Treewidth et séparateurs

Walid ASTAOUI

Juin 2022

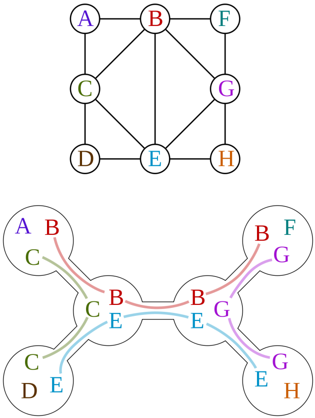
Graphes d'étude

Tous les graphes $G = (V, E)$ étudiés par la suite sont non orientés et simples. Pas de boucles $(u, u) \in E$ et il existe au plus une arête $(u, v) \in E$ pour chaque couple de sommets $u \neq v \in V$.

Définition : Tree decomposition

Une décomposition arborescente (ou tree-decomposition) d'un graphe $G = (V, E)$ est une paire $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$, avec T un arbre dont chaque noeud t représente un sac $X_t \subseteq V$, qui vérifie :

1. *Chaque sommet $v \in V$ est contenu dans un sac.*
2. *Pour toute arête $e = (u, v) \in E$, il existe un sac contenant u et v.*
3. *Pour un sommet $u \in V$, l'ensemble des noeuds t tels que $u \in X_t$ est un sous-arbre connexe de T.*



- Chaque sommet est dans l'ensemble des sacs de cet arbre T.
- Chaque arrête y est aussi.
- L'ensemble des sacs contenant un sommet donné est un sous-arbre connexe de T.

$\Longrightarrow$ C'est une décomposition arborescente du graphe.

Définition : Tree-Width

La largeur d'une tree decomposition est définie par $max_{t \in \mathcal{T}} |X_t| - 1$.
La largeur arborescente ou treewidth d'un graphe G *est le minimum des largeurs de toutes les décompositions arborescentes. On la note :* $tw(G)$.

Exemples ad-hoc

On a les treewidths suivants :

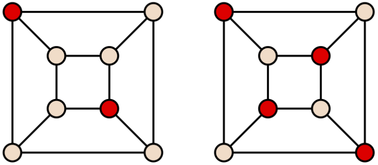
- *Arbre : tw=1*
- *Cycle : tw=2*
- *Clique* K_n *: tw=n-1*
- *Graphe biparti complet* $K_{a,b}$ *: tw=min(a,b)*

Définition : FPT

Un problème est dit $FPT(k)$ si la complexité de la solution est de la forme $\mathcal{O}(f(k) \times n^c)$ avec c une constante et $n = |V|$.

Exemple de problème FPT

Un Independent Set est un ensemble de sommets dans un graphe tels qu'aucun n'est voisin de l'autre.
Le problème Max Independent Set consiste à trouver le plus grand Independent Set possible dans un graphe donné.



NP-Complet
Résoluble en $\mathcal{O}(2^{\mathcal{O}(tw)} \times poly(n))$

Notations

Soit $S \subset V$ et un graphe $G = (V, E)$. On note :

1. $G[V/S] = (V - S, E - \{(s, v) \in E | s \in S, v \in V\})$
2. $G[S] = (S, \{(s, t) \in E | s, t \in S^2\})$

Définition : Séparateur A-B

Soient $A, B, S \subset V$. On dit que S sépare A de B dans $G = (V, E)$ lorsque :

- *(A, S, B) partitionne V.*
- *Le sous graphe $G[V/S]$ n'est pas connexe.*

Définition : Séparateur minimal

Soit $S \subset V$. On dit que S est un séparateur minimal de $G = (V, E)$ lorsque :

- *Le sous graphe $G[V/S]$ n'est pas connexe.*
- *$\forall S' \subset V : G[V/S']$ non connexe $\implies |S| \leq |S'|$*

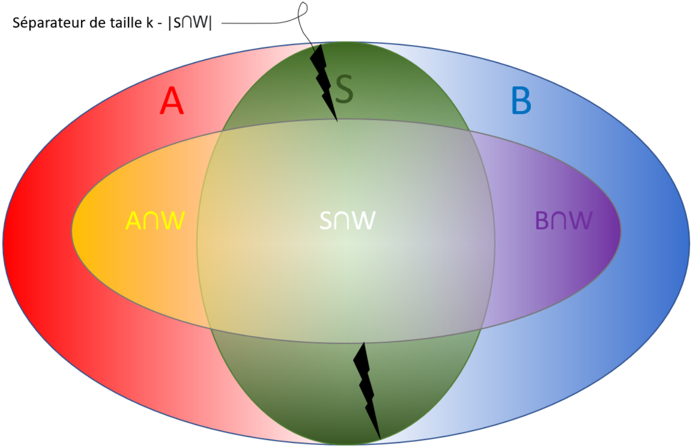
Définition : Partition équilibrée

On dit que (A, S, B) *est une partition équilibrée suivant* $W \subset V$ *d'un graphe* $G = (V, E)$ *avec* $tw(G) \leq k$ *lorsque :*

- S *sépare* A *de* B *dans* G.
- $|S| \leq k + 1$
- $|A \cap W| \leq \frac{2}{3}|W|$, $|B \cap W| \leq \frac{2}{3}|W|$

Étapes de la recherche d'une partition équilibrée (A, S, B) selon W et k :

- Distribution de W sur (A, S, B)
- Recherche d'un séparateur entre $A \cap W$ et $B \cap W$ de taille $k - |S \cap W|$ dans le sous-graphe $G[V/(S \cap W)]$

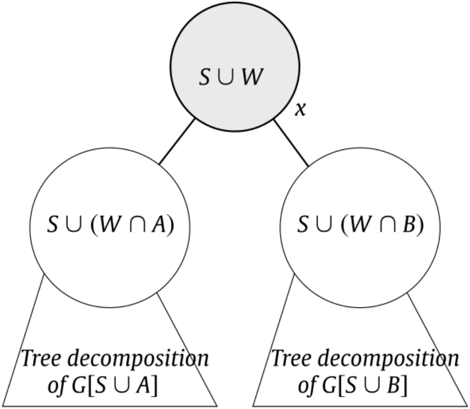


Théorème

Etant donné W et k, le problème de la recherche d'une partition équilibrée (A, S, B) d'un graphe $G = (V, E)$ est $FPT(|W|)$.

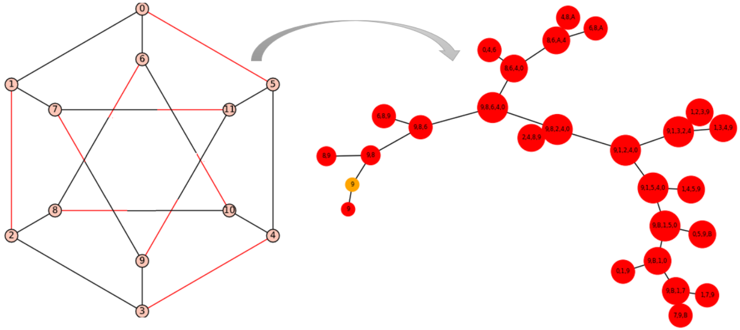
Preuve : Pour chaque distribution possible de W sur (A, S, B), la recherche d'un séparateur entre $(A \cap W)$ et $(B \cap W)$ est une variante Unoriented s-t Vertex-Cut du problème s-t cut de Floyd-Warshall résolu par recherche de flot maximal en compléxité $O(k(|V| + f))$ avec f le max-flow. Or, les capacités des arêtes ici sont unitaires, alors $f \leq |V|$. Donc, la recherche d'une partition équilibrée est en $O(3^{|W|}k|V|)$

La construction récursive d'une tree-decomposition avec séparateurs suit le schéma suivant :



Théorème

L'algorithme précise que le graphe d'entrée a une treewidth strictement supérieure à k, sinon renvoie une décomposition arborescente de largeur bornée par $4k + 3$.



Ecart d'unité de la tree-width 3

Théorème

La construction d'une décomposition arborescente avec séparateurs d'un graphe $G = (V, E)$ est $FPT(tw(G))$.

Comme la construction commence au départ par $W = \emptyset$, la complexité finale dépend de la taille maximale atteinte par W qui croit avec les appels récursifs.

Lemme

$$|W| \leq 3k + 3$$

Résultat direct des inégalités de taille fournies par la définition d'une partition équilibrée suivant W et k.

Pour améliorer la borne sur la largeur d'une décomposition arborescente construite par partition équilibrée, on procède par une transformation en graphe auxiliaire avant d'introduire des séparateurs plus forts.

Définition : Graphe auxiliaire

Le graphe auxiliaire $H_i = (X_i, E_i)$ d'un graphe $G = (V, E)$ par respect d'un sac X_i d'une décomposition arborescente $T = (I, F)$ vérifie :

1. $(u, v) \in E \wedge (u, v \in X_i^2) \implies (u, v) \in E_i$
2. $\forall j \neq i, X_j$ *voisin de* X_i *dans* T : $u, v \in (X_j \cap X_i)^2 \implies (u, v) \in E_i$

On commence par repérer le sac le plus problématique X_i dans la décomposition arborescente qu'on souhaite raffiner, ce sac doit vérifier :

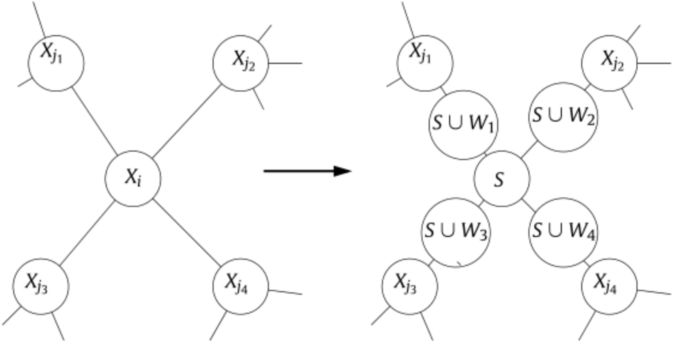
1. $G[X_i]$ n'est pas un clique.
2. $\forall j \neq i$: $G[X_j]$ n'est pas un clique $\implies |X_j| \leq |X_i|$.

On construit ensuite le graphe auxiliaire H_i par respect de ce sac X_i.

On calcule le séparateur minimal S du graphe auxiliaire H_i de $G = (V, E)$. Ce calcul se fait en complexité $O(|S||V|^{\frac{3}{2}}|E|)$, qui devient finalement $O(|V|^{\frac{1}{2}}|E|^2)$. Explications :

1. L'algorithme essaie chaque couple possible de sommets (s, t) du graphe H_i en pivotant sur s. Il essaie de tomber sur un sommet $s^* \notin S$ (afin qu'un $t^* \notin S$ permet la s^*-t^*séparation qui renvoie S). L'algorithme fait donc au pire $|S| + 1$ essais pour atteindre un tel s^*,et s'arrête juste après, d'où un facteur $|S||V|$ dans la complexité.
2. L'algorithme bénéficie de la structure 0-1 flow network de type 2 des graphes où il lance l'algorithme de Floyd-Warshall ou de Dinitz.Ce sont des graphes de flot à capacité unitaire qui vérifient en plus que chaque arête a une seule arête sortante ou bien une seule arête entrante. La complexité de Dinitz pour ces graphes est en $O(|V|^{\frac{1}{2}}|E|)$ pour chaque couple (s, t).
3. Si un sommet u est de degré $deg(u)$, couper ses $deg(u)$ arêtes permet de déconnecter le graphe, donc $|S| \leq \min_u deg(u)$. Comme $\sum_u deg(u) = 2E$, on a : $|S| \leq \dfrac{2|E|}{|V|}$.

S sépare H_i en r composantes connexes $(W_j)_{1 \leq j \leq r}$.
Finalement, reste l'opération la plus délicate : collage de S avec les X_j voisins de X_i dans T via r sacs supplémentaires correspondants aux W_r.



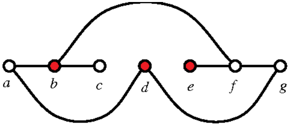
Voici comment l'algorithme de collage procède :

1. Collage de S avec chaque sac $S \cup W_j$.
2. Recherche de $1 \leq q_j \leq r$ vérifiant : $(X_i \cap X_j) \subset (S \cup W_{q_j})$, et ce pour tout X_j voisin de X_i.
3. Collage de chaque X_j avec le sac $S \cup W_{q_j}$.

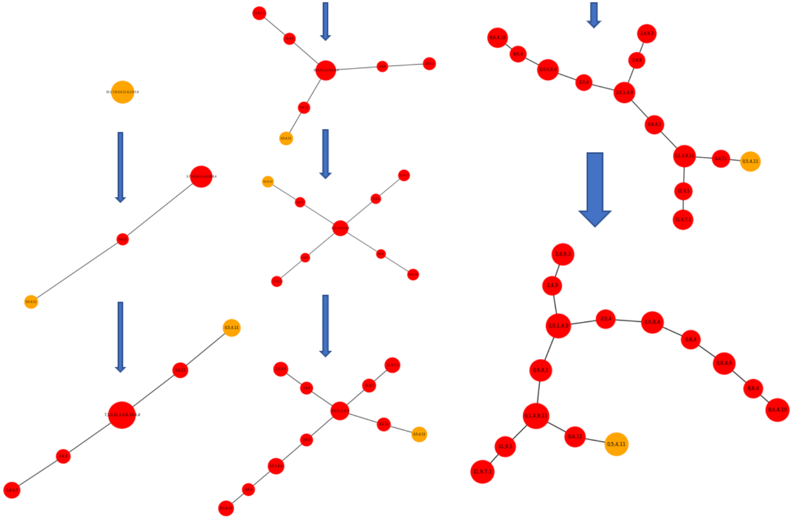
Remarques :

- Un unique q_j existe car $X_i \cap X_j$ est un clique dans H_i et donc la propriété 3 d'une décomposition arborescente assure que les sommets de $X_i \cap X_j$ ne peuvent être partagés entre 2 composantes connexes différentes de $H_i[X_i/S]$.
- Le nombre de sacs X_j auxquels un sac intermédiaire $S \cup W_l$ est collé peut être nul. La fonction $j \longmapsto q_j$ est surjective sur $[1, r]$.

- L'algorithme peut être appliqué efficacement de manière récursive sur un sac initial contenant tous les sommets du graphe $G = (V, E)$.
- L'algorithme est terminal car chaque étape permet de réduire la taille du plus grand sac en le substituant avec au moins 3 sacs strictement plus petits avant d'effectuer le collage.
- Toutefois, la seule borne prouvée à présent sur la largeur k de telles décompositions est $k \leq 8.an(G)$ où $an(G)$ est le nombre astéroïdal d'un graphe G : la largeur maximale d'un sous ensemble A de sommets non voisins tq l'élimination du voisinage fermé $N(a)$ d'un sommet $a \in A$ ne déconnecte pas le reste des sommets de A dans G.



Exemple d'exécution sur un graphe de Peterson



Généralisation d'une composante connexe.

Définition : Module

$M \subset V$ *est un module* $\Leftrightarrow \forall v \in V : v$ *voisin de tout sommet de* M *OU* v *n'est voisin d'aucun sommet de* M

Séparateur de modules.

Définition : Splitter

$S \subset V$ *est splitter d'un module* $M \Leftrightarrow \forall s \in S : s$ *voisin d'un sommet de* M *ET* s *n'est pas voisin d'un sommet de* M.

Les modules sont analogues aux sacs d'une décomposition arborescente : celui dont la taille est la plus grande définit la modular-width $mw(G)$.

1. Contrairement aux décompositions arborescentes, la construction d'une décomposition modulaire est polynomiale en taille du graphe.
2. Les séparateurs des décompositions arborescentes visent à être réduits en taille entre chaque appel récursif, tandis que les splitters visent à devenir plus grands entre chaque itération pour séparer des modules plus petits.
3. Aucune garantie sur la taille des splitters contrairement aux séparateurs des partitions équilibrées. Les gros cliques sont rares dans les réseaux quotidiens, inversement pour les gros modules premiers.
4. Toute la complexité se concentre donc au niveau de l'application. Exemple Independent Set : $O(2^{mw(G)}|V|^2)$

Conclusion

- Le concept de la tree-width permet une résolution plus simple et efficace de problèmes FPT(tw).
- Compromis entre la qualité de la construction et la rapidité de construction : Séparation équilibrée VS Raffinement récursif.
- Contraste entre complexité en amont et en aval : Tree-Width VS Modular-Width