

Couplages contraints dans les graphes

Rapport de stage de L3

Zoé Varin
ENS de Lyon

Encadré par:
Julien Bensmail et **Nicolas Nisse**
Équipe COATI, Université Côte d'Azur, Inria, CNRS, I3S, France

Juin–juillet 2019

Résumé

Le calcul de couplage maximum sur un graphe est un problème polynomial très connu, dont une des méthodes de résolution classique est fondée sur l'utilisation de chemins augmentants. La question de borner voire fixer la taille de ces chemins augmentants est par contre un problème ouvert, et plus difficile. C'est sur ce problème que j'ai travaillé pendant mon stage.

Table des matières

1	Introduction	2
1.1	Motivation	2
1.2	Introduction du problème théorique et des notations générales	2
1.3	État de l'art	4
1.4	Contribution	4
2	Les arbres	5
2.1	Définitions et notations spécifiques	5
2.2	Construction de l'algorithme	5
2.2.1	Les cinq problèmes	5
2.2.2	Résolution par induction structurelle	7
2.2.3	Conclusion et complexité de l'algorithme	10
2.3	Correction de l'algorithme	10
3	Split graphs	12
3.1	Cas des soleils	12
3.1.1	Présentation	12
3.1.2	L'algorithme	13
3.1.3	Preuve de correction	14
3.2	Pistes pour le cas général	17
4	Conclusion	19

1 Introduction

1.1 Motivation

Dans un aéroport, on affecte chaque avion qui atterrit ou décolle à une piste et un créneau horaire : un *slot*. Les affectations possibles peuvent être modélisées par un graphe biparti (on relie chaque avion aux slots acceptables). On veut alors affecter au plus un avion à chaque slot. L'établissement du planning correspond donc à la donnée d'un couplage sur le graphe biparti associé, concrètement un ensemble de paires avion-slot deux à deux disjointes.

Si un avion A a du retard, on souhaite le réaffecter à un slot sans pour autant modifier tout le planning. On pose alors des règles simples (qui sont inspirées des règles en vigueur en Australie par exemple) : on peut affecter l'avion A à un slot disponible, ou alors à un slot S déjà affecté à un avion B , à condition que l'avion B puisse être réaffecté à un slot disponible S' . La figure 1 représente un exemple de scénario. Ce problème revient à chercher à maximiser le couplage (ensemble de paires avion-slot) en utilisant uniquement des chemins augmentants de longueur inférieure ou égale à 3. C'est ce problème qui a introduit le problème théorique que nous allons présenter et étudier.

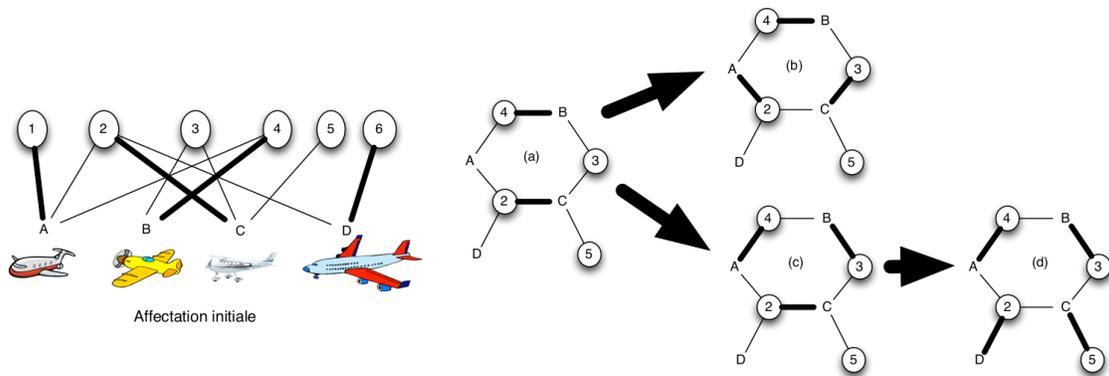


FIGURE 1 – Sur cet exemple (tiré de [NSW15a]), on commence à gauche avec le graphe biparti associé aux affectations possibles, et on en déduit un couplage (arêtes en gras). Mais les slots 1 et 6 ne sont plus accessibles. On est alors dans la configuration (a). On a deux possibilités. Soit on augmente le chemin (A2C3), i.e. on affecte l'avion A au slot 2, et l'avion C au slot 3. On est alors dans la configuration (b), et il n'y a plus d'augmentation possible, l'avion D n'est donc affecté à aucun slot. Soit on augmente les chemins (A4B3) puis (D2C5), ce qui donne finalement la configuration (d), dans laquelle tous les avions ont de nouveau une affectation.

1.2 Introduction du problème théorique et des notations générales

On considère un graphe non orienté $G = (V, E)$.

Un *couplage* est un ensemble $M \subseteq E$ d'arêtes disjointes : $\forall e, e' \in M, e \neq e' \implies e \cap e' = \emptyset$. On dit qu'un couplage M est *maximum* s'il n'existe pas de couplage M' tel que $|M'| > |M|$. On note $\mu(G)$ la taille d'un couplage maximum.

On dit qu'un sommet $u \in V$ est *couvert* (par un couplage M , implicite s'il n'y a pas d'ambiguïté) s'il existe $v \in V$ tel que $uv \in M$. Dans le cas contraire, il est dit *exposé*. On note V_M l'ensemble des sommets couverts par M .

Un chemin *M-augmentant* ou *M-améliorant* (ou simplement augmentant ou améliorant s'il n'y a pas d'ambiguïté sur le couplage considéré), est un ensemble de sommets distincts v_0, v_1, \dots, v_n tels que n impair, $v_0 \notin V_M, v_n \notin V_M, \forall 0 \leq i < n, \{v_i, v_{i+1}\} \in E$, et $\forall 0 \leq j \leq \frac{n-1}{2}, \{v_{2j}, v_{2j+1}\} \notin M$, et $\{v_{2j+1}, v_{2j+2}\} \in M$.

On parle d'augmenter un chemin *M-augmentant* P quand on remplace M par

$$M \Delta E(P) = (M \cup E(P)) \setminus (M \cap E(P)) \quad (\text{où } E(P) = \{\{v_i, v_{i+1}\}, \forall 0 \leq i < n\}).$$

Intuitivement, cela revient à échanger les arêtes de P étant dans M avec celles ne l'étant pas. Berge a prouvé qu'un couplage est maximum si et seulement s'il n'admet pas de chemin augmentant [Ber57].

Edmonds en a déduit un algorithme polynomial pour calculer un couplage maximum d'un graphe quelconque, le "*Blossom algorithm*", qui utilise justement cette propriété [Edm65]. Cependant, dans cet algorithme les chemins augmentants considérés peuvent être arbitrairement grands. Comme nous l'avons vu dans la partie précédente, cela peut être gênant dans certaines applications. C'est pourquoi on s'intéresse à la recherche d'un couplage maximum atteint en n'utilisant que des chemins améliorants de longueur bornée, voire fixée. Le cas où on s'intéresse aux chemins de taille inférieure ou égale à 3 correspond au problème pratique exposé juste avant. Les articles [NSW15b] et [BGN18] étudient ces problèmes, et servent donc de point de départ à mon stage.

On note alors, pour tout k impair, $\mu_{\leq k}(G, M)$ la taille d'un couplage de taille maximale qu'on puisse obtenir à partir d'un sous-ensemble $M \subseteq E$ en n'utilisant que des chemins augmentants de longueur inférieure ou égale à k . De la même façon, on note $\mu_{=k}(G, M)$ la taille d'un couplage de taille maximale qu'on puisse obtenir à partir d'un sous-ensemble $M \subseteq E$ et n'utilisant que des chemins augmentants de longueur égale à k . Par exemple, sur la figure 1, le couplage de la configuration (d) est bien maximum, on ne l'a obtenu qu'avec des chemins de longueur 3, et on ne peut faire mieux, que ce soit avec des chemins de longueur inférieure ou égale à 3 ou exactement 3. Donc on a, en notant M_a le couplage initial et M_d le couplage de la configuration (d) : $\mu_{=3}(G, M_a) = \mu_{\leq 3}(G, M_a) = |M_d|$.

On choisit de définir $\mu_{\leq k}(G, M)$ et $\mu_{=k}(G, M)$ quel que soit $M \subseteq E$, avec la convention $\mu_{\leq k}(G, M) = \mu_{=k}(G, M) = -1$ si M n'est pas un couplage. Par ailleurs, si $G' = (V', E')$ est un sous-graphe de G , et si M' est tel que $M' = M \cap E'$ (M' est la restriction de M à G'), alors $\mu_{\leq k}(G', M) := \mu_{\leq k}(G', M')$ et $\mu_{=k}(G', M) := \mu_{=k}(G', M')$.

On parle d'*augmenter* une arête (ou de l'améliorer) quand celle-ci est l'arête centrale d'un chemin augmentant de longueur 3 utilisé.

On définit enfin plusieurs structures sur lesquelles nous allons étudier le problème :

Définition (Chenille). *Soit $T = (V, E)$ un arbre. T est une chenille si, en notant F l'ensemble des feuilles de T , alors T' induit par la restriction de T à $V \setminus F$ est un chemin.*

Définition (Split graph). *Un split graph est un graphe $G = (I \cup K, E)$ tel que $\forall i, j \in I, i \neq j \implies ij \notin E$, et $\forall k, l \in K, k \neq l \iff kl \in E$. Autrement dit, K induit une clique et I induit un stable.*

Définition (Soleil). *On nomme soleils les split graphs tels que tout sommet de I possède au plus un voisin dans K . Formellement, si on note $N(u)$ le voisinage d'un sommet quelconque u , alors un soleil $S = (I \cup K, E)$ vérifie :*

$$\begin{cases} \forall i, j \in I, i \neq j \implies ij \notin E \\ \forall k, l \in K, k \neq l \iff kl \in E \\ \forall i \in I, |N(i)| \leq 1 \end{cases}$$

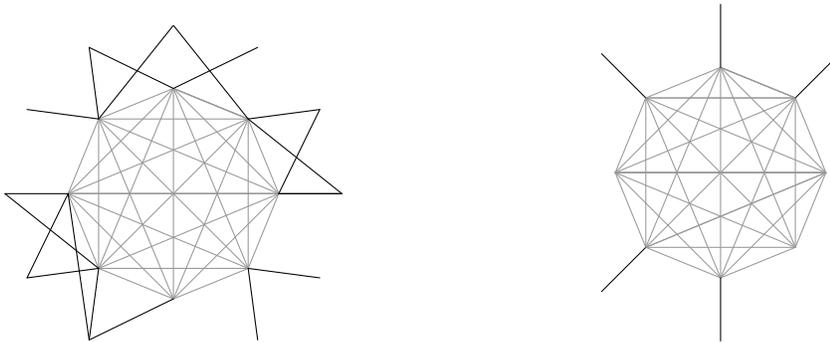


FIGURE 2 – Un exemple de split graph (à gauche) et de soleil (à droite).

Définition (Graphe d'intervalles). *Un graphe d'intervalles est un graphe $G = (V, E)$ tel que $V = \{I_1, I_2, \dots, I_n\}$ est un ensemble d'intervalles de \mathbb{R} , et pour tout i, j , $\{I_i, I_j\} \in E$ si et seulement si $i \neq j$ et $I_i \cap I_j \neq \emptyset$.*

1.3 État de l'art

Le problème est NP-complet en général, mais reste polynomial dans certains cas particuliers (dans certaines classes d'arbres, ou quand on s'intéresse à $\mu_{\leq k}$ avec $k \leq 3$). Voici donc un certain nombre de résultats, prouvés dans les articles [NSW15b] et [BGN18], qui ont guidé ma réflexion.

Il existe des algorithmes permettant de résoudre plusieurs cas particuliers.

Théorème 1.1. *Il existe un algorithme polynomial qui calcule $\mu_{\leq 3}(G, M)$ et renvoie un couplage correspondant, pour tout graphe G et couplage M .*

Théorème 1.2 (Chemins). *Il existe un algorithme qui calcule $\mu_{\leq k}(P_n, M)$ et renvoie une séquence de chemins augmentants permettant d'atteindre un couplage de cette taille, en partant de M , pour tout entier k impair, pour tout chemin $P_n = (v_0, v_1, \dots, v_n)$ et pour tout couplage $M \subseteq E(P_n)$, en temps linéaire par rapport à n .*

Soit k un entier naturel. Un arbre k -sparse est un arbre tel que les sommets de degré strictement supérieur à 2 soient à distance au moins k les uns des autres.

Théorème 1.3 (Arbres k -sparse). *Soit k un entier naturel impair. Il existe un algorithme polynomial qui calcule $\mu_{\leq k}(T, M)$ et renvoie une séquence de chemins augmentants permettant d'atteindre un couplage de cette taille, en partant de M , pour tout T arbre k -sparse et pour tout couplage $M \subseteq E(T)$.*

Théorème 1.4 (Chenilles). *Soit k un entier naturel impair. Il existe un algorithme qui calcule $\mu_{\leq k}(C, M)$ pour toute chenille C et pour tout couplage M de C , en temps linéaire en $|V(C)|$.*

Théorème 1.5. *Soient T un arbre de degré maximum Δ , M un couplage sur T , et k un entier impair. Le calcul de $\mu_{\leq k}(T, M)$ peut se faire en temps $O(f(k + \Delta) \cdot |V(T)|)$, où f est une fonction calculable.*

On a par ailleurs des résultats de NP-complétude :

Théorème 1.6. *Pour tout entier k impair, $k \geq 5$, le calcul de $\mu_{\leq k}(G, M)$ est NP-complet dans la classe des graphes bipartis planaires de degré maximum 3, en partant d'un couplage M quelconque.*

Pour le cas des chemins augmentants de longueur exactement k , le problème semble devenir encore plus compliqué, comme l'attestent les résultats suivants.

Théorème 1.7. *Pour tout entier k impair, $k \geq 3$, le calcul de $\mu_{=k}(G, M)$ est NP-complet dans la classe des graphes bipartis planaires de degré maximum 3, en partant d'un couplage M quelconque.*

Théorème 1.8. *Le calcul de $\mu_{=k}(G, M)$ en prenant en entrée un graphe G , un couplage M et un entier impair $k \geq 1$ est NP-complet, même en se restreignant aux instances dans lesquelles G est un arbre.*

Enfin, voici deux lemmes assez spécifiques qui seront utiles par la suite :

Lemme 1.9. *Soient $G = (V, E)$ un graphe, $M \subseteq E$ un couplage, et P un chemin M -augmentant quelconque. Si $v \in V$ est couvert dans M , alors il est couvert dans $M \Delta E(P)$, c'est à dire après augmentation du chemin P .*

Lemme 1.10. *Soient G un graphe, $M \subseteq E$ un couplage de G , et $S = (P_1, \dots, P_r)$ une séquence de chemins augmentants. Soit $i < r$ tel que P_i et P_{i+1} n'aient aucun sommet en commun. Alors, $(P_1, \dots, P_{i-1}, P_{i+1}, P_i, P_{i+2}, \dots, P_r)$ est également une séquence de chemins augmentants, qui aboutit au même couplage que la séquence S .*

1.4 Contribution

L'état actuel des recherches sur le sujet montre que ces problèmes sont difficiles, même sur certaines classes de graphes assez restreintes. Cela nous laisse donc plusieurs pistes de réflexion : poursuivre vers des résultats de NP-complétude, ou alors se concentrer sur d'autres classes de graphes classiques et éventuellement chercher d'autres algorithmes polynomiaux sur celles-ci.

Ces recherches ont abouti, au cours de mon stage, à ces deux théorèmes, que nous prouverons dans les parties suivantes :

Théorème. *Il existe un algorithme prenant en entrée un arbre T et un couplage M , et calculant $\mu_{=3}(T, M)$ avec une complexité $O(|V(T)|)$.*

Théorème. *Il existe un algorithme prenant en entrée un soleil S et un couplage M , et renvoyant un couplage de taille $\mu_{=3}(S, M)$ que l'on peut obtenir à partir de M en n'augmentant que des chemins de longueur 3.*

Une petite remarque pour la suite : une légende générale des illustrations est disponible en annexe E.

2 Les arbres

Ici, on s'intéresse à la résolution de $\mu_{=3}(T, M)$, pour tout arbre T et tout sous-ensemble d'arêtes M .

Le théorème 1.5 peut être bon un point de départ. La preuve de ce théorème est fondée sur la description d'un algorithme de programmation dynamique calculant $\mu_{\leq k}(T, M)$. L'idée de cet algorithme est simplement de tester tous les cas, d'où une complexité exponentielle en le degré de T . Cette fois, on cherche à s'affranchir de cette dépendance en le degré de l'arbre. Une idée est alors de chercher à traiter les sous-arbre indépendamment les uns des autres.

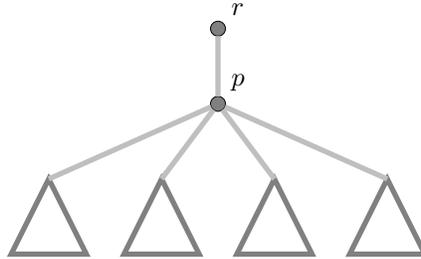
Cette section est dédiée à la preuve du théorème suivant :

Théorème. *Il existe un algorithme prenant en entrée un arbre T et un couplage M , et calculant $\mu_{=3}(T, M)$ avec une complexité $O(|V(T)|)$.*

On construit un tel algorithme dans la partie 2.2, et on démontre sa correction dans la partie 2.3.

2.1 Définitions et notations spécifiques

On commence par enraciner l'arbre en une feuille arbitraire r , de voisin p , de sorte que l'arbre en r ait la forme suivante, de même que tous les sous-arbres (qui sont schématisés par des triangles).



Pour tout sommet $u \in V$, on note T_u le sous-arbre enraciné en u , et si $u \neq r$, en notant v le parent de u , $\tilde{T}_u = T_u \cup \{v\}$. On note $F(u)$ l'ensemble des enfants d'un sommet $u \in V$. Par ailleurs on note $P(u)$ le parent d'un sommet $u \in V$, et on autorise la composition pour $P : \forall k, P^k(u) = P(P(\dots P(u)\dots))$.

Notons que chaque T_u peut éventuellement être constitué uniquement du sommet u .

Dans la suite, on parlera simplement de couplage maximum pour faire référence à un couplage maximum obtenu en n'utilisant que des chemins augmentants de longueur exactement 3.

2.2 Construction de l'algorithme

Soient $T = (V, E)$ un arbre enraciné en r , $M \subseteq E$ un couplage, et $u \in V \setminus \{r\}$

On distingue 5 sous-problèmes, et on veut montrer que la résolution de ces problèmes pour chaque $\tilde{T}_v, v \in F(u)$ permet de résoudre ces 5 problèmes pour \tilde{T}_u .

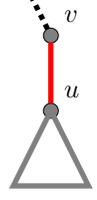
2.2.1 Les cinq problèmes

— Problème A_u :

On pose

$$a_u := \begin{cases} \mu_{=3}(\tilde{T}_u, M \cup \{uv\}) & \text{si } u \notin V_M \\ -1 & \text{sinon} \end{cases}$$

Ce problème sert à modéliser le cas où l'arête est apparue avec l'augmentation d'un chemin passant par uv . C'est pourquoi il n'y a pas de solution ($a_u = -1$) si initialement $uv \in M$ ou s'il existe $w \in F(u)$ tel que $uw \in M$.

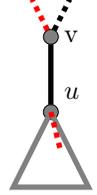


- Problème B_u :
On pose

$$b_u := \begin{cases} -1 & \text{si } u \in V_M, \text{ ou si } \forall w \in F(u), w \in V_M \\ \max_{w \in F(u), w \notin V_M} \mu_{=3}(T_u, M \cup \{uw\}) & \text{sinon} \end{cases}$$

On modélise ici le cas où l'arête uv a été augmentée, après sa création à partir d'un chemin augmentant (x, y, v, u) (intuitivement, on dira que le développement du couplage s'est fait "par le haut").

Notons qu'on a bien $b_u = -1$ si u est une feuille.

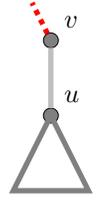


- Problème C_u :
On pose :

$$c_u := \begin{cases} \mu_{=3}(T_u, M) & \text{si } uv \notin M \\ -1 & \text{sinon} \end{cases}$$

On cherche ici à calculer la taille d'un couplage maximum dans \tilde{T}_u , sachant que v est couvert mais $uv \notin M$.

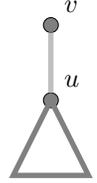
Remarquons au passage que $c_u \geq 0$ ssi $uv \notin M$.



- Problème D_u :
On pose :

$$d_u := \begin{cases} \mu_{=3}(\tilde{T}_u, M) & \text{si } uv \in M, \text{ ou si } v \notin V_M \\ -1 & \text{sinon} \end{cases}$$

On cherche donc à calculer la taille d'un couplage maximum dans \tilde{T}_u (soit v est exposé, soit $uv \in M$).

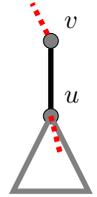


- Problème E_u : Avec ce problème on cherche à calculer la taille d'un couplage maximum dans \tilde{T}_u , sachant qu'à une étape l'arête uv a été augmentée.
Il apparaît dans deux cas :

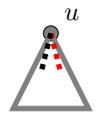
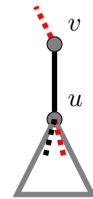
- Soit $uv \in M$, et on pose alors :

$$e_u := \begin{cases} \max_{w \in F(u), w \notin V_M} \mu_{=3}(T_u, M \setminus \{uv\} \cup \{uw\}) & \text{si } F(u) \setminus V_M \neq \emptyset \\ -1 & \text{sinon} \end{cases}$$

Cela revient à chercher un couplage maximum parmi tous ceux atteignables par une séquence de chemins dont l'un augmente d'arête centrale uv .



- Soit $uv \notin M$ (premier dessin), et on suppose qu'on a atteint un couplage couvrant uv , puis qu'on a utilisé un chemin augmentant d'arête centrale uv . Alors, dans T_u , cela revient à chercher un couplage maximum sans être passé par l'augmentation de uv (deuxième dessin). On cherche le meilleur couplage vérifiant ces conditions. Autrement dit, on cherche la taille maximale d'un couplage obtenu en utilisant un chemin augmentant de la forme (x, y, u, z) (avec $x \in F(y), y, z \in F(u)$).



Formellement, on pose $e_u := \max \{|S| \mid S \in \mathcal{S}\}$

avec $\mathcal{S} = \{(P_1, \dots, P_n), \text{ séquence de chemins augmentants de longueur 3 en partant du couplage } M\}$.

On peut montrer que

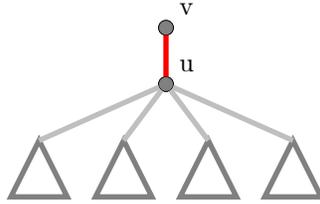
$$e_u = \max_{w \in T_u, \{P(w), w\} \in M} \max \left\{ \mu_{=3}(T_u, M \setminus \{wP(w)\} \cup_k \{w_k P^k(w)\} \mid \forall k, w_k \in F(P^k(w)), w_k \neq P^{k-1}(w), w_k \notin V_M \right\}.$$

2.2.2 Résolution par induction structurelle

On cherche maintenant à résoudre nos problèmes dynamiquement, c'est à dire que pour $u \neq r$ donné, on veut résoudre les 5 problèmes en u à partir des résolutions des problèmes en w , pour tout $w \in F(u)$.

Soit $u \in V \setminus \{r\}$. Posons $v := P(u)$. On distingue trois cas, en fonction des arêtes initialement présentes dans M .

- Initialement, l'arête uv est couverte par le couplage :



On a alors deux "évolutions" possibles, et on en déduit la résolution de certains problèmes.

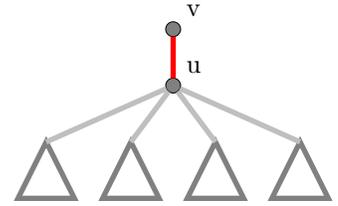
- Soit il ne se passe rien :

Il suffit alors de regarder un couplage maximum pour chaque sous-arbre indépendamment. Chaque sous-arbre est dans la situation du problème C_w , et donc :

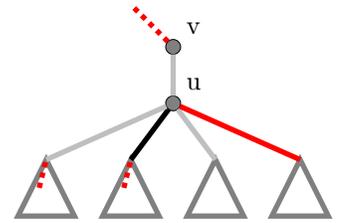
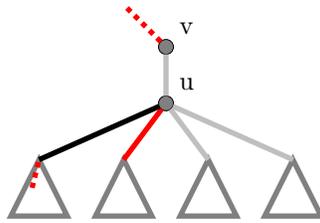
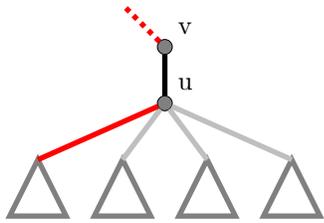
$$d_u = 1 + \sum_{w \in F(u)} c_w$$

(car $uw \in M$ et M couplage, donc pour tout $w \in F(u)$, $uw \notin M$, et donc $c_w \neq -1$).

On a bien $d_u = 1$ si u est une feuille.



- Soit cette arête est "améliorée" au moins une fois :



On a alors seulement à choisir sur quel fils on s'arrête (par exemple sur le fils droit pour la dernière image, ce fils est dans la configuration du problème A), et ensuite on regarde par quels sommets on a éventuellement pu passer (problème B si on y est passé, C sinon).

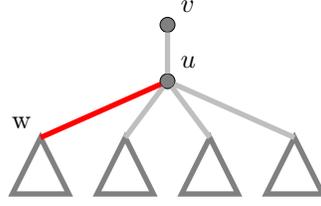
D'où,

$$e_u = \max \left\{ a_u + \sum_{x \in F(u), x \neq w} \max(b_x, c_x) \mid w \in F(u), a_w \geq 0 \right\} \cup \{-1\}.$$

Remarquons qu'on a bien $e_u = -1$ si u est une feuille.

Dans ce cas-là on a alors, par incompatibilité des problèmes A_u , B_u et C_u avec les conditions initiales, $a_u = -1$, $b_u = -1$ et $c_u = -1$.

- Si initialement il existe $w \in F(u)$ tel que $uw \in M$: (dans ce cas u n'est jamais une feuille)
Cette fois, on a quatre évolutions possibles du couplage.
On note w le sommet tel que $uw \in M$.

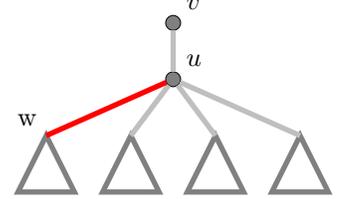


Notons que $c_w = -1$, et $\forall x \in F(u) \setminus \{w\}, c_x \geq 0$, et donc $\max(b_x, c_x) \geq 0$.

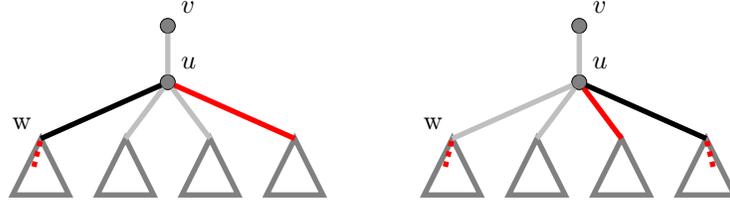
— Il ne se passe rien :

Dans ce cas, en résolvant pour chaque sous-arbre, on résout ce cas :

$$r_{u,1} := d_w + \sum_{x \in F(u), x \neq w} c_x.$$



— On développe plusieurs fois sans toucher à l'arête mère, par exemple :



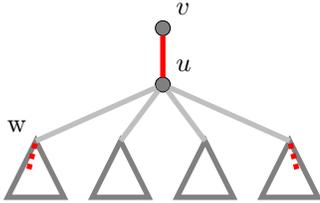
On pose :

$$M_u := \left\{ a_x + \sum_{y \in F(u) \setminus \{w, x\}} \max(b_y, c_y) \mid x \in F(u) \setminus \{w\}, a_x \neq -1 \right\}$$

et on a donc :

$$r_{u,2} := \begin{cases} b_w + \max M_u \cup \{-1\} & \text{si } b_w \geq 0 \text{ et } \max M_u \cup \{-1\} \geq 0 \\ -1 & \text{sinon} \end{cases}$$

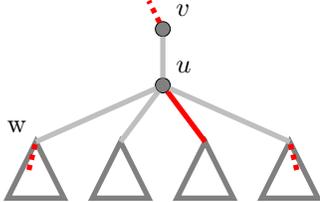
— On développe plusieurs fois mais cette fois en finissant sur l'arête mère, par exemple :



On calcule le maximum pour chaque sous-arbre, en passant éventuellement par chacun, et on en déduit un couplage maximum pour T_u dans cette configuration :

$$r_{u,3} := \begin{cases} b_w + \sum_{x \in F(u) \setminus \{w\}} \max(b_x, c_x) & \text{si } b_w \geq 0 \\ -1 & \text{sinon} \end{cases}$$

— On développe plusieurs fois, y compris l'arête mère, et on termine ailleurs, par exemple :



Ici, pour T_u on a le même résultat que dans le deuxième cas (on distingue seulement les deux cas car ils ne résolvent pas le même problème) :

$$r_{u,4} := r_{u,2}$$

Finalement, on déduit de ces sous-cas que :

$a_u = -1$ et $b_u = -1$ (incompatibilité de ces problèmes avec ce cas)

$c_u = \max(r_{u,1}, r_{u,2})$

$$d_u = \max(r_{u,1}, r_{u,2}, r_{u,3})$$

$$e_u = r_{u,4}$$

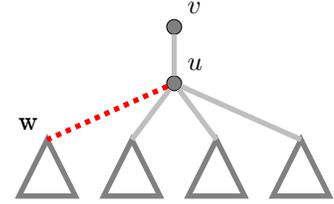
— Enfin, si on n'avait initialement aucune arête adjacente à u dans le couplage, d'éventuelles nouvelles arêtes peuvent provenir d'un sous-arbre, ou alors arriver par l'arête mère. On en déduit les 6 cas suivants :

- On laisse u exposé pour un de ses fils uniquement, fils qu'on note w .
On a alors dans ce premier sous-cas :

$$r_{u,1} := \max \left\{ d_w + \sum_{x \in F(u), x \neq w} c_x \mid w \in F(u) \right\} \cup \{-1\}$$

Notons que dans le cas présent, pour tout sommet $w \in F(u)$, on a $d_w \geq 0$, et $d_w \geq c_w$, donc le cas où aucun des fils n'a accès à u est bien pris en compte. Par ailleurs, pour tout $x \in F(u)$, on a bien $c_x \geq 0$.

De plus, on a bien $r_{u,1} = -1$ ssi u est une feuille.

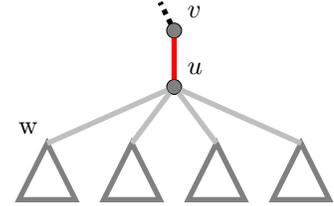


- L'arête mère est recouverte par un couplage "développé par le haut".
On pose alors (puisqu'on restreint l'accès à u pour tous les sous-arbres) :

$$r_{u,2} := \sum_{x \in F(u)} c_x$$

(car pour tout $x \in F(u)$, $c_x \geq 0$).

Si u est une feuille, $r_{u,2}$ vaut 0 par convention.



- Tout un couplage "arrive par le haut", par exemple :



On pose alors :

$$r_{u,3} := \max \left\{ a_x + \sum_{y \in F(u) \setminus \{x\}} \max(b_y, c_y) \mid x \in F(u), a_x \geq 0 \right\} \cup \{-1\}$$

(pour tout $y \in F(u)$, $\max(b_y, c_y) \geq 0$ puisque $c_y \geq 0$).

- Une nouvelle arête apparue suite à une évolution dans un sous-arbre a été développée, en terminant sur l'arête mère, par exemple :

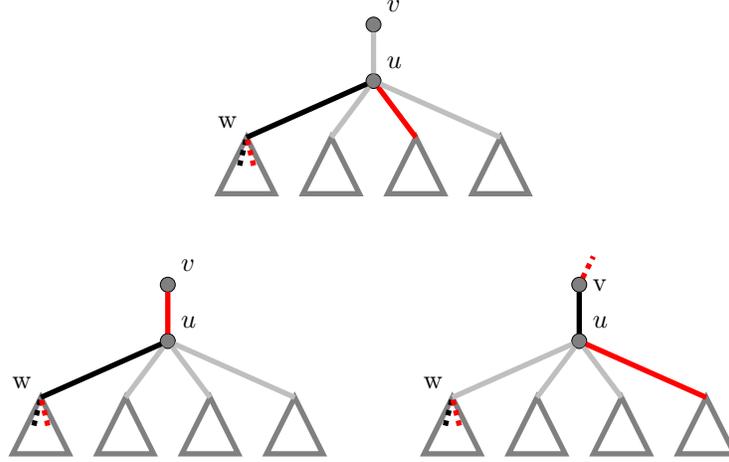


On pose cette fois :

$$r_{u,4} := \max \left\{ e_x + \sum_{y \in F(u) \setminus \{x\}} \max(b_y, c_y) \mid x \in F(u), e_x \geq 0 \right\} \cup \{-1\}$$

($\forall y \in F(u), \max(b_y, c_y) \geq 0$ est toujours vraie)

- Une nouvelle arête apparue suite à une évolution dans un sous-arbre a été développée, en passant éventuellement par l'arête mère, voici deux exemples :



Alors on pose, pour tout $w \in F(u)$,

$$m_w := \max \left\{ a_x + \sum_{y \in F(u) \setminus \{w, x\}} \max(b_y, c_y) \mid x \in F(u) \setminus \{w\}, a_x \geq 0 \right\} \cup \{-1\}$$

On pose ensuite :

$$r_{u,5} := \max \{ e_w + m_w \mid e_w \geq 0, m_w \geq 0 \} \cup \{-1\}$$

Finalement, pour ce cas :

$$a_u = r_{u,2}$$

$$b_u = r_{u,3}$$

$$c_u = \max(r_{u,1}, r_{u,5})$$

$$d_u = \max(r_{u,1}, r_{u,4}, r_{u,5})$$

$$e_u = r_{u,5}$$

2.2.3 Conclusion et complexité de l'algorithme

Rappelons que p est l'unique voisin de r . On calcule alors $d_p = \mu_{=3}(T, M)$.

Puisqu'au cours de l'algorithme on considère chaque arête au plus deux fois (en tant qu'arête "mère" et comme arête d'un sous-arbre), on a une complexité en $O(m)$, en notant m le nombre d'arêtes. Et étant donné qu'on travaille sur des arbres, on a donc une complexité finale en $O(|V|)$.

2.3 Correction de l'algorithme

Théorème 2.1. *Si $w \in M$, alors*

$$d_u = 1 + \sum_{w \in F(u)} c_w$$

$$e_u = \max \left\{ a_w + \sum_{x \in F(u), x \neq w} \max(b_x, c_x) \mid w \in F(u), a_w \geq 0 \right\} \cup \{-1\}$$

et

$$a_u = b_u = c_u = -1$$

La preuve de cet algorithme est rédigée en annexe C.

Théorème 2.2. *Si il existe $w \in F(u)$ tel que $uw \in M$, alors $a_u = b_u = -1$.*

Par ailleurs, si on pose

$$\begin{aligned}
r_{u,1} &:= d_w + \sum_{x \in F(u), x \neq w} c_x \\
M_u &:= \left\{ a_x + \sum_{y \in F(u) \setminus \{w, x\}} \max(b_y, c_y) \mid x \in F(u) \setminus \{w\}, a_x \neq -1 \right\} \\
r_{u,2} &:= \begin{cases} b_w + \max M_u \cup \{-1\} & \text{si } b_w \geq 0 \text{ et } \max M_u \cup \{-1\} \geq 0 \\ -1 & \text{sinon} \end{cases} \\
\text{et } r_{u,3} &:= \begin{cases} b_w + \sum_{x \in F(u) \setminus \{w\}} \max(b_x, c_x) & \text{si } b_w \geq 0 \\ -1 & \text{sinon} \end{cases}
\end{aligned}$$

alors,

$$\begin{aligned}
c_u &= \max(r_{u,1}, r_{u,2}) \\
d_u &= \max(r_{u,1}, r_{u,2}, r_{u,3}) \\
e_u &= r_{u,2}
\end{aligned}$$

Théorème 2.3. *Si $u \notin V_M$, on pose*

$$\begin{aligned}
r_{u,1} &:= \max \left\{ d_w + \sum_{x \in F(u), x \neq w} c_x \mid w \in F(u) \right\} \\
r_{u,2} &:= \sum_{x \in F(u)} c_x \\
r_{u,3} &:= \max \left\{ a_x + \sum_{y \in F(u) \setminus \{x\}} \max(b_y, c_y) \mid x \in F(u), a_x \geq 0 \right\} \cup \{-1\} \\
r_{u,4} &:= \max \left\{ e_x + \sum_{y \in F(u) \setminus \{x\}} \max(b_y, c_y) \mid x \in F(u), e_x \geq 0 \right\} \cup \{-1\} \\
\forall w \in F(u), m_w &:= \max \left\{ a_x + \sum_{y \in F(u) \setminus \{w, x\}} \max(b_y, c_y) \mid x \in F(u) \setminus \{w\}, a_x \geq 0 \right\} \cup \{-1\} \\
r_{u,5} &:= \max \{ e_w + m_w \mid e_w \geq 0, m_w \geq 0 \} \cup \{-1\}
\end{aligned}$$

alors,

$$\begin{aligned}
a_u &= r_{u,2} \\
b_u &= r_{u,3} \\
c_u &= \max(r_{u,1}, r_{u,5}) \\
d_u &= \max(r_{u,1}, r_{u,4}, r_{u,5}) \\
e_u &= r_{u,5}
\end{aligned}$$

Les preuves des théorèmes 2.2 et 2.3 sont similaires à celle du théorème 2.1, avec cependant plus de cas à considérer.

De ces théorèmes, on déduit immédiatement le suivant :

Théorème 2.4. *L'algorithme 1 (dont le pseudo-code est dans l'annexe B) prend un arbre T et un couplage M en entrée et calcule $\mu_{=3}(T, M)$ avec une complexité $O(|V(T)|)$.*

3 Split graphs

L'objectif est ici de calculer $\mu_{=3}(G, M)$ pour tout split graph $G = (I \cup K, E)$ et tout couplage $M \subseteq E$. On commence par se restreindre aux "soleils", un cas particulier des split graphs. Puis on étudie des pistes de généralisation.

3.1 Cas des soleils

3.1.1 Présentation

On se place dans un cas particulier des split graphs, les soleils (qui ont été définis dans la partie 1.2). À titre d'illustration, le graphe sur la figure 3 est un soleil.

On appelle *arête extérieure* une arête reliant un sommet de I à un sommet de K , et par opposition les arêtes de la clique sont les arêtes intérieures.

On remarque que si une arête extérieure est dans le couplage, elle ne pourra jamais être augmentée, donc cela revient à supprimer cette arête et les deux sommets qui la composent. On peut donc supposer que les arêtes de M sont uniquement des arêtes intérieures.

Par ailleurs, étant donné que deux arêtes extérieures ayant le même voisin ne pourront jamais être toutes les deux dans M , on peut supposer que chaque sommet $k \in K$ a au plus un voisin dans I : $\forall k \in K, |N(k) \cap I| \leq 1$.

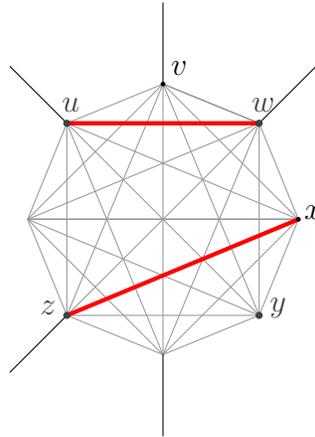


FIGURE 3 – Un exemple de soleil avec un couplage initial $M = \{xz, uw\}$.

On attribue des types aux sommets de K :

- $k \in K$ est de type *A* si $|N(k) \cap I| = 1$ et $\exists l \in K, kl \in M$, i.e. si k a un voisin dans I , et est couvert par M . Par exemple sur la figure 3 les sommets u, w et z sont de type *A*.
- $k \in K$ est de type *B* si $|N(k) \cap I| = 0$ et $\exists l \in K, kl \in M$, i.e. si k n'a pas de voisin dans I et est couvert par M . Par exemple, x est de type *B*.
- $k \in K$ est de type *C* si $|N(k) \cap I| = 0$ et $\forall l \in K, kl \notin M$, i.e. si k n'a pas de voisin dans I et est exposé pour M . Par exemple, y est de type *C*.
- $k \in K$ est de type *D* si $|N(k) \cap I| = 1$ et $\forall l \in K, kl \notin M$, i.e. si k a un voisin dans I et est exposé pour M . Par exemple, v est de type *D*.

On en déduit des types pour les arêtes intérieures, par exemple une arête de type *AA* est une arête (de M) reliant deux sommets de type *A*. On utilisera également les types *AB* et *BB*.

On pose $A_M := \{k \in K \mid |N(k) \cap I| = 1 \text{ et } \exists l \in K, kl \in M\}$, c'est à dire l'ensemble des sommets de type *A*. On construit de même B_M, C_M et D_M l'ensemble des sommets de types respectifs *B, C* et *D*.

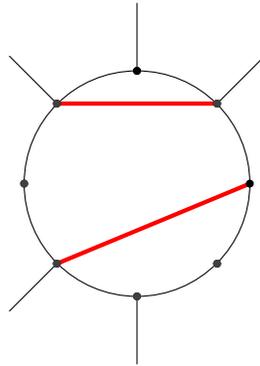
On pose également $AA_M := \{xy \in M \mid x \in A_M, y \in A_M\}$, $BB_M := \{xy \in M \mid x \in B_M, y \in B_M\}$. On pose enfin $AB_M := \{(x, y) \mid xy \in M, x \in A_M, y \in B_M\}$. Notons que AB_M est un ensemble de couples, et non d'arêtes.

Pour tout $x \in K$ tel que $|N(x) \cap I| = 1$, on note x' l'unique sommet $i \in I$ vérifiant $ix \in E$.

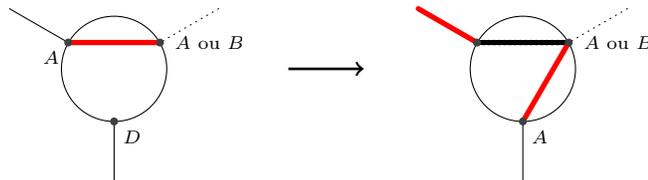
3.1.2 L'algorithme

Voici une description informelle de l'algorithme. Le pseudo-code de l'algorithme est disponible dans l'annexe D.

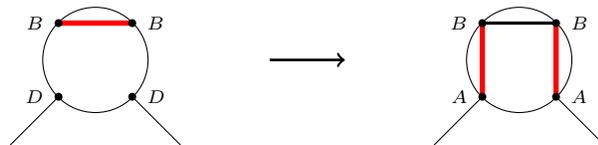
Commençons par adopter une représentation plus simple de nos soleils, dans laquelle les arêtes de la clique sont implicites.



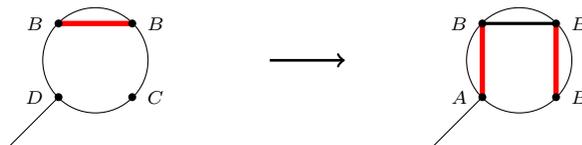
- Tant qu'il y a des sommets de type D :
 - S'il y a des sommets de type A :



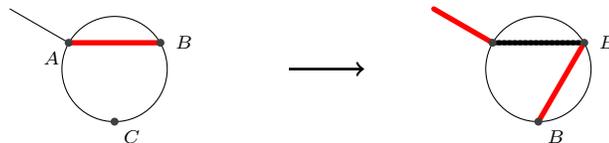
- Sinon, il y a des sommets de type B :
- S'il y a deux sommets de type D :



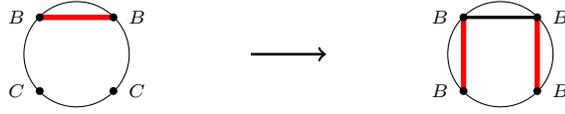
- Sinon, s'il y a un sommet de type D et un sommet de type C :



- Sinon, on ne peut plus rien faire.
- Tant qu'il y a des sommets de type C :
- S'il y a une arête de type AB :



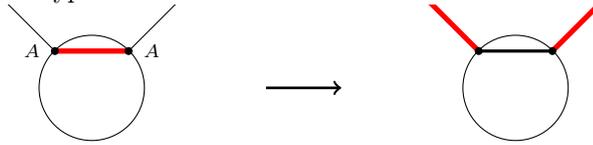
— Sinon, s'il y a une arête de type BB et au moins deux sommets de type C :



— Sinon, s'il y a une arête de type AA :



— Tant qu'il y a des arêtes de type AA :



Notons n le nombre de sommets du soleil S . Chaque boucle a une complexité en $O(1)$, et il y a au plus $\mu_{=3}(S, M) - |M| = O(n)$ appels de boucle (puisqu'à chaque boucle on augmente de 1 la taille du couplage), d'où une complexité finale en $O(n)$.

3.1.3 Preuve de correction

On suppose $M \neq \emptyset$ (en effet, sinon $\mu_{=3}(S, M) = 0$).

Lemme 3.1. *Si $A_M \neq \emptyset$, alors l'algorithme termine et renvoie $\mu_{=3}(S, M)$. Par ailleurs,*

$$\text{si } |AB_M| - |C_M| > 0, \text{ alors } \mu(S) - \mu_{=3}(S, M) = \left\lfloor \frac{|AB_M| - |C_M|}{2} \right\rfloor$$

et sinon, $\mu(S) = \mu_{=3}(S, M)$.

Dans tous les cas, $\mu_{\geq 3}(S, M) = \mu_{=3}(S, M)$.

Démonstration. Supposons $A_M \neq \emptyset$.

Commençons par le cas où $|AB_M| \geq |C_M|$. Montrons qu'alors l'algorithme renvoie un couplage qui couvre tous les sommets sauf éventuellement un.

Il suffit en fait de considérer l'exécution de l'algorithme. La première partie de l'algorithme permet de "se débarrasser" des sommets de type D sans changer ni $|AB_M|$ ni $|C_M|$.

Si $|AB_M| > 0$. Comme $|AB_M| \geq |C_M|$, toutes les arêtes de type AB peuvent être augmentées. Enfin, s'il existe deux sommets de type C exposés, on peut sélectionner une arête de type BB pour l'augmenter vers ces deux sommets. On augmente alors les arêtes de type AA . Alors toutes les arêtes extérieures sont couvertes, et il reste au plus un sommet de type C . On a le résultat.

Sinon, $|AB_M| = 0$. Alors $|C_M| = 0$, donc tous les sommets sont bien couverts.

Supposons maintenant que $|AB_M| > |C_M|$. On est dans le cas de la Figure 4.

Considérons l'exécution de l'algorithme du théorème 1.1 (calculant $\mu_{\leq 3}(G, M)$ pour tout graphe G et couplage initial M , introduit dans l'article [NSW15b]) sur le graphe S .

L'algorithme commence par retirer au graphe G les arêtes n'appartenant pas à M mais dont les deux sommets sont couverts. Cela ne change pas la valeur du couplage optimal puisque ces arêtes ne pourront de toute façon jamais faire partie d'un chemin augmentant de taille 3. On obtient alors un graphe S' . On calcule alors M' , de taille $\mu(S')$. Il a été prouvé dans l'article [NSW15b] que $\mu_{\leq 3}(S, M) = \mu(S')$. Il suffit donc de prouver que notre algorithme atteint $\mu(S')$, et de calculer $\mu(S) - \mu(S')$.

On sait déjà que :

$$\mu(S) = |A_M| + |D_M| + \left\lfloor \frac{|C_M| + |B_M|}{2} \right\rfloor = 2|AA_M| + |AB_M| + |D_M| + |BB_M| + \left\lfloor \frac{|AB_M| + |C_M|}{2} \right\rfloor$$

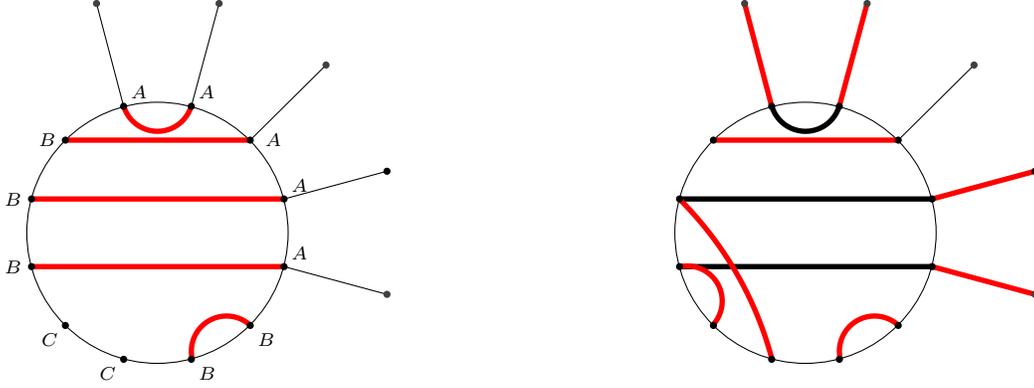


FIGURE 4 – Cas où $|AB_M| < |C_M|$: couplage initial M , puis un couplage optimal obtenu à partir de M .

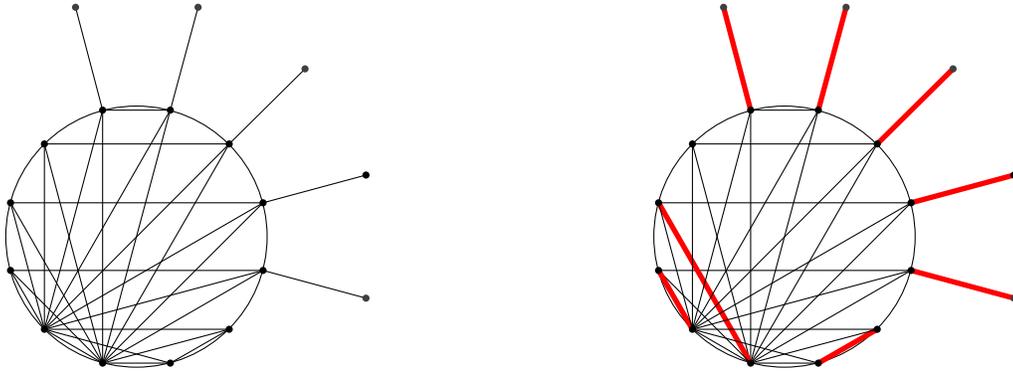


FIGURE 5 – Exécution de l'algorithme pour $\mu_{\leq 3}(S, M)$: S' et couplage de taille $\mu(S')$.

Montrons maintenant que :

$$\mu(S') = 2|AA_M| + |AB_M| + |D_M| + |C_M| + |BB_M|$$

On peut déjà remarquer que, si on note F l'ensemble des arêtes extérieures de S , et E' l'ensemble des arêtes de S' , alors $E' = F \cup AA_M \cup \{xy | (x, y) \in AB_M\} \cup BB_M \cup (C_M \times \{x | \exists y, xy \in M\})$.

Soit un couplage $M' \subseteq E'$ de taille $\mu(S')$.

Soit uv une arête extérieure, avec $u \in K$ et $v \in I$. Soit $uv \in M'$, soit il existe $w \in K$ tel que $uw \in M'$ (car sinon u et v sont exposés et on peut rajouter uv au couplage), auquel cas on peut échanger uw avec uv . On peut donc supposer que toutes les arêtes extérieures sont couvertes, ce qui implique que $M' = F \cup N$, où $N \subseteq BB_M \cup C_M \times \{b | \exists x, (x, b) \in AB_M \text{ ou } xb \in BB_M\}$ (car les deux sommets des arêtes de type AA sont couverts par des arêtes de F).

Montrons que si on a $cb \in N$, avec $c \in C_M$, alors on peut supposer qu'il existe x tel que $(x, b) \in AB_M$. En effet, si pour tout x , $(x, b) \notin AB_M$ et si de plus il existe y tel que $yb \in M$, alors comme $|AB_M| > |C_M|$, il existe z non couvert ($\forall c \in C_M, cz \notin N$), et donc on peut remplacer cb par cz . Donc, on peut supposer que $N \subseteq BB_M \cup (C_M \times \{b | \exists x, (x, b) \in AB_M\})$. Il est assez évident qu'il existe un ensemble d'arêtes deux à deux disjointes de taille $|BB_M| + |C_M|$ dans $BB_M \cup (C_M \times \{b | \exists x, (x, b) \in AB_M\})$, et qu'on ne peut faire mieux, puisque $|C_M| < |AB_M|$. Donc $|N| = |C_M| + |BB_M|$, et comme de plus $|F| = 2|AA_M| + |AB_M| + |D_M|$, on a finalement $\mu(S') = |F| + |N| = 2|AA_M| + |AB_M| + |D_M| + |BB_M| + |C_M|$.

On sait par la preuve de l'algorithme pour $\mu_{\leq 3}$ que :

$$\mu_{=3}(S, M) \leq \mu_{\leq 3}(S, M) = \mu(S') = 2|AA_M| + |AB_M| + |D_M| + |BB_M| + |C_M|$$

On peut voir que notre algorithme renvoie un couplage de cette taille, et donc on a égalité, d'où :

$$\mu(S) - \mu_{=3}(S, M) = \mu(S) - \mu(S') = \left\lfloor \frac{|AB_M| + |C_M|}{2} \right\rfloor - |C_M| = \left\lfloor \frac{|AB_M| - |C_M|}{2} \right\rfloor$$

□

Lemme 3.2. Si $A_M = \emptyset$, alors l'algorithme termine et renvoie $\mu_{=3}(S, M)$. Par ailleurs,

$$\mu_{\leq 3}(S, M) - 1 \leq \mu_{=3}(S, M) \leq \mu_{\leq 3}(S, M)$$

et $\mu_{=3}(S, M) = \mu_{\leq 3}(S, M) - 1$ si et seulement si $|D_M| > 0$ et $|C_M| = 0$.

Démonstration. On distingue deux cas, en fonction de $|C_M|$.

- Si $|C_M| = 0$:
 - Si $|D_M| = 0$, le couplage est maximum, donc $\mu_{=3}(S, M) = |M| = \mu(S)$, le lemme est vrai dans ce cas.
 - Si $|D_M| = 1$ (cf Figure 6), il n'y a pas de chemin augmentant de taille 3, donc $\mu_{=3}(S, M) = |M|$, et $\mu_{\leq 3}(S, M) = \mu(S) = |M| + 1$, le lemme est également vrai dans ce cas.

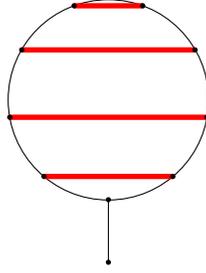


FIGURE 6 – Illustration du cas où $|C_M| = 0$ et $|D_M| = 1$.

- Supposons maintenant $|D_M| \geq 2$.

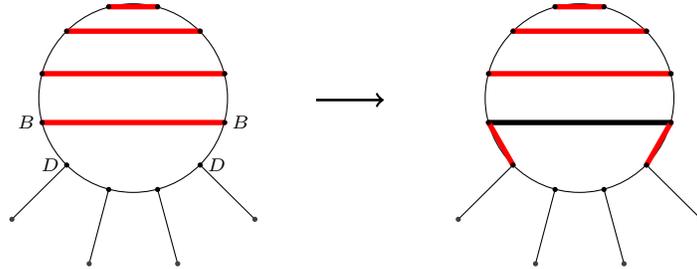


FIGURE 7 – Exemple de cas dans lequel $|AA_M| = 0, |C_M| = 0$: on peut seulement augmenter une arête BB vers des sommets de type D .

Il est évident que M n'est pas maximum. Une séquence pour obtenir un couplage maximum commence nécessairement par un chemin de la forme (x, y, z, t) , avec $x, t \in D_M, x \neq t$, et $yz \in BB_M$ (c'est la seule augmentation possible, cf figure 7; et c'est ce que fait l'algorithme). Par symétrie, on peut fixer x, y, z, t tels que décrits, et en déduire que $\mu_{=3}(S, M) = \mu_{=3}(S, M \setminus \{yz\} \cup \{xy, zt\})$. Posons donc $M' := M \setminus \{yz\} \cup \{xy, zt\}$. M' vérifie $|AB_{M'}| = 2, |C_{M'}| = 0$. On peut appliquer le lemme 3.1 : l'algorithme calcule $\mu_{=3}(S, M')$, et on a par ailleurs $\mu_{=3}(S, M) = \mu_{=3}(S, M') = \mu(S) - \left\lfloor \frac{|AB_{M'}| - |C_{M'}|}{2} \right\rfloor = \mu(S) - 1$.

Le lemme est alors prouvé, puisque $\mu_{\leq 3}(S, M) = \mu(S)$, ce qui est évident puisqu'il suffit d'ajouter les arêtes extérieures au couplage M pour obtenir un couplage de taille $\mu(S) = |D_M| + |B_M| = \frac{n}{2}$ (en notant n le nombre de sommets de S).

— Si $|C_M| > 0$:

Considérons l'exécution de l'algorithme, et montrons qu'il atteint $\mu(S)$ dans tous les cas.

— Si $|D_M| \geq 2$: l'algorithme commence par augmenter une arête de type BB vers deux sommets de type D . On a donc $b_1b_2 \in BB_M$, $d_1, d_2 \in D_M$, $d_1 \neq d_2$, et l'algorithme calcule alors un couplage pour $M' := (M \setminus \{b_1b_2\}) \cup \{d_1b_1, b_2d_2\}$. Or, $|A_{M'}| = |AB_{M'}| = 2$. On a alors deux cas.

Soit $|C_{M'}| = 1$, auquel cas $|AB_{M'}| > |C_{M'}|$, et alors par le lemme 3.1 l'algorithme renvoie :

$$\mu_{=3}(S, M') = \mu(S) - \left\lfloor \frac{|AB_{M'}| - |C_{M'}|}{2} \right\rfloor = \mu(S).$$

Comme on a nécessairement $\mu(S) \geq \mu_{=3}(S, M) \geq \mu_{=3}(S, M') = \mu(S)$, on a bien le résultat.

Soit $|C_{M'}| \geq 2$, auquel cas $|AB_{M'}| \leq |C_{M'}|$, et donc par le lemme 3.1 $\mu_{=3}(S, M') = \mu(S)$. On en déduit de même que $\mu_{=3}(S, M) = \mu(S)$.

— Si $|D_M| = 1$: ce cas se traite exactement de la même manière que le cas précédent, en utilisant également le lemme 3.1.

— Si $|D_M| = 0$: tant qu'il reste plus de deux sommets de type C , l'algorithme augmente une arête de type BB vers des sommets de type C . C'est la seule possibilité (toutes les arêtes jouent des rôles symétriques, de même pour les sommets de type C) et ne pas le faire est sous-optimal, donc l'algorithme est bien optimal dans ce cas. Et à la fin tous les sommets sont couverts sauf éventuellement un. □

Ce lemme démontre au passage que dans tous les cas, $\mu_{\leq 3}(S, M) = \mu(G)$.

De ces deux lemmes, on déduit immédiatement le théorème suivant :

Théorème 3.3. *Pour tout soleil S et tout couplage M de S , l'algorithme 2 calcule, avec une complexité linéaire en le nombre de sommets de S , un couplage de taille $\mu_{=3}(S, M)$ que l'on peut obtenir à partir de M en n'augmentant que des chemins de longueur 3.*

3.2 Pistes pour le cas général

L'objectif maintenant est de résoudre $\mu_{=3}(G, M)$ pour tout split graph G . On cherche à adapter le cas des soleils. Je n'ai abordé ce problème qu'en fin de stage et je n'ai donc pas obtenu de résultats, je présente seulement quelques pistes qui m'ont parues pertinentes.

On peut commencer par le cas où les arêtes du couplage initial sont toutes dans la clique. Une possibilité est, pour éviter d'avoir trop de cas à traiter et garder une bonne complexité, de chercher à précalculer quelles arêtes extérieures seront utilisées, pour se ramener à un soleil (par exemple, voir figure 8). Pour cela, on cherche à maximiser le nombre d'arêtes extérieures indépendantes, et ensuite à privilégier certaines arêtes. Mettre des poids sur les arêtes et chercher un couplage de poids maximal dans le graphe biparti associé (ce qui se fait polynomialement, avec l'algorithme hongrois par exemple [Kuh10]) est une première piste. On peut en voir un exemple sur la figure 9. Mais cela ne peut pas se faire uniquement de cette manière, car cette méthode ne privilégie aucun de deux cas de la figure 10, ce qui pose problème.

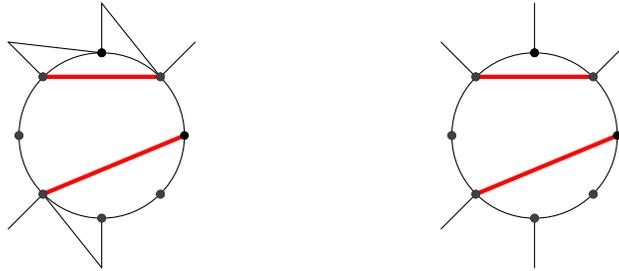


FIGURE 8 – Un exemple simple : à partir d'un split graph avec un couplage uniquement dans la clique (dessin de gauche), on se ramène à un soleil dont le couplage maximum aura la même taille.

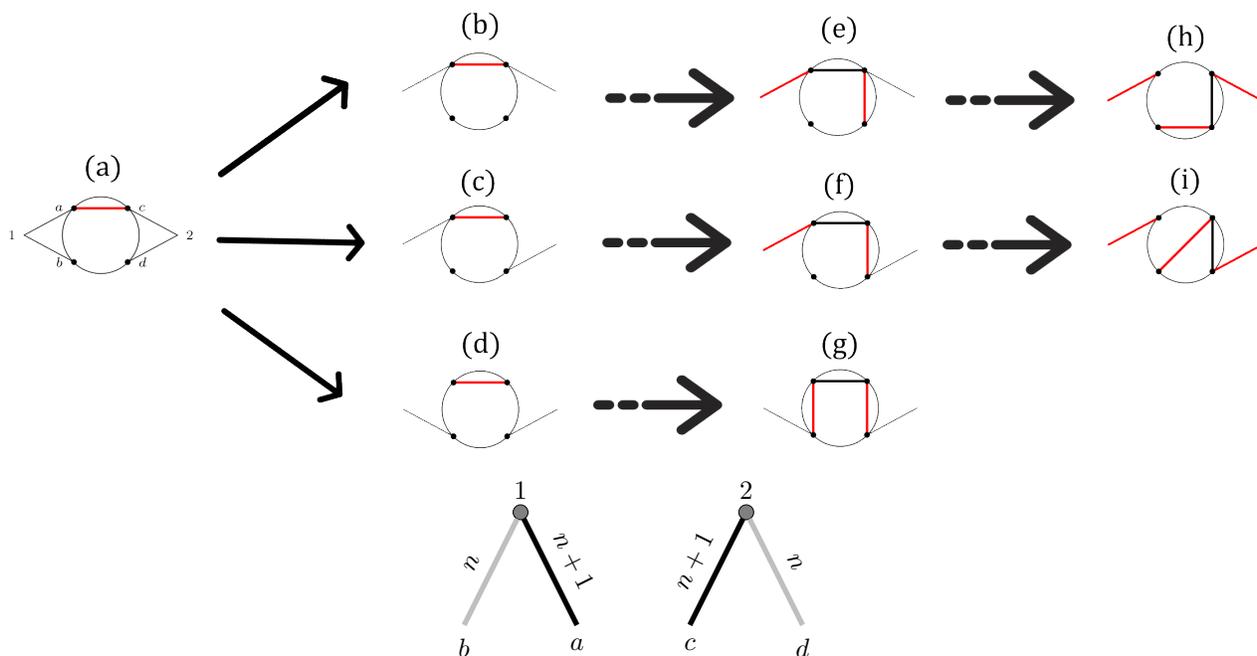


FIGURE 9 – Un deuxième exemple : à partir de la configuration (a), par symétrie, on a 3 configurations possibles (et intéressantes, car prendre seulement une arête, voire aucune, est sous-optimal) pour se ramener à un soleil : (b), (c) et (d). Les configurations (b) et (c) peuvent amener à un couplage de taille 3 (configurations respectives (h) et (i)), alors que la configuration (d) n’offre que la configuration (g) comme possibilité, qui donne un couplage de taille uniquement 2. Or, si on considère le graphe biparti associé à la configuration (a), en favorisant les arêtes adjacentes à un sommet de type A (on mets un poids $n + 1$ sur ces arêtes, et n sur les autres, avec n plus grand que le nombre de sommets de la clique), c’est alors la configuration (b) qui est favorisée, qui mène bien à un couplage maximum. Le choix de n plus grand que le nombre de sommets de la clique permet de favoriser les configurations de soleils avec plus d’arêtes, ce qui semble être toujours plus intéressant.

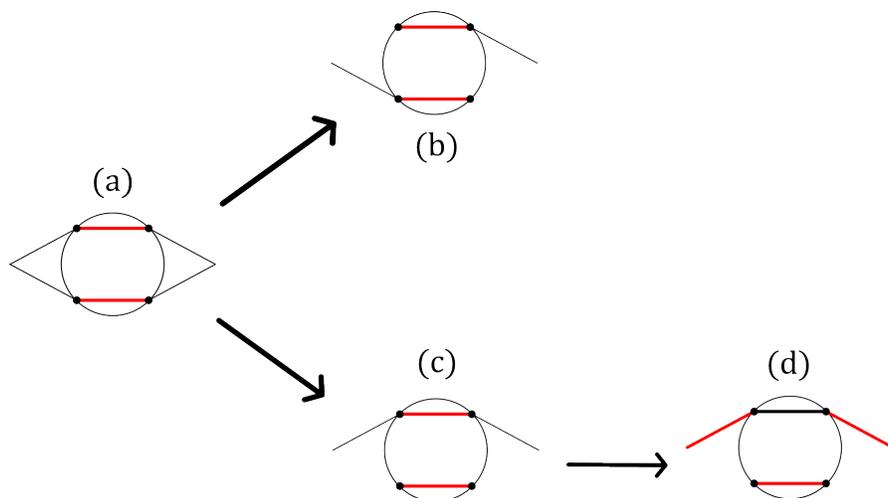


FIGURE 10 – Un autre exemple : ici, par symétrie, avec la configuration initiale (a), on peut sélectionner deux arêtes n’ayant pas la même arête du couplage pour voisine (b) auquel cas on ne peut rien faire de plus, ou sélectionner deux arêtes faisant partie d’un chemin augmentant de longueur 3 (c), ce qui permet d’augmenter le nombre d’arêtes du couplage (d).

Pour le cas où certaines arêtes du couplage sont des arêtes extérieures, le problème se complique encore. On peut chercher à précalculer lesquelles on devra augmenter, en se restreignant aux augmentations vers d’autres arêtes extérieures (on peut montrer que pour un couplage maximum quelconque, on peut trouver un couplage de même taille atteignable en utilisant au plus un chemin augmentant de la forme (x, y, z, t) , avec zt ou xy dans la clique). Mais ce précalcul, même restreint, reste compliqué.

4 Conclusion

Mon objectif pour ce stage était de m’approprier ces résultats et de prolonger les recherches, soit avec de nouvelles preuves de NP-complétude sur des problèmes encore plus restreints, soit en trouvant des algorithmes polynomiaux pour des classes de graphes plus larges que celles considérées dans les articles précédents. Je me suis surtout intéressée à la deuxième perspective, qui me paraissait plus concrète. J’ai commencé par m’intéresser aux calculs de $\mu_{=k}(G, M)$ pour k quelconque, que j’ai finalement restreint au cas $k = 3$, déjà suffisamment complexe et intéressant.

J’ai commencé par les chemins, les chenilles et des structures de graphes appelées lobster (un prolongement des chenilles), pour réussir à prolonger jusqu’aux arbres en général, par un algorithme de programmation dynamique qui est présenté dans la partie 2.

Une autre piste était l’étude du problème dans la classe des graphes d’intervalles. Elle n’a pas abouti à un résultat sur cette classe, mais nous avons dans la partie 3 des résultats sur la classe des split graphs, qui est une autre sous-classe des graphes cordaux. En particulier, nous avons pu construire un algorithme linéaire pour trouver un couplage maximum dans le cas des soleils.

Une première piste de réflexion envisageable est donc la poursuite du cas pour les split graphs puis les graphes d’intervalles, comme je l’ai expliqué dans la partie 3.2.

Il serait par ailleurs intéressant de poursuivre les recherches sur la complexité des problèmes de calcul de $\mu_{\leq k}$ et $\mu_{=k}$ dans des classes de graphes plus restreintes, par exemple pour le calcul de $\mu_{\leq k}$ dans les arbres.

D’un point de vue plus personnel, ce stage au sein de l’équipe COATI m’a permis de véritablement me rendre compte du monde de la recherche. J’ai pu en apprécier les bons côtés – beaucoup de dialogues, une bienveillance globale agréable, des personnes particulièrement intéressantes, intellectuellement et socialement – comme les moins bons, notamment la difficulté de formaliser et prouver ses résultats, qui m’a pénalisée plusieurs semaines. Mais le travail n’en a été que plus enrichissant.

Références

- [Ber57] C. Berge. Two theorems in graph theory. *Proceedings of the National Academy of Sciences of the United States of America*, 43(9) :842–844, September 1957.
- [BGN18] Julien Bensmail, Valentin Garnero, and Nicolas Nisse. On improving matchings in trees, via bounded-length augmentations. *Discrete Applied Mathematics*, 250 :110–129, 2018.
- [Edm65] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17 :449–467, 1965.
- [Kuh10] Harold W. Kuhn. The hungarian method for the assignment problem. In *50 Years of Integer Programming 1958-2008 : From the Early Years to the State-of-the-Art*, pages 29–47, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [NSW15a] Nicolas Nisse, Alexandre Salch, and Valentin Weber. Comment appliquer les chaînes augmentantes pour atterrir à l’heure ? In *ALGOTEL 2015 - 17èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, Beaune, France, June 2015.
- [NSW15b] Nicolas Nisse, Alexandre Salch, and Valentin Weber. Recovery of disrupted airline operations using k-Maximum Matching in Graphs. Research Report RR-8679, Inria Sophia Antipolis ; INRIA, February 2015.

Annexes

Annexe A : L'équipe

J'ai été accueillie pour ce stage par l'équipe COATI (pour Combinatoire, Optimisation et Algorithmes pour les Télécommunications), qui est une équipe regroupant des chercheurs de l'INRIA et du laboratoire I3S. Cette équipe étudie diverses problématiques autour des réseaux (de transports en particulier), ainsi que des problèmes algorithmiques et combinatoires, en théorie des graphes notamment.

Elle est constituée d'une dizaine de chercheurs permanents, et autant de doctorants et stagiaires qui ont donc des domaines de recherches très variés. Cette équipe est très accueillante et dynamique, ce qui est génial pour un premier stage. J'ai ainsi pu discuter avec de nombreuses personnes aux profils variés, de leurs sujets de recherches mais aussi de la vie de chercheur d'une manière plus globale, ce qui a été très enrichissant.

J'ai notamment beaucoup partagé avec d'autres stagiaires, ce qui est vraiment un plus puisqu'on a pu discuter de nos sujets, y apporter de nouveaux regards, et aussi s'entraider pour des aspects plus pratiques (aide en LaTeX, transports, relecture des rapports...).

Enfin, j'ai été encadrée par deux chercheurs : Julien Bensmail, également maître de conférence à l'université de Nice, et Nicolas Nisse, chargé de recherche qui de plus enseigne en classes préparatoires. Ces deux personnes ont été présentes malgré un calendrier chargé, et m'ont vraiment bien encadrée, réfléchissant parfois avec moi et me donnant toujours des retours très pertinents sur mon travail. J'en profite donc pour les remercier, ainsi que toutes les personnes dont j'ai parlé précédemment, car elles m'ont permis de faire ce stage dans les meilleures conditions possibles. Enfin, un remerciement particulier à Nicolas Nisse qui nous a fait, aux autres stagiaires et à moi, un cours condensé de complexité paramétrée, ce qui était pour nous une découverte très instructive.

Annexe B : L'algorithme de programmation dynamique

On peut supposer que pour tout arbre T , caractérisé par V l'ensemble de ses sommets, T est représenté par F et r tels que T enraciné en r , r est une feuille, et pour tout u , $F(u)$ représente la liste des enfants de u . Par ailleurs, on suppose construit P tel que pour tout u , $P[u]$ est le père de u (par convention $P[r] = None$; le coût de cette construction est en $O(|V|)$). On suppose finalement que les sommets sont ordonnés de sorte que pour tout u , $u < P[u]$ (il suffit de considérer qu'on retire récursivement les feuilles de l'arbre).

Algorithme 1 RESOLUTIONARBRESK=3(T, M)

Entrée : arbre T , caractérisé par V, r, F et P décrits juste avant couplage M

```
1:  $n \leftarrow |V|$ 
2:  $A, B, C, D, E \leftarrow$  tableau de taille  $n$  initialisé à -1
3: pour  $u \in V \setminus \{r\}$  faire
4:   // ce parcours est fait dans l'ordre décrit précédemment
5:   si  $F[u]$  est vide alors
6:     si  $\{u, v\} \in M$  alors
7:        $D[u] \leftarrow 1$ 
8:       // les autres valent -1
9:     sinon
10:      //  $\{u, v\} \notin M$ 
11:       $A[u] \leftarrow 1$ 
12:       $C[u] \leftarrow 0$ 
13:       $D[u] \leftarrow 0$ 
14:    fin si
15:  sinon
16:    //  $A[w], B[w], \dots$  déjà calculés, pour tout  $w \in F[u]$ 
17:    si  $\{u, P[u]\} \in M$  alors
18:      // Calcul de  $D[u]$ 
19:      // pour tout  $w \in F[u]$ ,  $C[w] \geq 0$ 
```

```

20:       $D[u] \leftarrow 1 + \sum_{w \in F(u)} C[w]$ 
21:      // Calcul de  $E[u]$ 
22:      pour  $w \in F[u]$  faire
23:          si  $A[w] \geq 0$  et  $\forall x \in F[u] \setminus \{w\}, \max(B[x], C[x]) \geq 0$  alors
24:               $E[u] \leftarrow \max \left( E[u], A[w] + \sum_{x \in F(u) \setminus \{w\}} \max(B[x], C[x]) \right)$ 
25:          fin si
26:      fin pour
27:      //  $A[u] = -1, B[u] = -1D[u] = -1$ 
28:      sinon si il existe  $w \in F[u]$  tel que  $\{u, w\} \in M$  alors
29:          Soit  $w$  tel sommet
30:          // Calcul de  $r_{u1}$ 
31:           $r_{u1} \leftarrow D[w] + \sum_{x \in F(u) \setminus \{w\}} \max(B[x], C[x])$ 
32:          // Calcul de  $r_{u2}$ 
33:           $r_{u2} \leftarrow -1$ 
34:          si  $B[w] \geq 0$  alors
35:              pour  $x \in F[u] \setminus \{w\}$  faire
36:                  si  $A[x] \geq 0$  et  $\forall y \in F[u] \setminus \{w, x\}, \max(B[y], C[y]) \geq 0$  alors
37:                       $r_{u2} \leftarrow \max \left( r_{u2}, B[w] + A[x] + \sum_{y \in F(u) \setminus \{w, x\}} \max(B[y], C[y]) \right)$ 
38:                  fin si
39:              fin pour
40:          fin si
41:          // Calcul de  $r_{u3}$ 
42:           $r_{u3} \leftarrow -1$ 
43:          // rappel :  $r_{u2} = r_{u4}$ 
44:           $C[u] \leftarrow \max(r_{u1}, r_{u2})$ 
45:           $D[u] \leftarrow \max(r_{u1}, r_{u2}, r_{u3})$ 
46:           $E[u] \leftarrow r_{u2}$ 
47:      sinon
48:          // il n'existe pas  $w \in V$  tel que  $\{u, w\} \in M$ 
49:          // Calcul de  $r_{u1}$ 
50:           $r_{u1} \leftarrow -1$ 
51:          pour  $w \in F[u]$  faire
52:              si  $D[w] \geq 0$  alors
53:                   $r_{u1} \leftarrow \max \left( r_{u1}, D[w] + \sum_{x \in F(u) \setminus \{w\}} C[x] \right)$ 
54:              fin si
55:          fin pour
56:          // Calcul de  $r_{u2}$ 
57:           $r_{u2} \leftarrow \sum_{x \in F(u)} C[x]$ 
58:          // Calcul de  $r_{u3}$ 
59:           $r_{u3} \leftarrow -1$ 
60:          pour  $x \in F[u]$  faire
61:              si  $A[x] \geq 0$  alors
62:                   $r_{u3} \leftarrow \max \left( r_{u3}, A[x] + \sum_{y \in F(u) \setminus \{x\}} \max(B[y], C[y]) \right)$ 
63:              fin si
64:          fin pour
65:          // Calcul de  $r_{u4}$ 
66:           $r_{u4} \leftarrow -1$ 

```

```

67:     pour  $x \in F[u]$  faire
68:         si  $E[x] \geq 0$  alors
69:              $r_{u4} \leftarrow \max \left( r_{u4}, E[x] + \sum_{y \in F(u) \setminus \{x\}} \max(B[y], C[y]) \right)$ 
70:         fin si
71:     fin pour
72:     // Calcul de  $r_{u5}$ 
73:      $r_{u5} \leftarrow -1$ 
74:     pour  $w \in F[u]$  faire
75:          $m_w \leftarrow -1$ 
76:         pour  $x \in F[u] \setminus \{w\}$  faire
77:             si  $A[x] \geq 0$  alors
78:                  $m_w \leftarrow \max \left( m_w, A[x] + \sum_{y \in F(u) \setminus \{w, x\}} \max(B[y], C[y]) \right)$ 
79:             fin si
80:         fin pour
81:         si  $E[w] \geq 0$  et  $m_w \geq 0$  alors
82:              $r_{u5} \leftarrow \max(r_{u5}, E[w] + m_w)$ 
83:         fin si
84:     fin pour
85:     // Bilan
86:      $A[u] \leftarrow r_{u2}$ 
87:      $B[u] \leftarrow r_{u3}$ 
88:      $C[u] \leftarrow \max(r_{u1}, r_{u5})$ 
89:      $D[u] \leftarrow \max(r_{u1}, r_{u4}, r_{u5})$ 
90:      $E[u] \leftarrow r_{u5}$ 
91: fin si
92: fin si
93: fin pour
94: retourner  $D[F[r][0]]$ 

```

Annexe C : Preuve du théorème 2.1

On démontre ici le théorème suivant :

Théorème. Si $uv \in M$, alors

$$d_u = 1 + \sum_{w \in F(u)} c_w$$

$$e_u = \max \left\{ a_w + \sum_{x \in F(u), x \neq w} \max(b_x, c_x) \mid w \in F(u), a_w \geq 0 \right\} \cup \{-1\}$$

et

$$a_u = b_u = c_u = -1$$

Démonstration. Par définition, $a_u = b_u = c_u = -1$.

Calculons $d_u = \mu_{=3}(\tilde{T}_u, M)$.

Puisque v est une feuille de \tilde{T}_u , l'arête uv ne pourra jamais être développée. Cela revient donc à supprimer les sommets u et v . Il suffit alors de calculer un couplage maximum pour chaque sous-arbre indépendamment :

$$d_u = \mu_{=3}(\tilde{T}_u, M) = 1 + \sum_{w \in F(u)} \mu_{=3}(T_w, M) = 1 + \sum_{w \in F(u)} c_w$$

Calculons maintenant e_u .

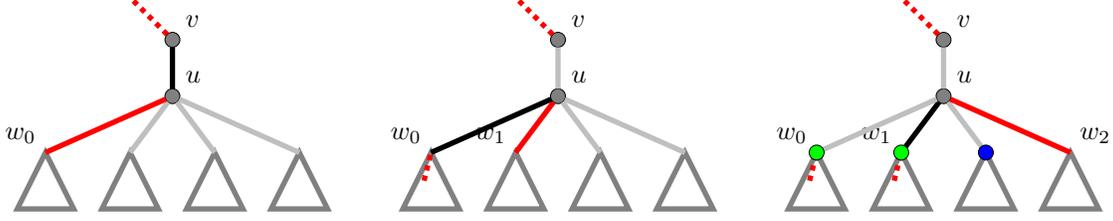


FIGURE 11 – Illustration du développement d'un couplage (départ, augmentation de P_{l_1} , puis de P_{l_2}). Les sommets verts sont dans Γ , les sommets bleus dans Δ . w_2 n'est ni dans Γ ni dans Δ .

Posons

$$z := \max \left\{ a_w + \sum_{x \in F(u), x \neq w} \max(b_x, c_x) \mid w \in F(u), a_w \geq 0 \right\} \cup \{-1\}$$

et montrons que $e_u = z$.

— Montrons d'abord que $e_u \leq z$.

Si $e_u = -1$, l'inégalité est vraie.

Sinon, il existe $w \in F(u), w \notin V_M$ et S une suite de chemins (P_1, P_2, \dots, P_s) tels que le résultat de leur augmentation en partant de $M \setminus \{uv\} \cup \{uw\}$ donne un couplage M^* de taille $e_u =$

$$\max_{w \in F(u), w \notin V_M} \mu_{=3}(T_u, M \setminus \{uv\} \cup \{uw\}).$$

Il existe $k \in \mathbb{N}$, $w_0, w_1, \dots, w_k, w'_0, w'_1, \dots, w'_{k-1} \in V$, $l_0, l_1, \dots, l_{k-1} \in \llbracket 0, s \rrbracket$ tels que :

- $\forall 0 \leq i \leq k, w_i \in F(u)$
- $\forall 0 \leq i < k, w'_i \in F(w_i)$
- $\forall 0 \leq i < k, P_{l_i} = (w'_i, w_i, u, w_{i+1})$ (ou (w_{i+1}, u, w_i, w'_i)).

On peut en effet les construire récursivement :

On pose $w_0 := w$.

Soit $i > 0$. On suppose construits w_0, w_1, \dots, w_i , ainsi que $w'_0, w'_1, \dots, w'_{i-1}$ et l_0, l_1, \dots, l_{i-1} . Si $uw_i \in M$, alors on pose $k := i$ et on a terminé. Sinon, on pose w_{i+1} et w'_i les uniques sommets tels que $w_{i+1} \in F(u)$ et $w'_i \in F(w_i)$ et $(w'_i, w_i, u, w_{i+1}) \in S$. On sait que ces sommets existent puisque l'arête uw_i n'est pas dans M , elle a donc été augmentée. On pose par ailleurs l_i l'unique entier tel que $P_{l_i} = (w'_i, w_i, u, w_{i+1})$.

On a la propriété suivante :

$$\forall i < k, \forall l_{i-1} < j < l_i, P_j \text{ et } P_{l_i} \text{ n'ont pas de sommets en commun.}$$

En effet, puisque après augmentations des chemins $P_1, \dots, P_{l_{i-1}}$, w'_i et w_{i+1} sont exposés, ils n'ont jamais été couverts (par contraposée du lemme 1.9), donc pour tout $j < l_i$, $w'_i \notin P_j$ et $w_{i+1} \notin P_j$. De plus, si on note M' le couplage obtenu en utilisant la séquence $P_1, \dots, P_{l_{i-1}}$ à partir de M , alors $uw_i \in M'$, et uw_i ne peut être augmentée que par P_{l_i} , donc $u \notin P_j, w_i \notin P_j$. On peut donc augmenter P_{l_i} juste après $P_{l_{i-1}}$ (on obtient ce résultat par des applications successives du lemme 1.10).

On en déduit, en appliquant cette propriété à P_{l_1} , puis P_{l_2}, \dots , jusqu'à $P_{l_{k-1}}$ qu'on peut réordonner les chemins de la façon suivante (en "commençant" par les P_{l_i}) :

$$P_{l_1}, P_{l_2}, P_{l_3}, \dots, P_{l_{k-1}}, P_1, P_2, \dots, P_{l_1-1}, P_{l_1+1}, \dots, P_{l_i-1}, P_{l_i+1}, \dots$$

On note M' le couplage obtenu après augmentation des chemins $P_{l_1}, P_{l_2}, P_{l_3}, \dots, P_{l_{k-1}}$. Par ailleurs, pour tout $w \in F(u)$, on note M_w (respectivement M_w' et M_w^*) est égal à la restriction de M (respectivement de M' et de M^*) aux arêtes de T_w . Ainsi, par exemple, $M_w = M \cap T_w$.

On partitionne les sommets de $F(u) \setminus \{w_k\}$ en $\Gamma := \{w_i \mid i < k\}$ et $\Delta := F(u) \setminus (\Gamma \cup \{w_k\})$ (voir illustration Figure 11).

Pour tout $w_i \in \Gamma$, $M'_{w_i} = M_{w_i} \cup \{w_i w'_i\}$, et $|M_{w_i}^*| \leq \mu_{=3}(T_{w_i}, M'_{w_i})$. En effet, $M_{w_i}^*$ peut être obtenu à partir de M'_{w_i} en utilisant les chemins de S agissant sur T_{w_i} . Donc, $|M_{w_i}^*| \leq \mu_{=3}(T_{w_i}, M_{w_i} \cup \{w_i w'_i\}) = b_{w_i}$.

Par ailleurs, pour w_k , $M'_{w_k} = M_{w_k} \cup \{uw_k\}$. On en déduit par le même raisonnement que $|M_{w_k}^*| \leq \mu_{=3}(T_{w_k}, M_{w_k} \cup \{uw_k\}) = a_{w_k}$.

Enfin, pour tout $w \in \Delta$, $M'_w = M_w$ et de la même façon : $|M_w^*| \leq \mu_{=3}(T_w, M_w) = c_w$.

Finalement, $e_u = |M^*| = \sum_{w \in F(u)} |M_w^*| \leq \sum_{w \in \Gamma} b_w + a_{w_k} + \sum_{w \in \Delta} c_w \leq z$.

— Réciproquement, montrons que $z \leq e_u$.

Si $z = -1$, c'est immédiat.

Sinon, soit $w \in F(u)$ tel que $a_w \geq 0$. On pose par ailleurs $\Gamma' := \{x \in F(u) \setminus \{w\} \mid b_x \geq c_x\}$ et $\Delta' := F(u) \setminus (\Gamma \cup \{w\})$.

Si $\Gamma' \neq \emptyset$: soit $y_0 \in \Gamma'$. Montrons que $z \leq \mu_{=3}(T_u, M \setminus \{uv\} \cup \{uy_0\})$. Il suffit de montrer que l'on peut obtenir un couplage de taille z dans T_u à partir de $M_{y_0} \setminus \{uv\} \cup \{uy_0\}$.

On ordonne les éléments de $\Gamma' \setminus \{y_0\}$ en y_1, \dots, y_{k-1} . Pour tout i entre 0 et $k-1$, on choisit $y'_i \in F(y_i)$ tel que $b_{y'_i} = \mu_{=3}(T_{y_i}, M_{y_i} \setminus \{uy_i\} \cup \{y_i y'_i\})$. Il existe alors $S_{y'_i}$ séquence de chemins atteignant $b_{y'_i}$.

On pose alors, pour tout i entre 0 et $k-1$, $P_i := (y'_i, y_i, u, y_{i+1})$. On pose également $P_k := (y'_{k-1}, y_k, u, w)$. Pour tout $y \in \Delta'$, il existe une séquence de chemins S_y atteignant un couplage de taille $c_y = \mu_{=3}(T_y, M_y)$ partant de M_y , dans T_y .

Enfin, il existe une séquence de chemins S_w atteignant $a_w = \mu_{=3}(\tilde{T}_w, M_w)$, partant de M_w , dans T_w . Alors, la séquence consistant en l'augmentation des chemins $P_1, P_2, \dots, P_{k-1}, P_k$ puis des chemins des séquences $S_{y_1}, \dots, S_{y_{k-1}}, S_w$ et S_y , pour tout $y \in \Delta'$, est une séquence permettant d'obtenir un couplage de taille z .

Donc, dans ce cas, on a bien $z \leq e_u$.

Si au contraire $\Gamma' = \emptyset$: il existe une séquence de chemins S_w atteignant $a_w = \mu_{=3}(\tilde{T}_w, M_w)$, partant de M_w , dans T_w .

Pour tout $y \in F(u) \setminus \{w\}$, il existe une séquence de chemins S_y atteignant un couplage de taille $c_y = \mu_{=3}(T_y, M_y)$ partant de M_y , dans T_y .

Alors, la séquence constituée de la concaténation des séquences S_w et S_y pour tout $y \in F(u) \setminus \{w\}$ est une séquence permettant d'obtenir un couplage de taille z .

Donc ici également on a $z \leq e_u$. □

Annexe D : L'algorithme pour le cas des soleils

Algorithme 2 RESOLUTIONSOLEILSK=3(T, M)

Entrée : $S = (I \cup K, E)$ un soleil, $M \subseteq E$, $M \neq \emptyset$, M couplage

```

1: Soient  $A_M, B_M, C_M, D_M, AA_M, BB_M$  et  $AB_M$  les ensembles décrits précédemment
2: tant que  $D_M \neq \emptyset$  faire
3:   si  $A_M \neq \emptyset$  alors
4:     Soit  $a \in A_M$ , soit  $d \in D_M$ 
5:     Soit  $x \in K$  tel que  $ax \in M$  //  $x$  est de type  $A$  ou  $B$ , cela n'a pas d'importance
6:      $M \leftarrow (M \setminus \{ax\}) \cup \{aa', dx\}$ 
7:   sinon
8:     // Il y a des sommets de type  $B$ , puisque  $M \neq \emptyset$ 
9:     Soit  $b_1 b_2 \in BB_M$ 
10:    si  $|D_M| \geq 2$  alors
11:      Soient  $d_1, d_2 \in D_M, d_1 \neq d_2$ 
12:      // on augmente  $(d_1, b_1, b_2, d_2)$ 
13:       $M \leftarrow (M \setminus \{b_1 b_2\}) \cup \{d_1 b_1, b_2 d_2\}$ 
14:    sinon si  $|C_M| \geq 1$  alors
15:      Soit  $d \in D_M, c \in C_M$ 
16:      // on augmente  $(d, b_1, b_2, c)$ 
17:       $M \leftarrow (M \setminus \{b_1 b_2\}) \cup \{db_1, b_2 c\}$ 
18:    sinon
19:      // On ne peut plus rien faire
20:      retourner  $M$ 
21:  fin si

```

```

22:   fin si
23: fin tant que
24: tant que  $C_M \neq \emptyset$  faire
25:   si  $AB_M \neq \emptyset$  alors
26:     Soit  $(a, b) \in AB_M$ 
27:     Soit  $c \in C_M$ 
28:     // on augmente  $(a', a, b, c)$ 
29:      $M \leftarrow (M \setminus \{ab\}) \cup \{a'a, bc\}$ 
30:   sinon si  $BB_M \neq \emptyset$  et  $|C_M| \geq 2$  alors
31:     Soit  $b_1b_2 \in BB_M$ 
32:     Soient  $c_1, c_2 \in C_M, c_1 \neq c_2$ 
33:     // on augmente  $(c_1, b_1, b_2, c_2)$ 
34:      $M \leftarrow (M \setminus \{b_1b_2\}) \cup \{c_1b_1, b_2c_2\}$ 
35:   sinon
36:     // Il y a un arête de type AA
37:     Soit  $a_1a_2 \in AA_M$ 
38:     Soit  $c \in C_M$ 
39:     // on augmente  $(a'_1, a_1, a_2, c)$ 
40:      $M \leftarrow (M \setminus \{a_1a_2\}) \cup \{a'_1a_1, a_2c\}$ 
41:   fin si
42: fin tant que
43: tant que  $AA_M \neq \emptyset$  faire
44:   Soit  $a_1a_2 \in AA_M$ 
45:   // On augmente  $(a'_1, a_1, a_2, a'_2)$ 
46:    $M \leftarrow (M \setminus \{a_1a_2\}) \cup \{a'_1a_1, a_2a'_2\}$ 
47: fin tant que
48: retourner M

```

Annexe E : Légende générale des illustrations

D'une manière générale, les graphes sont représentés par leurs sommets et une partie de leurs arêtes, en noir et en gris. Les arêtes rouges sont des arêtes du couplage. Les arêtes en pointillés, qu'elles soient rouges ou noires, sont des arêtes qui existent mais qu'on ne sait pas forcément placer exactement. Enfin, certaines arêtes noires sont mises en évidences (trait plus épais), ce sont les arêtes qui "viennent d'être augmentées" lorsque l'on considère une succession d'augmentations de chemins de longueur 3.