



Suites de contractions et *twin-width*

Édouard Bonnet¹

Les théoriciens des graphes sont attirés par la beauté intrinsèque de questions ouvertes redoutables bien qu'exprimées dans le langage non-trivial le plus simple², mais aussi par les nombreuses applications et connexions avec le reste de l'informatique et des mathématiques, ainsi qu'avec pratiquement toutes les autres disciplines. La théorie des graphes entretient un lien évident et prolifique avec les algorithmes. La plupart des problèmes informatiques sont ou bien directement formulés en termes de graphes, ou bien, en totalité ou en partie, se réduisent naturellement à de tels problèmes. Mieux nous comprenons les graphes, mieux nous sommes équipés pour résoudre ces problèmes plus rapidement ou plus précisément.

Bien souvent, on ne peut pas raisonnablement espérer un algorithme uniformément performant sur tous les graphes. La bonne nouvelle est que les entrées typiques des problèmes de graphes de la vie réelle, comme on peut en trouver en biologie computationnelle, en planification de mouvement, en science des réseaux, en linguistique, etc., ne sont pas arbitraires et présentent une certaine « structure ». Les théories algorithmique et structurelle des graphes — deux branches sœurs de la théorie des graphes — visent à détecter et exploiter cette structure. Concrètement, cela passe par la décomposition des graphes, c'est-à-dire à leur (dé)construction à partir d'objets et de règles plus simples. Dans cette veine, l'exemple le plus célèbre est celui de la *treewidth*.

En première approximation, la *treewidth* est un paramètre de graphe qui mesure sa proximité topologique à un arbre. Elle s'accompagne d'une décomposition arborescente optimale, et en est sa largeur. La *treewidth* a été introduite par Bertelè et

1. Chargé de recherche CNRS en informatique au laboratoire de l'informatique du parallélisme de Lyon.

2. Fait d'une seule relation binaire.

Brioschi [4] et popularisée sous son nom actuel par l'œuvre monumentale de Robertson et Seymour [23]. Dans les pas de Bertelè et Brioschi, il a été démontré que de nombreux problèmes NP-difficiles admettent des algorithmes polynomiaux lorsqu'ils sont restreints aux graphes de *treewidth* bornée. Courcelle a donné une raison unifiée à ce phénomène, prouvant que cela s'applique en effet à tout problème de graphes exprimable en logique monadique du second ordre³ [13]. La *treewidth* a été utilisée par la suite pour résoudre efficacement des problèmes dans des classes de *treewidth* non bornée, comme par exemple les graphes planaires avec la technique de la bidimensionalité [18]. Le succès de la *treewidth* ne se limite pas à la conception d'algorithmes. La motivation originelle de Robertson et Seymour pour celle-ci était un plan de longue haleine qui les conduisit à résoudre la conjecture de Wagner (que la relation de mineur est un bel ordre sur les graphes finis).

D'autres paramètres de largeur de graphe associés à des décompositions plus générales (comme la *clique-width* [15], la *rank-width* [22], ou encore la *mim-width* [24]) ont vu le jour ces quarante dernières années. Toutes ces décompositions sont basées sur le même moule statique : un arbre étiqueté avec des règles plus ou moins permissives pour construire les graphes de largeur bornée. Avec Eun Jung Kim, Stéphan Thomassé et Rémi Watrigant, nous avons introduit un nouveau point de vue, dynamique celui-là, sur la décomposition de graphes. Nous inspirant d'une largeur et décomposition de permutation mise au point, il y a une dizaine d'années, par Guillemot et Marx [19], nous proposons de maintenir pas à pas une partition de plus en plus grossière des sommets d'un graphe, telle que la suite de partitions induisent des graphes auxiliaires de degré maximum borné, voire des graphes encore plus contraints⁴. Les suites de partitions peuvent être vues, de façon équivalente, comme des suites de contractions, où chaque partie d'une partition est amassée en un seul sommet. Les suites de contractions permettent d'introduire un nouveau paramètre (plus général que les *treewidth*, *clique-width* et *rank-width*), que nous avons nommé la *twin-width* [11], mais aussi de retrouver les paramètres classiques de largeur de graphe.

Nous souhaiterions récupérer, dans notre étude de la *twin-width*, des caractéristiques similaires au succès de la *treewidth* : un invariant avec une grande richesse d'utilisations et d'applications. Cet article survole les petits succès de la *twin-width*, notamment algorithmiques, en théorie des modèles et en combinatoire énumérative, et exhibe l'intérêt unificateur des suites de contractions.

3. Logique monadique du second ordre, bidimensionalité, mineur de graphes, bel ordre, clique-width, etc. Nous passons sous silence les définitions techniques de ces notions. Celles dont l'exposition aiderait à la lecture de l'article seront définies plus tard.

4. Nous verrons les définitions formelles dans la section suivante.

Suite de partitions ou de contractions

Une *suite de partitions* d'un graphe G à n sommets est une suite $\mathcal{S}$ de partitions $\mathcal{P}_n, \dots, \mathcal{P}_1$ des sommets $V(G)$ telle que $\mathcal{P}_n$ est la partition la plus fine, $\{\{v\} : v \in V(G)\}$, et que chaque partition $\mathcal{P}_{i-1}$ est obtenue en fusionnant deux parties de $\mathcal{P}_i$. Comme G possède n sommets, $\mathcal{P}_1$ est la partition la plus grossière, $\{V(G)\}$. Deux parties distinctes P, P' de $\mathcal{P}_i$ sont *homogènes* si entre P et P' apparaissent toute ou aucune arête. La *largeur* de la partition $\mathcal{P}_i$ est l'entier d maximum telle qu'une partie $P \in \mathcal{P}_i$ n'est pas homogène avec d autres parties de $\mathcal{P}_i$. Par extension, la *largeur* de la suite $\mathcal{S}$ est la largeur maximum d'une partition de $\mathcal{S}$. Enfin, la *twin-width* de G est la largeur minimum d'une suite de partitions de G .

On définit le quotient $G_i := G / \mathcal{P}_i$ de la manière suivante. Son ensemble de sommets $V(G_i)$ est $\mathcal{P}_i$, et deux parties $P, P' \in \mathcal{P}_i$ sont reliées par une arête *noire* quand, dans G , chaque sommet dans P est relié à chaque sommet dans P' , et par une arête *rouge* si ce n'est pas le cas mais qu'il y a quand même une paire de $P \times P'$ qui forme une arête de G . On appelle G_i un *trigraphe*, car considérant deux de ses sommets trois situations sont possibles : la présence d'une arête noire, d'une arête rouge, ou d'une non-arête. Son *graphe rouge* est simplement le graphe, sur le même ensemble de sommets, formé par les arêtes rouges. La suite $G = G_n, \dots, G_1$ est dite *suite de contractions*, une vision alternative à la suite de partitions (cf. figure 1).

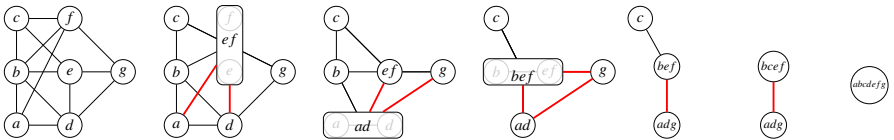


FIGURE 1. Suite de contractions impliquant que le graphe initial a *twin-width* au plus 2.

On peut passer d'un trigraphe G_i au suivant, G_{i-1} , sans faire référence aux partitions $\mathcal{P}_i$ et $\mathcal{P}_{i-1}$. Deux sommets u, v de G_i sont identifiés en un seul, disons w , de voisinage (par des arêtes rouges ou noires) l'union des voisinages de u et v privée de $\{u, v\}$. Un voisin z de w est relié à w par une arête noire si les paires uz et vz sont toutes les deux des arêtes noires, et par une arête rouge, autrement. Plus généralement, on définit la *twin-width* d'un trigraphe, de la même façon qu'on a défini celle d'un graphe. La *twin-width* s'étend naturellement aux structures binaires⁵ sur une signature finie.

5. Le lecteur peut penser les structures binaires comme des multigraphes dont les sommets (relations unaires) et les arêtes (relations binaires) sont colorés.

Si on impose la contrainte plus forte que dans tous les graphes rouges d’une suite de contractions, toutes les composantes connexes ont taille au plus $d' := d + 1$, on caractérise⁶ alors la *clique-width* [10]. En particulier, les classes de *twin-width* bornée contiennent les classes de *clique-width* bornée, et donc de *treewidth* bornée. Les chemins ont *twin-width* au plus 1, les arbres, au plus 2, et les grilles, qui ont *clique-width* non bornée, ont *twin-width* au plus 4. La figure 2 est une preuve par l’image de cette dernière assertion.

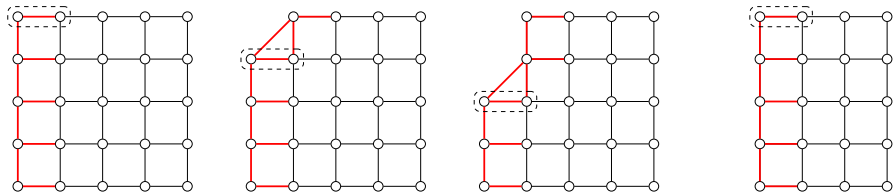


FIGURE 2. Contracter les arêtes horizontales entre la première et la deuxième colonnes, de haut en bas, nous ramène à une grille avec une colonne en moins ; et finalement au chemin rouge, qui a *twin-width* 2. Notons que la grande composante connexe du graphe rouge augmente avec la taille de la grille ; c’est inévitable car les grilles ont *clique-width* non bornée.

Parmi les exemples de graphes denses, les graphes complets ont *twin-width* 0, ainsi que leurs unions disjointes. En effet, il y a toujours dans ces graphes une paire de sommets *jumeaux* u, v , c’est-à-dire de même voisinage en dehors de $\{u, v\}$. La contraction de u et v ne crée ainsi aucune arête rouge, et reste dans la classe des unions disjointes de graphes complets. Cette propriété est vérifiée plus généralement par les *cographes* — graphes caractérisés par l’existence d’une paire de jumeaux dans chacun de leurs sous-graphes induits — qui sont précisément les graphes de *twin-width* 0.

Caractérisation par matrices d’adjacence

Nous allons voir dans cette section une caractérisation très pratique pour montrer qu’une classe de graphes a *twin-width* bornée. Cela ne donne cependant pas de bornes très précises. À quelques exceptions près, une fois que la *twin-width* d’une classe est prouvée bornée par cette méthode, une meilleure borne est ensuite montrée par une approche directe, mais souvent plus laborieuse.

6. Le lecteur peut donc prendre la valeur minimum de d' comme la définition de la *clique-width*, qui est en réalité comprise entre $\frac{d'+1}{2}$ et $d' + 1$ [1].

On dit qu'une matrice M a une *division de rang* k si on peut partitionner M en $k \times k$ blocs, chacun de rang au moins k . Le *rang de grille* de M est l'entier k maximum pour lequel M a une division de rang k . Le *rang de grille* d'un graphe G — ou d'une structure binaire — est le rang de grille minimum parmi les matrices d'adjacence de G .

Un résultat central de la théorie de la *twin-width* est le suivant : le rang de grille et la *twin-width* sont fonctionnellement équivalents⁷ pour les graphes, et plus généralement pour les structures binaires. Les suites de contractions sont des objets assez complexes, et il est parfois difficile d'en trouver directement qui aient faible largeur. La caractérisation par matrices d'adjacence simplifie notre tâche pour borner la *twin-width* de graphes. Il suffit d'ordonner leurs sommets de manière à ce que leur matrice d'adjacence correspondante soit sans grande division de rang.

La recette pour borner la *twin-width* d'une classe $\mathcal{C}$ est alors :

- trouver un bon schéma d'ordonnement des sommets basé sur les propriétés de $\mathcal{C}$;
- supposer que les matrices d'adjacence suivant l'ordre ont une grande division de rang ;
- utiliser cette division complexe pour contredire la « simplicité » de l'appartenance à $\mathcal{C}$;
- et conclure par l'équivalence fonctionnelle entre rang de grille et *twin-width*.

Par exemple, les graphes planaires, et plus généralement les classes de graphes excluant un *mineur*⁸ sont de *twin-width* bornée. Il suffit de considérer une variante de parcours en profondeur pour ordonner les sommets. Un autre exemple, au-delà des graphes, est la classe des ordres partiels $\prec$ partitionnables en au plus k chaînes, ou ordres totaux. Ceux-ci ont *twin-width* bornée (par une fonction de k), et une preuve rapide consiste à prendre pour ordonnancement les k chaînes mises bout à bout.

Contractions parallèles

La preuve de l'équivalence fonctionnelle entre *twin-width* et rang de grille [11, 7, 8] nous apprend quelque chose de plus, peut-être d'insoupçonné, sur les suites de contractions. Il y a une fonction f telle que tout graphe (ou structure binaire) de *twin-width* d , admet un *schéma arborescent* de suites de contractions de largeur au plus $f(d)$. Plus précisément, au lieu d'avoir un unique choix de contraction à chaque étape, on dispose d'un nombre linéaire — en le nombre de sommets restants — de

7. Une fonction de l'un(e) borne l'autre.

8. Un graphe G (ou une classe $\mathcal{C}$) contient un mineur H si le graphe H peut être obtenu à partir d'un sous-graphe de G (ou d'un graphe de $\mathcal{C}$) par la contraction d'arêtes ; et l'exclut sinon. La contraction d'une arête uv assimile u et v en un seul sommet relié aux voisins de u et de v , hors $\{u, v\}$. Contrairement à nos contractions (de sommets) dans les trigraphes, la contraction d'arêtes reste dans les graphes.

contractions possibles de paires disjointes ; et ce, inductivement. Cela nous donne à explorer un *arbre de trigraphes* de taille exponentielle : sa hauteur est le nombre initial n de sommets et son facteur de branchement pour les $n/2$ premiers niveaux est $\Omega(n)$. Le petit prix à payer est de passer de degré maximum d des graphes rouges à potentiellement $f(d)$. De plus, les $\Omega(n')$ contractions spécifiées pour tout trigraphe à n' sommets de cet arbre peuvent être réalisées *en parallèle*. Par cela, on entend que n'importe quel séquentialisation des $\Omega(n')$ contractions ne dépasse pas la borne $f(d)$ pour le degré maximum des graphes rouges, et termine sur un trigraphe possédant lui-même un schéma arborescent pour la même largeur $f(d)$.

Si on reprend notre exemple de la grille, une suite de contractions parallèles est donnée en figure 3. Notons au passage que chaque contraction suggérée produit un

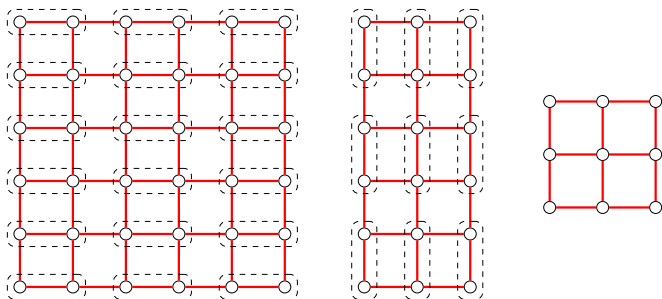


FIGURE 3. Schéma parallèle pour les grilles.

graphe rouge de degré maximum au plus 6, légèrement au-dessus de la borne de 4 de la figure 2. D'ailleurs, toute séquentialisation allant du trigraphe de gauche à celui au centre ne passe que par des graphes rouges de degré maximum au plus 6, et il y a une séquentialisation qui n'excède pas 5.

En combinatoire énumérative, les croissances⁹ simplement exponentielle et factorielle ont un statut particulier. Il y a souvent un saut entre croissance au plus c^n , pour une constante c , et croissance $\Omega(n!)$. Le théorème de Marcus–Tardos [21], un résultat important en combinatoire qui se trouve être au cœur de la théorie¹⁰ de la *twin-width*, implique par une observation préalable de Klazar [20] que toute classe de permutations¹¹ a croissance simplement exponentielle sauf la classe de toutes les permutations (qui, elle, a croissance factorielle).

9. La croissance d'une classe est une fonction qui compte, ou à défaut, borne le nombre de structures à n éléments présentes dans la classe.

10. En particulier, l'équivalence entre *twin-width* et rang de grille est crucialement basée sur ce théorème.

11. Une *classe de permutations* est un ensemble $\mathcal{C}$ de permutations tel que si σ est dans $\mathcal{C}$, toute permutation dont la matrice est une sous-matrice de celle de σ est aussi dans $\mathcal{C}$.

Il y a deux notions de croissance pour les classes de graphes : soit on compte à isomorphisme près (croissance non étiquetée), soit on force l'ensemble des sommets à être $\{1, \dots, n\}$ et on compte à égalité près de l'ensemble des arêtes (croissance étiquetée). Une classe est dite *minuscule* si sa croissance non étiquetée est simplement exponentielle, et *petite* si sa croissance étiquetée est au plus $n!c^n$ pour une constante c . Une conséquence du schéma arborescent est que les classes de *twin-width* bornée sont petites [7]. On peut avantageusement borner le nombre de tri-graphes à n sommets ayant un tel schéma en fonction du nombre de ceux à $n - 1$ sommets, en s'appuyant sur le fait qu'un nombre linéaire de sommets font partie d'une paire à contracter. Une caractérisation par la logique implique que les classes de *twin-width* bornée sont même minuscules [12]. Ces résultats unifient et étendent la même propriété constatée pour les classes de *treewidth* bornée [2], celles excluant un mineur [5], et le théorème de Marcus–Tardos [21] (en codant les permutations par des graphes).

À l'inverse (mais aussi à l'instar) de la caractérisation par matrices d'adjacence pour montrer qu'une classe a *twin-width* bornée, la petitesse de ces classes est un outil puissant pour établir que la *twin-width* est non bornée. Par exemple, les graphes de degré maximum 3 ont croissance labellée $(n!)^{3/2+o(1)}$, et donc, ne constituent pas une petite classe. Par conséquent, la *twin-width* de ces graphes diverge. De façon amusante, nous ne savons toujours pas construire une famille explicite de graphes de degré borné et de *twin-width* grandissante; même si choisir uniformément aléatoirement pour chaque entier naturel n , un graphe à n sommets de degré maximum 3 donne presque sûrement une telle famille.

Algorithmes basés sur des suites de partitions

Les classes dont on a mentionné que la *twin-width* est bornée admettent toutes un algorithme polynomial qui prend un graphe de la classe en entrée et renvoie une suite de contractions de largeur constante. Néanmoins, calculer la *twin-width* est en général NP-complet [3], et l'approximer est le problème ouvert le plus important dans ce contexte. On suppose désormais qu'on sait calculer une bonne suite de partitions, ou bien que celle-ci nous est donnée.

On va commencer avec une suite de partitions dont les graphes rouges ont toutes leurs composantes connexes de taille au plus d . On sait que cela est équivalent à avoir *clique-width* bornée. Donc on peut résoudre efficacement un plus vaste ensemble de problèmes, ceux exprimables en logique monadique du second ordre [14]. Dans cette logique, on peut quantifier sur des sommets ou des ensembles de sommets. Prenons l'exemple de la 3-COLORATION, problème NP-difficile dans les graphes planaires [16], et donc dans les graphes de *twin-width* bornée.

Soit G le graphe d'entrée et $R_n, \dots, R_1$ la suite de graphes rouges, dont les sommets sont des parties de $V(G)$. Pour chaque composante connexe C de chaque R_i , on

appelle *profil* une fonction λ qui à chaque sommet P de C associe un ensemble non-vidé de $\{1, 2, 3\}$ (les couleurs) telle qu'il y a une 3-coloration c de $G[\bigcup_{P \in V(C)} P]$ ¹² vérifiant $c(P) = \lambda(P)$ pour chaque $P \in V(C)$. La question est donc de savoir si la composante singleton $\{V(G)\}$ de R_1 a un profil. Chaque sommet de R_n constitue sa propre composante rouge. On a alors pour chaque sommet, trois profils, à savoir, colorer le sommet par 1 (rouge), 2 (bleu), ou 3 (vert), suivant les conventions de la figure 4.

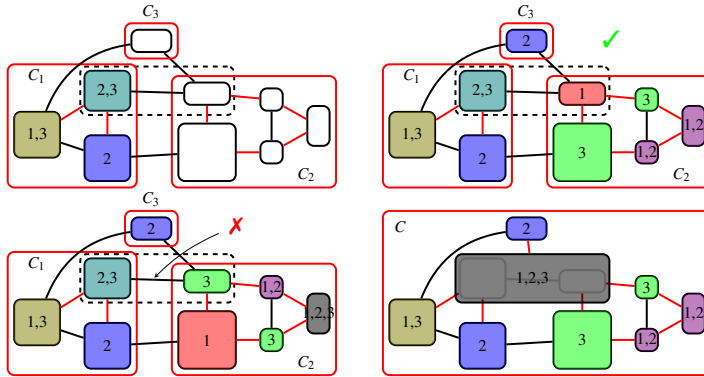


FIGURE 4. Exemple de profil de la composante rouge C_1 (haut, gauche), profils compatibles des composantes C_1, C_2, C_3 unies en une composante rouge C après la fusion des parties entourées en pointillé (haut, droite), profil correspondant de la nouvelle composante C (bas, droite) et profils incompatibles (bas, gauche).

L'algorithme est une simple programmation dynamique pour passer des profils de R_i à ceux de R_{i-1} . Disons qu'une nouvelle composante rouge est formée dans R_{i-1} à partir des composantes $C_1, \dots, C_h$ de R_i , avec $h \leq d+1$; dans l'illustration de la figure 4, $h = 3$. On calcule le produit cartésien des profils de $C_1, \dots, C_h$. Certaines combinaisons sont compatibles (haut à droite) et donnent lieu à un profil pour C (bas à droite), tandis que d'autres combinaisons n'en donnent pas (bas à gauche), où l'arête noire entre des sommets colorés 2 ou 3 et des sommets colorés uniquement 3 souligne un conflit.

Le temps d'exécution est linéaire en le nombre de profils essayés, soit au plus $7^{d+1}n$, multiplié par $(d+1)^2$ pour tester si une arête noire relie deux ensembles de couleurs s'intersectant. De façon intéressante, le temps obtenu, $O(7^d d^2 n)$, est possiblement bien inférieur au nombre d'arêtes de G .

12. On dénote par $G[X]$ le sous-graphe de G induit par $X \subseteq V(G)$.



En maintenant des solutions partielles plus « systématiques »¹³ que nos profils de l'exemple précédent, on retrouve le théorème de Courcelle-Makowsky-Rotics [14], que tout problème exprimable par une formule monadique du second ordre (MSO_1) peut être résolu en temps linéaire sur les graphes de *clique-width* bornée. Le célèbre théorème de Courcelle [13], que tout problème exprimable par une formule monadique du second ordre sur le graphe d'incidence (MSO_2) peut être résolu en temps linéaire sur les graphes de *treewidth* bornée, en est un cas particulier.

On va maintenant donner un algorithme efficace pour k -STABLE à *twin-width* bornée, toujours en supposant que la suite de partitions de faible largeur nous est donnée. Le problème k -STABLE demande si le graphe d'entrée possède k sommets distincts deux à deux non adjacents. Dans les graphes généraux à n sommets, un algorithme résolvant ce problème en temps $f(k)n^{o(k)}$ est peu vraisemblable quelle que soit la fonction f , car cela impliquerait un algorithme sous-exponentiel pour la satisfiabilité de formules booléennes.

À nouveau, on nous donne une suite de contractions $G = G_n, \dots, G_1$, de graphes rouges $R_n, \dots, R_1$. L'idée est de maintenir pour chaque trigraphe G_i , et chaque ensemble X d'au plus k sommets induisant un sous-graphe connexe dans R_i et aucune arête noire dans G_i , un stable maximum S_X de $G[\bigcup_{P \in X} P]$ qui intersecte chaque $P \in X$. Un stable intersectant au moins k parties est lui-même de taille au moins k , et peut être retourné comme solution globale. Le problème revient à déterminer si pour la composante singleton $\{V(G)\}$ de R_1 , on a un stable de taille au moins k . L'initialisation dans G_n se fait en observant que pour chaque composante singleton $\{v\}$ de R_n , le stable $S_{\{v\}}$ est simplement $\{v\}$.

13. Il faut retenir, pour chaque composante rouge, toute formule avec au plus k quantificateurs imbriqués qui est vérifiée par le graphe partitionné sous-jacent.

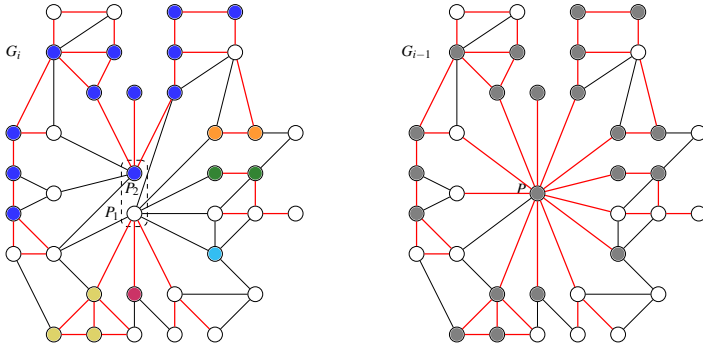


FIGURE 5. Comment obtenir S_X pour le $X \subseteq V(G_{i-1})$ représenté par les parties grisées ? On garde la meilleure des trois options dans G_i : en l'occurrence, celle intersectant P_2 mais pas P_1 (la seule à marcher à cause des arêtes noires entre P_1 et les sommets bleu, orange et vert). Cela partitionne X en au plus $d + 2$ ensembles connexes Y dans R_i , chacun représenté dans une couleur différente, pour lesquels on a déjà calculé S_Y .

Après la fusion de deux parties P_1, P_2 en P , on calcule pour chaque ensemble $X \ni P$ de taille au plus k , connexe dans R_{i-1} , un stable maximum S_X grâce aux stables S_Y de G_i . On compare les trois possibilités : S_X intersecte P_1 mais pas P_2 , ou P_2 mais pas P_1 , ou les deux (cf. figure 5), on vérifie l'absence d'arêtes noires entre les Y agglomérés, et on garde l'union des S_Y correspondants.

Combien y a-t-il d'ensembles connexes de taille au plus k contenant une partie fixée dans un des graphes rouges ? Pas plus de d^{2k} car leur degré maximum est d'au plus d . Le temps d'exécution est donc $O(d^{2k}k^2n)$ [6] qui, quand d et k sont petits comparés à n , est bien meilleur que l'algorithme en $n^{\Theta(k)}$, le plus rapide dans les graphes généraux.

On peut étendre ce schéma d'algorithme à la vérification de modèles en logique du premier ordre. Pour une structure binaire G à n sommets, donnée avec une suite de partitions de largeur d , on peut décider si G satisfait φ en temps $f(d, |\varphi|) \cdot n$ [11]. L'algorithme combine ce qu'on a vu à *clique-width* bornée (pour MSO_1) et la localité de la logique du premier ordre. Un corollaire particulièrement intéressant est que toute transduction¹⁴ d'une classe de *twin-width* bornée a aussi *twin-width* bornée [11].

14. Une *transduction* redéfinit par des formules du premier ordre les nouvelles relations d'une structure dans une expansion monadique, c'est-à-dire après ajout d'un nombre constant de relations unaires.

Les classes qui ne permettent pas de produire la classe de tous les graphes par transduction sont dites *monadiquement dépendantes*. Parmi les classes *monadiquement dépendantes*, celles qui ne permettent pas de transduire la classe de tous les ordres totaux sont dites *monadiquement stables*. La famille des classes de *twin-width* bornée et celle des classes monadiquement stables sont incomparables, et incluses dans la famille des classes monadiquement dépendantes. Il est conjecturé que la vérification de modèles en logique du premier ordre a un algorithme efficace dans une classe $\mathcal{C}$ si $\mathcal{C}$ est monadiquement dépendante, et la réciproque a été prouvée récemment [17]. Présentement, cette conjecture est vérifiée pour les classes de *twin-width* effective¹⁵ bornée [11] et pour toute classe monadiquement stable [17].

Graphes ordonnés et matrices

Les structures binaires *ordonnées* ont une de leurs relations binaires interprétée comme un ordre total. Par exemple, un graphe ordonné a deux relations binaires : les arêtes du graphe $E(\cdot, \cdot)$ et un ordre total sur ses sommets $\prec$. Dans les graphes ordonnés, la *twin-width* passe d'un paramètre intéressant à une mesure cruciale. Pour toute classe de structures binaires ordonnées close par sous-structure induite, il est équivalent, et ces équivalences sont montrées par la théorie de la *twin-width* [8] :

- d'avoir *twin-width* bornée ;
- d'admettre un algorithme efficace pour la vérification de modèles au premier ordre¹⁶ ;
- d'être monadiquement dépendante ;
- d'avoir croissance simplement exponentielle ;
- d'avoir croissance sous-factorielle.

De plus, il y a un algorithme efficace d'approximation pour calculer des suites de partitions [8]. Donc, la *twin-width* joue le même rôle pour la logique du premier ordre dans les graphes ordonnés que la *clique-width* pour MSO_1 dans les graphes.

Les résultats de la section précédente s'étendent à la logique du premier ordre avec des quantificateurs modulaires ; logique où pour un entier p fixé on dispose de quantifications $\exists^p x \varphi(x)$ (pour $0 \leq i \leq p-1$) avec pour sémantique le fait que le nombre de sommets $v \in V(G)$ tels que $\varphi(v)$ est satisfait dans G est congru à i modulo p . On peut décrire le produit matriciel sur un corps fini par une interprétation au premier ordre avec quantificateurs modulaires [9]. Combiné à une structure de donnée compacte à *twin-width* bornée, cela permet de multiplier des matrices $n \times n$ sur un corps fini en temps quasi-linéaire en n , c'est-à-dire bien en dessous de leur nombre de coefficients.

15. Celles pour lesquelles on sait efficacement trouver des suites de contractions de largeur constante.

16. Cet item n'implique les autres que sous une hypothèse standard de complexité.

Références

- [1] Ambroise Baril, Miguel Couceiro, and Victor Lagerkvist. Linear bounds between cliquewidth and component twin-width and applications, 2023. URL : <https://ramics20.lis-lab.fr/slides/slidesAmbroise.pdf>.
- [2] Lowell W Beineke and Raymond E Pippert. The number of labeled k -dimensional trees. *Journal of Combinatorial Theory*, 6(2) :200–205, 1969.
- [3] Pierre Bergé, Édouard Bonnet, and Hugues Déprés. Deciding twin-width at most 4 is NP-complete. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 18 :1–18 :20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.18.
- [4] Umberto Bertelè and Francesco Brioschi. On non-serial dynamic programming. *J. Comb. Theory, Ser. A*, 14(2) :137–148, 1973.
- [5] Robin L. Blankenship. *Book embeddings of graphs*. Louisiana State University and Agricultural & Mechanical College, 2003.
- [6] Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width III : Max Independent Set, Min Dominating Set, and Coloring. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 35 :1–35 :20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.35.
- [7] Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width II : small classes. *Combinatorial Theory*, 2(2), 2022. URL : <https://escholarship.org/uc/item/9cs265b9>, doi:<http://dx.doi.org/10.5070/C62257876>.
- [8] Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, Pierre Simon, Stéphan Thomassé, and Szymon Torunczyk. Twin-width IV : ordered graphs and matrices. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22 : 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 924–937. ACM, 2022. doi:10.1145/3519935.3520037.
- [9] Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, and Stéphan Thomassé. Twin-width V : linear minors, modular counting, and matrix multiplication. In Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté, editors, *40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, March 7-9, 2023, Hamburg, Germany*, volume 254 of *LIPIcs*, pages 15 :1–15 :16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. URL : <https://doi.org/10.4230/LIPIcs.STACS.2023.15>, doi:10.4230/LIPIcs.STACS.2023.15.
- [10] Édouard Bonnet, Eun Jung Kim, Amadeus Reinald, and Stéphan Thomassé. Twin-width VI : the lens of contraction sequences. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1036–1056. SIAM, 2022.
- [11] Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I : tractable FO model checking. *J. ACM*, 69(1) :3 :1–3 :46, 2022. doi:10.1145/3486655.
- [12] Édouard Bonnet, Jaroslav Nesetril, Patrice Ossona de Mendez, Sebastian Siebertz, and Stéphan Thomassé. Twin-width and permutations. *CoRR*, abs/2102.06880, 2021. URL : <https://arxiv.org/abs/2102.06880>, arXiv:2102.06880.
- [13] Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1) :12 – 75, 1990. URL : <http://www.sciencedirect.com/science/article/pii/089054019090043H>, doi:10.1016/0890-5401(90)90043-H.

- [14] Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2) :125–150, 2000. doi:10.1007/s002249910009.
- [15] Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discret. Appl. Math.*, 101(1-3) :77–114, 2000. doi:10.1016/S0166-218X(99)00184-5.
- [16] David P Dailey. Uniqueness of colorability and colorability of planar 4-regular graphs are np-complete. *Discrete Mathematics*, 30(3) :289–293, 1980.
- [17] Jan Dreier, Ioannis Eleftheriadis, Nikolas Mählmann, Rose McCarty, Michal Pilipczuk, and Szymon Torunczyk. First-order model checking on monadically stable graph classes. *CoRR*, abs/2311.18740, 2023. URL : <https://doi.org/10.48550/arXiv.2311.18740>, arXiv:2311.18740, doi:10.48550/ARXIV.2311.18740.
- [18] Fedor V. Fomin, Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Bidiimensionality. In *Encyclopedia of Algorithms*, pages 203–207. 2016. doi:10.1007/978-1-4939-2864-4_47.
- [19] Sylvain Guillemot and Dániel Marx. Finding small patterns in permutations in linear time. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 82–101. SIAM, 2014. doi:10.1137/1.9781611973402.7.
- [20] Martin Klazar. The Füredi-Hajnal conjecture implies the Stanley-Wilf conjecture. In *Formal power series and algebraic combinatorics*, pages 250–255. Springer, 2000.
- [21] Adam Marcus and Gábor Tardos. Excluded permutation matrices and the Stanley-Wilf conjecture. *J. Comb. Theory, Ser. A*, 107(1) :153–160, 2004. doi:10.1016/j.jcta.2004.04.002.
- [22] Sang-il Oum and Paul D. Seymour. Approximating clique-width and branch-width. *J. Comb. Theory, Ser. B*, 96(4) :514–528, 2006. URL : <https://doi.org/10.1016/j.jctb.2005.10.006>, doi:10.1016/J.JCTB.2005.10.006.
- [23] Neil Robertson and Paul D. Seymour. Graph minors. III. planar tree-width. *J. Comb. Theory, Ser. B*, 36(1) :49–64, 1984. doi:10.1016/0095-8956(84)90013-3.
- [24] Martin Vatshelle. New width parameters of graphs. 2012.