

chapitre5: Découverte de topologie par les corrélations

Wilfried Ehounou

July 5^e 2018

Dans le chapitre précédent, nous avons déterminé la matrice de corrélation M_c du graphe G d'un réseau électrique. Cette matrice peut contenir des cases erronnées. Une case erronnée $M_c[i, j]$ est un coefficient de corrélation proche de 1 (resp. de 0) entre les arcs i et j alors que ces arcs ne partagent aucune extrémité (resp. ces arcs ont une extrémité commune).

Nous considérons une matrice M de dimension identique à celle de M_c telle que, pour toute valeur de seuil $s \in [0, 1]$ et toute paire d'arcs (i, j) , $M[i, j] = 1$ si et seulement si $M_c[i, j] \geq s$. La matrice M est la matrice d'adjacence d'un graphe non orienté G_c dit *graphe de corrélation*. Cette matrice peut également contenir des cases erronnées. Une case erronnée $M[i, j] = 1$ désigne la présence d'arêtes dans G_c alors qu'il n'existe aucune arête entre les sommets i et j dans le line-grahe du graphe non-orienté sous jacent au DAG G . De même, l'absence d'une arête entre i et j dans G_c alors qu'elle est présente dans le line-grahe du graphe non-orienté sous jacent à G est aussi une case erronné $M[i, j] = 0$.

S'il n'existe aucune case erronnée dans la matrice M , alors G_c est le line-grahe de graphe non-orienté sous jacent au DAG G et le line-grahe de G est isomorphe à G_c . Notre but est de déduire le DAG G à partir de G_c en deux étapes :

- Déterminer si G_c est un line-grahe. Si c'est le cas, déduire le graphe dont G_c est le line-grahe.
- Dans le cas où G_c n'est pas un line-grahe, proposer un algorithme qui modifie G_c de tel sorte qu'il devient un line-grahe et ensuite déduire le graphe dont le graphe G_c modifié est le line-grahe.

Nous désignons ce problème par *Proxi-Line*.

Nous allons, dans un premier temps, décrire les caractéristiques d'un line-grahe et le problème *Proxi-Line*. Ensuite nous présentons les algorithmes qui traitent ce problème et enfin nous expliquons la reconstruction de la topologie à partir du line-grahe découvert par nos algorithmes.

0.1 Line-graphes : caractéristiques et propriétés

Dans la théorie des graphes, un line-grahe est aussi appelé un *graphe adjoint* et ce terme est introduit par l'article de Harary et Norman [?]. Le line-grahe d'un graphe non orienté G est un graphe qui représente la relation d'incidence entre les arêtes de G . Nous allons définir formellement le line-grahe d'un graphe et présenter ses propriétés.

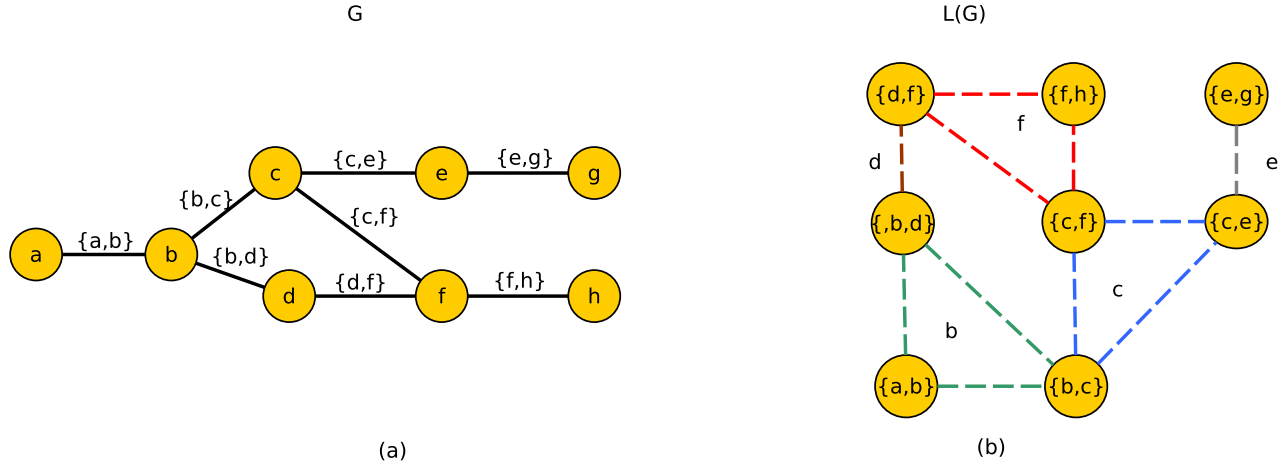
0.1.1 Caractéristiques d'un line-grahe

Définition 1. Soit $G = (V, E)$ un graphe non orienté.

Le line-grahe de G est un graphe non-orienté $L(G) = (V', E')$ dans lequel $V' = E$ et une paire $[a, a']$ de sommets de $L(G)$ est une arête si et seulement si a et a' ont une extrémité commune dans G .

Le graphe G est appelé le graphe racine de $L(G)$.

Si G est un DAG alors $L(G)$ est le line-grahe du graphe non-orienté sous jacent à G . Étant donné que G a n sommets et m arcs (sans arcs symétriques), le graphe $L(G)$ a m sommets et $|E'| = \sum_{i=1}^n d_i(d_i - 1)/2$ arêtes avec d_i le degré de chaque sommet i de G .

Figure 1: Le graphe G et son line-graphe $L(G)$.

La figure 1(a) présente le graphe $G = (V, E)$ dans lequel l'ensemble V contient 8 sommets $V = \{a, b, c, d, e, f, g, h\}$ et l'ensemble E contient 8 arêtes $E = \{\{a, b\}, \{b, c\}, \{b, d\}, \{c, f\}, \{d, f\}, \{f, h\}, \{c, e\}, \{e, g\}\}$. Chaque arête de E devient un sommet de $L(G)$ dans la figure 1(b). Lorsque deux arêtes de E ont une extrémité commune alors leurs sommets respectifs dans $L(G)$ sont adjacents. Par exemple, dans la figure 1(b), les sommets $\{b, d\}$ et $\{d, f\}$ sont liés par une arête à cause du sommet $d \in V$. Nous construisons ainsi le graphe $L(G)$ qui contient 8 sommets et 11 arêtes. Nous constatons qu'un sommet de G correspond à une clique dans $L(G)$. En effet, les sommets de la clique $\{\{a, b\}, \{b, c\}, \{b, d\}\}$ dans $L(G)$ concourent à un point b qui est un sommet de $G[V]$. Le sommet b de G identifie la clique $\{\{a, b\}, \{b, c\}, \{b, d\}\}$ dans $L(G)$. Le graphe $L(G)$ est le line-graphe de G et G est le *graphe racine*.

La notion de *line-graphe* a été introduite par *Whitney* [?] en se basant sur la notion d'isomorphisme. Il montre que si deux line-graphes sont isomorphes et connexes alors leurs graphes racines sont aussi isomorphes à l'exception des graphes triangle K_3 et étoile $K_{1,3}$.

Proposition 1. [?] *Le graphe étoile $K_{1,3}$ n'est pas un line-graphe.*

Preuve 1. Supposons que $K_{1,3}$ est le line-graphe de H ($K_{1,3} = L(H)$). Alors H est un graphe connexe de quatre arêtes. Tous les graphes connexes de quatres arêtes sont représentés dans la figure 2. Comme $L(C_4) = C_4$ et $L(K_{1,3} + e) = K_4 + e$ (voir figure 2), $L(H)$ ne peut être que l'un des trois arbres P_4 , $K_{3,2}$ et $K_{1,4}$. Ce qui est contraire à notre hypothèse de départ ($K_{1,3} = L(H)$).

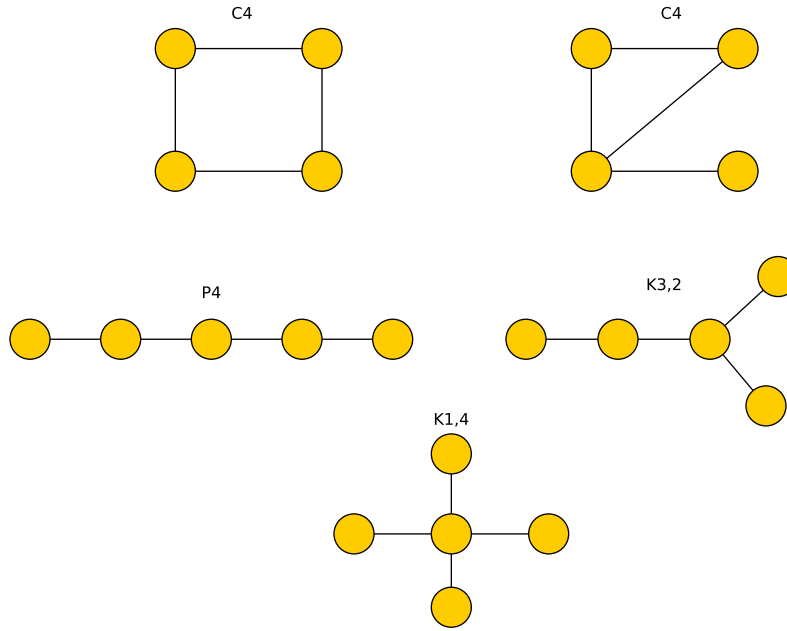


Figure 2: Les graphes racines possibles de $K_{1,3}$ de quatres arêtes.

Le graphe étoile ($K_{1,3}$) a un rôle important dans la caractérisation des line-graphes. La première caractéristique provient des travaux de *Krausz* [?] et elle est relative au partitionnement du line-graphe en sous-graphes. La seconde caractéristique, éditée par *van Rooij and wilf* [?], décrit la structure de base d'un graphe pour être un line-graphe. Et enfin, la dernière caractéristique présentée par *Beineke*[?] et *Hemminger* a déterminé les neufs sous-graphes ne pouvant pas être les sous-graphes induits de line-graphes.

Théorème 1. [?] Soit H un graphe. Les affirmations suivantes sont équivalentes.

- (a) H est un line-graphe.
- (b) Les arêtes de H peuvent être partitionnées en sous-graphes complets appelés cliques tel qu'aucun sommet n'est contenu dans plus de deux sous-graphes.
- (c) H ne contient pas $K_{1,3}$ comme sous-graphe et si deux triangles ont une arête commune alors le sous-graphe induit est K_4 .
- (d) Aucun des neufs sous-graphes de la figure 3 ne peut être un sous-graphe du line-graphe H .

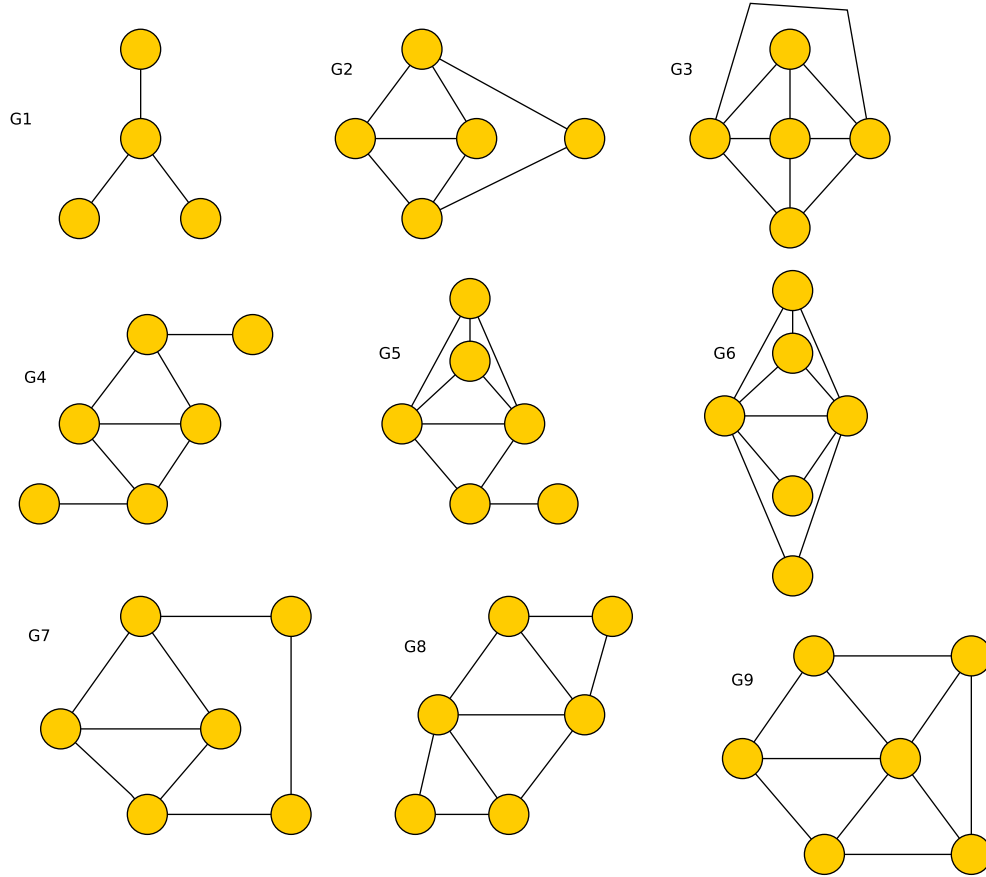


Figure 3: Les 9 sous-graphes interdits dans un line-graphe.

Conclusion : soit H un graphe et G un graphe non-orienté. Le graphe H est un line-graphe de G si le théorème 1 est respecté. Les graphes H et $L(G)$ sont isomorphes et le graphe racine de H est $L^{-1}(H)$. Si H un line-graphe d'un graphe non orienté G , alors le graphe H admet un partitionnement en cliques et chaque clique correspond à un sommet de G . L'ensemble de cliques est appelé une *couverture de corrélation* et est noté $\mathcal{CC}(G)$.

0.1.2 Line-graphes ambigus

Soient G un graphe non orienté et H l'unique line-graphe de G . D'après le théorème 1(b), les arêtes de H se partitionnent en cliques telles que chaque clique correspond à un sommet de G . Si le graphe de correction G_c est sans erreur, alors il est un line-graphe et son partitionnement en cliques forme une *couverture de corrélation* notée $\mathcal{CC}(G_c)$. Existe-t-il plusieurs *couvertures de corrélation* de G_c c'est-à-dire G_c a-t-il plusieurs graphes racines qui sont isomorphes ? Pour répondre à cette question, nous définissons la notion d'*ambiguïté*.

Définition 2. Soient G un graphe non orienté et $L(G)$ le line-graphe de G .

Une *ambiguïté* dans $L(G)$ est un graphe isomorphe à l'un des graphes de la figure 4. Le sommet X est appelé le **point d'ambiguïté**.

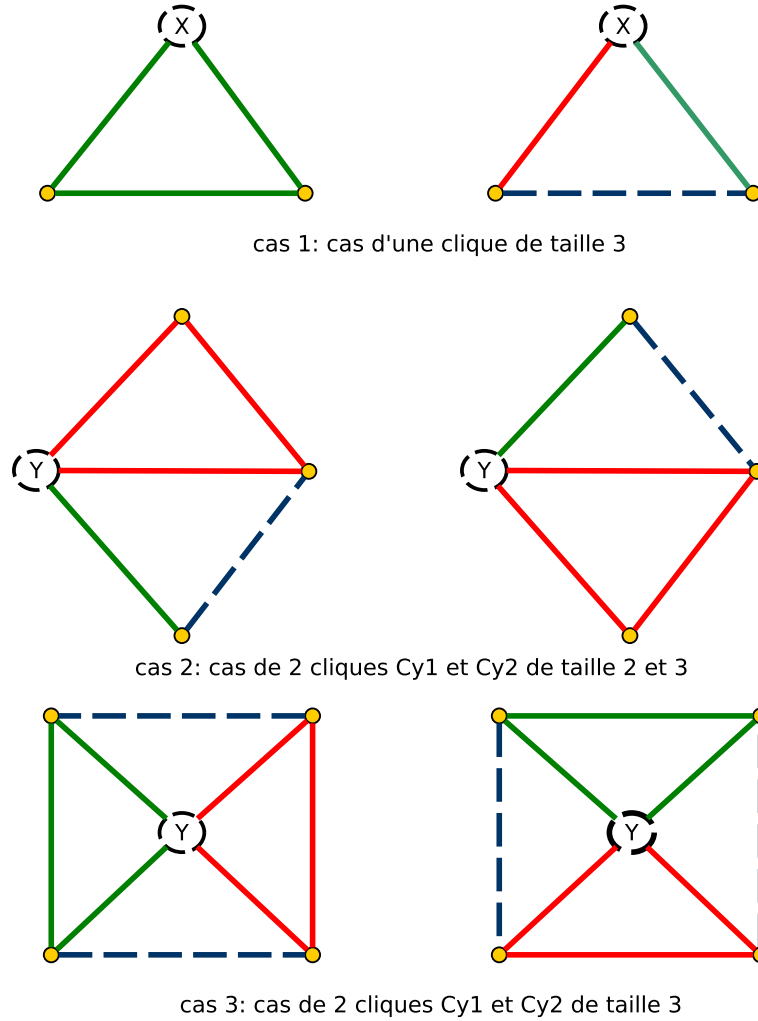


Figure 4: Configurations possibles d'une ambiguïté au sommet X.

Lemme 1. Soient $G = (V, E)$ un line-graphe et u un sommet de $V(G)$.

Si G admet deux couvertures de corrélation, alors il existe au moins un sommet u de G tel que $G[\{u\} \cup \Gamma_G(u)]$ est une ambiguïté dans laquelle u est le point ambiguïté.

Preuve 2. Considérons deux couvertures de corrélation $\mathcal{CC}(G)$ et $\mathcal{CC}'(G)$ de G . Il existe au moins un sommet $v \in V[G]$ qui n'est pas couvert par la (ou les) même(s) clique(s) dans $\mathcal{CC}(G)$ et $\mathcal{CC}'(G)$. Soient deux cliques c_1 et c_2 (potentiellement vide) partitionnant $\{v\} \cup \Gamma_G(v)$ dans $\mathcal{CC}(G)$. Considérons deux autres cliques c_3 et c_4 différentes de c_1 et c_2 partitionnant également $\{v\} \cup \Gamma_G(v)$ dans $\mathcal{CC}(G)$.

Notons $c_{i,j}$ l'intersection de c_i et c_j pour tout $i \in \{1, 2\}$ et $j \in \{3, 4\}$. Chaque sommet $w \in c_{i,j}$ est couvert par au plus deux cliques de G dans $\mathcal{CC}(G)$, dont la clique c_i . Puisque c_j est une clique alors ce sommet w est voisin de tous les sommets de $c_{i',j}$, pour $i' \neq i$. Les arêtes entre ces sommets sont dans $c_{i'}$, donc chaque arête $[w, z]$ pour tout sommet $z \in c_{i',j}$ forme une clique dans le réseau de flots. Ainsi, le cardinal de chaque ensemble $c_{i,j}$ est égal à 1.

Appelons $v_{i,j}$ le seul sommet présent dans $c_{i,j}$. Il est possible d'avoir $v_{1,3} = v_{1,4}$ ou $v_{2,3} = v_{2,4}$. Si les deux égalités sont vérifiées, le sommet v est alors couvert non pas par deux cliques mais par une seule de cardinalité 3. Ainsi, les seuls cas possibles sont alors résumés par la figure 5. Le sommet

v est bien le point d'une ambiguïté isomorphe à $G[\{u\} \cup \Gamma_G(u)]$.

Nous déduisons le corollaire suivant :

Corollaire 1. *Soit G un line-graphe.*

Si G admet deux couvertures de corrélation différentes, alors il est isomorphe à l'un des graphes de la figure 5.

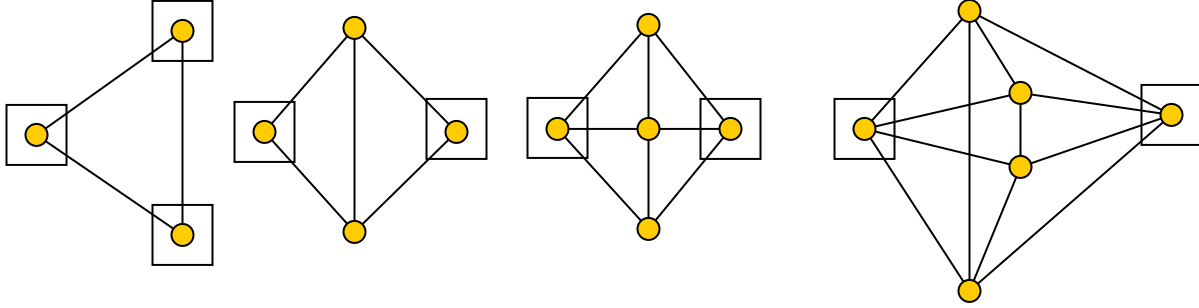


Figure 5: Les graphes possibles avec deux couvertures de corrélation avec les points d'ambiguïtés encadrés.

En effet, si $G[\{u\} \cup \Gamma_G(u)]$ a une ambiguïté, chaque arête, qui n'est pas liée au point d'ambiguïté, doit être une arête d'une et une seule autre ambiguïté de G . Et chaque sommet d'une ambiguïté, qui n'est pas un point d'ambiguïté, doit appartenir à une et une seule autre ambiguïté de G dont il n'est pas non plus le point d'ambiguïté. De plus, chaque arête, n'étant pas couverte par les deux configurations de cliques possibles dans une ambiguïté (les arêtes en pointillées dans la figure 4), doit être dans une autre ambiguïté à laquelle elle appartient. Ces contraintes font que si un graphe contient une ambiguïté induite, alors il ne peut être que dans un cas de la figure 5.

Définition 3. *Soient G un graphe et u un sommet de G et $\Gamma_G(u)$ les sommets voisins de u .*

*Une partition de $\Gamma_G(u)$ en deux cliques C_{u1}, C_{u2} est **cohérente** si et seulement si chaque sommet v de C_{u1} (resp. C_{u2}) a au plus un voisin dans C_{u2} (resp. C_{u1}).*

Conclusion : nous avons montré que la *couverture de corrélation* d'un line-graphe est unique à l'exception des situations ambiguïtés. Les cas d'ambiguïtés sont présentés dans la figure 5.

0.2 Formulation du Problème *Proxi-Line*

Soient G un graphe non-orienté d'un DAG, G_c un graphe de corrélation de G et M la matrice d'adjacence de G_c .

Notre problème est de déterminer G à partir de G_c . Pour ce faire, nous décidons de nous servir de la *couverture de corrélation*. On a 3 cas :

- Soit G_c est isomorphe à $L(G)$. Nous trouverons la couverture de corrélation unique qui donne G .
- Soit G_c est un line-graphe non isomorphe à $L(G)$. Modifier la matrice d'un line-graphe peut en effet le transformer en un autre line-graphe. Ce cas arrive rarement notamment lorsqu'il y'a peu d'arêtes erronnées dans G_c .

- Soit G_c n'est pas un line-graphe. Dans ce cas, l'idée est de corriger G_c en ajoutant ou supprimant le minimum d'arêtes.

Nous résolvons le 3^{ième} cas en introduisant le problème suivant :

0.2.1 Problème

Étant donné un graphe G' qui a des arêtes en plus ou en moins par rapport à un autre graphe G de même ensemble de sommets.

Définition 4. Soient G et G' deux graphes non-orientés ayant le même ensemble de sommets. La distance de Hamming entre les graphes G et G' notée $DH(G, G')$ est le nombre d'arêtes présentes dans G et pas G' et inversement.

Une distance de Hamming égale à k ($k \in \mathbb{N}$) signifie qu'il existe k cases différentes entre les matrices d'adjacence des graphes G et G' .

Définition 5. Soit G un graphe non orienté.

On appelle distance line de G , notée $DL(G)$, la plus petite distance de Hamming entre G et G' , un line-graphe ayant le même ensemble de sommets que G .

Nous considérons le problème suivant.

Problème Proxi-Line

Données : Un graphe $G = (V, E)$, un entier k .

Question : $DL(G) \leq k$?

Conjecture 1. *Proxi-Line est NP-complet.*

Ce problème généralise le problème *NP-complet* défini et montré dans [?], c'est-à-dire étant donné un graphe G et un entier k , le problème de savoir s'il existe un line-graphe G' sous-graphe couvrant de G et tel que $dH(G, G') \leq k$ (c'est-à-dire, le problème Proxi-Line dans lequel seule la suppression d'arêtes est autorisée); une solution de programmation linéaire en nombres entiers dans [?]. Récemment, il a été montré au sein du laboratoire DAVID que ce problème est aussi NP-complet si seul l'ajout d'arêtes est autorisé.

0.3 Algorithmes de découverte de topologie

Le problème considéré ici est, étant donné un graphe, de déterminer s'il est un line-graphe et dans ce cas donner un graphe racine. Nous débutons par l'état de l'art des algorithmes de couverture en cliques puis présentons nos algorithmes tout en spécifiant leurs particularités par rapport aux méthodes existantes.

Recherche de Couverture en cliques

Différents travaux ont été réalisés sur la découverte de *couverture en cliques* dans les line-graphes. Parmi lesquelles, nous citons l'algorithme de *Roussopoulos* [?] qui utilise une propriété des line-graphes provenant des travaux de *Krausz* [?]. Il affirme que le graphe G est un line-graphe si ces

arêtes peuvent être partitionnées en cliques de telle sorte qu'aucun sommet ne soit couvert par plus de deux cliques. L'algorithme proposé détecte si G est un line-graphe et il fournit, en plus, son graphe racine en temps linéaire $O(\max(\{m, n\}))$, avec n et m les nombres respectives de sommets et d'arêtes.

Un autre algorithme, proposé par *Klauss Simon* et *Daniele Degiorgi* [?], est une simplification du problème de reconnaissance de line-graphes. Basé sur la preuve de *Oystein Ore* [?] du théorème de *Whitney* [?], il stipule que deux line-graphes connexes avec plus de quatre sommets sont isomorphes si et seulement si leurs graphes racines sont aussi isomorphes et que ces graphes doivent être différents de $K_{1,3}$ et K_3 . Il détermine en un temps linéaire une couverture étant mise à jour sommet après sommet. L'inconvénient de cette méthode est le traitement de sommets appartenant à des cliques ayant déjà été découverts.

L'algorithme de *Philippe Lehot* [?] a une complexité en $O(n) + m$ avec n le nombre de sommets de G et m le nombre d'arêtes de $L(G)$. Il recherche les 9 sous-graphes de la figure 3 et il utilise le théorème de *Van Rooij et Wilf* [?] qui énonce qu'un graphe G est un line-graphe si G ne contient pas de sous-graphe induit $K_{1,3}$ et si deux graphes triangles *impairs* ont une arête commune, alors le sous-graphe induit par ces sommets est une clique K_4 . Rappelons qu'un graphe triangle (c'est-à-dire un cycle de longueur 3) $\{a_1, a_2, a_3\} \subseteq V(L(G))$ est *impair* s'il existe un sommet $e \in V(G)$ adjacent à au moins un des sommets $\{a_1, a_2, a_3\}$. Ce triangle est *pair* si chaque sommet de ce triangle est adjacent à 0 ou 2 autres sommets. Cet algorithme est détaillé dans la section 0.3.1.

Tous les algorithmes existants ne retournent aucun résultat lorsque le graphe de corrélation G_c possède des cases erronées c'est-à-dire qu'il n'est pas un line-graphe. Cependant, la méthode proposée par *Halldórsson and al.* [?] a pour but de corriger un graphe de corrélation pour en obtenir un line-graphe. En effet, il propose une méthode de découverte de la généalogie de population en se basant sur les haplotypes partagés entre les génomes des individus. Un haplotype est un groupe d'allèles [?] dans un organisme qui sont transmis ensemble par un parent. Les haplotypes sont uniques et sont les arêtes d'un graphe dit *Clark Consistency* [?]. Quant aux sommets, ils sont formés par les individus. Cette méthode recherche le graphe racine induit par le graphe *Clark Consistency* si celui-ci est un line-graphe. Dans le cas où le graphe *Clark Consistency graph* n'est pas un line-graphe, l'algorithme suppose qu'il existe des sommets superflus dans le graphe *Clark Consistency*, va les supprimer afin que le graphe devienne un line-graphe et enfin retourner le graphe racine. Le graphe *Clark consistency (CC)* a été proposé dans la méthode d'identification d'haplotypes par *Andrew Clark*. En effet, *Andrew Clark* considère un ensemble de génomes d'individus qui ont des haplotypes homozygotes et hétérozygotes. Il suppose que deux génomes n'ont pas les mêmes paires d'haplotypes. Les sommets du graphe CC sont les génomes des individus et une arête de graphe CC entre deux génomes existe s'ils partagent le même haplotype (homozygote ou hétérozygote). Les arêtes de ce graphe sont formées par des individus partageant les mêmes haplotypes. Le problème de découverte de line-graphes étant *NP-Complet*, la solution proposée réalise un algorithme de suppression de sommets et d'arêtes. L'algorithme de suppression de sommets est une 6-approximation alors que celui des arêtes est de complexité $O(n * m)$ avec m le nombre de sommets et n le nombre d'arêtes. Dans le cas où des suppressions sont effectuées, le line-graphe fourni est le plus proche possible du line-graphe de l'arbre généalogique. La particularité de la solution est l'absence d'arêtes ajoutées dans le line-graphe et cela implique que cette solution est inapplicable dans notre problème où il existe des arêtes inconnues dans notre graphe de corrélation. En plus, l'ensemble de nos sommets dans le graphe de corrélation est connu contrairement à l'algorithme de *Halldórsson et al.* qui suppose que les sommets doivent être supprimés pour atteindre un line-graphe.

Nous nous basons sur l'algorithme de *Lehot* parce qu'il s'exécute en un temps linéaire en effectuant

un traitement sommet par sommet pour la reconnaissance de sous-graphes complets. Ce traitement permet de sélectionner les cliques existantes et les sommets, n'appartenant à aucune clique, qui nécessitent une modification de leur voisinage.

0.3.1 Algorithme de couverture

L'algorithme de *couverture* que nous proposons est une amélioration de celui de *Lehot*. Ainsi, nous présentons tout d'abord brièvement le principe de l'algorithme de couverture en cliques de *Lehot* [?].

Soient H et G deux graphes. Nous supposons que H est le line-graphe de G . Le but de cet algorithme est d'identifier le graphe racine $L^{-1}(H)$ de H . L'algorithme va construire G au fur et à mesure en identifiant les cliques dans H . Les arêtes et les sommets de H et G peuvent avoir au cours de l'exécution plusieurs états :

- Sommet “bien-défini” : un sommet découvert de G tel que la clique correspondante dans H a été trouvée et identifiée.
- Sommet “à moitié-nommé” : un sommet de H tel que l'arête correspondante dans G a une extrémité “bien-définie”.
- Sommet “pleinement-nommé” : un sommet de H tel que l'arête correspondante dans G a des extrémités “bien-définies”.
- Sommet “basique” : un sommet de H tel que chaque sommet est une arête de G . Ces sommets sont notés $x - y$ dans H avec x et y des sommets découverts de G .
- Sommet “partagé” : sommet de H partageant une arête avec des sommets “basiques” adjacents. Ce sommet est une extrémité commune entre des arêtes de G incidentes.

L'idée de cet algorithme est de déterminer une couverture de corrélation de H en détectant, selon 3 cas [?] dans H , les sommets partagés adjacents à un sommet “basique” qui forment une clique. Nous illustrons le fonctionnement de l'algorithme avec l'exemple suivant illustré par la figure 6. L'algorithme sélectionne deux sommets “basiques” $1 - 2, 2 - 3$ et l'ensemble X des sommets adjacents aux sommets “basiques”. Si $X = \emptyset$ alors il n'existe pas de sommet partagé dans G et les sommets $1 - 2, 2 - 3$ sont étiquetés “à moitié nommé” (figure 6(a)). Si $X = \{x\}$ alors le sommet x est un sommet partagé dans H si $x = 2 - 4$ car le triangle $x, 1 - 2, 2 - 3$ est impair. Dans le cas où $x = 1 - 3$, le triangle $x, 1 - 2, 2 - 3$ est pair et aucun sommet découvert dans G n'est incident aux sommets du triangle $x, 1 - 2, 2 - 3$ (figure 6(b)). Le sommet x dans H est étiqueté “pleinement-nommé” et les autres sommets $1 - 2, 2 - 3$ dans H sont étiquetés “à moitié nommés”. Si $X = \{x, y\}$, il n'existe aucun sommet partagé dans H si x et y sont adjacents dans H . Dans le cas où ils ne sont pas adjacents alors ils forment deux triangles avec $1 - 2, 2 - 3$. Si $x = 2 - 4$ et $y = 1 - 3$ alors le triangle $\{2 - 4, 1 - 2, 2 - 3\}$ est impair et x est le sommet partagé dans H (figure 6(c)). Enfin pour $|X| = |\{a, b, c, \dots\}| = 3$, le sommet b dans H est étiqueté sommet partagé si a est adjacent à b sinon le sommet a devient le sommet partagé. La dernière étape sélectionne aléatoirement un sommet “à moitié-nommé” dans H qui est adjacent à un sommet “pleinement-nommé” dans H . Si ce sommet n'est pas déjà couvert par une clique alors il est “pleinement-nommé” et il est un sommet partagé dans H . À la fin de cette étape, tous les sommets sont étiquetés à “pleinement-nommé” dans H et ils deviennent des sommets partagés dans H .

L'algorithme s'exécute en $O(m) + m'$ avec m le nombre d'arêtes dans G et m' le nombre d'arêtes dans $L(G)$. Il retourne la liste de cliques découvertes dans H dans laquelle chaque clique correspond à un sommet de G . Cette liste de cliques est appelée *couverture de corrélation*. Malheureusement, si H n'est pas un line-graphe alors il ne retourne pas de couverture partielle.

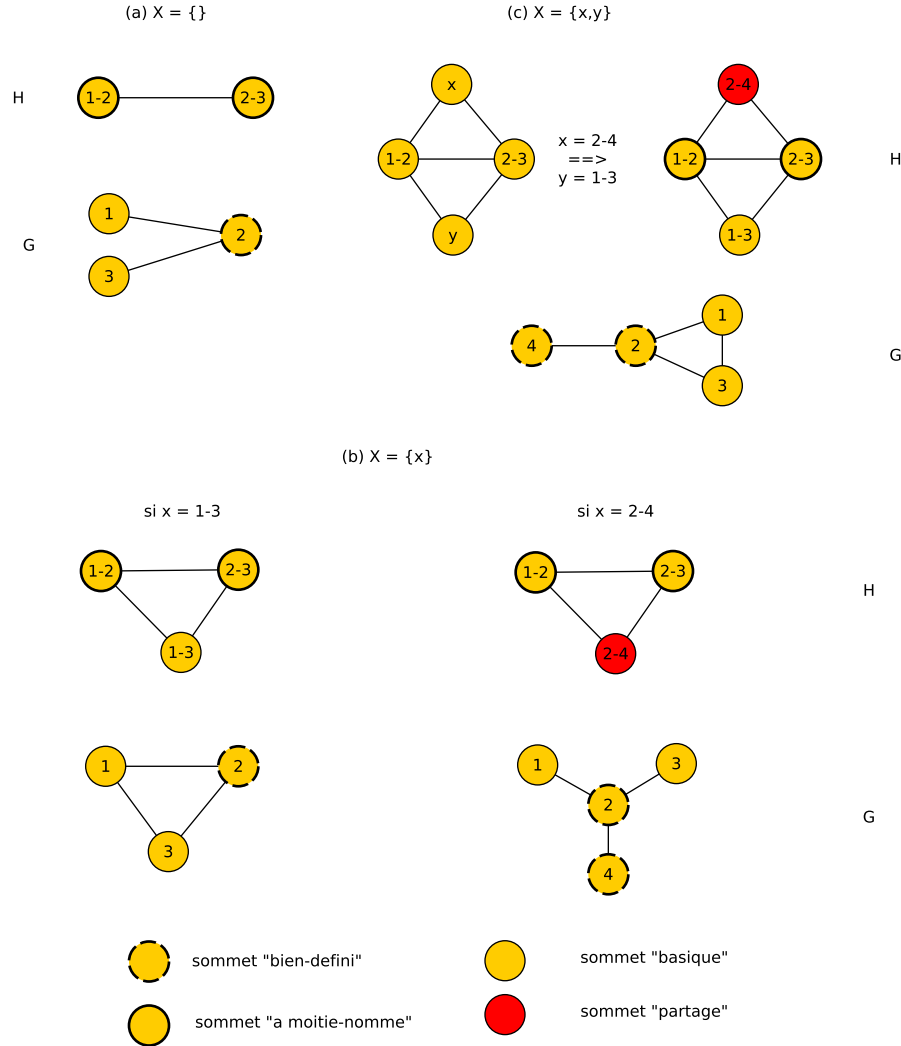


Figure 6: Identification des sommets partagés dans le graphe H et nommage des sommets de G .

Description de l'algorithme de couverture

Nous proposons un algorithme en lien avec celui de *Lehot* qui couvre autant que possible les sommets du graphe de corrélation G_c par une ou deux cliques. Notre algorithme retourne la *couverture de corrélation* de G_c si la matrice d'adjacence de G_c ne contient aucune case éronnée sinon il renvoie une *couverture de corrélation partielle* de G_c .

Soient $G_c = (V, E)$ un graphe de corrélation et $Cliq(v)$ l'état de chaque sommet v de G_c . L'algorithme va construire une couverture de corrélation $\mathcal{CC}(G_c)$ de G_c en ajoutant des cliques découvertes dans $\mathcal{CC}(G_c)$. Une clique est un ensemble de sommets qui induit un sous-graphe

complet. Si un sommet appartient à une clique alors il est couvert par cette clique. De même, si deux sommets u et v appartiennent à une même clique, alors la clique couvre l'arête $[x, y]$. Initialement $\mathcal{CC}(G_c)$ est vide et chaque sommet de $v \in V$ a un état $Cliq(v) = 0$.

À chaque étape de l'algorithme, chaque sommet v a 5 états possibles :

- $Cliq(v) = 0$: le sommet v n'est couvert par aucune clique. Il correspond à un sommet "basique" dans l'algorithme de *Lehot*.
- $Cliq(v) = 1$: le sommet v est couvert par une clique ou deux cliques. Dans le cas où il est couvert par deux cliques, l'intersection de ces cliques donne le sommet v . Ce sommet est étiqueté "pleinement-nommé" dans l'algorithme de *Lehot*.
- $Cliq(v) = 2$: le sommet v est couvert par une clique et peut être couvert par une seconde clique. Ce sommet est "bien-nommé" dans l'algorithme de *Lehot*.
- $Cliq(v) = 3$: le sommet v est un sommet ambigu. L'algorithme doit identifier la clique à laquelle il appartient pour qu'elle devienne un sommet partagé dans l'algorithme de *Lehot*.
- $Cliq(v) = -1$: le sommet v est couvert par plus de deux cliques. Il est contenu dans l'ensemble \mathcal{C} et doit être corrigé par l'algorithme de correction.

Nous choisissons un sommet v de degré minimum qui n'appartient à aucune clique ou qui est un sommet ambigu. S'il existe une partition cohérente de ce sommet et de son voisinage $\{v\} \cup \Gamma_{G_c}(v)$ en deux cliques C_1, C_2 , alors ces deux cliques sont contenues dans la couverture de corrélation $\mathcal{CC}(G_c)$ en cours de construction. Les sommets v et u (avec $u \in \Gamma_{G_c}(v)$), appartenant à C_1 ou C_2 , ont leurs états modifiés à chaque étape de l'algorithme de la manière suivante :

- $Cliq(v) = 1$ si son état précédent est égal à 0 et la clique C_2 est vide.
- $Cliq(v) = 3$ si son état précédent est égal à 0 et la clique C_2 est non vide.
- $Cliq(v) = 2$ si son état précédent est différent de 0.
- $Cliq(u) = 1$ si son état précédent est égal à 0 et l'ensemble des arêtes incidentes à u est vide.
- $Cliq(u) = 2$ si son état précédent est égal à 3 et l'ensemble des arêtes incidentes à u est vide.
- $Cliq(u) = 3$ si son état précédent est égal à 0 et l'ensemble des arêtes incidentes à u est non vide.
- $Cliq(u) = -1$ si son état précédent est égal à 3 et l'ensemble des arêtes incidentes à u est non vide.

Dans le cas où il n'existe aucune partition cohérente (voir définition 3) au sommet v , son état est à $Cliq(v) = -1$.

L'algorithme de recherche de couverture de corrélation $\mathcal{CC}(G_c)$ considère, tant qu'il en existe, un sommet v dont les arêtes incidentes sont couvertes par une seule clique. Il affecte à ce sommet l'état $Cliq(v) = 1$, sauvegarde cette clique dans \mathcal{H} puis supprime ces arêtes incidentes dans le graphe G_c . Les voisins de ce sommet passe à l'état 2.

Si au cours de l'exécution un tel sommet v n'existe pas alors l'algorithme considère un sommet u dont son voisinage peut être couvert par deux cliques et qui n'a pas été précédemment couverte par

une clique de $\mathcal{CC}(G_c)$. Si cette partition en deux cliques est unique, l'algorithme affecte $Cliq(u) = 1$ à ce sommet et supprime les arêtes. Dans le cas où ce sommet appartient à un des cas de la figure 4 (cas d'un sommet encadré), nous avons montré que ces graphes sont les seuls line-graphes pour lesquels deux partitions possibles existent. Nous utilisons la fonction de décision *Verif - correl* (section ??) pour lever l'ambiguïté.

Un état est affecté aux autres sommets v de la clique couvrant v selon l'un des trois cas. Dans le premier cas, l'état actuel appartient à $Cliq(v) \in \{2, 3\}$ si les sommets v ont des arêtes incidentes non encore couvertes par une clique et l'état précédent est $Cliq(v) \in \{3, 0\}$. Dans le second cas, $Cliq(v) = 1$ est attribué aux sommets v si l'ensemble des arêtes incidentes est vide. Enfin, dans le dernier cas, l'algorithme affecte $Cliq(v) = -1$ à ces sommets si leur état précédent est $Cliq(v) \in \{2, 3\}$ et leurs arêtes incidentes ne forment pas une clique.

À la fin de l'exécution de cet algorithme, tous les sommets v dont l'état courant est $Cliq(v) = -1$ n'ont pas été couverts. Ces sommets appartiennent à l'ensemble \mathcal{C} des sommets à corriger par l'algorithme de correction.

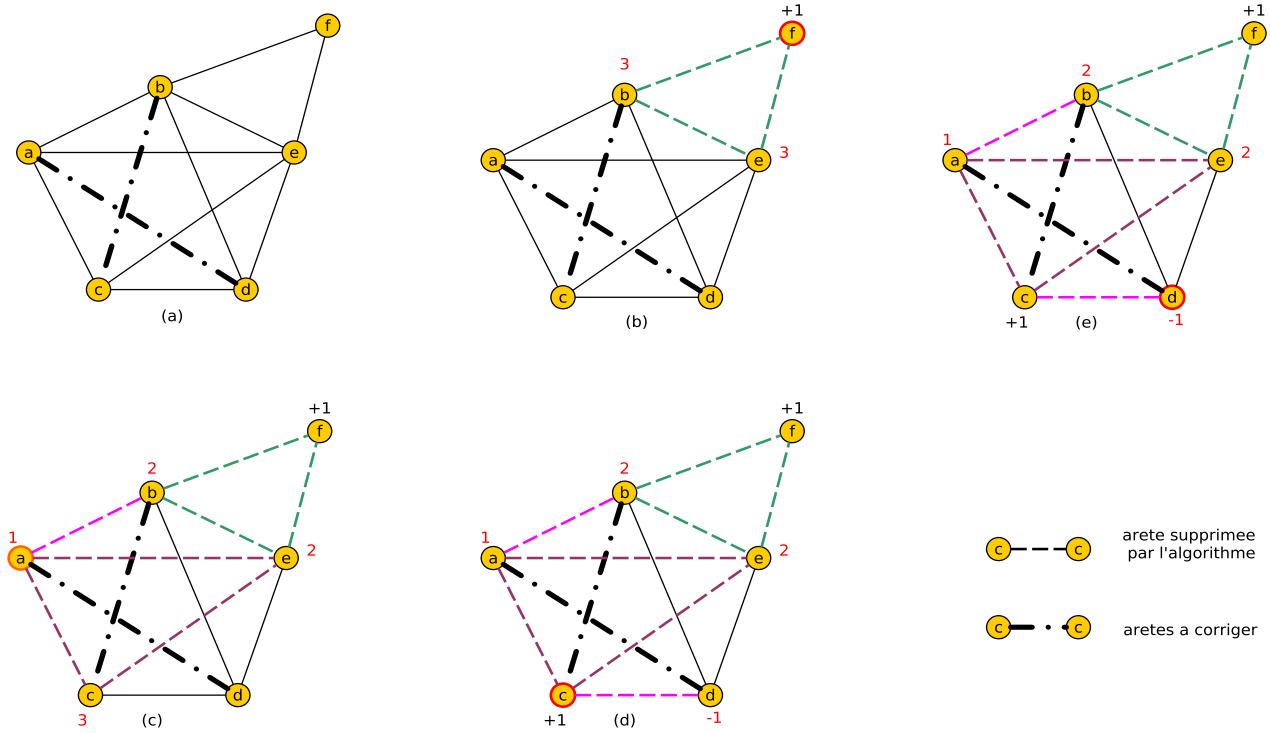


Figure 7: Les différentes étapes de la couverture en cliques du graphe G .

La figure 7 détaille l'algorithme de couverture sur un graphe $G = (V, E)$ dans lequel nous avons supprimé deux arêtes. L'objectif de la suppression est d'obtenir des sommets à corriger à la fin de la couverture. Nous sélectionnons le sommet f car il est de degré minimum. Il forme une clique avec son voisinage alors la clique $\{f, b, e\}$ est ajoutée à l'ensemble $\mathcal{CC}(G)$ puis les arêtes (f, b) , (b, e) , (f, e) sont supprimées de E . L'état de f est $Cliq(f) = 1$ et les sommets b et e ont $Cliq(b) = Cliq(e) = 3$ (voir figure 7(c)).

Le second sommet traité est a car son état est $Cliq(a) = 0$. Il existe deux partitions cohérentes $\{a, b\}$ et $\{a, e, c\}$ au voisinage de a . Ces deux partitions sont ajoutées à $\mathcal{CC}(G)$ et les arêtes de ces cliques sont supprimées de E . L'algorithme attribue

- Au sommet a , l'état $Cliq(a) = 1$,
- Aux sommets b et e , les états $Cliq(b) = Cliq(e) = 2$ car leur état précédent était $Cliq(b) = Cliq(e) = 3$ et ces sommets ont encore un voisin,
- Au sommet c , l'état $Cliq(b) = 3$.

On traite les autres sommets (c) de la même manière jusqu'à ce qu'on sélectionne le sommet d . Ce sommet a deux partitions $\{d, b\}$ et $\{d, e\}$ non cohérentes (voir définition 3) parce que la fonction *Verif – correl* (section ??) appliquée à ces partitions retourne 0. Ce sommet est donc à corriger $\mathcal{C} = \{d\}$ (voir figure 7(e)).

Algorithme 1 couverture_cliques

DEBUT

1. **Si** G_c est isomorphe à un graphe double (voir figure 5), **alors** le traiter avec *Verif_correl*⁽¹⁾

Sinon

2. **Tant que** il existe un sommet u t.q $Cliq(u) \in \{0, 3\}$

Faire

3. choisir u de degré minimum

4. **Si** $\{u\} \cup \Gamma_{G_c}(u)$ peut être couvert par deux cliques C_1 et C_2 cohérentes,
 C_1 maximale et $C_2 = \emptyset$ si $Cliq(u) = 3$ ⁽²⁾

alors

5. **Si** $Cliq(u) = 0$ et $C_2 \neq \emptyset$ **Alors** $Cliq = 3$

6. **Sinon Si** $Cliq = 0$ et $C_2 = \emptyset$ **Alors** $Cliq(u) = 1$

7. **Sinon** $Cliq(u) = 2$

8. **FinSi**

9. **FinSi**

10. $\epsilon_u = E(G_c[C_1]) \cup E(G_c[C_2])$

11. **Pour tout** $w \in \Gamma_{G_c}(u)$ **Faire**

12. $\alpha(w) = \text{card}\{[w, x] \in E - \epsilon_u\}$

13. **Si** $\alpha_w > 0$ **Alors**

14. **Si** $Cliq(w) = 0$ **Alors** $Cliq(w) = 3$

15. **Sinon Si** $Cliq(w) = 3$ **Alors** $Cliq(w) = -1$

16. **FinSi**

17. **Sinon Si** $Cliq(w) = 0$ **Alors** $Cliq(w) = 1$

18. **Sinon Si** $Cliq(w) = 3$ **Alors** $Cliq(w) = 2$

19. **FinSi**

20. **FinPourTout**

21. $E = E - \epsilon_u$

22. **Sinon** $Cliq(u) = -1$

FinSi

23. **FinTant que**

24. **Fin Si**

FIN

¹ : chaque graphe de la figure 5 admet deux couvertures de corrélation, souvent isomorphes, mais une seule de ces couvertures de corrélation peut correspondre au DAG du réseau électrique

sous-jacent. Dans ce cas, on utilise la fonction *Verif – correl* afin de déterminer la couverture de corrélation la plus probable.

² : le sommet u choisi (s’il existe) ne sera pas prioritairement un sommet tel que $Cliq(u) = 0$ et u est un point d’ambiguïté. Si lors d’une étape, seul un tel choix est possible et qu’il n’y a aucun sommet u tel que $Cliq(u) = -1$, c’est que chaque sommet du graphe initial G_c est un point d’ambiguïté. Dans ce cas, G_c est une union de composantes connexes isomorphes à un des graphes de la figure 5. Dans ce cas, n’importe quel choix conduit à une couverture de corrélation correcte.

Si le graphe $G_c = (V_c, E_c)$ est un graphe de corrélation alors l’algorithme de couverture en détermine la couverture de corrélation $\mathcal{CC}(G_c)$. En effet, si G_c est un line-graphe, par récurrence sur l’ensemble des sommets et à chaque étape, il existe un sommet non encore couvert qui :

- Soit est couvert par une clique appartenant à $\mathcal{CC}(G_c)$ et son voisinage restant peut être convert par une nouvelle clique.
- Soit n’est couvert par aucune clique de $\mathcal{CC}(G_c)$ et son voisinage restant peut être couvert par une ou deux nouvelles cliques.
- Soit est dans une ambiguïté alors on a recours à la fonction *Verif – correl* (section ??) pour déterminer les bonnes partitions de ce sommet.

Complexité de l’algorithme de couverture

La complexité de l’algorithme est au pire en $O(m \times \Delta(G_c))$ avec m le nombre d’arêtes et $\Delta(G_c)$ le degré maximum du graphe. Le facteur $\Delta(G_c)$ dans notre algorithme est du à la recherche de la partition en deux cliques C_1 et C_2 et est nécessaire à la détermination de l’ensemble des sommets v tels que $Cliq(v) = -1$, en nombre le plus petit possible.

Conclusion : si le graphe G_c est un line-graphe, tous ses sommets v sont labellisés à $Cliq(v) = 1$ et l’algorithme de couverture trouvera une partition du voisinage d’un sommet en une ou deux cliques de façon unique (voir les lemmes précédents). Une fois ce sommet et ses arêtes incidentes supprimés, le graphe restant est toujours un line-graphe, et la propriété se propage. Ainsi G_c qui possède des sommets v aux états $Cliq(v) = -1$ n’est pas un line-graphe. Nous proposons l’algorithme de correction qui retourne le line-graphe le plus proche de G_c .

0.3.2 Algorithme de correction

Nous l’avons vu, si $G_c = (V_c, E_c)$ n’est pas un line-graphe, certains sommets ne peuvent pas être convert par 1 ou 2 cliques. Dans l’algorithme de couverture, ces sommets v sont labellisés par $Cliq(v) = -1$. L’ensemble des cliques $\mathcal{CC}(G_c)$ ne contient alors que des cliques dont les sommets v labellisés à $Cliq(v) = 1$ qui sont couverts par 1 ou 2 cliques. Les sommets v dont l’état $Cliq(v) = -1$ appartiennent à l’ensemble \mathcal{C} des sommets à corriger et ce sont ces sommets qui sont traités par l’algorithme suivant.

Nous proposons l’*algorithme de correction* qui va modifier l’ensemble initial E_c par l’ajout et la suppression d’arêtes dans le but d’obtenir un *line-graphe*. Dans cet algorithme, nous traitons

un sommet de \mathcal{C} après l'autre sachant que chaque sommet peut modifier \mathcal{C} . Soit z_i le i^{ieme} sommet traité dans \mathcal{C} . Certaines expériences réalisées dans le chapitre ?? montrent que le choix des sommets à traiter, à chaque étape de correction, peut avoir une influence sur le line-graphe fourni parce que la correction modifie le voisinage des sommets. Nous notons alors E_c^i l'ensemble des arêtes de G_c après le traitement du $(i - 1)^{ieme}$ sommet de \mathcal{C} et $\mathcal{CC}^i(G_c)$ l'ensemble des cliques de G_c à l'étape i . Ainsi $E_c^1 = E_c$ et $\mathcal{CC} = \mathcal{CC}(G_c) = \mathcal{CC}^1(G_c)$. Nous notons \mathcal{CC} pour désigner $\mathcal{CC}(G_c)$ dans la suite de cette section.

Soient z_i le i -ième sommet et $\mathcal{CC}(z_i) = \{C_1, \dots, C_k\}$ l'ensemble des cliques maximales de \mathcal{CC}^i de taille supérieure ou égale à 3 auxquelles le sommet z_i appartient. Notons que, par définition et par construction, chaque paire de cliques dans $\mathcal{CC}(z_i)$ n'a que z_i comme sommet commun et que $S(z_i)$ est l'union des voisins v de z_i dans des cliques $\{v, z_i\} \in \mathcal{CC}^i$ de taille 2 et des voisins v de z_i tels que l'arête $[z_i, v]$ n'est couverte par aucune clique de \mathcal{CC}^i .

$$C(z_i) = \{C_i, i \in [1, k] \mid |C_i| \geq 3 \ \& \ C_i \in \mathcal{CC}^i\} \quad (1)$$

$$S(z_i) = \{v \in \Gamma_{G_c}(z_i) \mid \{v, z_i\} \in \mathcal{CC}^i\} \cup \{v \in \Gamma_{G_c}(z_i) \mid \nexists C \in \mathcal{CC}^i, [z_i, v] \in E_c(C)\} \quad (2)$$

Définition 6. Soient \mathcal{CC}^i la couverture de corrélation après le traitement des $(i - 1)^{ieme}$ sommets de \mathcal{C} et $\mathcal{CC}^i(z_i)$ l'ensemble des cliques contenant le i^{ieme} sommet z_i .

Deux cliques C et C' de $\mathcal{CC}^i(z_i)$ sont **contractables** si aucune arête $[u, v]$ de E_c^i telle que $u \in C$ et $v \in C'$ n'est couverte par une clique (autre que u, v) dans \mathcal{CC}^i . Un ensemble de cliques de \mathcal{CC}^i est contractable si tous les cliques sont deux à deux contractables.

Dans la figure 8(a), les paires de cliques $(C3, C4)$, $(C2, C3)$ sont contractables car il n'y a aucune arête entre les sommets 5 et 6 dans la première paire et dans la seconde paire, les sommets 3 et 4 n'ont aucune arête entre eux. Cependant, la paire $(C4, C6)$ n'est pas contractable car l'arête $[z_i, 10]$ est couverte par la clique $C5$. De même, la clique $C1$ n'appartenant pas à $C(z_i)$ entraîne que les cliques $C1$ et $C2$ ne sont pas contractables.

Définition 7. Soient \mathcal{CC}^i la couverture de corrélation après le traitement des $(i - 1)^{ieme}$ sommets de \mathcal{C} et $\mathcal{CC}^i(z_i)$ l'ensemble des cliques contenant le i^{ieme} sommet z_i .

Une clique $C \in \mathcal{CC}^i$ est **voisine** de z_i si $C \notin \mathcal{CC}^i(z_i)$ et $\text{card}(C \cap S(z_i)) \geq 1$. La dépendance d'une clique C voisine de z_i est l'ensemble $D_{z_i}(C) \subset \mathcal{CC}^i(z_i)$ tel que $C' \in D_{z_i}(C)$ si et seulement si $C' \cap C \cap \Gamma_{G_c}(z_i) \neq \emptyset$.

Une clique C est **augmentante** pour le sommet z_i si et seulement si elle est voisine de z_i et $D_{z_i}(C)$ est vide ou $D_{z_i}(C) \cup \{C\}$ est contractable.

$$\text{voisine}(z_i) = \{C \in \mathcal{CC}^i \mid C \notin C(z_i) \ \& \ \text{card}(C \cap S(z_i)) \geq 2\} \quad (3)$$

$$D_{z_i}(C) = \{C' \in C(z_i) \mid C' \cap C \cap \Gamma_{G_c}(z_i) \neq \emptyset\} \quad (4)$$

On appelle **augmentation** du sommet z_i l'union d'une clique augmentante C pour z_i et d'une contraction de cliques de $D_{z_i}(C)$.

Dans notre exemple, considérons $\bar{C}(z_i) = \{C1, C6\}$ les cliques n'appartenant pas à $C(z_i)$ et $S(z_i) = \{10, 1\}$. L'ensemble des cliques voisines à z_i est $\text{voisine}(z_i) = \{C1, C6\}$ parce que l'intersection de $C1$ et $S(z_i)$ donne un sommet $\{1\}$ et celui de $C6$ et $S(z_i)$ donne un sommet $\{10\}$ ($C1 \cap S(z_i) = \{1, 2, 11\} \cap \{10, 1\} = \{1\}$, $C6 \cap S(z_i) = \{8, 9, 10\} \cap \{10, 1\} = \{10\}$). Par ailleurs,

la dépendance de la clique $C1$ est $D_{z_i}(C1) = C2$ ($C1 \cap C2 \cap \Gamma_{G_c}(z_i) = \{2\}$) et celle de $C6$ est $D_{z_i}(C6) = C4$ ($C6 \cap C4 \cap \Gamma_{G_c}(z_i) = \{8\}$). Nous en déduisons que la clique $C1$ est *augmentante* car $C1$ est contractable avec $C2$ et est voisine de z_i . De même, la clique $C6$ est *augmentante* car $C6$ est voisine de z_i et puisque l'arête $[z_i, 10]$ forme la clique $C5$, la paire $(C6, C4)$ est contractable. Une *augmentation* de z_i est soit $\{z_i\} \cup C1 \cup C2$ ou soit $\{z_i\} \cup C4 \cup C6$.

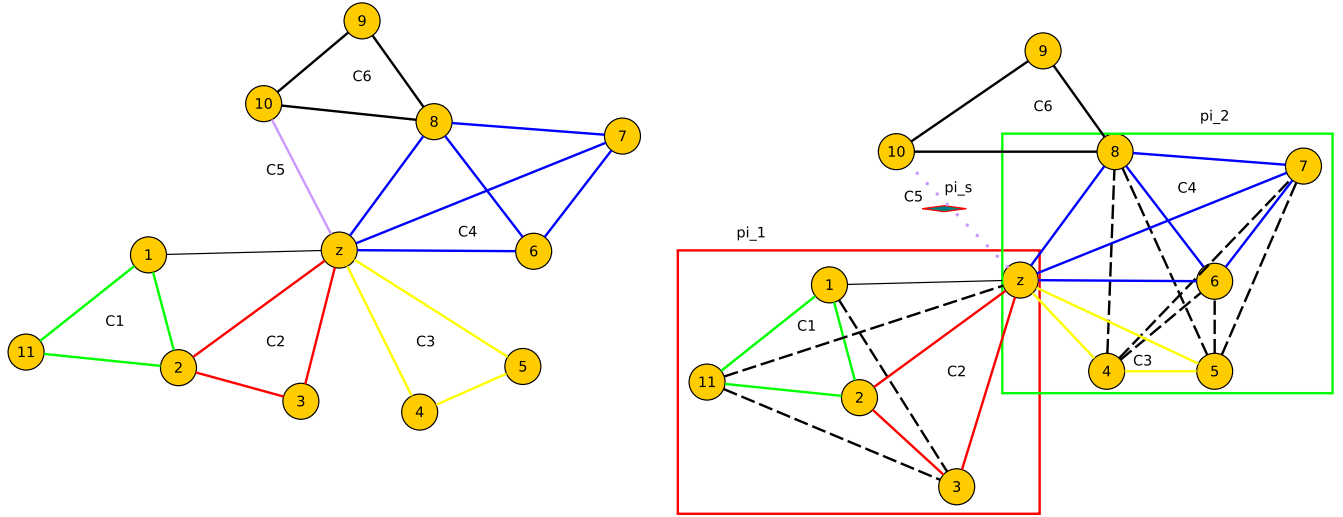


Figure 8: (a) Le sommet z et son voisinage avec les cliques qui le couvrent, (b) un exemple de compression de cliques : les sommets à l'intérieur des rectangles rouges et verts forment les nouvelles cliques couvrant z . $\pi_1 = \{1, 2, 3, z, 11\}$, $\pi_2 = \{z, 4, 5, 6, 7, 8\}$, $\pi_s = \{10\}$

Soient π_1, π_2 une bipartition du voisinage du sommet z_i en cliques et π_s un ensemble d'arêtes à retirer du graphe G_c .

Définition 8. Soient \mathcal{CC}^i la couverture de corrélation après le traitement des $(i-1)^{ieme}$ sommets de \mathcal{C} et $\mathcal{CC}^i(z_i)$ l'ensemble des cliques contenant le i^{ieme} sommet z_i .

On appelle compression du sommet z_i un triplet (π_1, π_2, π_s) défini par :

- π_1 (resp. π_2) peut être d'une des formes suivantes :
 1. l'union de z_i , d'un sous-ensemble C_1 (resp. C_2) de cliques de $\mathcal{CC}^i(z_i)$ telle que toute paire (C, C') de C_1 (resp. C_2) est contractable et d'un sous-ensemble S_1 (resp. S_2) de sommets $v \in S(z_i)$ n'appartenant à aucune clique de C_1 (resp. C_2) tel que

$$\forall v \in S_1, \forall x \in C_1, \nexists C' \in \mathcal{CC}^i \text{ t.q. } \text{card}(C') > 2 \text{ et } \{v, x\} \subset C'$$

(ce qui fait que $\{v, x\}$ peut être une clique de \mathcal{CC}^i).

2. une augmentation du sommet z_i

- L'intersection entre π_1 et π_2 est réduite $\{z_i\}$ ($\pi_1 \cap \pi_2 = \{z_i\}$),
- $\pi_s = \Gamma_{G_c}(z_i) - ((\pi_1 \cap \Gamma_{G_c}(z_i)) \cup (\pi_2 \cap \Gamma_{G_c}(z_i)))$ tel que l'ensemble des arêtes $\{[z_i, v] \in E_c : v \in \pi_s\}$ n'est pas déconnectant.
- le triplet $\pi_1 \cap \Gamma_{G_c}(z_i), \pi_2 \cap \Gamma_{G_c}(z_i), \pi_s \cap \Gamma_{G_c}(z_i)$ est une 3-partition de $\Gamma_{G_c}(z_i)$

Il existe toujours une telle compression, ne serait-ce que $\pi_1 = \{z_i\} \cup C_i \in C(z_i)$, $\pi_2 = \emptyset$, $\pi_s = \Gamma_{G_c}(z_i) - (\Gamma_{G_c}(z_i) \cup C_i)$ si $\mathcal{CC}^i(z_i)$ n'est pas vide. Sinon, $\pi_1 = \{z_i\} \cup \{v \in \Gamma_{G_c}(z_i)\}$, $\pi_2 = \emptyset$, $\pi_s = \Gamma_{G_c}(z_i) - \{v\}$ est aussi une compression. Un exemple de compression est aussi donné dans la figure 8. Le coût $c(T)$ d'une compression π_1, π_2, π_s est défini par :

$$c(T) = |\{\{u, v\} \in \pi_1 : [u, v] \notin E_c^i\}| + |\{\{u, v\} \in \pi_2 : [u, v] \notin E_c^i\}| + |\pi_s|$$

L'exemple de compression qui est donné dans la figure 8(b) est $\pi_1 = C1 \cup C2$ (une augmentation), $\pi_2 = C3 \cup C4$ (ces deux cliques étant contractables), et $\pi_s = \{x\}$. Les cliques $C1$ et $C2$ sont compressées en ajoutant les arêtes $[1, 3]$ et $[z_i, 11]$. De même, les cliques $C3$ et $C4$ sont compressées en ajoutant les arêtes $[4, 8]$, $[5, 8]$, $[7, 4]$, $[7, 5]$, $[6, 4]$ et $[6, 5]$. La clique $C5$ est supprimée afin que z_i ne soit pas couvert par trois cliques. Le coût de cette compression est 10, 10 étant le nombre d'arêtes en pointillées plus l'arête supprimée $[x, z_i]$.

Soit $c(T)$ le coût minimum d'une compression T de z_i . Le but est de modifier G_c afin que z_i puisse être couvert par une ou deux cliques issues de π_1 et π_2 . Pour cela, le coût de cette modification $c(T)$ tient compte des arêtes à ajouter (liées à π_1 et π_2) et à supprimer (liées à π_s).

$$c(T) = \sum_{\{u, v\} \subseteq \pi_1 : [u, v] \notin E_c^i} \phi^+(u, v) + \sum_{\{u, v\} \subseteq \pi_2 : [u, v] \notin E_c^i} \phi^-(u, v) + \sum_{v \in \pi_s} \phi^-(u, v) \quad (5)$$

Avec ϕ^+ le coût de l'opération *ajouter une arête* et ϕ^- le coût de l'opération *supprimer une arête*. Nous évaluons les performances des différents couples de fonctions ϕ^+ et ϕ^- dans la section ??.

Ainsi, **appliquer une compression** $T = \pi_1, \pi_2, \pi_s$ consiste à ajouter dans E_c^i les arêtes définies par les ensembles de paires $\{\{u, v\} \in \pi_1 : [u, v] \notin E_c^i\}$ (qui seront couvertes par la clique π_1) et $\{\{u, v\} \in \pi_2 : [u, v] \notin E_c^i\}$ (qui seront couvertes par la clique π_2) et à supprimer les arêtes $\{[z_i, v] \in E_c^i : v \in \pi_s\}$.

Dès lors, le sommet z_i appartient aux deux cliques π_1 et π_2 . On procède alors aux mises à jour suivantes pour obtenir \mathcal{CC}^{i+1} et E_M^{i+1} :

- Supprimer toutes les cliques $\mathcal{CC}^i(z_i)$ couvertes par π_1 dans \mathcal{CC}^i .
- Supprimer toutes les cliques $\mathcal{CC}^i(z_i)$ couvertes par π_2 dans \mathcal{CC}^i .
- Supprimer toutes les cliques de cardinalité 2 couvertes par π_1 et π_2 dans \mathcal{CC}^i .
- Ajouter π_1 et π_2 dans \mathcal{CC}^i , supprimer de E_c^{i+1} toutes les arêtes $\{[z_i, v] \in E_c^i : v \in \pi_s\}$.
- Affecter $Cliq(z)$ à 1 (si π_1 ou π_2 est vide) ou 2 (sinon).

Cette procédure a les propriétés suivantes :

Propriété 1. *Considérons l'application d'une compression.*

Soit \mathcal{CC}^{i+1} l'ensemble obtenu à partir de \mathcal{CC}^i après la mise à jour selon cette application.

- *Tout sommet de G_c couvert par une ou deux cliques dans \mathcal{CC}^i le reste dans \mathcal{CC}^{i+1} .*
- *Toute arête couverte par une et une seule clique dans \mathcal{CC}^i et qui n'est pas supprimée le reste dans \mathcal{CC}^{i+1} .*

- Le sommet z_i est couvert par une ou deux cliques dans \mathcal{CC}^{i+1} (le nombre de sommets ainsi couverts augmente de 1 par rapport à celui dans \mathcal{CC}^i).

Ainsi, pour chaque sommet z_i , on considère une compression de coût minimum c_m^i et on l'applique. La propriété ci-dessus garantit qu'à la fin du processus, on obtient un graphe de corrélation $G_c^t = (V_c, E_c^t)$ dont l'ensemble \mathcal{CC}^i modifié est une couverture de corrélation. Considérons la distance de correction $DC(G_c^0, G_c^t) = |(E_c^0 \cup E_c^t) - (E_c^0 \cap E_c^t)|$ qui est le nombre de cases modifiées dans la matrice d'adjacence du graphe G_c . La distance-line vérifie

$$DL(G_c^0, G_c^t) \leq DC(G_c^0, G_c^t)$$

Notons que lors d'une étape $j > 1$, le sommet z_j et son voisinage se retrouve être couvert par une ou deux cliques suite au traitement des $j - 1$ sommets précédents, aucune compression ne lui est appliquée (on considère la compression identité) et donc $c_m^j = 0$.

0.3.3 Complexité des algorithmes

L'algorithme de correction traite au plus une fois chaque sommet du graphe. La complexité de traitement de chaque sommet est exponentiel en fonction du degré de chaque sommet et des cliques auxquelles il appartient, la encore en fonction de son degré en taille et en nombre. L'algorithme global (couverture et correction) est donc pseudo-polynomial en fonction du degré du graphe.

Nous mettons une conjecture sur le comportement de l'algorithme. Etant donné un graphe de départ, une exécution de l'algorithme est un ordre dans lequel seront traités les sommets dans l'algorithme de couverture, puis la sélection des sommets z à traiter dans \mathcal{C} .

Considérons un graphe de corrélation G_c n'étant pas isomorphe à un graphe de la figure 5. On dira que G_c est non-ambigu.

Deux arêtes $[u, v]$ et $[u', v']$ de G_c seront dit **clique-indépendantes** si et seulement si il n'existe pas de cliques C dans la couverture de corrélation de G_c telle que $C \cap \{u, v\} \cap \{u', v'\} \neq \emptyset$

Conjecture 2. Si $G' = (V, E')$ est un graphe obtenu en supprimant un ensemble d'arêtes deux à deux clique-indépendantes d'un graphe de corrélation non-ambigu $G_c = (V, E_c)$, alors il existe une exécution de l'algorithme qui transforme G' en G_c .

0.3.4 Conclusion de la description des algorithmes

Dans cette section, nous décrivons deux algorithmes. Le premier algorithme est *l'algorithme de couverture* qui attribue un état à un sommet du graphe de corrélation en fonction des cliques qui le couvrent. L'ensemble de cliques est la couverture de corrélation ou couverture de corrélation \mathcal{CC} . La particularité de la couverture de corrélation est que chaque sommet appartient à 1 ou 2 cliques. Lorsqu'un sommet n'est pas couvert par 1 ou 2 cliques, cela signifie que le graphe de corrélation n'est pas un line-graphe et ces sommets sont regroupés dans l'ensemble \mathcal{C} de sommets à corriger.

Le second algorithme est *l'algorithme de correction*. Il consiste à ajouter ou à supprimer des arêtes au voisinage d'un sommet $u \in \mathcal{C}$ afin que la partition de ce sommet et son voisinage forme deux cliques. Pour assurer ces opérations d'ajout et de suppression, il utilise une phase d'augmentation et de compression. En effet, La phase d'augmentation détermine les cliques de \mathcal{CC} contenant u , les cliques de \mathcal{CC} dont u partage une arête avec un sommet de la clique (cliques

voisines), les cliques contractables (cliques de \mathcal{CC} dont l'intersection retourne le sommet u) et les cliques dépendantes (cliques contenant u dans lesquelles un des sommets partagent une arête avec une clique de la couverture de corrélation \mathcal{CC}). Puis elle effectue le produit cartésien de ces ensembles de cliques. Chaque élément de ce produit est noté π_1 ou π_2 . Quant à la phase de compression, elle sélectionne deux éléments du produit cartésien qu'elle note π_1 et π_2 , puis elle ajoute des arêtes à π_1 et π_2 dans l'ensemble des arêtes initiales du graphe de correction pour en faire des cliques. Elle crée aussi l'ensemble π_s des arêtes à supprimer pour que le sommet u ne soit couvert que par deux cliques. Les cliques π_1 et π_2 sont ajoutées à la couverture de corrélation \mathcal{CC} . Avec la découverte de la couverture de corrélation \mathcal{CC} , nous allons construire le graphe racine de ce line-graphe dans la section suivante.

0.4 Détermination de la topologie du réseau énergétique

Soient $\mathcal{CC}(G_c)$, l'ensemble des cliques du line-graphe G_c et le graphe non-orienté $G' = (V', E')$ sous-jacent du DAG G .

Nous considérons que chaque clique de $\mathcal{CC}(G_c)$ est un sommet dans G' . Si l'intersection de deux cliques $c_1, c_2 \in \mathcal{CC}(G_c)$ retourne l'arête a_i alors nous ajoutons l'arête a_i dans E' entre les sommets $c_1, c_2 \in G'$. Dans le cas où un sommet de G_c n'appartient qu'à une seule clique $c \in \mathcal{CC}(G_c)$, nous ajoutons un nouveau sommet (noté ext_c) dans V' puis nous ajoutons une arête (ext_c, c) dans E' . Nous obtenons le graphe G' non orienté connexe. Nous utilisons la figure 9 pour illustrer la construction de G' . La couverture de corrélation de G_c est $\mathcal{CC}(G_c) = [c_1 = \{\{a, b\}, \{b, c\}, \{b, d\}\}, c_2 = \{\{d, f\}, \{f, h\}, \{c, f\}\}, c_3 = \{\{b, c\}, \{c, e\}, \{c, f\}\}, c_4 = \{\{c, e\}, \{e, g\}\}, c_5 = \{\{b, d\}, \{d, f\}\}]$. Les sommets du G' sont $V' = \{c_1, c_2, c_3, c_4, c_5\}$. L'intersection des cliques ci-dessous est non vide et le sommet d'intersection est l'identifiant d'une arête dans G' .

$$c_1 \cap c_3 = \{b, c\}, \quad c_1 \cap c_5 = \{b, d\}, \quad c_3 \cap c_2 = \{c, f\}, \quad c_3 \cap c_4 = \{c, e\}, \quad c_2 \cap c_5 = \{d, f\}$$

Pour les sommets de G_c couverts par une seule clique, nous créons le sommet ext_x dans G' avec x le nom d'une clique de $\mathcal{CC}(G_c)$ puis nous ajoutons une arête entre le sommet ext_x et le sommet de G' correspondant à la clique. Par exemple le sommet $\{a, b\}$ de G_c est couvert par la clique c_1 . Nous relions ext_c_1 de G' avec le sommet c_1 de G' . Nous répétons la même opération pour les sommets $\{f, h\}, \{e, g\}$ de G_c . Le graphe G' (figure 9(c)) est ainsi construit et est isomorphe au graphe G de la figure 9(a).

0.4.1 Orientation du graphe G'

Nous supposons que le graphe non-orienté $G' = (V', E')$ sous-jacent au DAG G a déjà été obtenu. Nous orientons les arêtes de G' afin de découvrir le DAG cible.

Soient un sommet v , son voisinage $N(v)$ et une bipartition $p(v), s(v)$ de v .

Si $Verif - correl(p(v), s(v)) = 1$ alors la partie $p(v)$ est l'ensemble $p(v)$ des arcs entrants de v et la partie $s(v)$ est l'ensemble $s(v)$ des arcs sortants de v . Sinon la bipartition n'est pas la bonne et nous testons une autre bipartition. Nous testons ainsi dans le pire des cas, de l'ordre de $2^{d(v)}$ bipartitions avec $d(v) = |N(v)|$.

Soit une séquence $S = v_1, v_2, \dots, v_k$ de sommets de G tel que chaque arête est incidente à $\{v_1, \dots, v_k\}$ et que $\{v_1, \dots, v_{k-1}\}$ n'a pas cette propriété.

Soit d_i le nombre d'arêtes liant à v_i à un sommet de $V - \{v_1, \dots, v_{i-1}\}$.

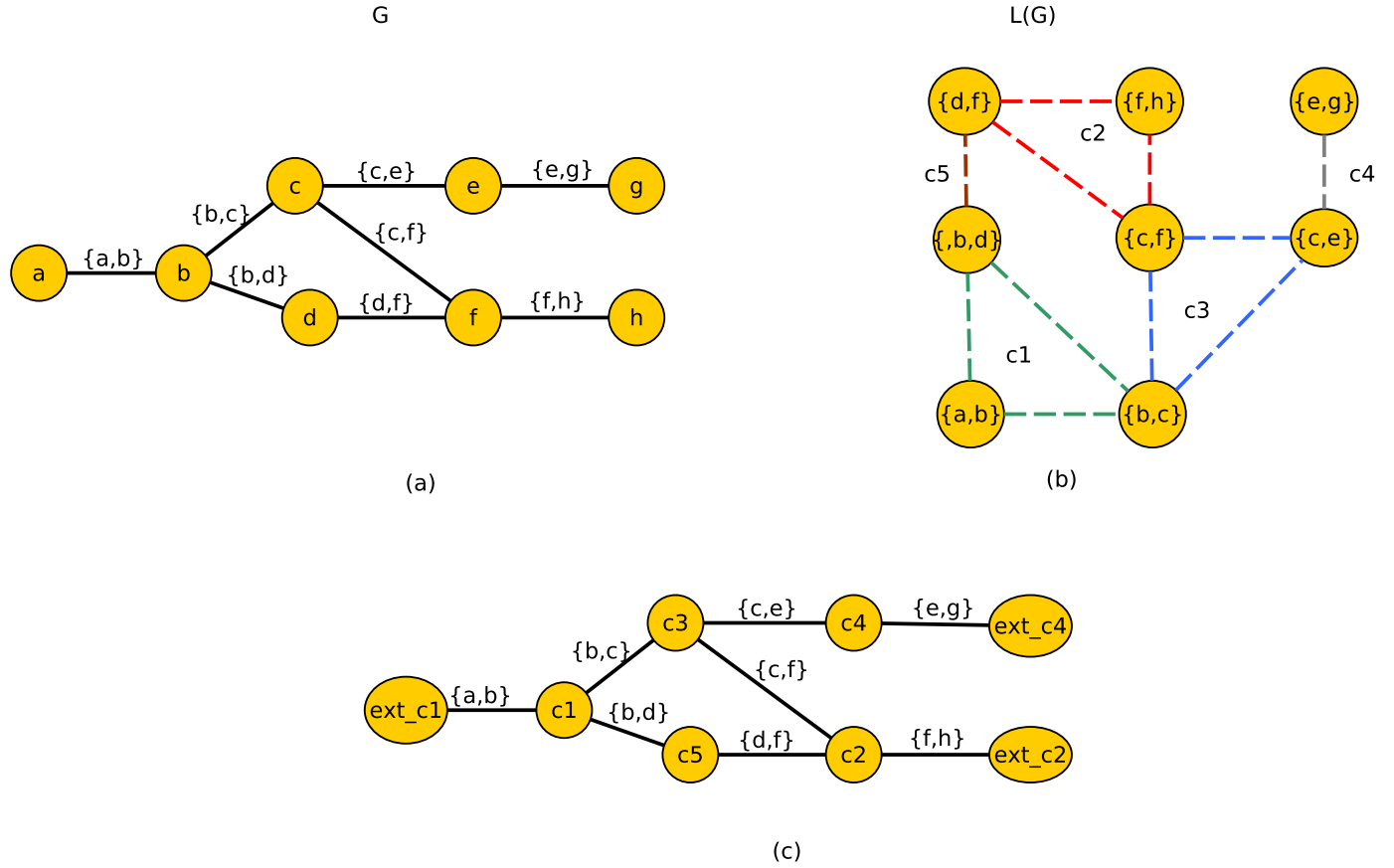


Figure 9: Construction de la topologie non orienté de G_c . (a) : réseau initial modélisé par G . (b) : line graphe de G . (c) : graphe G' reconstruit.

Nous allons prendre dans l'ordre chaque sommet de la séquence S et nous allons traiter ses arêtes pour trouver une bipartition correcte. Si une arête a déjà été traitée pour un sommet, elle n'est plus prise en compte dans les arêtes incidentes des sommets suivants dans la séquence S . Le traitement du sommet v_i va nécessiter 2^{d_i} opérations et le nombre total d'opérations est alors

$$2^{d_1} + 2^{d_2} + \dots + 2^{d_k}$$

Notre problème est de trouver une telle liste telle que cette somme est minimum. L'heuristique est, à chaque étape, de choisir le sommet tel que le nombre d'arêtes incidentes non encore traitées est minimum. Par exemple, on commence par le sommet v_1 comme sommet de degré minimum. À chaque étape, on prend le sommet v_i tel que son degré d_{v_i} est minimum.

Cependant, cette heuristique n'est pas optimale et voici un contre-exemple illustré dans la figure 10.

Soit le graphe H composé de 3 cliques K_5 et d'un sommet v ayant une arête incidente avec un seul sommet dans chaque clique K_5 . Le sommet v est de degré minimum. Nous considérons 2 séquences de sommets différents pour dénombrer les bipartitions. La première séquence suit l'heuristique et la seconde séquence débute par un sommet de K_5 de degré 4.

Soit $c(S)$ la somme des bipartitions possibles. En considérant l'heuristique, nous débutons par v .

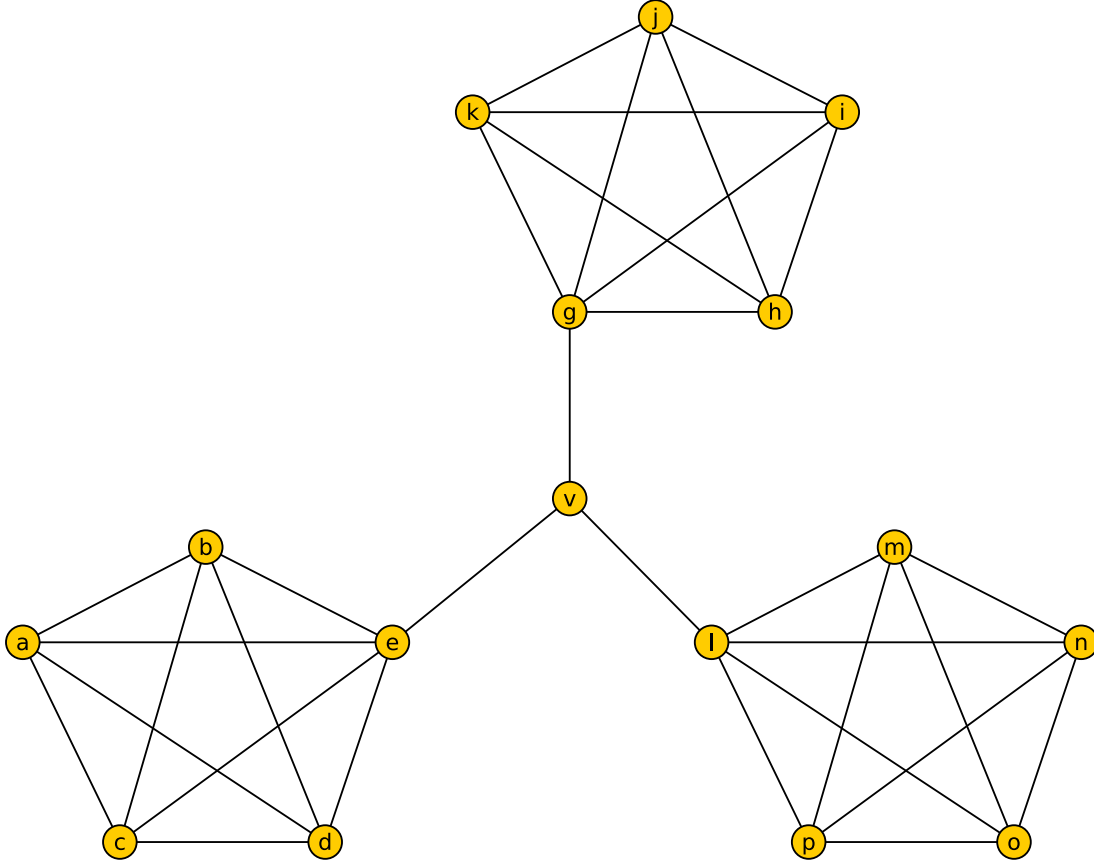


Figure 10: Un contre-exemple de l'heuristique choisie pour l'orientation des arêtes. On choisit un sommet de degré minimum dans une clique K_5 . Ensuite, on traite tous les sommets de la clique et on se sert du degré minimum pour choisir les sommets de la séquence. Puis on passe à une clique et on reprend le traitement jusqu'à ce qu'il ne reste plus d'arêtes dans une seule clique.

Puis le sommet suivant de degré minimum est un sommet de K_5 . On traite tous les sommets de cette clique K_5 avant de passer à une autre clique K_5 . Le nombre de bipartitions traitées avec l'heuristique est

$$c(S_H) = 2^3 + 3(2^4 + 2^3 + 2^2 + 2^1) = 98$$

En considérant la seconde séquence, on choisit le sommet de K_5 de degré minimum. On traite ce sommet en 2^4 bipartitions. Le sommet suivant est encore dans cette clique mais le nombre de bipartitions baisse à 2^3 . Après les deux sommets traités, le nombre de bipartitions est $2^4 + 2^3$. On répète le traitement des sommets jusqu'à ce que tous les sommets soient traités avant de passer à une autre clique K_5 . Dans cette clique, on reprend à nouveau la seconde séquence. Le nombre de bipartitions traitées avec la seconde séquence est

$$c(S_2) = 3(2^4 + 2^3 + 2^2 + 2^1) = 96$$

Nous remarquons que le nombre de bipartitions avec la seconde séquence est minimum. Cet exemple confirme que la solution de l'heuristique n'est pas minimale car il existe une autre séquence de choix de ces sommets (ici la seconde séquence) qui minimise le nombre de bipartitions.

Conclusion : l'algorithme d'orientation que nous avons proposé choisit le sommet de degré minimum à chaque étape de l'algorithme et cette solution n'est pas optimal. Nous conjecturons que la séquence S de traitement des sommets est *NP-complet*.

0.5 Conclusion du chapitre

Nous avons décrit la relation existante entre la matrice de corrélation et la notion de line-graphe. En effet, la matrice de corrélation appliquée à une valeur de seuil est la matrice d'adjacence d'un line-graphe si elle ne contient aucune erreur de corrélation. Dans le cas où elle possède des cases erronées, il est impossible de déterminer la *couverture de corrélation* du graphe de corrélation. Nous avons alors défini le problème *Proxi-Line* dont l'objectif est de trouver le line-graphe qui a le moins d'arêtes différentes avec le graphe de corrélation.

Ensuite, nous avons présenté les propriétés d'un line-graphe et les travaux existants dans la découverte de couverture en cliques sur des line-graphes. Nous avons retenu l'algorithme de Lehot [?] comme la base de notre algorithme de couverture parce qu'il attribue des états à tous les sommets à chaque étape de l'algorithme. Cette opération permet de connaître les sommets non couverts des sommets déjà couverts.

Dans le but de résoudre le problème *Proxi-Line*, nous proposons ainsi deux algorithmes.

Le premier algorithme est *l'algorithme de couverture* et il couvre tous les sommets par une ou deux cliques à partir des états des sommets. Nous avons distingué trois types d'états. En effet, un sommet couvert par une clique et qui possède des arêtes incidentes est à l'état 2. Un sommet couvert par une clique ou deux cliques et qui n'a aucune arête incidente est à l'état 1. Un sommet couvert par deux cliques ayant des arêtes incidentes ou un sommet ayant des arêtes incidentes qui ne forment pas une clique est à l'état par -1 . L'ensemble des sommets v à l'état -1 est l'ensemble \mathcal{C} de sommets à corriger. À la fin de l'algorithme, il retourne les cliques découvertes (couverture de corrélation \mathcal{CC}).

Le second algorithme proposé est *l'algorithme de correction*. Il se base sur l'ensemble \mathcal{C} et les cliques découvertes pendant l'algorithme de couverture. En effet, cet algorithme sélectionne chaque sommet $u \in \mathcal{C}$ et le corrige en procédant par une phase d'augmentation et de compression. La phase d'augmentation détermine les cliques de \mathcal{CC} contenant u , les cliques de \mathcal{CC} dont u partage une arête avec un sommet de la clique (cliques voisines), les cliques contractables (cliques de \mathcal{CC} dont l'intersection retourne le sommet u) et les cliques dépendantes (cliques contenant u dans lesquelles un des sommets partagent une arête avec une clique de la couverture de corrélation \mathcal{CC}). Puis elle effectue le produit cartésien de ces ensembles de cliques. Chaque élément de ce produit est noté π_1 ou π_2 . Quant à la phase de compression, elle sélectionne deux éléments du produit cartésien qu'elle note π_1 et π_2 , puis elle ajoute des arêtes à π_1 et π_2 dans l'ensemble des arêtes initiales du graphe de correction pour en faire des cliques. Elle crée aussi l'ensemble π_s des arêtes à supprimer afin que le sommet u ne soit couvert que par deux cliques. Les cliques π_1 et π_2 sont ajoutées à la couverture de corrélation \mathcal{CC} . La complexité des algorithmes de couverture et de correction est *pseudo-polynomial* en fonction du degré du graphe de corrélation.

Enfin, la dernière section présente la construction de la topologie du graphe racine et son orientation. Pour la construction de la topologie, nous nous servons principalement de la couverture de corrélation. En effet, chaque clique de cette couverture de corrélation est un sommet dans le graphe racine. Si l'intersection de deux cliques est non vide alors il existe une arête entre les

sommets correspondant à ces cliques dans le graphe racine. Concernant l'orientation des arêtes, nous nous servons de la fonction *Verif – correl* (section ??) qui teste tous les bipartitions possibles afin de trouvant les arcs incidents entrants et sortants d'un sommet du graphe racine. Nous avons montré qu'il existe un ordre de sommets qui réduit le nombre de bipartitions à tester. Toutefois cette solution n'est pas optimale.