TER – Couverture minimale de graphes par collection de graphes

Clémence DUMOULIN

Encadrants : Dominique BARTH – Coralie ZENS

23 mai 2024

Sommaire

Figures

Tableaux

1 Définition du sujet

1.1 Définitions des notions de graphes

Définition 1 Soit $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ deux graphes non orientés. G_1 et G_2 sont **isomorphes** s'il existe une bijection $\phi : V_1 \to V_2$ telle que :

$$[i, j] \in E_1 \iff [\phi(i), \phi(j)] \in E_2$$

Définition 2 Étant donné un graphe G, une chaîne de G, $C_e(G)$ contenant une arête e est dite **maximum** pour e ssi pour toute autre chaîne $C'_e(G)$ contenant e, $|C'_e(G)| \leq |C_e(G)|$.
On pose également $L(G) = \min_{e \in E} |C_e(G)|$.

Définition 3 Étant donné un graphe $G = (V, E)$, **l'excentricité** $Ex(u)$ d'un sommet u de G est le maximum des distances entre u et tout autre sommet v de G. C'est-à-dire, soit $d(u, v)$ la distance minimale entre deux sommets $u, v \in V$, $Ex(u) = \max_{v \in V} d(u, v)$.
On pose également $\mu(G)$ tel que pour toute feuille f de G, $\mu(G) = \min_{f \in V} Ex(f)$.

Définition 4 Soit $G = (V, E)$ un graphe non orienté. G est un graphe **pseudo-étoile** s'il existe un unique sommet $s \in V$ tel que son degré $deg(s) \geq 3$. On appelle alors une **branche** b toute chaîne dont les extrémités sont une feuille f et ce sommet s.

1.2 Définition des objets et métriques

On étudie des graphes connexes non orientés simples.
Soit $\mathcal{F} = \{g_1 = (V_1, E_1), ..., g_k = (V_k, E_k)\}$ un ensemble de graphes, avec $m_i = |E_i|$.
Un graphe $G = (V, E)$ est dit $\mathcal{F}$-couvrable ssi il existe un ensemble de sous-graphes **induits** $H = \{h_1, ..., h_x\}$ de G tel que :
- pour tout $1 \leq i \leq x$, il existe $g_j \in \mathcal{F}$, tel que h_i isomorphe à g_j,
- pour toute arête $e \in E$, il existe (au moins) un sous-graphe $h_j \in H$ dont e est une arête.

On appelle H une $\mathcal{F}$-couverture de G.

Étant donné une famille $\mathcal{F}$ et un graphe G, on s'intéresse à deux problèmes :
1. Existe-t-il $\mathcal{F}$-couverture pour un graphe G ?
2. Si une $\mathcal{F}$-couverture de G existe, quel est le poids minimum de celle-ci ?

Pour ce second problème, on peut utiliser différentes métriques pour définir le poids d'une couverture :
- **Le cardinal de H** : $|H|$, c'est-à-dire le nombre de sous-graphes de H utilisés pour la couverture,
- **La charge totale** : $C_{Totale} := \sum_{e \in E} |\{h_j \in H : e \in E(h_j)\}|$, c'est-à-dire le nombre total d'arêtes des sous-graphes de H utilisées,
- **La charge maximum** : $C_{Max} := \max_{e \in E} |\{h_j \in H : e \in E(h_j)\}|$, c'est-à-dire le nombre maximum de recouvrement sur une arête.

Remarque : Soit $\mathcal{F} = \{g_1 = (V_1, E_1)\}$, c'est-à-dire que la famille $\mathcal{F}$ est composée d'une unique chaîne. Alors **le cardinal de H** est équivalent à **la charge totale**, $|H| = \sum_{e \in E} |\{h_j \in H : e \in E(h_j)\}|$.

Dans le cadre de notre TER, on utilisera la charge totale C_{Totale}.

1.3 Différences avec des travaux de recherches existants

Il existe d'autres travaux de recherches sur la couverture minimale de graphes par collection de graphes. Par exemple, l'article [1] traite également de ce sujet.
Selon [1], un graphe $G = (V, E)$ est dit $\mathcal{F}$-couvrable ssi il existe un ensemble de sous-graphes **partiels** $H = \{h_1, ..., h_x\}$ de G. On peut dans ce cas dire qu'étant donné $\mathcal{F}$ une famille contenant

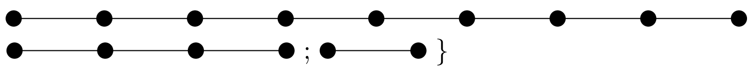
uniquement des chaînes de longueur au moins $|k|$ et G un graphe avec un cycle de taille $|k|$, il existe une $\mathcal{F}$-couverture de G.
L'article [1] présente, sur des graphes $\mathcal{F}$-couvrables, différentes métriques pour définir le poids minimum d'une couverture. Cette article recense des bornes supérieures et inférieures de poids minimum pour chaque métrique selon le type de graphes de la famille $\mathcal{F}$ (étoiles, graphes d'intervalles...) et selon le type de graphe G (planaire, planaire biparti,...).
Pour répondre à ce problème, une des métriques de poids utilisée est **le cardinal de H** : $|H|$.

Cependant, nous traitons dans ce TER des sous-graphes **induits**. C'est par ailleurs pour cette différence, qu'étant donné $\mathcal{F}$ une famille contenant uniquement des chaînes de longueur au moins k et G un graphe avec un cycle de taille k, il n'existe **pas** de $\mathcal{F}$-couverture de G.
De plus, nous étudions deux problèmes, les conditions d'existence d'une $\mathcal{F}$-couverture et son optimisation.
Enfin, nous utilisons **la charge totale** comme poids de la $\mathcal{F}$-couverture à minimiser.

Dans l'exemple ci-dessous, nous montrons qu'il existe des instances où l'optimal diffère selon la métrique utilisée pour calculer la valeur d'une $\mathcal{F}$-couverture.

Étant donné $G = (V, E)$ une chaîne de longueur 8 et $\mathcal{F}$ une famille de deux chaînes de longueur 3 et 1 :

$G := \{$ $\}$
$\mathcal{F} := \{$; $\}$



- Selon [1], une solution optimale serait de prendre 3 chaînes de 3 arêtes, comme sur le schéma ci-dessous. En effet, toutes les arêtes sont couvertes et **l'une d'elles** est couverte deux fois. Dans ce cas, $|H| = 3$ et $C_{Totale} = 9$. Cette solution n'est donc pas optimale selon la charge totale C_{Totale}.

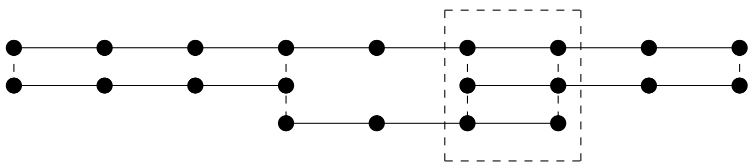


Figure 1: Une F-couverture optimale selon le cardinal de H

- Une solution optimale en utilisant la charge totale serait de prendre 8 chaînes d'une arête, comme sur le schéma ci-dessous. Il n'y a donc aucun recouvrements. $C_{Totale} = 8$ et $|H| = 8$. Cette solution n'est pas optimale selon le cardinal de H.

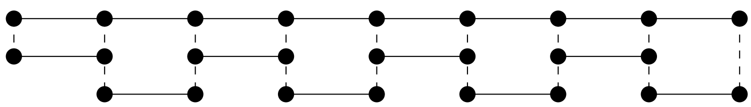


Figure 2: Une F-couverture optimale selon la charge totale

Les objets et les métriques étant différents, les solutions obtenues dans cet article ne sont donc pas applicables pour répondre à notre second problème.

D'après cette comparaison avec l'article [1], on en déduit que la couverture minimale de graphes par collection de graphes est un thème bien étudié en algorithme de graphes. Néanmoins il reste encore des branches de ce sujet à explorer.

Nous allons donc désormais nous intéresser au premier problème en étudiant l'existence d'une $\mathcal{F}$-couverture de G, étant donné une famille $\mathcal{F}$ et un graphe G.

2 Conditions d'existence

Tout d'abord, réduisons le problème des conditions d'existence au cas où $\mathcal{F}$ est une famille contenant uniquement des chaînes, c'est-à-dire :

Étant donné une famille $\mathcal{F} = \{g_1 = (V_1, E_1), ..., g_k = (V_k, E_k)\}$ un ensemble de chaînes, on cherche les conditions d'existence d'une $\mathcal{F}$-couverture d'un graphe G.
Posons $g_p = (V_p, E_p)$ la chaîne la plus courte de cette famille, de longueur $m_p = |V_p| - 1 = |E_p|$.

2.1 Chaînes et chaîne

Dans un premier temps, on définit des conditions d'existence d'une $\mathcal{F}$-couverture d'une chaîne G. On définit donc G de la manière suivante :

Théorème 1. *Soient $\mathcal{F}$ un ensemble de chaînes g_i de longueur $m_i \geq m_p$ et $G = (V, E)$ une chaîne de longueur $|V| - 1$.*
G est dit $\mathcal{F}$-couvrable ssi $m_p \leq |V| - 1$

Démonstration.

G est dit $\mathcal{F}$-couvrable $\implies m_p \leq |V| - 1$

Montrons que $m_p > |V| - 1 \implies G$ n'est pas $\mathcal{F}$-couvrable.
Si $m_p > |V| - 1$, $|V_p| > |V|$. Il n'existe donc pas de sous-graphes de G isomorphes à g_p.
G n'est pas $\mathcal{F}$-couvrable.
Et donc G est dit $\mathcal{F}$-couvrable $\implies m_p \leq |V| - 1$.

G est dit $\mathcal{F}$-couvrable $\impliedby m_p \leq |V| - 1$

G est une chaîne composée d'une suite de sommets $u_1 u_2 ... u_{|V|}$.

Comme $m_p \leq |V| - 1$, alors $|V| - m_p \geq 1$. Il existe un ensemble de sous-graphe **induits** H_{couvr} de G définit comme suit :
Soit $H_{couvr} = \{h_1, h_2, ..., h_{|V|-m_p}\}$, pour tout $1 \leq i \leq |V| - m_p$, $h_i = (V_i, E_i)$ tel que $V_i = \{u_i, u_{i+1}, ..., u_{m_p+i}\}$ et $E = \{[u_i, u_{i+1}], [u_{i+1}, u_{i+2}], ..., [u_{m_p-1+i}, u_{m_p+i}]\}$.
Pour tout $1 \leq i \leq |V| - m_p$, h_i est une chaîne, sous-graphe de G de longueur m_p.
Par la suite, on écrira la chaîne h_i comme une suite de sommets : $h_i := u_i u_{i+1} ... u_{m_p+i}$.

Ainsi, comme g_p est une chaîne et que pour tout $1 \leq i \leq |V| - m_p$, h_i est de longueur m_p, alors h_i est isomorphe à g_p.
En effet, on a :
$h_1 := u_1 u_2 ... u_{m_p+1}$, $h_2 := u_2 u_3 ... u_{m_p+2}$ et $h_{|V|-m_p} := u_{|V|-m_p} u_{|V|-m_p+1} ... u_{|V|}$ isomorphes à g_p.

Par exemple, dans la figure 3, pour $|V| = 9$ et $|V_p| = 3$ (c'est-à-dire $m_p = 2$), on a $|H_{couvr}| = |V| - m_p = 7$.
Et donc G est dit $\mathcal{F}$-couvrable $\impliedby m_p \leq |V| - 1$.
Donc G est dit $\mathcal{F}$-couvrable ssi $m_p \leq |V| - 1$. □

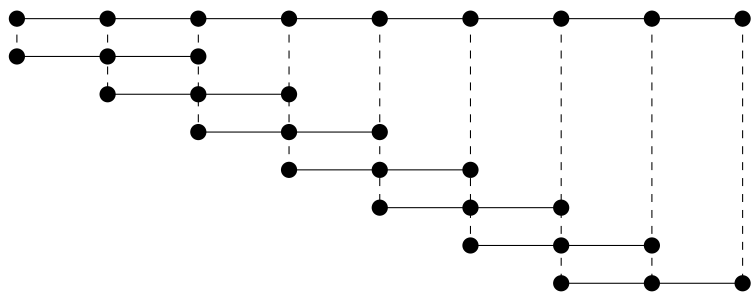


Figure 3: F-couverture d'une chaîne par une chaîne

2.2 Chaînes et arbre

Dans un second temps, on définit des conditions d'existence d'une $\mathcal{F}$-couverture sur un arbre T.

Pour ce faire, on rappelle $C_e(T)$ une chaîne maximum pour une arête e.

Théorème 2. *Soient $\mathcal{F}$ un ensemble de chaînes g_i de longueur $m_i \geq m_p$ et $T = (V, E)$ un arbre. T est dit $\mathcal{F}$-couvrable ssi $m_p \leq \min_{e \in E} |C_e(T)|$*

Démonstration.

Montrons que T est dit $\mathcal{F}$-couvrable ssi $m_p \leq L(T)$

<u>T est dit $\mathcal{F}$-couvrable $\implies m_p \leq L(T)$</u>

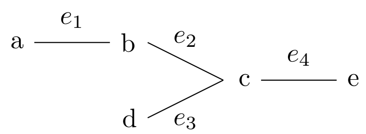
Montrons que $m_p > L(T) \implies T$ n'est pas $\mathcal{F}$-couvrable.
Comme $L(T) = \min_{e \in E} |C_e(T)|$, par définition il existe une arête $e' \in E$ telle que $L(T) = |C_{e'}(T)|$.
De plus $m_p > L(T) = |C_{e'}(T)|$.
D'après le **théorème 1**, $m_p > |C_{e'}(T)| \implies C_{e'}(T)$ ne peut être couverte par g_p. Or par définition de la chaîne maximum, il n'existe aucune autre chaîne contenant e' de longueur supérieure à $|C_{e'}(T)|$.
Donc il existe une arête $e' \in E$ ne pouvant pas être couverte par g_p.
T n'est donc pas $\mathcal{F}$-couvrable.

Et donc T est dit $\mathcal{F}$-couvrable $\implies m_p \leq L(T)$

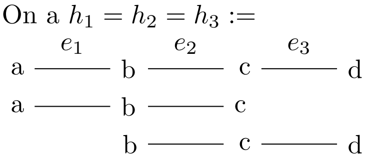
<u>T est dit $\mathcal{F}$-couvrable $\impliedby m_p \leq L(T)$</u>

Soit $H_{couvr} = \{h_1, h_2, ..., h_{|V|-1}\}$, tel que pour toute arête $e \in E$, $h_i = (V_i, E_i)$ est une chaîne maximum $C_e(T)$ pour cette arête, de longueur $|C_e(T)|$. h_i est un sous-graphe de T.
Pour tout $e \in E$, par définition de $L(T)$, on a $L(T) \leq |C_e(T)|$. C'est-à-dire, pour tout $h_i \in H_{couvr}$, $L(T) \leq |V_i| - 1$.
On a donc pour tout $1 \leq i \leq |V| - 1$, $m_p \leq L(T) \leq |V_i| - 1$.
Or pour toute chaîne $h_i = (V_i, E_i)$, d'après le **théorème 1**, $m_p \leq |V_i| - 1 \implies h_i$ est $\mathcal{F}$-couvrable.
Dans la figure 4 est présenté un exemple de $\mathcal{F}$-couverture, pour $|V| = 5$, $L(T) = 3$ et $m_p = 2$.

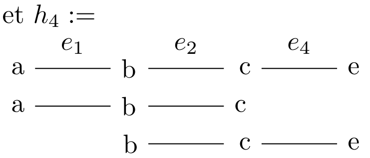
Et donc T est dit $\mathcal{F}$-couvrable $\impliedby m_p \leq L(T)$
Ainsi T est dit $\mathcal{F}$-couvrable ssi $m_p \leq L(T)$ □



On a $h_1 = h_2 = h_3 :=$



et $h_4 :=$



On couvre donc e_1 quatre fois, e_2 huit fois, e_3 trois fois et e_4 une fois, toutes les arêtes sont bien couvertes.

Figure 4: F-couverture d'un arbre par une chaîne

3 Algorithmes pour établir l'existence d'une F-couverture

3.1 Chaînes et arbre

Étant donné un arbre T et une famille $\mathcal{F}$, on veut désormais obtenir la valeur de $L(T)$ en un temps raisonnable pour confirmer l'existence d'une $\mathcal{F}$-couverture.

3.1.1 Algorithme trivial pour déterminer l'existence d'une F-couverture pour un arbre T

Soient un arbre T, et une famille F.
Pour obtenir la valeur $L(T)$, on peut faire un parcours en largeur à partir de chaque extrémités de chaque arêtes.
Soient :
Une arête $[u, v]$,
h_u la hauteur maximale du parcours en largeur de $G \setminus [u, v]$ depuis u,
h_v la hauteur maximale du parcours en largeur de $G \setminus [u, v]$ depuis v.
Ainsi, la chaîne maximum de l'arête $[u, v]$ vaut h_u+h_v+1.

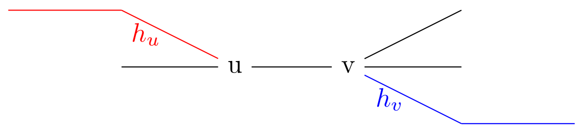


Figure 5: Calcul de la chaîne maximum $C_e(G)$ de $e = [u, v]$

Enfin, on récupère le minimum de toutes les valeurs calculées pour obtenir $L(T)$.

Complexité : Un parcours en profondeur se fait en $O(n)$ avec n le nombre de noeuds. On ferait ce parcours pour chaque arête, qui sont au nombre de $n - 1$.
La complexité de l'algorithme pour trouver toutes les chaînes maximales serait donc de $O(n^2)$.

3.1.2 Algorithme optimisé pour déterminer l'existence d'une F-couverture pour un arbre T

Il est possible d'améliorer le précédent algorithme en obtenant $L(T)$ sans calculer la longueur $|C_e(T)|$ pour chaque arête $e \in E$ de T.

Pour ce faire, on veut utiliser la propriété suivante :

Théorème 3. *Soit T un arbre, alors $L(T) = \mu(T)$.*

Démonstration.

Montrons que $L(T) = \mu(T)$. C'est-à-dire que, $\min_{e \in E} |C_e(T)| = \min_{f \in V} Ex(f)$.

Cas $L(T) > \mu(T)$:

Par définition de $L(T)$, $L(T) > \mu(T) \implies$ pour toute arête $e \in E$, $|C_e(T)| > \mu(T)$.

D'une part, pour toute arête $e \in E$, $C_e(T)$ est une chaîne maximum donc aussi une chaîne maximale. C'est-à-dire que $C_e(T)$ ne peut pas être allongée. De plus, comme T est un arbre, alors les extrémités de $C_e(T)$ sont des feuilles.

D'autre part, par définition de l'excentricité, pour tout sommet $u, v \in V$, $Ex(u) = \max_{v \in V} d(u, v)$.
Il existe donc un sommet a tel que pour tout $u \in V$, $\mu(T) = Ex(a) = \max_{u \in V} d(a, u)$.

Ainsi, considérons l'arête $e' = [a, a'] \in E$, dont la chaîne $C_{e'}(T)$ a pour extrémités a, une feuille telle que $Ex(a) = \mu(T)$, et b une autre feuille quelconque.
T étant un arbre, la distance entre deux sommets $u, v \in V$ est unique et correspond à la distance minimale $d[u, v]$. Donc $|C_{e'}(T)| = d(a, b)$.
Or $|C_{e'}(T)| > \mu(T)$ donc $d(a, b) > \max_{u \in V} d(a, u)$ nous menant alors à une contradiction.

On a donc $L(T) \leq \mu(T)$.

Cas $L(T) < \mu(T)$:

Soit $C_e(T)$, une chaîne maximum pour e, la plus petite de T. Cette chaîne étant maximum, elle a pour extrémités deux feuilles, f_1 et f_2.
Si $\mu(T) > |C_e(T)|$, alors $Ex(f_1) > |C_e(T)|$ et $Ex(f_2) > |C_e(T)|$. C'est-à-dire qu'il existe deux sommets s_1 et s_2, tels que les chaînes $f_1...s_1...f_3$ et $f_2...s_2...f_4$ sont de longueur supérieure à $|C_e(T)|$.
Si s_1 est placé après l'arête e depuis f_1 (sur la figure 6, à droite de l'arête e), alors $|C_e(T)|$ n'est pas maximum. De même pour s_2 placé après e depuis f_2 (sur la figure 6, à gauche de l'arête e). C'est pourquoi, s_1 et s_2 sont placés comme sur la figure 6. Cette figure représente le contexte proche de cette chaîne $C_e(T)$.

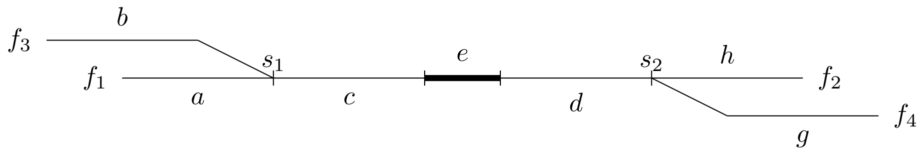


Figure 6: Contexte d'une chaîne maximum C_e d'extrémités f_1 et f_2

On a donc $|C_e(T)| = a + c + e + d + h$, $Ex(f_1) = a + b$ et $Ex(f_2) = g + h$.
Par définition de l'excentricité :

$$\begin{aligned} a + b &> a + c + e + d + h \\ \text{et } g + h &> a + c + e + d + h \end{aligned}$$

On a donc :

$$\begin{aligned} b &> c + e + d + h && (1) \\ \text{et } g &> a + c + e + d && (2) \end{aligned}$$

De plus, par définition de la chaîne maximum :

$$\begin{aligned} |C_e(T)| &> b + c + e + d + g \\ a + c + e + d + h &> b + c + e + d + g \\ a + h &> b + g && (3) \end{aligned}$$

Ainsi :

$$\begin{aligned} \text{avec (1), on a : } b + g &> c + e + d + h + g \\ \text{par (3), on obtient : } a + h > b + g &> c + e + d + h + g \\ a + h &> c + e + d + h + g \\ a &> c + e + d + g \\ \text{d'après (2), on peut écrire : } c + e + d + g &> c + e + d + a + c + e + d \\ a > c + e + d + g &> c + e + d + a + c + e + d \\ a &> 2 \cdot (c + e + d) + a && (4) \end{aligned}$$

Or $c \geq 0$, $d \geq 0$ et $e > 0$. En effet, s_1 et s_2 peuvent être les extrémités de l'arête e.
L'équation (4) nous mène donc à une contradiction.

On a donc $L(T) \geq \mu(T)$. □

Algorithme optimisé permettant de déterminer si un arbre T est F-couvrable

Comme $L(T) = \mu(T)$, il suffit de faire un parcours en largeur à partir d'un sommet et de récupérer la hauteur maximale de ce parcours pour obtenir l'excentricité de ce sommet.
En faisant, un parcours en largeur pour chaque feuille, on peut extraire l'excentricité de chaque feuille et en retenir le minimum. On obtient ainsi la valeur de $L(G)$ *(voir programme en annexe).*

Complexité : La complexité de cet algorithme serait donc en $O(n \times f)$ avec n le nombre de noeuds et f le nombre de feuilles.
Or dans le pire des cas, $f = n-1$ (graphe étoile), on obtient donc une complexité en $O(n^2)$ comme celle du premier algorithme.
Dans le meilleur des cas, il y a deux feuilles (une chaîne), la complexité est au minimum en $O(n)$.

4 Limites des propriétés d'existence

On ne peut pas appliquer cette condition si pour chaque arête $e \in E$ du graphe G qu'on souhaite couvrir, e fait partie d'un cycle.
En effet, H étant un ensemble de sous-graphes **induits** de G, rien ne garantit qu'entre plus de 2 sommets, il n'y a pas de cycles.

Par exemple, si $m_p > 2$ alors K_i avec $i \geq 3$ n'est pas $\mathcal{F}$-couvrable. En effet, tout ensemble de 3 sommets forme un cycle dans un graphe complet et donc n'est pas $\mathcal{F}$-couvrable par une chaîne de plus d'une arête (à cause de l'induction).

Nous allons maintenant nous intéresser au second problème en regardant le poids minimum d'une $\mathcal{F}$-couverture de G si elle existe.

5 Algorithmes d'optimisation de la charge totale

Tout d'abord, réduisons le problème d'optimisation au cas où $\mathcal{F}$ est une famille contenant uniquement des chaînes, c'est-à-dire :

Étant donné une famille $\mathcal{F} = \{g_1 = (V_1, E_1), ..., g_k = (V_k, E_k)\}$ un ensemble de chaînes, on cherche une $\mathcal{F}$-couverture d'un graphe G dont le poids est minimum.
Posons $g_p = (V_p, E_p)$ la chaîne la plus courte de cette famille, de longueur $m_p = |V_p| - 1$.

5.1 Chaînes et chaîne

Dans un premier temps, pour un graphe G et une famille $\mathcal{F}$ respectant les contraintes d'existence d'une $\mathcal{F}$-couverture définies antérieurement, on cherche **la charge totale** minimum de cette couverture, c'est-à-dire :

Étant donné une chaîne $G = (V, E)$ de longueur $|V| - 1$, telle que $m_p \leq |V| - 1$, on cherche la valeur minimale de C_{Totale} et un algorithme permettant d'obtenir cette valeur.

Valeur optimale de C_{Totale}
Soit $m_p = |E_p| = |V_p| - 1$, le nombre d'arête de la chaîne g_p, et $m = |E| = |V| - 1$, le nombre d'arête de G. On a $m \geq m_p$.
$\frac{m}{m_p}$ correspond au nombre d'occurrences de g_p pour couvrir G, si m_p divise m. Or une chaîne est entièrement utilisée ou inutilisée. Ainsi, dans le cas où m_p ne divise pas m, il faut utiliser la chaîne entièrement pour couvrir les quelques arêtes non couvertes. Ce qui se traduit par la partie entière supérieure : $\lceil \frac{m}{m_p} \rceil$.
On a donc $\lceil \frac{m}{m_p} \rceil$ le nombre de pose minimal de g_p. On multiplie cette valeur par m_p pour obtenir C_{Totale}.

On obtient donc $C_{Totale} = \lceil \frac{m}{m_p} \rceil \times m_p$ comme valeur minimum.

Algorithme pour créer une $\mathcal{F}$-couverture de poids minimum

On pose les occurrences de la chaîne de la famille $\mathcal{F}$ à partir des extrémités de G jusqu'à arriver à une collision.

Par exemple, étant donné $G = (V, E)$ une chaîne de longueur $m = 8$ et $\mathcal{F}$ une famille d'une de longueur $m = 3$:

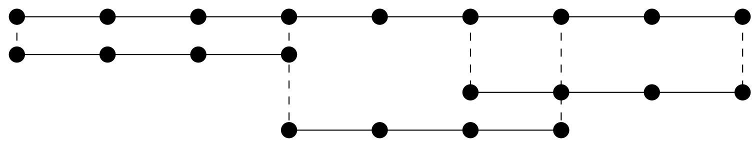


Figure 7: $\mathcal{F}$-couverture de poids minimum d'une chaîne

Entrée : m_p : nombre d'arêtes de g_p
Sortie : *poids* : poids de la $\mathcal{F}$-couverture
Fn `Poids_couverture():`

```
    ext1 ← 0
    ext2 ← 0
    poids ← 0
    tant que ext1 < ext2 − m_p OU ext2 > ext1 + l faire
        si ext1 + l < ext2 alors
            ext1 ← ext1 + m_p  // On pose g_p à partir de la 1ere extrémité de G
            poids ← poids + m_p
        si ext2 + l < ext1 alors
            ext1 ← ext1 − m_p  // On pose g_p à partir de la 2e extrémité de G
            poids ← poids + m_p
    si ext1 < ext2 alors
        // La couverture n'est pas complète donc on rajoute une dernière fois
           g_p pour couvrir les arêtes manquantes
        ext1 ← ext1 + m_p
        poids ← poids + m_p
    renvoyer poids
```

Algorithme 1 : Optimisation : couverture d'une chaîne par une autre chaîne

A la fin de la boucle "tant que", on obtient $\lfloor \frac{m}{m_p} \rfloor$ poses de g_p. Si m_p divise m, alors la couverture est complète et on ne pose pas une nouvelle occurrence de g_p. Sinon on ajoute une occurrence de g_p. On obtient donc $\lfloor \frac{m}{m_p} \rfloor + 1 = \lceil \frac{m}{m_p} \rceil$.
A chaque itération, on ajoute m_p, on obtient donc à la fin de l'algorithme $poids = \lceil \frac{m}{m_p} \rceil \times m_p$, la formule de la valeur minimum.

5.2 Chaînes et arbre

Dans un second temps, pour un arbre T et une famille $\mathcal{F}$ respectant les contraintes d'existence d'une $\mathcal{F}$-couverture définies antérieurement, on cherche **la charge totale** minimum de cette couverture, c'est-à-dire :

Étant donné un arbre $T = (V, E)$, tel que $m_p \leq L(T)$, on cherche la valeur minimale de C_{Totale} et un algorithme permettant d'obtenir cette valeur.

5.2.1 Pseudo-étoile

On s'intéresse ici au cas particulier des pseudo-étoiles pour la recherche d'un algorithme permettant d'obtenir la valeur minimale de C_{Totale} et une $\mathcal{F}$-couverture associée.
Soit T un graphe pseudo-étoile. On souhaite décomposer ce graphe en un ensemble de chaînes distinctes.
Par exemple, si on a 4 branches, on aura 2 chaînes distinctes comme dans le figure 8 :

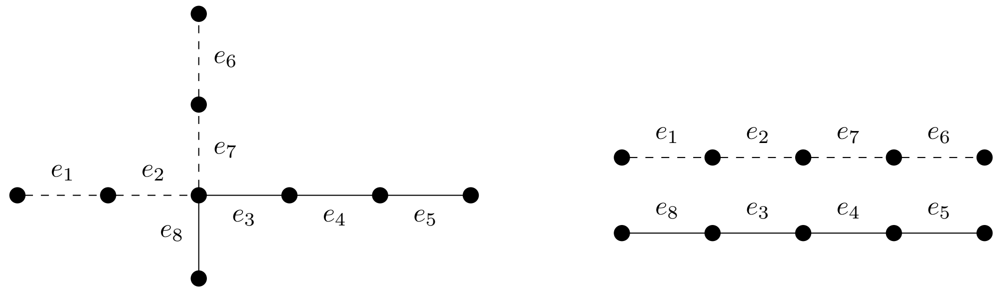
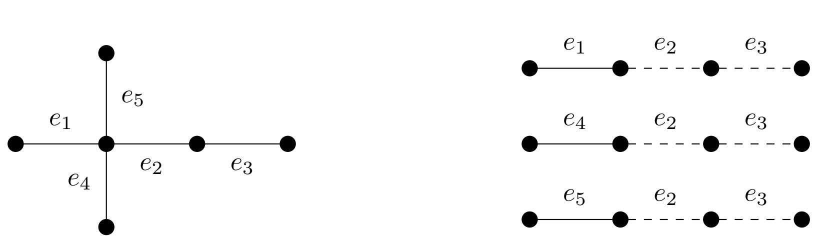


Figure 8: Décomposition d'une pseudo-étoile en un ensemble de chaînes

On se demande alors si couvrir chaque chaîne séparément avec l'algorithme 1 permet d'obtenir **la charge totale** C_{Totale} minimum.

On remarque qu'il n'est pas toujours possible de décomposer la pseudo-étoile en un ensemble de chaînes distinctes de longueur supérieure ou égale à m_p.
Par exemple, dans la figure 9, si la chaîne la plus courte de la famille est de longueur $m_p = 5$, la pseudo-étoile est $\mathcal{F}$-couvrable mais pas avec une décomposition en chaînes distinctes.

La branche e_2e_3 est communes deux chaînes.



La branche e_2e_3 est commune aux trois chaînes .

Figure 9: Conte-exemples d'une décomposition en chaînes distinctes

Lorsqu'une décomposition valide existe, on suppose, pour l'instant, que la charge totale minimum est la charge totale obtenue en appliquant l'algorithme 1 sur chaque chaîne de la décomposition.

5.2.2 Solution approchée

Au départ, on souhaitait trouver un algorithme en temps polynomial, pour le cas général des arbres, permettant d'obtenir une $\mathcal{F}$-couverture de poids minimum. Néanmoins, trouver un tel algorithme correspond à un problème plus difficile que ce que l'on imaginait.
De plus, l'application d'une méthode exacte sur toutes les $\mathcal{F}$-couvertures possible est trop coûteuse. Ainsi, notre objectif actuel est d'utiliser une solution approchée sur laquelle on pourrait appliquer une méthode exacte.

Algorithme pour créer une $\mathcal{F}$-couverture de poids minimum

A l'opposé d'un algorithme naïf pouvant couvrir la même arête de nombreuses fois inutilement, l'algorithme 2 présenté ci-après priorise les chaînes à couvrir en partant des feuilles de l'arbre. Ainsi, on évite de couvrir trop souvent les arêtes au centre de l'arbre, diminuant alors la charge totale.

Pour cet algorithme, il nous faut définir nos objets :

Soit $T = (V, E)$ un arbre qui peut être couvert par des chaînes de longueur k (dites k-chaînes). Une chaîne $\mathcal{C}(T)$ de T est dite candidate si et seulement si elle est de :

- **Type 1** : Ses extrémités sont des feuilles, elle contient au plus un sommet s tel que $deg(s) > 2$ dans T.
- **Type 2** : Une de ses extrémités est une feuille et l'autre est un sommet s tel que $deg(s) \geq 2$ dans T. Pour tout sommet interne a de la chaîne, $deg(a) = 2$ dans T.

Entrée : T : Arbre couvrable par une chaîne de longueur m_p, m_p : longueur de la chaîne couvrante
Sortie : H : ensemble de chaînes représentant la $\mathcal{F}$-couverture de T

Fn `Poids_couverture_arbre():`
  tant que *T a des arêtes non marquées* **faire**
    si *il existe une chaîne $\mathcal{C}(T)$ de type 1 de longueur m_p* **alors**
      Supprimer $\mathcal{C}(T)$ dans T
      $H \leftarrow$ Ajout($\mathcal{C}(T)$)
    sinon
      si *il existe une chaîne $\mathcal{C}(T)$ de type 2 de longueur m_p* **alors**
        Supprimer $\mathcal{C}(T)$ dans T
        $H \leftarrow$ Ajout($\mathcal{C}(T)$)
      sinon
        $\mathcal{C}(T) \leftarrow$ RechercheExhaustive(T, m_p)
        `// On fait une recherche exhaustive à partir des feuilles de T, si on ne trouve aucune chaîne de type 1 ou de type 2, et on récupère la chaîne avec le moins d'arêtes marquées`
        si *une chaîne $\mathcal{C}(T)$ de longueur m_p existe* **alors**
          Supprimer $\mathcal{C}(T)$ dans T
          $H \leftarrow$ Ajout($\mathcal{C}(T)$)
    si *T n'est pas $\mathcal{F}$-couvrable* **alors**
      Annuler la suppression de $\mathcal{C}(T)$
      Marquer les arêtes de $\mathcal{C}(T)$ dans T
  renvoyer H

Algorithme 2 : Optimisation : couverture d'un arbre par une chaîne

Pour trouver les chaînes $\mathcal{C}(T)$ de type 1 ou de type 2, on fait un parcours en largeur à profondeur m_p à partir de chaque feuille de l'arbre T. Ces parcours en largeur renvoient une chaîne de type 1 ou de type 2.

6 Conclusion

Étant donné une famille $\mathcal{F}$ et un graphe G, on a réussi à définir les conditions d'existence d'une $\mathcal{F}$-couverture pour un graphe G, lorsque G est une chaîne ou un arbre. On peut donc répondre au premier problème avec un algorithme en $O(n \times f)$.
De plus, si une $\mathcal{F}$-couverture de G existe, on peut trouver une de poids minimum pour les chaînes.

Poursuite du TER

Dans le cas des arbres, on trouve un algorithme, donnant une solution proche de l'optimale, qu'il faudrait continuer à explorer, notamment en l'associant avec une méthode totale.

En outre, dans le cas particulier des pseudo-étoiles, il faudrait prouver que lors d'une décomposition possible en chaînes distinctes, la solution optimale est celle obtenue par l'algorithme 1 sur chacune des chaînes de la décomposition.

Notations utiles

Dans le tableau 1, on retrouve les notations utilisées tout au long de ce travail, associées à leur signification.

Notations	**Définitions**
m_i	$\lvert E_i \rvert$, le nombre d'arêtes du graphe g_i de la famille $\mathcal{F}$
$C_e(G)$	Une chaîne de G **maximum** pour un arête e
$L(G)$	$\min_{e \in E} \lvert C_e(G) \rvert$
$Ex(u)$	L'excentricité d'un sommet u du graphe G
$\mu(G)$	$\min_{f \in V} Ex(f)$ avec f une feuille
C_{Totale}	$\sum_{e \in E} \lvert \{h_j \in H : e \in E(h_j)\} \rvert$, le nombre total d'arêtes de sous-graphes de H utilisées

Table 1: Tableau récapitulatif des notations

Bibliographie

[1] Torsten Ueckerdt Kolj Knauer. “Three ways to cover a graph”. In: *Elsevier* (2015).