cas 2.2, extrémité gauche juste après une extrémité droite :

Pour tout d de d_{max} à f_m :

– Cas 2.2.1, extrémité droite sur une extrémité droite :

Pour tout f de $d + 1$ à f_m :

– $\mathcal{I}' = \mathcal{I}' \cup \{\text{DECALE}(d, S) \cup [d + 1, f + 1]\}$;

– Cas 2.2.2, extrémité droite juste après une extrémité droite :

Pour tout f de d à f_m :

– $\mathcal{I}' = \mathcal{I}' \cup \{\text{DECALE}(d, \text{DECALE}(f, S)) \cup [d + 1, f + 2]\}$;

DECALE(entier f_{min} , ensemble de couples d'entiers \mathcal{I}) \rightarrow ensemble de couples d'entiers

Pour tout $(d, f) \in \mathcal{I}$:

– si $f > f_{min}$ alors $\mathcal{I} = \mathcal{I} \setminus \{(d, f)\} \cup \{(d, f + 1)\}$;

retourner \mathcal{I} ;

Le calcul qui suit se fonde sur une autre construction récursive non ambiguë des séquences arc-annotées suggérée par Dominique Rossin : une séquence arc-annotée à e extrémités est une séquence arc-annotée à $e - 1$ extrémités à laquelle on a ajouté un certain nombre d'arcs partant d'une extrémité supplémentaire qu'on ajoute à la fin, illustrée en figure 6.3(a) ou c'est la séquence constituée uniquement d'arcs ayant la dernière extrémité comme extrémité droite, comme en figure 6.3(b). Ce dernier cas est donc une séquence à n arcs et $e = n + 1$ extrémités.



FIG. 6.3 – Génération récursive, en ajoutant une extrémité, des séquences annotées. Soit un certain nombre x d'arcs partent de l'extrémité ajoutée (I^1 , I^2 et I^3 en (a)), et les autres arcs proviennent de la précédente étape de récurrence. Soit tous les arcs partent de l'extrémité ajoutée (b).

En notant $AA(n, e)$ le nombre de séquences arc-annotées à n arcs et e extrémités, on peut décomposer $AA(n)$ en fonction de e :

$$AA(n) = \sum_{e=1}^{2n} AA(n, e). \quad (6.1)$$

On utilise alors la construction récursive en appelant x le nombre d'arcs qui partent de l'extrémité la plus à droite de la séquence (I^1 , I^2 et I^3 dans la figure 6.3a) donc dans cet exemple $x = 3$), et en distinguant parmi ceux-ci les y arcs dont l'extrémité gauche est une extrémité gauche partagée avec un autre arc (I^1 et I^3 , donc $y = 2$ dans cet exemple), et les $x - y$ dont l'extrémité gauche n'est pas partagée, pour $n \geq 2$ et pour $1 \leq e \leq 2n$:

$$AA(n, e) = \mathbf{1}_{e=n+1} + \sum_{x=1}^{e-1} \sum_{y=0}^x AA(n-x, e-1-y) \binom{e-1}{y} \binom{e-1-y}{x-y}. \quad (6.2)$$

Ainsi, en remarquant que $AA(n, e) = 0$ si $e < 1$ ou $e > 2n$ et que $AA(1, 2) = 1$, on peut calculer récursivement tout $AA(n)$. En Java, le plus grand entier `longint` est rapidement atteint ($AA(15) = 2^{64}$) et le temps de calcul avec ces grands entiers est assez long. En revanche, grâce à une transcription du code en Mathematica, en figure 6.4, on arrive à calculer en quelques secondes les 20 premières valeurs de $AA(n)$, montrées en figure 6.5.

```

a[n_, e_] :=
  a[n, e] = Which[n <= 0, 0, And[n == 1, e != 2], 0, And[n == 1, e == 2], 1,
    And[n >= 2, e < 1], 0, And[n >= 2, e > 2n], 0,
    And[n >= 2, 1 <= e <= 2n], KroneckerDelta[e, n+1]
    + Sum[Sum[a[n-x, e-1-y] Binomial[e-1, y]
      Binomial[e-1-y, x-y], {y, 0, x}], {x, 1, e-1}]
aa[n_] := Sum[a[n, e], {e, 1, 2n}]

```

FIG. 6.4 – Le code Mathematica de calcul de $AA(n)$ (transcription par Ahmed Youssef).

Depuis la réalisation de ce travail en juillet 2006, la séquence a été proposée à l'encyclopédie en ligne des entiers [Slo]¹ par Vladeta Jovović, comme le nombre de graphes ayant un nombre quelconque de sommets étiquetés non isolés et n arêtes².

¹ <http://www.research.att.com/~njas/sequences/A121251>

² En effet, la bijection entre de tels graphes et les séquences arc-annotées est immédiate en considérant les extrémités comme des sommets et les arcs comme des arêtes du graphe.

n	$AA(n)$
1	1
2	6
3	62
4	900
5	16824
6	384668
7	10398480
8	324420840
9	11472953760
10	453518054216
11	19815916826160
12	948348447031440
13	49334804947402800
14	2771902062752597520
15	167281797371598801136
16	10791777047497882651296
17	741135302021991803931360
18	53983717302568691555767360
19	4156878464543273462779074240
20	337395644452189799264161979424

FIG. 6.5 – Les vingt premiers termes de $AA(n)$.

Il propose de calculer $AA(n)$ comme une somme de série convergente, à termes positifs :

$$AA(n) = \sum_{v \geq 0} \frac{\binom{\frac{v(v-1)}{2}}{n}}{2^{v+1}}. \quad (6.3)$$

Il avait auparavant proposé une autre formule pour $AA(n, e)$ ³, pour e fixé et n variant de 0 à $\frac{e(e-1)}{2}$:

$$AA(n, e) = \sum_{i=0}^n (-1)^{(n-i)} \binom{n}{i} \binom{\frac{i(i-1)}{2}}{e}. \quad (6.4)$$

Détaillons les calculs pour aboutir à ces deux formules, qui s'inspirent de ceux de l'article [CPS06], consacré à l'énumération de certaines matrices d'adjacence.

Tout d'abord, générons tous les graphes à v sommets étiquetés et n arêtes (parmi les $\frac{v(v-1)}{2}$ arêtes possibles) : il y en a $\binom{\frac{v(v-1)}{2}}{n}$.

On peut aussi les générer en les distinguant selon leur nombre e de sommets non isolés : on a alors $\binom{v}{e}$ possibilités pour choisir de tels sommets, et $AA(n, e)$ possibilités pour choisir le graphe à e sommets étiquetés et n arêtes. Ainsi,

$$\binom{\frac{v(v-1)}{2}}{n} = \sum_{e=0}^v \binom{v}{e} AA(n, e). \quad (6.5)$$

Par une inversion de Möbius [BG75], on arrive à l'équation 6.4. Pour obtenir l'équation 6.3, on

³ <http://www.research.att.com/~njas/sequences/A054548>

injecte dans l'équation 6.1 la formule $1 = \sum_{v=e}^{+\infty} \frac{1}{2^{v+1}} \binom{v}{e}$ et donc :

$$\begin{aligned}
 AA(n) &= \sum_{e=1}^{2n} AA(n, e) \sum_{v=e}^{+\infty} \frac{1}{2^{v+1}} \binom{v}{e} \\
 &= \sum_{v=0}^{+\infty} \frac{1}{2^{v+1}} \sum_{e=0}^v AA(n, e) \binom{v}{e} \quad \left(\begin{array}{l} \text{en réindiquant et en utilisant le fait que} \\ AA(n, e) = 0 \text{ si } e < 1 \text{ ou } e > 2n \end{array} \right) \\
 &= \sum_{v=0}^{+\infty} \frac{1}{2^{v+1}} \binom{\frac{v(v-1)}{2}}{n} \quad \text{(en utilisant l'équation 6.5).}
 \end{aligned}$$

6.4 Dénombrement à nombre fixé d'extrémités

On appelle $AA(\bullet, e)$, le nombre de séquences arc-annotées à e extrémités et un nombre quelconque d'arcs non étiquetés. Ce nombre est aussi celui de graphes à e sommets étiquetés et non isolés. Or pour les générer, on peut procéder de la manière suivante :

- générer tous les $2^{\frac{e(e-1)}{2}}$ graphes à e sommets.
- retirer tous les graphes ayant k sommets isolés : pour chacun, le sous-graphe induit par les $e - k$ sommets non isolés est un graphe à $e - k$ sommets étiquetés et non isolés. Il y a donc $\binom{e}{k} AA(\bullet, e - k)$ graphes à enlever pour k entre 1 et e .

Ainsi :

$$AA(\bullet, e) = 2^{\frac{e(e-1)}{2}} - \sum_{k=1}^e \binom{e}{k} AA(\bullet, e - k)$$

Cette séquence est présente dans l'encyclopédie en ligne des entiers [Slo]⁴ :

1, 4, 41, 768, 27449, 1887284, 252522481, 66376424160, 34509011894545 ...

⁴ <http://www.research.att.com/~njas/sequences/A006129>

Chapitre 7

Conclusion

Ce stage m'a permis de me plonger dans le monde des 2-intervalles et des séquences arc-annotées, afin de trouver certaines propriétés de chacun, notamment diverses décompositions des séquences arc-annotées pour déterminer plusieurs méthodes d'énumération, et quelques petites propriétés sur les réalisations de graphes 2-intervallaires utiles pour prouver l'existence d'un contre-exemple de graphe 2-intervallaire ne pouvant être équilibré.

D'autre part, la synthèse des connaissances sur diverses classes de graphes apparentées aux graphes 2-intervallaires a permis d'améliorer la complexité de la résolution d'une variante du problème 2-IP sur des ensembles de 2-intervalles à support disjoint (ou de façon équivalente, les séquences arc-annotées).

Toutefois, l'utilisation des approches synthétisées afin de lever le dernier point d'interrogation sur le problème 2-IP n'a pas abouti, tant pour les approches par la théorie des graphes, qui ont buté sur les difficultés à caractériser la classe des graphes $\{<, \square\}$ -(1,1)-intervallaires, que celles par la programmation dynamique.

D'autre part, les diverses réductions tentées en particulier depuis le problème 3-SAT afin de prouver la NP-complétude du problème ont aussi échoué. C'est pourtant cette voie qui semble privilégiée avec le résultat récent de NP-complétude d'un problème similaire sur les mêmes objets [LL06].

D'autre part, je reste en contact avec Yun-Kang Ahn de l'IRCAM en ce qui concerne les expérimentations en analyse musicale afin de déterminer l'utilité éventuelle des diverses relations entre 2-intervalles, et donc de l'utilisation des algorithmes pour le problème 2-IP, dans ce domaine.

7.1 Remerciements

Je tiens à remercier Michel Habib, qui en plus de m'encadrer et de me guider, notamment par des suggestions de lectures très profitables, m'a fait profiter de sa bonne humeur et de son enthousiasme communicatifs pour rendre ce stage très agréable.

Stéphane Vialette était toujours disponible pour répondre à mes questions et me transmettre ses connaissances expertes sur les 2-intervalles. Je remercie aussi chaleureusement Derek Corneil, Dominique Rossin, et Fabien de Montgolfier, avec qui les discussions m'ont fait économiser des heures de travail, et ont abouti à des pistes de réflexion intéressantes. Merci aussi à Mathilde Bouvel, toujours disponible pour partager son savoir en combinatoire et dénombrement, Guylain Naves le sien en théorie des graphes, Yun-Kang Ahn sa passion pour la musique et l'analyse musicale et Ahmed Youssef sa maîtrise de Mathematica.

Enfin, je remercie les membres du LIAFA pour leur accueil, et la gestion de toutes les questions pratiques, de secrétariat ou de matériel. Merci en particulier aux si(x) sympathiques camarades de bureau qui ont contribué à l'excellent souvenir qui me restera de ce stage.

Index

- $<$, précédence de 2-intervalles, 9
- $<_{[]}$, précédence d'intervalles, 9
- \int , entrelacement de 2-intervalles, 10
- \cap , intersection non vide, 10
- $\nu(G)$, nombre de couplage, 41
- $\omega(G)$, nombre de clique, 41
- \sqsubset , emboîtement de 2-intervalles, 10
- $\tau(G)$, nombre transversal, 40
- (1,1)-intervallaire, 11
- 2-intervalle, 9
- 2-subdivision, 18, 39
- 3-SAT, 26, 29, 40

- A006129, 48
- A054548, 47
- A121251, 46
- $AA(\bullet, e)$, 48
- $AA(n)$, 14, 43
- $AA(n, e)$, 44
- adjacent, 37
- adjoint, voir graphe adjoint
- analyse transformationnelle, 13
- APX-dur, 27
- arbre, 38
- arc, 9
- arcs circulaires, 17
- arête, 37
- ARN, 12, 14
- AT-free, voir sans triplet astéroïde

- bicoloriage, 29
- biparti, 38, 41
- biparti complet, 38
- bitâche, 11

- chaine, voir chemin
- chemin, 37
- chenille, 17
- chevauchement, 9
- clause, 26, 40
- clique, 37, 41
- clique max, 18

- cographe, 18, 38
- coloration, 11, 18
- complémentaire, 14, 37
- complet, 37
- connexe, 37
- coupe, 42
- coupe maximum, 18, 42
- couplage, 41
- couplage maximal, 41
- couplage maximum, 27, 41
- couplage parfait, 41
- couverture de sommets, 40
- couvrant, voir intervalle couvrant
- cycle, 37

- degré, 10, 17, 37
- dénombrement, 14, 43
- disjoint, 25
- disjoints, 9
- domination, 18, 41

- E , arêtes, 37
- éclatement($\cap \mapsto \int$), 16
- éclatement($\cap \mapsto \sqsubset, <$), 16
- emboîtement, 14
- ensemble de précédence, 28
- ensemble dominant, 41
- ensemble indépendant, 38, 40
- ensemble stable, voir stable
- énumération, voir dénombrement
- équilibré, 31
- eulérien, 42
- extrémité, 9

- \overline{G} , le complémentaire de G , 37
- $G[S]$, voir sous-graphe
- $g(n)$, 16, 42
- graphe 2-intervallaire, 10, 17
- graphe adjoint, 11, 18, 27, 39, 41
- graphe d'arcs circulaires, 17
- graphe d'intersection, 10, 11, 17, 38
- graphe d'intervalles, 10

- graphe de 2-unions, 18
 graphe de chevauchement, 21, 38
 graphe de co-comparabilité, 38
 graphe de comparabilité, 15, 38
 graphe de cordes, 15, 21, 26, 30, 38
 graphe de croisement de trapézoïdes circulaires, 15, 21
 graphe de de trapézoïdes circulaires, 38
 graphe de permutation, 15, 19, 22, 38
 graphe de t -pistes, 11
 graphe de t -unions, 11
 graphe de trapézoïdes circulaires, 15, 21
 graphe parfait, 27, 41
 graphe planaire, 17
 graphe planaire extérieur, 17
 graphe t -intervallaire, 10
 graphe trapézoïdal, 15, 21, 26, 30, 38
- hamiltonien, 42
 hélice, 14
- index chromatique, 18, 41
 induit, voir sous-graphe
 intervalle couvrant, 9, 15, 29–31
 intervalles support, 9
 inversion, 12
 isolé, 37, 46
 isomorphe, 33, 37
- K -COLORIAGE, 41
 $K_{i,j}$, 38
 K_n , la clique, 37
- littéral, 40
 littéral, 26
- m , nombre d'arêtes, 37
 MAXCLIQUE, 41
 MAXCUT, 42
 MAXIMUMINDEPENDENTSET, voir stable maximum
 MAXIMUMWEIGHTEDINDEPENDENTSET, 40
 MINIMUMVERTEXCOVER, 40
 MIS, voir stable maximum
 multigraphe, 37
 MWIS, voir stable maximum
- \mathcal{N} , voisinage, 30, 37
 n , nombre de sommets, 37
 noeud, 37
 nombre chromatique, 18, 41
 nombre de clique, 41
- nombre de couplage, 41
 nombre intervallaire, 10, 18
 nombre transversal, 40
 NP-complet, 26, 40–42
- ordonnancement, 11
- permutation, 23, 38
 planaire extérieur, 17
 P_n , le chemin, 37
 précédence, 9, 14
 pseudo-noeud, 12
- réalisation, 10, 31
- sans, 37
 sans triplet astéroïde, 38
 séquence arc-annotée, 7, 9, 43
 S_n , le stable, 37
 sommet, 37
 sous-graphe, 19, 37, 41
 stable, 11, 37, 40
 stable maximum, 11–13, 26, 27
 structure primaire, 13
 structure secondaire, 12–14
 support, 9, 16, 25, 31
 support disjoint, 9, 11
 support unitaire, 9
- t -intervalle, 9
 T_2 , voir triplet astéroïde
 traduction, 12
 transcription, 12
 transposition, 12
 trapézoïdes circulaires, 21, 30
 trapèze, 22, 38
 trapézoïdal, voir graphe trapézoïdal
 triangulé, 38
 triplet astéroïde, 27, 38
- unitaire, 25
- V , sommets, 37
 voisin, 37
 voisinage, 37
 voisinage clos, 37
- WIS, voir stable maximum

Bibliographie

- [AAH92] Alberto Apostolico, Mikhail J. Atallah, and Susanne E. Hambruch. New clique and independent set algorithms for circle graphs. *Discrete Applied Mathematics*, 36(2):1–24, 1992. 30
- [Ber94] Claude Berge. Qui a tué le duc de Densmore? *Bibliothèque Oulipienne*, 67, 1994. réédition Castor Astral, 2000. 7
- [BFV04] Guillaume Blin, Guillaume Fertin, and Stéphane Vialette. New results for the 2-interval pattern problem. In *Proceedings of the 15th Symposium on Combinatorial Pattern Matching (CPM04)*, 2004. http://www.sciences.univ-nantes.fr/info/perso/permanents/fertin/PAPERS/Blin_Fertin_Vialette_CPM04.pdf. 7, 9, 25, 30
- [BG75] Edward A. Bender and Jay R. Goldman. On the applications of moebius inversion in combinatorial analysis. *American Mathematical Monthly*, 82(8):789–803, 1975. <http://www.citeulike.org/user/euclid/article/775065>. 47
- [Bie05] Therese Biedl. Graph-theoretic algorithms, 2005. <http://www.student.cs.uwaterloo.ca/~cs762/Notes/book.pdf>. 14
- [BKKM99] Hajo Broersma, Ton Kloks, Dieter Kratsch, and Haiko Müller. Independent sets in asteroidal triple-free graphs. *SIAM Journal on Discrete Mathematics*, 12(2):276–287, 1999. <http://epubs.siam.org/fulltext/SIDMA/volume-12/32634.pdf>. 27
- [BL76] Kellogg S. Booth and George S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comp. Sys. Sci.*, 13:335–379, 1976. 15
- [Bli05] Guillaume Blin. *Combinatoire et Bio-informatique : Comparaison de structures d'ARN et calcul de distances intergénomiques*. PhD thesis, Université de Nantes, 2005. http://igm.univ-mlv.fr/~gblin/recherches/Publis/Phd_Guillaume_Blin.pdf. 7, 10
- [BLS⁺] Andreas Brandstädt, Van Bang Le, Thomas Szymczak, Frank Siegemund, H.N. de Ridder, Stefan Knorr, Mirko Rzehak, M. Mowitz, and Natalia Ryabova. ISGCI: Information System on Graph Class Inclusions. <http://www.teo.informatik.uni-rostock.de/isgci/classes.cgi>. 17, 18
- [BLS99] Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph Classes: a Survey*. SIAM Monographs on Discrete Mathematics and Applications, 1999. 39
- [BNR96] Vineet Bafna, Babu O. Narayanan, and R. Ravi. Nonoverlapping local alignments (weighted independent sets of axis-parallel rectangles). *Discrete Applied Mathematics*, 71(1):41–54, 1996. <http://www.gsia.cmu.edu/afs/andrew/gsia/ravi/WWW/public/rectangle.ps>. 12, 26
- [BOP04] József Balogh, Pascal Ochem, and András Pluhár. On the interval number of special graphs. *Journal of Graph Theory*, 46:241–253, 2004. <http://www.inf.u-szeged.hu/~pluhar/ieven.ps>. 17

- [Bou06] Mathilde Bouvel. Autour des permutations séparables. Master's thesis, ENS Cachan, MPRI, 2006. <http://philippe.gambette.free.fr/LIAFA/Bouvel2006.htm>. 29
- [BYHN⁺02] Reuven Bar-Yehuda, Magnús M. Halldórson, Joseph Naor, Hadas Shachnai, and Irina Shapira. Scheduling split intervals. In *13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 732–741, 2002. <http://www.cs.technion.ac.il/~reuven/PS/BarHal04.ps>. 11, 27
- [CHLV05] Maxime Crochemore, Danny Hermelin, Gad M. Landau, and Stéphane Vialette. Approximating the 2-interval pattern problem. In *Algorithms - ESA 2005, 13th Annual European Symposium*, volume 3669 of *Lecture Notes in Computer Science*, pages 426–437, 2005. http://dx.doi.org/10.1007/11561071_39. 7, 26, 31
- [CLR90] Thomas Cormen, Charles Leiserson, and Ronald Rivest. *Introduction to Algorithms*. MIT Press, 1990. 41
- [COS97] Derek G. Corneil, Stephan Olariu, and Lorna Stewart. Asteroidal triple-free graphs. *SIAM Journal on Discrete Mathematics*, 1997. <http://epubs.siam.org/fulltext/SIDMA/volume-10/25012.pdf>. 38
- [CPS06] Peter Cameron, Thomas Prellberg, and Dudley Stark. Asymptotic enumeration of incidence matrices. *Journal of Physics: Conference Series*, 42:59–70, 2006. http://ej.iop.org/links/r8JtBU4n_/st_AleI52xG2ZV7Sav5vpA/jpconf6_42_007.pdf. 47
- [DM41] Ben Dushnik and Edwin W. Miller. Partially ordered sets. *American Journal of Mathematics*, 63:600–610, 1941. 38
- [dM03] Fabien de Montgolfier. *Décomposition modulaire des graphes, théorie, extensions et algorithmes*. PhD thesis, Université Montpellier II, 2003. <http://www.liafa.jussieu.fr/~fm/publications/memoire.pdf>. 32
- [Edm65] Jack Edmonds. Paths, trees and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965. 42
- [Eva99] Patricia A. Evans. *Algorithms and Complexity for Annotated Sequence Analysis*. PhD thesis, University of Victoria, 1999. <http://www.cs.unb.ca/profs/pevans/research/pae-thesis.ps>. 7
- [FF62] Lester R. Ford and Delbert R. Fulkerson. *Flows In Networks*. Princeton University Press, 1962. 42
- [Fis85] Peter C. Fishburn. *Interval Orders and Interval Graphs*. John Wiley & Sons, 1985. 7
- [FMW97] Stefan Felsner, Rudolf Müller, and Lorenz Wernisch. Trapezoid graphs and generalizations, geometry and algorithms. *Discrete Applied Mathematics*, 74(1):13–32, 1997. <ftp://ftp.inf.fu-berlin.de/pub/reports/tr-b-94-02.ps>. 30
- [Gal86] Zviv Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys*, 18(1):23–38, 1986. <http://doi.acm.org/10.1145/6462.6502>. 42
- [Gar05] Frédéric Gardi. *Ordonnancement avec exclusion mutuelle par un graphe d'intervalles ou d'une classe apprentée : complexité et algorithmes*. PhD thesis, Université de la Méditerranée - Aix-Marseille II, 2005. http://www.lif-sud.univ-mrs.fr/~gardif/downloads/These_Gardi.pdf. 7
- [GJ79] Michael R. Garey and David. S. Johnson. *Computers and Intractability, a guide to the theory of NP-completeness*. Bell Telephone Laboratories, Inc., 1979. 40, 41, 42

- [GLL82] Udaiprakash I. Gupta, D. T. Lee, and Joseph Y.-T. Leung. Efficient algorithms for interval graphs and circular-arc graphs. *Networks*, 12:459–467, 1982. 30
- [GLS81] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981. <http://www.zib.de/groetschel/pubnew/paper/groetschellovaszschrijver1981a.djvu>. 27
- [Gol80] Martin C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press Inc, 1980. 7, 14, 41
- [Gra04] Jens Gramm. A polynomial-time algorithm for the matching of crossing contact-map patterns. In *Proceedings of the 4th International Workshop on Algorithms in Bioinformatics (WABI 2004), Lecture Notes in Bioinformatics*, volume 3240, pages 38–49, 2004. <http://portal.acm.org/citation.cfm?id=1042198.1042361>. 29
- [GW79] Jerrold R. Griggs and Douglas B. West. Extremal values of the interval number of a graph. *SIAM Journal on Algebraic and Discrete Methods*, 1:1–7, 1979. 17
- [Gyá] András Gyárfás. Combinatorics of intervals, preliminary version. <http://www.math.gatech.edu/news/events/ima/newag.pdf>. 14
- [Har69] Frank Harary. *Graph Theory*. Addison-Wesley, 1969. 42
- [Hol81] Ian Holyer. The NP-completeness of edge-colouring. *SIAM Journal on Computing*, 10(4):718–720, 1981. <http://www.cs.bris.ac.uk/~ian/graphs/edge.pdf>. 41
- [HTC92] Ju Yuan Hsiao, Chuan Yi Tang, and Ruay Shiung Chang. An efficient algorithm for finding a maximum weight 2-independent set on interval graphs. *Information Processing Letters*, 43(5):229–235, 1992. 30
- [Kar05] Ragnar Karlsson. A survey of split intervals and related graphs. <http://www.hi.is/~rkk1/SplitSurvey.pdf>, 2005. 10
- [Ken69] David G. Kendall. Incidence matrices, interval graphs, and seriation in archaeology. *Pacific Journal of Mathematics*, 28:565–570, 1969. <http://projecteuclid.org/Dienst/UI/1.0/Summarize/euclid.pjm/1102983306>. 7
- [KMMS03] Dieter Kratsch, Ross M. McConnell, Kurt Mehlhorn, and Jeremy P. Spinrad. Certifying algorithms for recognizing interval graphs and permutation graphs. In *Proceedings of the fourteenth annual ACM-SIAM Symposium on Discrete algorithms (SODA'03)*, pages 158–167, 2003. <http://www.cs.colostate.edu/~rmm/certificates.pdf>. 15
- [KMR72] Richard M. Karp, Raymond E. Miller, and Arnold L. Rosenberg. Rapid identification of repeated patterns in strings, trees and arrays. In *Proceedings of the 4th Annual ACM Symposium*, pages 125–136, 1972. <http://doi.acm.org/10.1145/800152.804905>. 12
- [KW99] Alexandr V. Kostochka and Douglas B. West. Every outerplanar graph is the union of two interval graphs. *Congress Numerantium*, 139:5–8, 1999. <http://citeseer.ifi.unizh.ch/297772.html>. 17
- [Leh74] Philippe G. H. Lehot. An optimal algorithm to detect a line graph and output its root graph. *Journal of the ACM (JACM)*, 21(4):569–575, 1974. <http://doi.acm.org/10.1145/321850.321853>. 15
- [Lew87] David Lewin. *Generalized Musical Intervals and Transformations*. Yale University Press, 1987. 13

- [Lin06] Yaw-Ling Lin. Circular and circle trapezoid graphs. *Journal of Science and Engineering Technology*, 2(2):11–17, 2006. <http://journal.dyu.edu.tw/dyujournal/document/s2-2-11-17.pdf>. 38
- [LL06] Shuai-Cheng Li and Ming Li. On the complexity of the crossing contact map pattern matching problem. In *Proceedings of 6th Workshop on Algorithms in Bioinformatics (WABI 2006)*, 2006. 29, 49
- [Mäk99] Erkki Mäkinen. On the longest upsequence problem for permutations, 1999. <http://citeseer.ist.psu.edu/225912.html>. 30
- [MS94] Tze-Heng Ma and Jeremy P. Spinrad. On the 2-chain subgraph cover and related problems. *Journal of Algorithms*, 17(2):251–268, 1994. 15
- [MV80] Silvio Micali and Vijay V. Vazirani. An $O(\sqrt{|v||e|})$ algorithm for finding maximum matching in general graphs. In *Proceedings of the 21th Annual IEEE Symposium on Foundation of Computer Science*, pages 17–27, 1980. 27, 42
- [Pól37] George Pólya. Kombinatorische anzahlbestimmungen für gruppen, graphen, und chemische verbindungen. *Acta Mathematica*, 68:145–254, 1937. 42
- [RPP⁺82] Krijn Rietveld, R. Van Poelgeest, Cornelis W.A. Pleij, J.H. Van Boom, and Leendert Bosch. The tRNA-like structure at the 3' terminus of turnip yellow mosaic virus RNA, differences and similarities with canonical tRNA. *Nucleic Acids Research*, 10:1929–1946, 1982. 12
- [SB05] David W. Staple and Samuel E. Butcher. Pseudoknots: RNA structures with diverse functions. *PLoS Biology*, 3(6):109–133, 2005. <http://biology.plosjournals.org/perlserv/?request=get-document&doi=10.1371/journal.pbio.0030213>. 12
- [Slo] Neil J. A. Sloane. The on-line encyclopedia of integer sequences. <http://www.research.att.com/~njas/sequences/>. 42, 44, 46, 48
- [Spi94] Jeremy P. Spinrad. Recognition of circle graphs. *Journal of Algorithms*, 16(2):264–282, 1994. 15
- [TH79] William T. Trotter and Frank Harary. On double and multiple interval graphs. *Journal of Graph Theory*, 3:205–211, 1979. 17, 18
- [Tis06] Alexander Tiskin. Semi-local longest common subsequences and maximum cliques in circle graphs. In *Proceedings of CPM2006, Lecture Notes in Computer Science*, volume 4009, pages 271–282, 2006. <http://www.dcs.warwick.ac.uk/~tiskin/pub/2006/circle.pdf>. 30
- [Via01] Stéphane Vialette. *Aspects algorithmiques de la prédiction des structures secondaires d'ARN*. PhD thesis, Université Paris 7, 2001. <http://philippe.gambette.free.fr/LIAFA/Vialette2001.htm>. 7, 12, 18, 21, 25, 27, 30
- [Via04] Stéphane Vialette. On the computational complexity of 2-interval pattern matching problems. *Theoretical Computer Science*, 312(2-3), 2004. <http://dx.doi.org/10.1016/j.tcs.2003.08.010>. 7, 12, 30
- [WC53] James D. Watson and Francis H. Crick. Molecular structure of nucleic acids, a structure for deoxyribose nucleic acid. *Nature*, 1953. <http://www.annals.org/cgi/reprint/138/7/581.pdf>. 12
- [WS84] Douglas B. West and David B. Shmoys. Recognizing graphs with fixed interval number is NP-complete. *Discrete Applied Mathematics*, 8:295–305, 1984. 17