

chapitre3: les graphes adjoints ou line graphes

Wilfried Ehounou

March 8^c 2018

Chapter 1

Découverte de topologie par les corrélations

1.1 Description du problème

1.1.1 Propriétés de la matrice de corrélation

Soit la matrice de corrélation M_c désignant les probabilités de corrélation entre les arcs d'un graphe G . Chaque case de la matrice M_c définit le pourcentage de corrélation entre deux arcs. Une corrélation, proche de 1, indique que les arcs partagent un sommet tandis qu'une valeur, qui tend vers 0, signifie qu'il existe aucun sommet en commun entre ces arcs.

Erreurs de corrélation

Dans la figure 1.1, les arcs a_2 et a_3 partagent le sommet d'extrémité initiale et leur valeur de corrélation est de 0.85. De même, il n'existe aucun sommet commun aux arcs a_1 et a_6 impliquant que leur corrélation tend vers 0 (corrélation égale à 0.21). Toutefois, certains arcs n'ayant aucun sommet en commun, ont des valeurs de corrélations très élevées (proche de 1). C'est le cas des arcs a_5 et a_7 dont leur valeur de corrélation est de 0.89 (figure 1.1.b).

Ce type de corrélations, appelé *erreurs de corrélation*, est dû aux lissages des signaux électriques causés par la présence d'onduleurs ou des équipements, dans des branches différentes, ont des profils de consommation identiques sur la période de temps. En général, ces profils de consommation sont quasi-constants avec des variations très faibles.

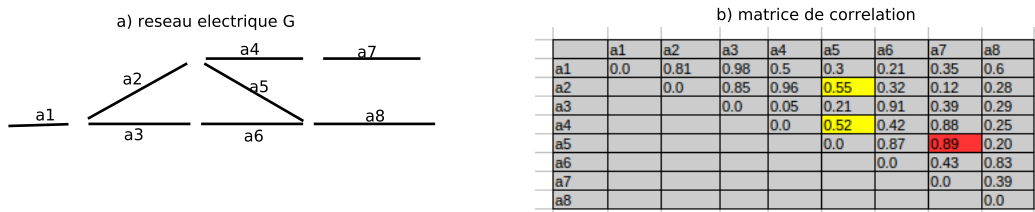


Figure 1.1: Le réseau électrique G et sa matrice de corrélation M_c . a) réseau électrique, b) matrice de corrélation associée. Les cases en jaunes et en rouges sont les corrélations erronées.

Matrice de corrélation : un graphe adjoint

Soient la valeur de seuil s et la matrice de corrélation $M_{c,s}$ appliquée avec ce seuil. Cette matrice $M_{c,s}$ est binaire avec 0 et 1 désignant respectivement aucune corrélation entre arcs (dont pas de sommets en commun) et l'existence d'un sommet entre ces arcs.

Chaque arc peut être représenté par un sommet et deux arcs ayant un sommet en commun (quelque soit son extrémité) ont leurs sommets respectifs adjacents. Cela implique que la matrice $M_{c,s}$ peut être considérée comme la matrice d'adjacence d'un graphe adjoint et ce graphe adjoint est celui sous-jacent au graphe non orienté du réseau électrique G .

Dans la littérature, le graphe adjoint est désigné par *linegraph*. Nous utilisons *line graphe* dans la suite du document pour indiquer un graphe adjoint.

Définition 1. Line graphe

Soit un graphe non orienté G . Le line graphe H d'un graphe G est un graphe dont chaque sommet correspond à une arête de G et deux sommets de H sont adjacents si et seulement si leurs arêtes respectives dans G ont un sommet en commun.

Avec G ayant n sommets et m arêtes, le graphe H a m sommets et E arêtes dont

$$E = \sum_{i=1}^n d_i(d_i - 1)/2$$

, d_i étant le degré de chaque sommet i de G .

Le graphe G est appelé le *graphe racine* de H . Un exemple de graphe racine et son line graphe est présenté dans la figure 1.2

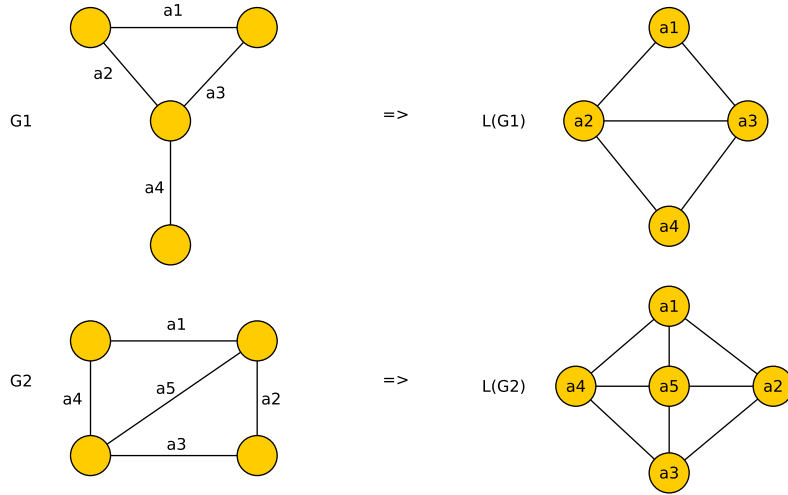


Figure 1.2: Le graphe racine G et son line graphe $L(G)$

Ce concept a été introduit par *Whitney* [?] qui montra que l'isomorphisme des arêtes implique l'isomorphisme pour des graphes connexes à l'exception des graphes K_3 (graphe triangle) et $K_{1,3}$ (graphe étoile).

Théorème 1. Si deux line graphes $L(G_1)$ et $L(G_2)$ sont isomorphes et connexes alors leurs graphes racines G_1 et G_2 sont aussi isomorphes sauf si $G_1 = K_3$ et $G_2 = K_{1,3}$.

Proposition 1. Le graphe étoile $K_{1,3}$ n'est pas un line graphe.

Proof. 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 1.3. Comme $L(C_4) = C_4$ par le théorème 1 et $L(K_{1,3} + e) = K_4 + e$ (voir figure 1.2), $L(H)$ 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 ($K_{1,3} = L(H)$). \square

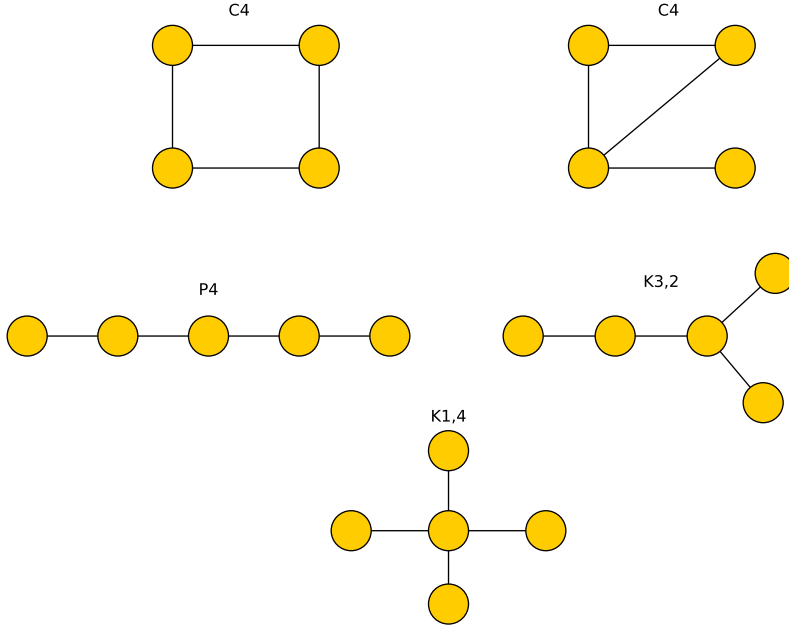


Figure 1.3: Les graphes racines possibles de $K_{1,3}$ de quatres arêtes

Beineke et Hemminger [?] affichent les neufs sous-graphes ne pouvant pas avoir de line graphes.

Théorème 2. *Aucun des neufs sous-graphes de la figure 1.4 ne peut être un sous graphe d'un line graphe G .*

Théorème 3. *Les arêtes d'un line graphe G sont partitionnées en sous graphes complets (Cliques) de tel sorte que chaque arête ne peut appartenir qu'à un seul sous graphe.*

Proof. Soit G le line graphe de H .

* de line graphe à des sous graphes : nous supposons que H est connexe. Alors les arêtes dans le graphe étoile de chaque sommet de H induit un clique dans G . Et chaque arête de G est couverte par exactement une clique. Pour chaque arete de H appartenant à deux graphes étoiles de deux sommets de H , aucun sommet de G est contenu dans plus de deux cliques.

** de sous graphes à un line graphe: étant donné une décomposition des arêtes de G en cliques C_1, C_2, \dots, C_n satisfaisant le théorème ci dessus, nous construisons le graphe H dont le line graphe est G . Les sommets de H correspondent à l'ensemble C de cliques de la décomposition avec l'ensemble U de sommets de G appartenant à une clique C_i uniquement. Alors $C \cup U$ est l'ensemble de sommets de H et deux de ses sommets sont adjacents s'il y a une intersection non vide. Alors H est un graphe d'intersection $\Omega(C \cup U)$. \square

Divers travaux existent dans la reconnaissance de line graphes et son graphe racine associé. Le premier algorithmé proposé est celui de LEHOT [?]. Il utilise la propriété définie par VAN ROOIJ and WILF [?] énonçant qu'un graphe G est un line graphe si G ne contient pas de sous

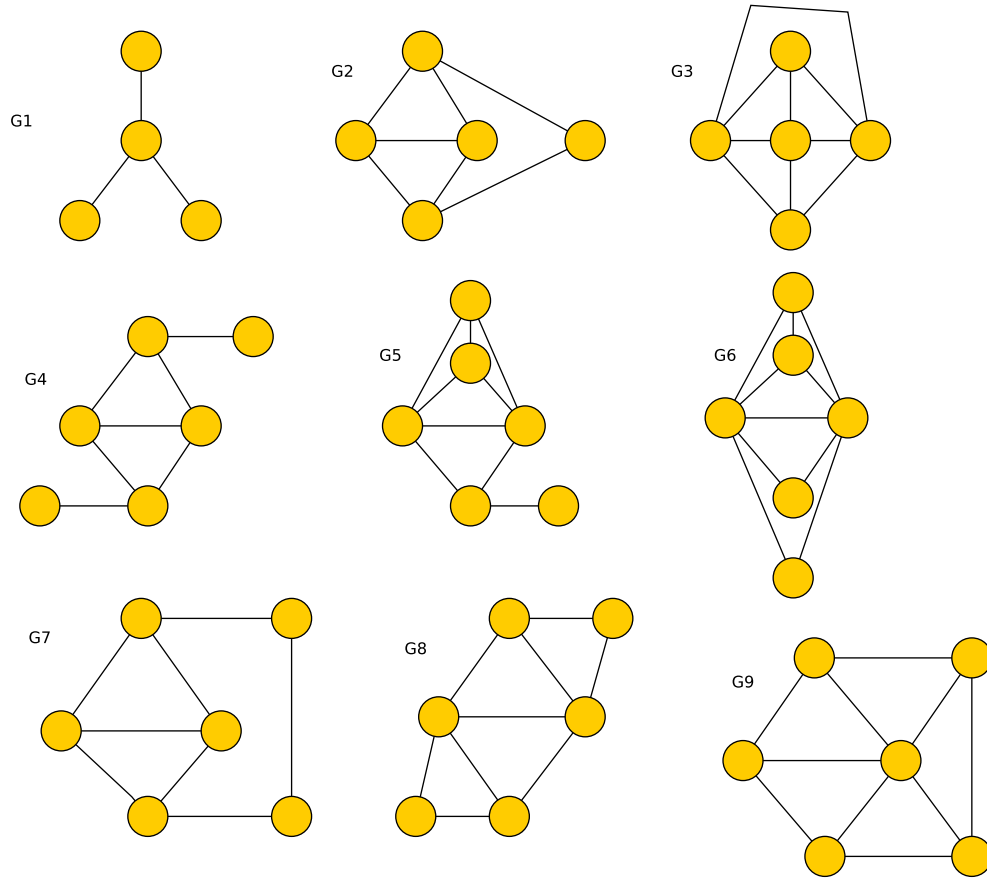


Figure 1.4: Line graphs et sous graphes interdits

graphe induit $K_{1,3}$ and si deux graphes triangles “Odd” ont une arête commune, le sous graphe induit par ces sommets est une clique K_4 . Son algorithme n’effectue que la reconnaissance de line graphe et la complexité est en $O(n) + E$ avec n le nombre de sommets de G et E le nombre d’arêtes de $L(G)$. Rappelons qu’une graphe triangle $\{a_1, a_2, a_3\} \subseteq V(L(G))$ est “Odd” s’il existe un sommet $e \in V(G)$ incident à au moins un des sommets $\{a_1, a_2, a_3\}$. Ce triangle est “Even” dans le cas contraire.

L’algorithme de ROUSSOPOULOS utilise une autre propriété des line graphes provenant des travaux de KRAUSZ [?]. Il affirme que G est un line graphe si les arêtes de G peuvent être partitionnées en cliques de tel sorte qu’aucun sommet ne soit couvert par deux cliques. Cette algorithme non seulement détecte si G est un line graphe mais fournit son graphe racine en temps linéaire $O(\max(\{m, n\}))$, avec m et n le nombre de sommets et d’arêtes respectivement.

Cette algorithme, basé sur la preuve de ORE [?] du théorème de Whitney [?] qui stipule que si deux graphes connexes avec plus de quatre sommets sont isomorphes en arêtes alors il existe exactement un isomorphisme en sommets qui induit un isomorphisme en arêtes. Proposé par Klauss Simon et Daniele Degiorgi [?], cette approche est une simplification du problème de reconnaissance de line graphes. Les deux précédents algorithmes utilisent les propriétés globales des line graphes tandis que celui ci utilisent uniquement les propriétés locales permettant une reconnaissance incrementale. Par ailleurs il permet de vérifier en temps linéaire si une modification locale (i.e l’ajout et la suppression) preserve les caractéristiques du line graphe.

Tous les algorithmes existants se limitent à un résultat négatif lorsque le graphe G n’est pas un line graphe. Les algorithmes, que nous soumettons, retournent la décomposition en cliques de G

si G est un line graphe ou dans le cas contraire, la décomposition du line graphe le plus proche de G .

1.1.2 Définition du problème

Nous définissons à présent le problème dont nous étudions. Soit le graphe $G_C = (V_C, E_C)$ obtenu entre les paires d'arcs du réseau de flot à déterminer.

Définition 2. Soient G et G' deux graphes ayant le même ensemble de sommets ordonné de la même façon. La distance de Hamming entre G et G' notée $DH(G, G')$ est le nombre d'éléments ayant une valeur différente dans chacune des deux matrices d'adjacence.

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

Définition 3. On appelle distance-line de G , notée $DL(G)$, la plus petite distance de Hamming entre G et G' , G' étant le line graphe d'un DAG 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.*

Proof.

□

Afin de traiter ce problème, nous utilisons l'approche de ROUSSOPOULOS [?] qui définit un line graphe par la couverture de ses arêtes par une clique et ses sommets par au moins deux cliques. On parle alors de *line-couverture*.

1.1.3 Définition de la line-couverture

Considérons un réseau électrique modélisé par un DAG et une matrice μ_c dans laquelle chaque ligne/colonne est associée à un arc du DAG. Une case de $\mu_c[i, j]$ contient le niveau de corrélation de mesures entre l'arc i et l'arc j , une valeur entre 0 et 1. Toutefois des erreurs de mesures peuvent apparaître.

Idéalement, si ces deux arcs partagent une extrémité en commun, soit l'extrémité initiale ou soit l'extrémité finale comme dans la figure 1.5, la valeur est proche de 1 sinon elle est proche de 0.

À partir de μ_c et d'une valeur de seuil choisie $s \in [0, 1]$, on considère la matrice M de même dimension que μ_c dans laquelle $M[i, j] = 1$ ssi $\mu_c[i, j] \geq s$, sinon $M[i, j] = 0$.

Soit $G_C = (V_C, E_C)$ le graphe dont M est la matrice d'adjacence. Idéalement, l'ensemble des liens ayant une extrémité commune forme une clique dans G_C .

Définition 4. Un graphe $G_C = (V_C, E_C)$ est un graphe de corrélation ssi il est le line graphe d'un DAG. L'ensemble \mathcal{C} est dit une **couverture de corrélation** de G_C .

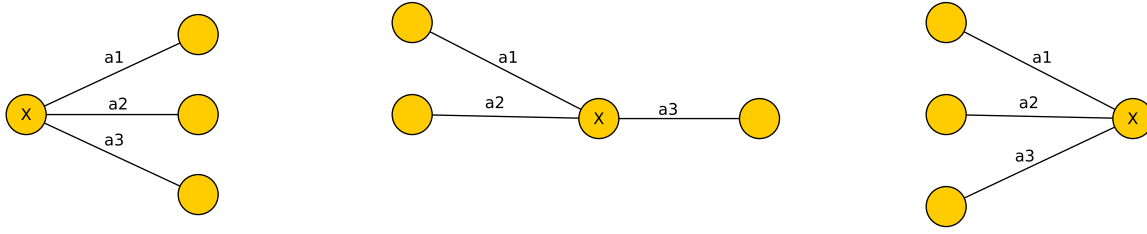


Figure 1.5: Sommet X partagé entre les arêtes. De la gauche vers la droite : sommet entrant, à droite sommet intermédiaire et sommet sortant

Détermination du graphe de corrélation et sa line-couverture

Définition 5. Une ambiguïté est un graphe isomorphe à l'un des graphes de la figure 1.6. Le sommet X est appelé le point d'ambiguïté.

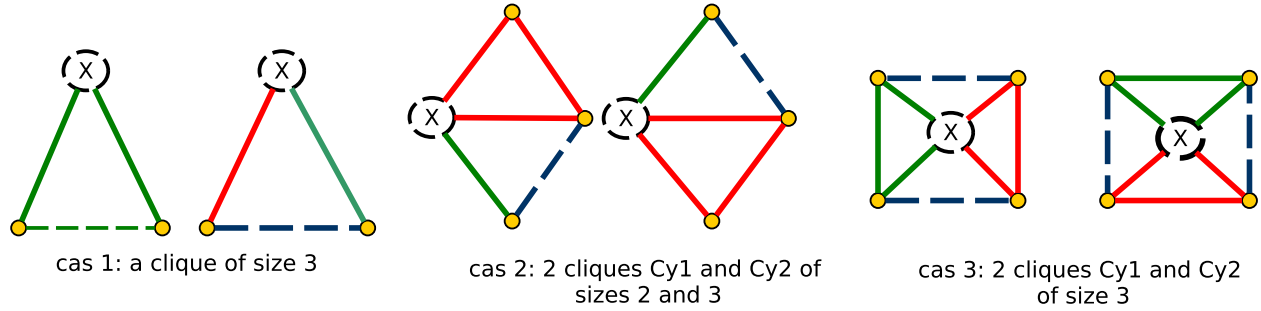


Figure 1.6: Configurations possibles d'une ambiguïté au sommet X

Lemme 1. Si un graphe de corrélation G_C admet deux couvertures de corrélation, alors il existe au moins un sommet u de G_C tel que $G_C[\{u\} \cup \Gamma_{G_C}(u)]$ est une ambiguïté dont u est le point.

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

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{C} , 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 correspondant 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 1.7. Le sommet v est bien le point d'une ambiguïté isomorphe à $G_C[\{u\} \cup \Gamma_{G_C}(u)]$. \square

On peut déduire le corollaire suivant :

Corollaire 1. Si un graphe de corrélation admet deux couvertures de corrélation, alors il est isomorphe à l'un des graphes de la figure 1.7.

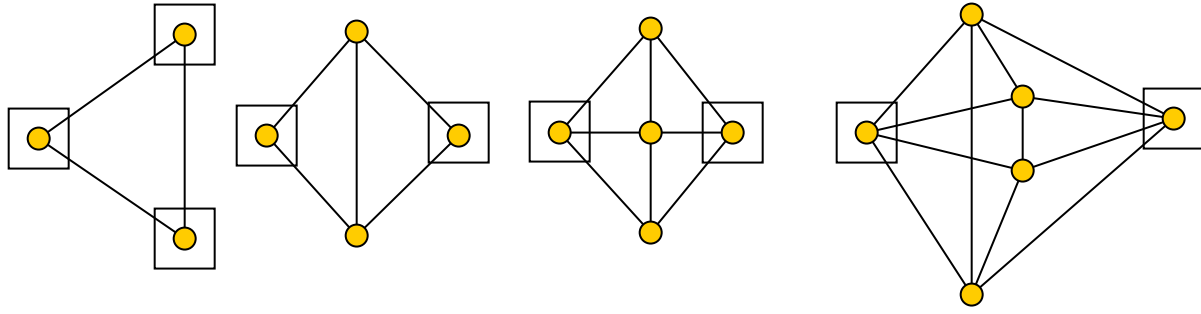


Figure 1.7: Graphes possibles avec deux couvertures avec les points d'ambiguïtés

En Effet, si $G_C[\{u\} \cup \Gamma_{G_C}(u)]$ est 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_C . 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_C 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 1.7), doivent être dans la même situation dans une autre ambiguïté à laquelle elles appartiennent. Ces contraintes font que si un graphe contient une ambiguïté induite, alors il ne peut être que dans un cas de la figure 1.7.

Définition 6. Soit G un graphe et u un sommet de G . Une partition de $\Gamma_G(v)$ en deux cliques C_{u1}, C_{u2} est **cohérente** ssi chaque sommet v de C_{u1} (resp C_{u2}) a au plus un voisin dans C_{u2} (resp C_{u1}).

Le résultat suivant est un corollaire direct de la preuve du lemme 2

Lemme 2. Soit G_C un line graphe, u un sommet de G_C et une partition cohérente de $u \cup \Gamma_{G_C}(v)$ en deux cliques C_{u1}, C_{u2} . Si l'une de ces deux cliques est de cardinal supérieur ou égal à 4, alors cette partition cohérente est unique.

Étant donné une partition cohérente C_{u1}, C_{u2} pour un sommet u de G_C , la fiabilité de cette cohérence est $F(C_{u1}, C_{u2}) = \min_{[x,y] \in E(C_{u1} \cup C_{u2})} \mu_C([x, y])$

Définition 7. Une **line-couverture** d'un graphe non orienté connexe G_C est un ensemble de cliques maximales de G_C telles que chaque sommet de G_C appartient à une ou deux de ces cliques et que chaque arête de G_C soit couverte par exactement une de ces cliques.

Afin de traiter ce problème, nous considérons un algorithme consistant à l'exécution consécutive de deux algorithmes : un algorithme de couverture et un algorithme de correction. L'algorithme de couverture est différent de celui de LEHOT [?] car notre objectif est de fournir la line-couverture si G est un line graphe ou dans le cas contraire, de couvrir le plus d'arêtes et de sommets de G par des cliques respectant les contraintes afin de minimiser autant que possible le nombre de sommets et d'arêtes à corriger ensuite.

1.2 Algorithmes de découverte de topologie

Il a été montré qu'un graphe G_C est un line graphe si et seulement si il admet une line-couverture. Basée sur cette idée et sur des sous-graphes exclus, un algorithme linéaire en fonction du nombre

d'arêtes a été proposé pour vérifier si un graphe est un line graphe et si oui en fournir la line-couverture. Cette algorithme est **l'algorithme de couverture**.

1.2.1 Algorithme de couverture

Étant donné un graphe $G_C = (V_C, E_C)$, l'algorithme suivant a pour but de déterminer la line-couverture si elle existe (notamment si la matrice de corrélation est sans erreur). Dans le cas contraire, l'algorithme couvrira autant que possible des sommets de G_C par une ou deux cliques. Considérons C un ensemble de cliques de G_C initialement vide et chaque sommet $v \in V_C$ a un état $Cliq(v)$ initialisé à 0.

Chaque sommet v a cinq états possibles:

- 0 : il n'est couvert par aucune clique. Il correspond aussi à l'état initial des sommets.
- 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 .
- 2 : le sommet est couvert par une clique et peut être couvert par une seconde clique.
- 3 : le sommet v est un sommet ambigu.
- -1 : le sommet v est couvert par plus de deux cliques. Il doit être corrigé par l'algorithme de correction.

Pour tout sommet v de degré minimum tel qu'il n'appartient à aucune clique ou qu'il est un sommet ambigu, S'il existe une partition cohérente de ce sommet et son voisinage $\{v\} \cup \Gamma_G(v)$ en deux cliques C_1, C_2 , alors ces deux cliques sont contenues dans la line-couverture. Les sommets v et u (avec $u \neq v$), appartenant à C_1 ou C_2 , a son état modifié de tel sorte que:

- $Cliq(v) = 1$ 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 au sommet v , son état est à $Cliq(v) = -1$.

Algorithme 1 couverture_cliques

DEBUT

1. **Si** G est isomorphe à un graphe double (voir figure 1.7), **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 minimum4. **Si** $\{u\} \cup \Gamma_G(u)$ 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. $\epsilon_u = E(G[C_1]) \cup E(G[C_2])$ 11. **Pour tout** $w \in \Gamma_G(u)$ **Faire**12. $\alpha(w) = card\{[w, x] \in E - \epsilon_u\}$ 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 1.7 admet deux line-couvertures, souvent isomorphes, mais (au plus) une seule de ces line-couvertures correspond au DAG du réseau électrique sous-jacent. Dans ce cas, on utilise les mesures de corrélation de la matrice de mesures μ_C afin de déterminer la plus probable entre les deux.

² : le sommet u choisi (s'il existe) ne sera pas prioritairement un sommettel 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 est un point d'ambiguïté. Dans ce cas, G_C est une union de composantes connexes isomorphes à un des graphes de la figure 1.7. Dans ce cas, n'importe quel choix conduit à une décomposition correcte. Dans tous les autres cas de sommet choisi u , le lemme 2 montre que le choix de décomposition est unique si G est un line graphe.

La partition cohérente se détermine au moyen des mesures physiques selon la loi de conservation des noeuds. En effet, tous les sommets d'une clique dans un line graphe sont des arêtes qui concourent un sommet du réseau électrique. Par les lois d'électricités, la différence entre les flots entrant et sortant dans ce sommet correspond aux pertes par effets joules EJ . Ainsi les deux cliques s'obtiennent si cette différence est inférieure aux pertes joules. Dans notre algorithme de couverture, nous supposons que la partition cohérente proposée est toujours exacte. Cela signifie que nous connaissons la valeur des pertes par effets joules dans le réseau électrique. Dans le cas ou ce paramètre est inconnu, quel est la valeur minimum de ces pertes notée ϵ à partir duquel la partition est toujours exacte? En d'autres termes, quel est la valeur ϵ pour laquelle la décision de l'ORACLE est un *vrai positif*, L'ORACLE étant la fonction retournant les partitions cohérentes. La decision de l'ORACLE est la valeur ϵ .

Impact des pertes par effets joules

Nous considérons que le réseau électrique est connu et que sa matrice de corrélation est correcte. Dans le but d'étudier le taux minimum des pertes par effets joules notée *epsilon* dans le réseau afin que la partition cohérente proposée (appelée ORACLE) soit toujours correcte, nous réalisons deux expériences:

- La première expérience consiste à fixer la décision de l'ORACLE $\epsilon = cte$ tout en générant des mesures dans le graphe. Ces mesures subissent les variations des pertes par effets joules de 0 à 1 par pas de 0.125 $EJ = [0, 1]$. Nous etudions l'évolution de la distance de Hamming (ou coefficient de similarité) en fonction de pertes joules EJ .
- La deuxième expérience consiste à faire varier la décision de l'ORACLE. Nous etudions les variations des pertes en fonctions de la decision de l'ORACLE. Les pertes par effets joules EJ de 0 à 1 par pas de 0.125.

Rappelons que $EJ = 0.1$ signifie qu'il existe une différence de flot par grandeurs entre les arcs extérieures (sortantes) et intérieures (entrantes) à chaque sommet. Ainsi $EJ = 0$ signifiant qu'il n'existe aucune perte tandis que $EJ = 1$ signifiant qu'il n'existe aucuns flots entre les arcs entrants et sortants d'un sommet du réseau de flots.

expérience 1 : On fixe la décision de l'ORACLE à $\epsilon = 0,75$. La figure 1.8 resume les variations de l'ORACLE. Le fait que la courbe de décision de l'ORACLE est décroissante conforte notre

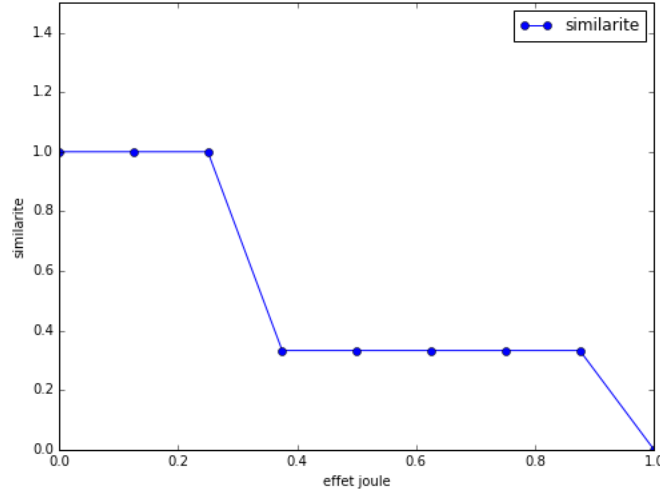


Figure 1.8: La distance de Hamming en fonction des pertes par *effets joules* pour $\epsilon = 0.75$

hypothèse selon laquelle il n'existe aucuns flots entre les arcs entrants et sortants d'un sommet lorsque les pertes par effets joules sont égales à 1. En effet, pour toute valeur de pertes par effets joules $EJ = [0, 0.3]$, le graphe proposé est identique au graphe du réseau électrique. Par contre, l'ORACLE se trompe deux sur trois sur les cliques fournies pour $EJ =]0.3, 0.9]$ parce que la distance de Hamming est de 0.38.

expérience 2 : Ici, on suppose la similarité égale à 1 et on cherche les variations de l'ORACLE ϵ en fonction des pertes par *effets joules*. Les pertes par *effets joules* varient de $[0, 1]$ par pas de 0.125 et nous distinguons 8 intervalles $[0, 0.125]$, $]0.125, 0.250]$, $]0.250, 0.375]$, $]0.375, 0.5]$, $]0.5, 0.625]$, $]0.625, 0.75]$, $]0.75, 0.875]$, $]0.875, 1]$. Pour chaque valeur ϵ , on compte les intervalles $EJ_x, x \in [1, 8]$ dans lesquelles la distance de Hamming est égale à 1 noté $X(\epsilon)$. Ainsi $X(\epsilon = 0.3) = 8$ signifie que les pertes par effets joules varient de 0 à 1 ($EJ = [0, 1]$) et $X(\epsilon = 0.3) = 2$ correspond à une variation de EJ sur l'intervalle $[0, 0.250]$. On crée ainsi la distribution des décisions de l'ORACLE en fonction des pertes par effets joules. La figure 1.9 résume cette distribution qui varie de 1 (correspondant à une variation sur $[0, 0.125]$) à 8 (correspondant à une variation sur $[0, 1]$). la courbe de cette distribution est constante de l'intervalle $\epsilon = [0, 0.125]$ puis décroissante de l'intervalle $\epsilon = [0.125, 1]$ et cette pente est très accentuée dans l'intervalle $\epsilon = [0.8, 1]$. Au delà de $\epsilon > 0.8$, les pertes EJ varient dans l'intervalle $EJ = [0, 0.2]$. On en conclut que le meilleur intervalle est $\epsilon = [0.8, 1]$ pour produire des graphes de distances de Hamming égale à 1 en présence de pertes par *effets joules* de 20%.

Ces deux expériences montrent que la décision de l'ORACLE (ϵ) a une relation inverse avec les pertes par effets joules (EJ). En effet, plus epsilon est petit plus les pertes par effets joules sont grandes et plus les decisions de l'ORACLE sont erronées.

$$\epsilon > 1 - EJ \quad (1.1)$$

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. On rappelle que l'algorithme de Lehot [?] a une complexité en $O(m)$.

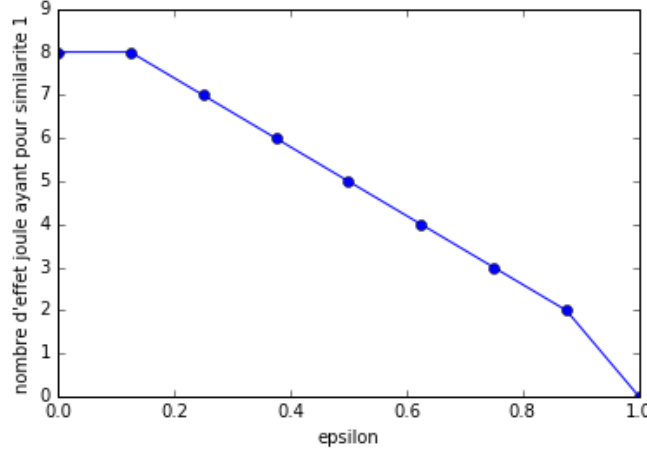


Figure 1.9: relation inverse entre epsilon et EJ: 8 en ordonné correspond à $[0,1]$, 7 à $[0, 0.875]$

Le facteur $\Delta(G_C)$ dans notre algorithme, du à la recherche de la decomposition en deux cliques C_1 et C_2 est nécessaire à la determination de l'ensemble des sommets v tels que $Clq(v) = -1$, en nombre le plus petit possible.

En conclusion, si le graphe G_C est bien un line graphe, l'algorithme trouvera une décomposition du voisinage d'un sommet en une ou deux cliques de façon unique (voir les lemmes précédents). Une fois ce sommet supprimé, le graphe restant est toujours un line graphe, et la propriété se propage. Donc, si G_C est un line graphe, l'algorithme en trouvera toujours la couverture de corrélation unique.

1.2.2 Algorithme de correction

Si le graphe $G_C = (V_C, E_C)$ est un graphe de corrélation alors l'algorithme de couverture determine sa line couverture \mathcal{C} . En effet, par récurrence sur l'ensemble des sommets, on montre à chaque étape qu'il existe un sommet non encore couvert qui soit :

- est couvert par une clique appartenant à \mathcal{C} et son voisinage restant et lui peuvent être convertis par une nouvelle clique.
- n'est couvert par aucune clique de \mathcal{C} et son voisinage restant et lui peuvent être couverts par une ou deux nouvelles cliques.

Dans le cas où la line-couverture de G_C ne peut être fournie à cause des erreurs de corrélations, nous avons des sommets couverts par soit aucune clique ou soit par plus de deux cliques. Ces sommets, labellisés à -1 , forment l'ensemble $sommets_1 = \{\exists z \in V, Clq(z) = -1\}$ et sont appelés *sommets à corriger*.

Nous proposons l'*algorithme de correction* qui va modifier l'ensemble initial E_C par ajout et suppression d'arêtes dans le but d'obtenir un *line graphe*. Nous allons considérer un ordre $O_z = [z_1, z_2, \dots, z_t]$ de sommets de $sommets_1$ qui correspond au mode de sélection de ceux-ci pendant la phase de correction. Il en suit que l'ordre a une influence sur le line graphe fourni parce que la correction modifie le voisinage des sommets. Il est montré dans la chapitre ??.

Soit E_C^i l'ensemble des arêtes de G_C après le traitement des $i - 1$ premiers sommets dans l'ordre O_z . De même, on note \mathcal{C}^i l'ensemble des cliques de G_C à l'étape i et donc $E_C^1 = E_C$ et $\mathcal{C} = \mathcal{C}^1$.

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

$$\mathcal{C}(z) = \{C_i, i \in [1, k] \mid |C_i| \geq 3 \ \& \ C_i \in \mathcal{C}^i\} \quad (1.2)$$

$$S(z) = \{v \in \Gamma_G(z) \mid \{v, z\} \in \mathcal{C}^i\} \cup \{v \in \Gamma_G(z) \mid \nexists C \in \mathcal{C}^i, [z, v] \in E_C(C)\} \quad (1.3)$$

Définition 8. Deux cliques C et C' de $\mathcal{C}(z)$ 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{C} . Un ensemble de cliques de \mathcal{C} est contractable si tous les cliques sont deux à deux contractables.

Définition 9. Une clique $C \in \mathcal{C}_i$ est voisine de z si $C \notin \mathcal{C}(z)$ et $\text{card}(C \cap S(z)) \geq 2$. La dépendance d'une clique C voisine de z est l'ensemble $D_z(C) \subset \mathcal{C}(z)$ tel que $C' \in D_z(C)$ si et seulement si $C' \cap C \cap \Gamma_G(z) \neq \emptyset$.

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

$$\text{voisine}(z) = \{C \in \mathcal{C}^i \mid C \notin \mathcal{C}(z) \ \& \ \text{card}(C \cap S(z)) \geq 2\} \quad (1.4)$$

$$D_z(C) = \{C' \in \mathcal{C}(z) \mid C' \cap C \cap \Gamma_G(z) \neq \emptyset\} \quad (1.5)$$

On appelle *augmentation* du sommet z l'union d'une clique augmentante C pour z et d'une contraction de cliques de $D_z(C)$.

Un exemple de clique augmentante $C1$ pour le sommet z est donné dans la figure 1.10, avec $D_z(C1) = \{C2\}$. Par contre, la clique $C6$ ne peut pas être augmentante à cause de l'appartenance de l'arête $[u, v]$ à la clique $C7$ de \mathcal{C}^i . Ce qui rend impossible toute contraction entre $C6$ et $C4$

Définition 10. On appelle compression du sommet z un triplet $(\pi_1, \pi_2 \text{ et } \pi_s)$ défini par :

- π_1 (resp. π_2) peut être chacun d'une des formes suivantes :

1. l'union de z , d'un sous-ensemble C_1 (resp. C_2) de cliques de $\mathcal{C}(z)$ tel que toute paire C et C' de C_1 (resp. C_2) est contractable et d'un sous-ensemble S_1 (resp. S_2) de sommets $v \in S(z)$ n'appartenant à aucune clique de C_1 (resp. C_2) tel que

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

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

2. une augmentation du sommet z

- π_1 et π_2 ne peuvent pas être simultanément réduits à $\{z\}$ et $\pi_1 \cap \pi_2 = \{z\}$,
- $\pi_s = \Gamma_G(z) - ((\pi_1 \cap \Gamma_G(z)) \cup (\pi_2 \cap \Gamma_G(z)))$ tel que l'ensemble des arêtes $\{[z, v] \in E_C^i : v \in \pi_s\}$ n'est pas déconnectant.

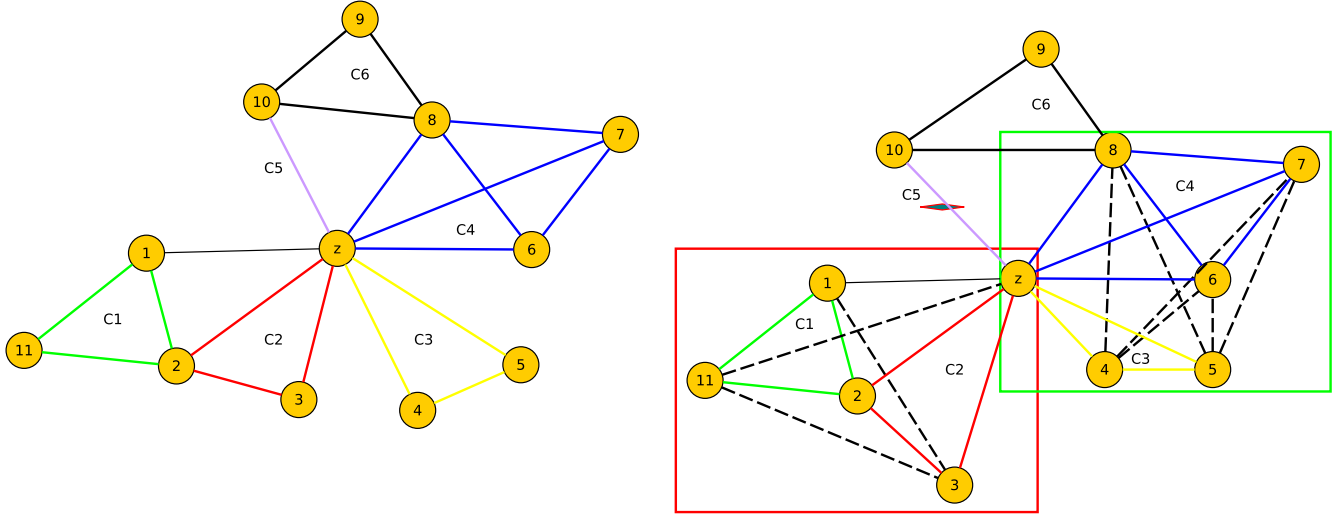


Figure 1.10: un exemple de compression de cliques

- le triplet $\pi_1 \cap \Gamma_G(z)$, $\pi_2 \cap \Gamma_G(z)$, $\pi_s \cap \Gamma_G(z)$ est une 3-partition de $\Gamma_G(z)$

Il existe toujours une telle compression, ne serait-ce que $\pi_1 = \{z\} \cup C_i \in \mathcal{C}(z)$, $\pi_2 = \emptyset$, $\pi_s = \gamma_G(z) - (\gamma_G(z) \cup C_i)$ si $\mathcal{C}(z)$ n'est pas vide. Sinon, $\pi_1 = \{z\} \cup \{v \in \gamma_G(z)\}$, $\pi_2 = \emptyset$, $\pi_s = \gamma_G(z) - \{v\}$ est aussi une compression. Un exemple de compression est aussi donné dans la figure 1.10. Le coût $c(T)$ d'une compression π_1, π_2, π_s est défini par :

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

Dans l'exemple de la figure 1.10(a), autour d'un sommet z , l'ensemble $\mathcal{C}(z)$ contient les cliques $C2$, $C3$, $C4$ et $C5$. Les cliques $C5$ et $C4$ ne sont pas contractables, à cause de l'existence de $C6$ dans \mathcal{C}_i . La clique $C1$ est voisine de z et $D(C1) = \{C2\}$. L'exemple de compression qui est donné dans la figure 1.10(b) est $\pi_1 = C1 \cup C2$ (une augmentation), $\pi_2 = C3 \cup C4$ (ces deux cliques étant contractables), et $\pi_s = \{x\}$. Le coût de cette compression est 10, 10 étant le nombre d'arêtes en pointillé plus l'arête supprimée $[x, z]$.

Soit $\text{Cost}(z)$ le coût minimum d'une compression de z . Le but est de modifier G_C afin que z 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). 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 $\{[z, v] \in E_C^i : v \in \pi_s\}$.

Des lors, le sommet z appartient aux deux cliques π_1 et π_2 . On procède alors aux mises à jour suivantes pour obtenir \mathcal{C}^{i+1} et E_C^{i+1} :

- supprimer toutes les cliques \mathcal{C}_z couvertes par π_1 dans \mathcal{C}^i .
- supprimer toutes les cliques \mathcal{C}_z couvertes par π_2 dans \mathcal{C}^i .
- supprimer toutes les cliques de cardinalité 2 couvertes par π_1 et π_2 dans \mathcal{C}^i .

- ajouter π_1 et π_2 dans \mathcal{C}^i , supprimer de E_C^{i+1} toutes les arêtes $\{[z, v] \in E_C^i : v \in \pi_S\}$.
- 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 une application d'une compression, Soit \mathcal{C}^{i+1} l'ensemble obtenu à partir de \mathcal{C}^i après mise à jour selon cette application.*

- *Tout sommet de G_C couvert par une ou deux cliques dans \mathcal{C}^i le reste dans \mathcal{C}^{i+1} .*
- *Toute arête couverte par une et une seule clique dans \mathcal{C}^i et qui n'est pas supprimée le reste dans \mathcal{C}^{i+1} .*
- *Le sommet z est couvert par une ou deux cliques dans \mathcal{C}^{i+1} (le nombre de sommets ainsi couverts augmente de 1 par rapport à celui dans \mathcal{C}^i).*

Ainsi, pour chaque sommet z_i pris dans l'ordre O_z , on considère une compression de coût minimum c_m^i et on l'applique. La propriété ci-dessus garantit qu'à l'afin du processus, on obtient un graphe de corrélation $G_C^t = (V, E_C^t)$ dont l'ensemble \mathcal{C} modifié est une couverture de corrélation. La distance-line vérifie

$$DL(G_C^0, G_C^t) \leq \sum_{1 \leq i \leq t} c_m^i$$

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 $c_m^i = 0$.

1.2.3 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 un ordre dans lequel seront pris les sommets $z \in \text{sommets}_1$. Considérons un graphe de corrélation G n'étant pas isomorphe à un graphe de la figure 1.7. On dira que G est non-ambigu.

Deux arêtes $[u, v]$ et $[u', v']$ de G seront dit **clique-indépendantes** si et seulement si il n'existe pas de cliques C dans la ligne couverture de G 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 $G = (V, E)$, alors il existe une exécution de l'algorithme qui transforme G' en G*

1.3 Détermination de la topologie du réseau énergétique

Soient \mathcal{C} , l'ensemble des cliques de la ligne graphe G_C et G le graphe racine de G_C .

Chaque clique $C_k \in \mathcal{C}$ correspond à un sommet du graphe racine G .

Soit $u = \{\text{"ENTRANT"}, \text{"SORTANT"}, \text{"NONE"}\}$, l'ensemble des marquages de chaque arête tel que les étiquettes "ENTRANT" et "SORTANT" sont associées respectivement à l'arête a_i entrante et sortante du sommet C_k .

Soient S_1 et S_2 l'ensemble des arcs entrants et sortantes du sommet C_k .

$$S_\alpha = \{S_{\alpha,i}, \forall i \leq 2^{\text{card}(C_k)+1}\} \quad (1.6)$$

avec i , le nombre de sous-ensemble de C_k et $\alpha = \{1, 2\}$. Le sous-ensemble $S_{\alpha,i}$ peut être l'ensemble vide \emptyset .

Définition 11. Soit la clique C_k de la couverture C . Un couple (S_1, S_2) de sous-ensembles de C_k est **valide** si

- $S_1 \neq S_2$
- $S_1 \cup S_2 = C_k$
- $S_1 \cap S_2 = \emptyset$

Définition 12. Soit une fonction binaire *ORACLE* définit de $S_1 \times S_2 \rightarrow \{0, 1\}$, avec le couple valide (S_1, S_2) . La fonction *ORACLE* renvoie 1 si et seulement si la loi de conservation autour du sommet $C_k \in G$ est respectée c'est-à-dire :

$$\forall a_i \in S_1, \forall a_j \in S_2, \sum_{a_i \in S_1} gp(a_i) - \sum_{a_j \in S_2} gp(a_j) \leq \epsilon \quad (1.7)$$

avec ϵ les pertes par effets joules, $gp(a_i)$ le flot dans l'arc $a_i \in E(G)$.

Propriété 2. Si $ORACLE(S_1, S_2) = 1$ alors l'ensemble S_1 est l'ensemble des arcs entrants et S_2 l'ensemble des arcs sortants du sommet $C_k \in G$.

Les arcs de S_1 et S_2 ne concourent pas à un sommet C_k lorsque $ORACLE(S_1, S_2) = 0$.

Théorème 4. Si un sommet $a_i \in G_C$ appartient à deux cliques $C_{k_1}, C_{k_2} \in \mathcal{C}$ alors l'arc $a_i \in E(G)$ est soit entrant de C_{k_1} et sortant de C_{k_2} ou soit entrant de C_{k_2} et sortant de C_{k_1} .

Proof. D'après la définition 7, le sommet a_i appartient à deux cliques C_{k_1} et C_{k_2} au maximum. Cela implique qu'il existe une arête entre les sommets C_{k_1} et C_{k_2} dans G . Soient $S_1, S_2 \subset C_{k_1}$ et $S_1, S_2 \subset C_{k_2}$ telles que $ORACLE(S_1, S_2) = 1$ et $ORACLE(S_3, S_4) = 1$. Comme $S_1 \cap S_2 = \emptyset$, $S_3 \cap S_4 = \emptyset$ et aussi $C_{k_1} \cap C_{k_2} = \{a_i\}$, on a quatre choix possibles : $a_i \in S_1 \cap S_3$, $a_i \in S_1 \cap S_4$, $a_i \in S_2 \cap S_3$, $a_i \in S_2 \cap S_4$.

Si a_i est entrant et sortant à C_{k_1} et C_{k_2} alors $a_i \in S_1 \cap S_3$ ou $a_i \in S_2 \cap S_4$. On en déduit qu'il existe une boucle sur le sommet C_{k_1} et C_{k_2} . Cela est impossible parce que G est un DAG.

Alors a_i est entrant à $C_{k_1} \Rightarrow a_i \in S_2 \cap S_3$ ou a_i sortant à $C_{k_2} \Rightarrow a_i \in S_1 \cap S_4$. \square

Soit A l'ensemble des cliques C_j tel que $card(C_j) = \min\{card(C_k), C_k \in \mathcal{C}\}$.

Soit la fonction f définie sur $V(G_C) \times A \rightarrow \{0, 1\}$ retournant 1 si le sommet a_i^u ne possède aucune étiquette c'est-à-dire labellisé à *None*.

$$f(a_i^u, C_j) = \begin{cases} 1 & \text{si } M(a_i^u, C_j) \neq \text{None} \\ 0 & \text{sinon} \end{cases}$$

La fonction MA calcule le nombre de sommets labellisé dans une clique C_k .

$$MA(C_k) = \sum_{a_i^u \in C_k} f(a_i^u, C_k)$$

Soient la clique C_m telle que $MA(C_m) = \max\{MA(C_j), C_j \in A\}$ et B l'ensemble des sommets de la clique C_m tel que $M(C_m, a_j^v) = \text{None}, \forall a_j^v \in B$.

Algorithme 2 Decouverte_graphe_racine

DEBUT

1. Initialisation des étiquettes des sommets a_i de G_C , et $C_k \in \mathcal{C}$ $M(C_k, a_i^u) = \text{None}$ 2. **Tant que** il existe une arête de G_C ($E(G_C)$) **Faire**3. choisir $A = \{C_j, \text{card}(C_j) = \min\{\text{card}(C_k), C_k \in \mathcal{C}\}\}$ 4. choisir C_k tel que $MA(C_k) = \max\{MA(C_j), C_j \in A\}$ 5. **Si** il existe S_1 et S_2 tel que $S_1 \cup S_2 = \emptyset$ et $S_1 \cup S_2 = C_k$ et $ORACLE(S_1, S_2, M) = 1$ 6. **alors**7. $S_1^k = \text{sommets_marques}(S_1, M)^{(1)}$;8. $S_2^k = \text{sommets_marques}(S_2, M)$;9. **Pour tout** $a_i^K \in S_1 - S_1^k$ **Faire**10. $M(C_k, a_i^K) = \text{"ENTRANT"}$ 11. **Pour tout** $a_i^K \in S_2 - S_2^k$ **Faire**12. $M(C_k, a_i^K) = \text{"SORTANT"}$ 13. $E(G_C) = E(G_C) - E(G_C[C_k])$ 14. $\mathcal{C} = \mathcal{C} - C_k$ 15. **FinSi**16. **FinTant que**17. **Pour tout** $a_i^u \in V(G_C)$ **Faire**18. $C_1, C_2 = \text{Couverture_Cliques}(a_i^u, M)^{(2)}$ 19. **Si** $C_1 \neq \emptyset$ et $C_2 \neq \emptyset$ **Alors**20. **Si** $M(C_1, a_i^u) == \text{"SORTANT"}$ **Alors**21. $\text{Mat}(C_1) += C_2$ ⁽³⁾22. **Fin Si**23. **Si** $M(C_2, a_i^u) == \text{"SORTANT"}$ **Alors**24. $\text{Mat}(C_2) += C_1$ ⁽³⁾25. **Fin Si**26. **Fin si**27. **Si** $C_1 \neq \emptyset$ et $C_2 == \emptyset$ **Alors**28. **Si** $M(C_1, a_i^u) == \text{"ENTRANT"}$ **Alors**29. $\text{Mat}(EXT_a_i^u) += C_1$ ⁽³⁾30. **Fin Si**31. **Si** $M(C_1, a_i^u) == \text{"SORTANT"}$ **Alors**32. $\text{Mat}(C_1) += EXT_a_i^u$ ⁽³⁾33. **Fin Si**34. **Fin si**35. **Fin Pour**36. **Return** Mat

FIN

¹ : Cette fonction retourne les sommets de l'ensemble S_1 ou S_2 labellisés à "ENTREE" ou "SORTANT". Initialement, Ces sommets sont étiquetés à "None".

² : Cette fonction renvoie les cliques couvrantes une arête a_i^u avec $C_1 \neq \emptyset$. Si a_i^u est couvert par une clique alors $C_2 = \emptyset$.

³ : $EXT_a_i^u$ est un sommet du graphe racine G .

L'algorithme *Decouverte_graphe_racine* débute par l'initialisation des sommets de G_C à "None". Tant qu'il existe une arête dans notre line graphe G_C , on choisit la plus petite clique C_k dont le maximum de sommets de cette clique est labellisé soit par "ENTRANT" ou soit par "SORTANT" (lignes 3 – 4). On partitionne la clique C_k en deux sous-ensembles valides S_1 et S_2 tels que S_1 et S_2 correspondent, respectivement, à l'ensemble des arcs entrants et sortants du graphe racine G . Les sommets labellisés de S_1 notés S_1^k portent l'étiquette "ENTRANT" tandis que ceux labellisés en "SORTANT" sont notés S_2^k (lignes 6 – 12). Ensuite nous mettons à jour l'ensemble des arêtes et la line-couverture de G_C (lignes 13 – 14).

Une fois terminé l'orientation des arêtes de G qui sont les sommets labellisés du line graphe G_C (lignes 2 – 16), nous construisons la liste d'adjacence de chaque sommet de G (lignes 17 – 36). Nous recherchons la couverture d'un sommet a_i^u de G_C .

Si le sommet a_i^u est couvert par une seule clique alors $C_2 = \emptyset$. Dans ce cas, cela signifie qu'il existe un arc entre le sommet C_1 et le sommet $EXT_a_i^u$ que nous avons créé. Cet arc a pour extrémité initiale C_1 si a_i^u est labellisé par "SORTANT" sinon pour extrémité initiale $EXT_a_i^u$ si a_i^u est labellisé par "ENTRANT" (lignes 27 – 34).

Dans le cas où le sommet a_i^u est couvert par deux cliques non vides C_1, C_2 , nous ajoutons un arc entre ces deux cliques. D'après le théorème 4 stipulant qu'un sommet de G_C est "ENTRANT" de C_1 et "SORTANT" de C_2 et vice-versa, nous définissons C_1 comme extrémité initiale de cet arc si a_i^u est labellisé par SORTANT dans cette clique C_1 sinon C_2 si a_i^u est labellisé par SORTANT dans cette clique C_2 (lignes 19 – 26).

1.3.1 Complexité de l'algorithme *Decouverte_graphe_racine*

La fonction *Couverture_Cliques* a une complexité constante $O(1)$ alors que la complexité de la fonction *sommets_marques* dépend du nombre de sommets dans S_1 . Dans le pire des cas, sa complexité est $O(n)$ avec $n = E(G)$ le nombre d'arcs de G . Donc les lignes 17 – 35 s'exécutent en $O(n)$ dans le pire des cas.

Considérons une line-couverture \mathcal{C} de taille K , une clique $C_i \in \mathcal{C}$ de taille p_i et k_i le nombre de sommets marqués dans C_i .

Le nombre de couples valides (S_1, S_2) pour la clique C_i avec k_i sommets marqués est $2^{p_i - k_i}$. Au traitement de la première clique de \mathcal{C} de taille minimale C_1 , il existe 0 sommets marqués. Le coût R_{cv_1} de couples valides est $R_{cv_1} = 2^{p_1}$. Au traitement de la seconde clique C_2 , il existe $k_2 \geq 0$ et son coût est $R_{cv_2} = 2^{p_2 - k_2}$. Il existe une clique C_α à partir de laquelle $k_\alpha > 0, \alpha \leq K$. On remarque que $p_\alpha - k_\alpha$ est décroissant car k_α augmente après chaque traitement de cliques. Cela signifie qu'à la selection de la dernière clique C_K , son coût est de $R_{cv_K} = 1$. Le coût est décroissant à chaque traitement c'est-à-dire

$$2^{p_\alpha - k_\alpha} \geq 2^{p_{\alpha+1} - k_{\alpha+1}} \geq \dots \geq 2^{p_K - k_K}$$

Le coût des couples valides de \mathcal{C} est

$$R_{cv} = O(2^\gamma), \gamma = \max\{\text{card}(C_i), \forall C_i \in \{C_1, \dots, C_\alpha\}\}$$

car il depend de la clique $C_i \in \{C_1, \dots, C_\alpha\} \subset \mathcal{C}$.

Le coût des lignes 3 – 4 depend de p_i et K ($O(p_i) + O(K)$) et celui des lignes 7 – 14 est de $O(p_i^2)$. Soit le coût R de la boucle *Tant que*. Il est de

$$R = O(p_i) + O(K) + O(p_i^2) + O(2^\gamma) \simeq O(2^\gamma)$$

La complexité de *Decouverte_graphe_racine* est *pseudo-exponentielle* car l'exposant γ est la taille d'une clique de taille intermédiaire.

1.4 Correction particulière de Graphes : Graphes Iourtes

On se doit de considérer le cas où dans le graphe de corrélation initial G_C , il existe des sommets z tels que $Cliq(z) = -1$ et qu'aucun tel sommet z n'est contenu dans une clique et qu'aucun de ses voisins n'est contenu par une clique. Ce qui implique que tout sommet du graphe est dans la situation de z .

Lemme 3. *Pour tout entier $n \geq 12$, il existe un graphe G d'au moins n sommets dans lequel aucun sommet n'est couvert par une ou deux cliques.*

Proof. Ce lemme est tout d'abord basé sur le fait que le graphe G_0 de la figure 1.11 le vérifie. On obtient, à partir de G_0 , un graphe avec 3 sommets supplémentaires vérifiant toujours cette propriété en réalisant la modification suivante. Ce graphe G distingue trois sommets A, B, C . On remplace chaque arête $[A, B]$, $[B, C]$, $[A, C]$ par une chaîne de longueur 2. Les sommets ainsi créés deviennent les nouveaux sommets distingués A, B, C que l'on relie par un cycle de longueur 3. On peut ainsi propager cette modification autant de fois que nécessaire pour atteindre n . Ce qui clôt la preuve du lemme. \square

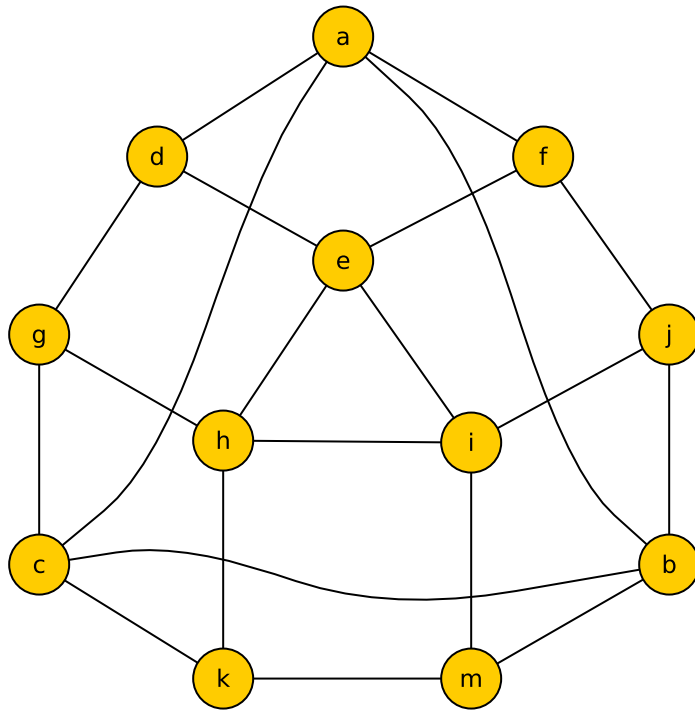
Notons G_0^k avec $k \geq 1$ le graphe obtenu à partir de k opérations de propagation à partir de G_0 , graphe contenant donc $3k + 12$ sommets, $6k + 21$ arêtes et de degré maximum égal à 4.

Théorème 5. *La distance-line de G_0^k vérifie*

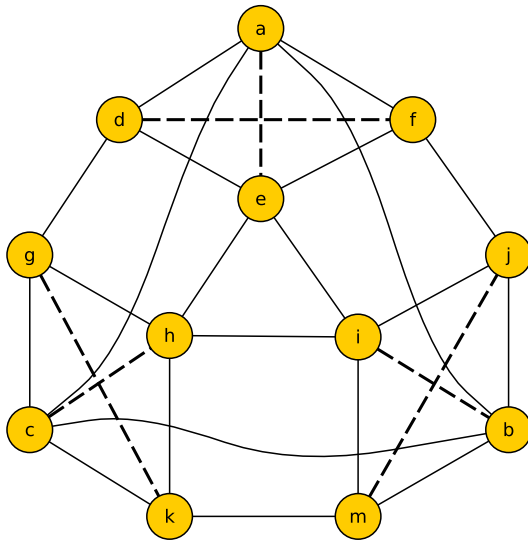
$$DL(G_0^k) \leq 3k + 6$$

Proof. La figure 1.12 décrit un line graphe LG_0 couvrant G_0 avec 6 arêtes ne figurant pas dans G_0 . Le line graphe LG_0^k couvrant et à distance $3k + 6$ de G_0^k est obtenu à partir de LG_0^{k-1} en supprimant une arête sur deux dans le cycle de taille 6 formé par les anciens sommets A, B et C de G_0^{k-1} et les nouveaux sommets A, B et C de G_0^k (c'est-à-dire la suppression de 3 arêtes). Chaque arête restante sur ce cycle forme une nouvelle clique de taille 2 et le nouveau cycle de taille 3 entre A, B et C forme une nouvelle clique de taille 3 (qui remplace la précédente dans LG_0^{k-1}).... \square

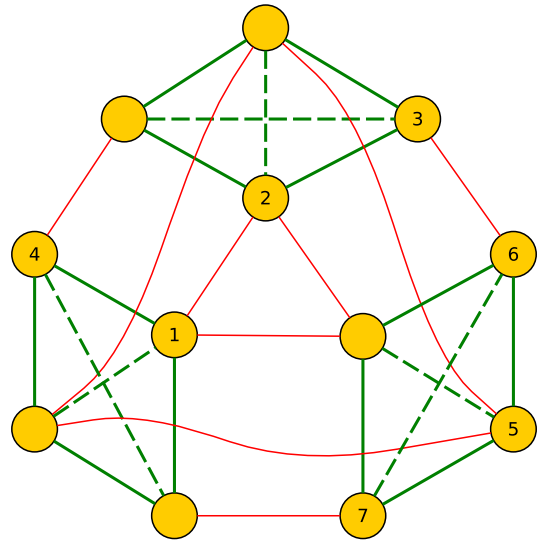
————-; PAS ENCORE TERMINER/REDIGER

Figure 1.11: Graphes G_0 où aucun sommet n'est couvert par aucune clique

$$C = \{ \{A,d,e,f\}, \{C,g,h,k\}, \{B,m,i,j\}, \\ \{e,h,i\}, \{d,g\}, \{f,j\}, \{k,m\}, \{A,B,C\} \}$$



(a)



(b)

Figure 1.12: (a) line graphe LG_0 à distance de Hamming 6 de G_0 avec une couverture par cliques maximales C , et (b) une exécution (parmi celles possibles) de l'algorithme de compression transformant G_0 en LG_0