

Dislocation de graphes

Benjamin Auder & Julien Haguenaer

Juin 2006

Cas Extrémaux et Applications, Algorithmes et Complexité

encadré par Julien Moncel (G-SCOP, Grenoble).

Lexique :

C_n : cycle d'ordre n .

CH_n : chaîne de longueur n .

K_n : graphe complet d'ordre n .

PK_Δ : graphe constitué de sous graphes complets d'ordre Δ , connectés comme décrit en 3.1.2.

$E_{n,m}$: pseudo-étoile à n sommets et m arêtes.

H_n : graphe à $2n$ sommets, obtenue à partir d'un K_n auquel on a déplacé le cycle extérieur.

$P_{n,p}$: puissance k^{eme} de cycle à n sommets tous de degré p .

CK_n : graphes complets d'ordre n connectés entre eux (en fasciagraphe ou rotagraphe).

Table des matières

1	Introduction	4
1.1	Définition formelle	4
1.2	Exemples de Δ – <i>dislocations</i>	5
1.3	Historique du problème	7
1.4	Contribution :	7
2	Dislocation des Arbres	8
2.1	Cas d’une chaîne	8
2.2	Cas de l’arbre complet	9
2.3	Autres arbres	14
3	Recherche de cas extrémaux	18
3.1	Graphes faciles à disloquer	18
3.1.1	Cas connexe	19
3.1.2	Cas disconnexe	22
3.2	Chaîne et cycle : difficiles à disloquer ?	24
3.3	Une généralisation du cycle	29
3.4	Comparaison à une autre classe de graphes	33
3.5	Conclusion	35
4	Applications	37
4.1	Stoppage de la propagation d’un virus	37
4.2	L’invasion au jeu de go	37
5	Algorithmes	37
5.1	Stratégies utilisées, algorithmes probabilistes	38
5.1.1	Une heuristique plus efficace	39
5.1.2	L’algorithme exact	41
5.2	Expérimentations et limitations	41
6	Complexité	42
6.1	APX et NP-complétude	42
6.2	Un algorithme d’approximation efficace ?	43
6.3	Conclusion	44

1 Introduction

On connaît le problème classique de recherche d'un transversal dans un graphe : on recouvre le graphe avec un nombre minimal de sommets de manière à ce que toutes les arêtes soient en contact avec au moins un sommet couvrant. Il s'agit ici de généraliser cette idée, en imposant non plus que toute arête ait une extrémité dans l'ensemble solution, mais que toute partie connexe de cardinal $\Delta + 1$ possède un sommet dans l'ensemble solution, que l'on notera S tout au long de cet article.

Bien sûr pour $\Delta = 1$ on retrouve la définition du transversal, car une partie connexe à deux éléments n'est rien d'autre qu'une arête.

1.1 Définition formelle

Définition Étant donné un graphe $G = (V, E)$ et un entier naturel Δ , on appelle Δ -*dislocation* un ensemble $S \subseteq V$ de sommets de G ssi toute composante connexe de $G - S$ contient au plus Δ sommets.

Remarque 1 : En général, il n'y a pas d'unique Δ -*dislocation*, mais il existe plusieurs solutions au problème, et même éventuellement plusieurs solutions optimales. Notamment, il est simple de constater que si $\Delta' \leq \Delta$, alors une Δ' -*dislocation* est aussi une Δ -*dislocation*.

Remarque 2 : Bien sûr pour $\Delta = 1$ on retrouve la définition du transversal, car une partie connexe à deux éléments n'est rien d'autre qu'une arête.

Éléments de complexité : Posons le problème de décision correspondant :

Δ -*DISLOCATION* :

Entrée : $G = (V, E)$ un graphe, et $k \in \mathbb{N}$

Sortie : Existe-t-il un ensemble $S \subseteq V$ de Δ -*dislocation* du graphe G tel que $|S| \leq k$

D'après la remarque faite plus haut, il est clair que pour $\Delta = 1$, le problème se réduit au problème de décision *TRANSVERSAL*, qui est NP-Complet. Par ailleurs il est montré dans [GMP04] qu'il existe une réduction du problème Δ -*DISLOCATION* à *TRANSVERSAL* pour tout Δ qui montre donc que Δ -*DISLOCATION* est NP-Complet.

Dans la suite de cet article, nous nous intéresserons évidemment plus particulièrement aux problèmes de Δ -*DISLOCATION* optimale (ou bien

minimale, *i.e.* avec $|S|$ minimum.)

1.2 Exemples de Δ – *dislocations*

On représentera l'ensemble S de dislocation en noir dans les graphes. On constatera que les composantes connexes en blanc ne seront jamais de taille supérieure à Δ

$\Delta = 1$: **Transversal**

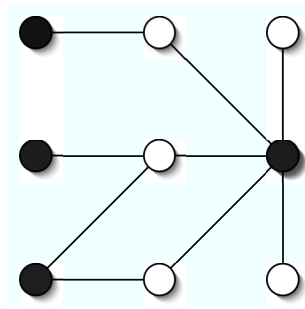


FIG. 1 – Une solution sur un graphe quelconque pour $\Delta = 1$

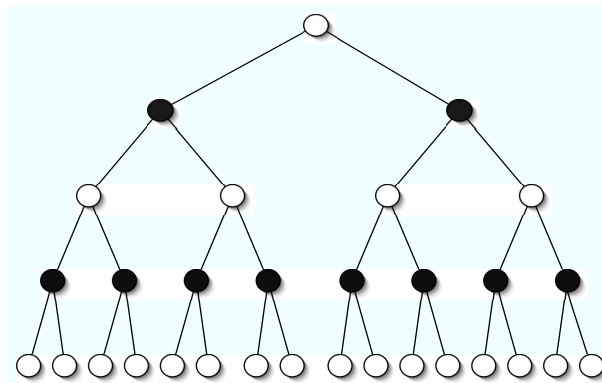


FIG. 2 – Une solution sur un arbre pour $\Delta = 1$

$\Delta = \dots$

Sur les mêmes graphes, on obtient :

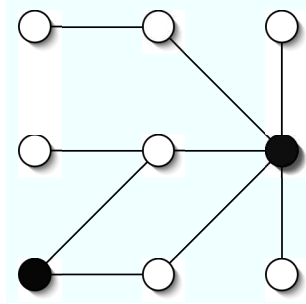


FIG. 3 – Une solution le même graphe quelconque pour $\Delta = 2$

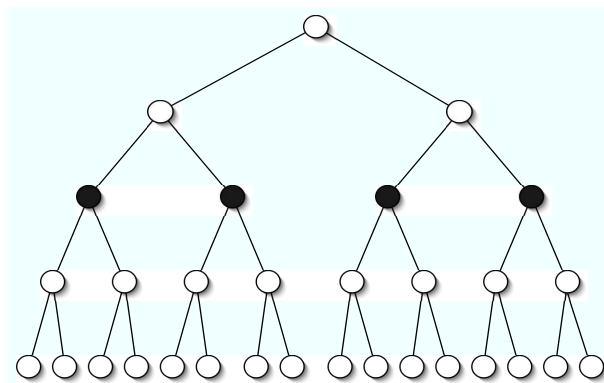


FIG. 4 – Une solution sur le même arbre pour $\Delta = 3$

1.3 Historique du problème

Le problème de la dislocation des graphes a été formulé en 2004 par Sylvain Gravier, Julien Moncel et Charles Payan dans [GMP04], dans le cadre de l'étude du problème d'exclusion des pentominos. Le problème d'exclusion des pentominos, posé par Solomon Golomb dans un cadre "mathématiques récréatives" (cf. [GOL94]) est le suivant : *Trouver le nombre minimum de cases à retirer d'un damier $n \times k$ pour exclure tous les pentominos.* Il s'agit donc pour nous d'une 4 – *dislocation*, sur un graphe désigné par l'appellation *grille*.

Jusqu'à présent, seuls cinq articles ont été rédigés sur le sujet :

- [GOL94] est l'ouvrage de S. Golomb datant de 1994 où le problème est initialement posé
- [BOS98] est un court article de Robert A. Bosch dans le magazine Optima, qui présente le problème et présente également les résultats obtenus à l'aide de la programmation linéaire. (avec CPLEX, version 4.0.9)
- [GP01] est un article de deux chercheurs du laboratoire Leibniz, Sylvain Gravier et Charles Payan. En introduisant la notion de densité d'un ensemble de cases, ils montrent des résultats asymptotiques, et notamment le résultat exact pour un plateau $k \times n$ où $k \leq 4$
- [GMP04] est un article rédigé avec la même équipe, à laquelle il faut rajouter Julien Moncel, qui a encadré ce présent projet. Comme nous l'avions mentionné plus haut, ce papier généralise le problème de S. Golomb, en posant celui de la Δ – *dislocation*. Il y est démontré que ce problème est NP-Complet, mais il y est aussi donné un algorithme déterministe linéaire qui résout le problème sur certaines familles de graphes.
- [ZER04] est un article de Janez Žerovnik qui s'intéresse au problème de la Δ – *dislocation* sur certaines familles de graphes (fasciagraphes et rotagraphes) pour lesquels il est donné un algorithme en temps constant qui donne une solution optimale.

1.4 Contribution :

Dans cet article, on s'intéresse d'abord au cas des arbres, et on donnera notamment un formule donnant la solution optimale sur les arbres complets (quelque soit leur arité).

On étudie ensuite les problèmes extrémaux, c'est à dire les graphes "faciles" ou "difficiles" à "disloquer". En particulier, on exhibera deux familles de

graphes qui sont des cas extrêmes : les étoiles "itérées" (faciles à disloquer) et les puissances de cycles (durs à disloquer).

On donnera ensuite quelques applications possibles au problème de dislocation, qui reposent sur l'idée de "seuil limite".

Ensuite, on ira expliquer les algorithmes que nous avons testé à l'aide du langage C++. Nous présenterons les différentes stratégies appliquées pour l'approximation (heuristique, recuit simulé) et leur précision, et enfin un algorithme exact.

Enfin, on tentera d'apporter quelques éléments nouveaux sur la complexité en posant la question d'un algorithme d'approximation efficace.

2 Dislocation des Arbres

Comme dans de nombreux problèmes en théorie des graphes, le cas des arbres a de grandes chances d'être assez facile à traiter. En effet on peut alors se donner un sens de parcours naturel des feuilles vers la racine, et ce qui est probablement plus important, l'arbre a toujours localement la même structure, inspirant des raisonnements par induction. Il est donc logique de commencer par ces graphes particuliers.

2.1 Cas d'une chaîne

La chaîne semble être l'arbre le plus simple à disloquer algorithmiquement, étant donnée sa structure régulière. On commence donc par étudier, à Δ fixé, le nombre $\rho_{n,\Delta}$ de Δ -dislocation d'une chaîne à n sommets (et donc $n - 1$ arêtes).

On remarque que $\rho_{n,\Delta}$ est une fonction croissante de n , car on peut toujours déduire un ensemble solution sur une chaîne de longueur n à partir d'une solution optimale pour une chaîne de longueur $m > n$ par restriction.

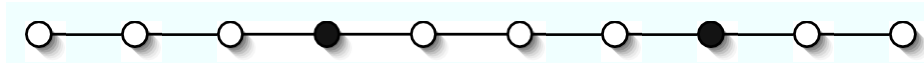


FIG. 5 – Une solution sur une chaîne pour $\Delta = 3$

Intuitivement, il suffit de "découper" la chaîne en morceaux de longueur $\Delta + 1$, puis de retirer un sommet à chaque jonction. Cette condition est en fait également nécessaire, la démonstration utilisant l'ensemble des sous-chaînes

et un raisonnement par induction que l'on ne détaille pas ici.

Théorème :

Soit G une chaîne à n sommets (et donc $n - 1$ arêtes). $\forall \Delta \in \mathbb{N}^*$ on a :

$$\rho_{n,\Delta} = \left\lfloor \frac{n}{\Delta+1} \right\rfloor$$

Démonstration :

Comme toute partie connexe (sous-chaîne de G) de cardinalité $\Delta + 1$ contient au moins un élément de S , on obtient facilement :

$$\rho_{n,\Delta} \geq \left\lfloor \frac{n}{\Delta+1} \right\rfloor,$$

car il existe toujours au moins $\left\lfloor \frac{n}{\Delta+1} \right\rfloor$ sous-chaînes de longueur $\Delta + 1$.

Il est ensuite aisé de montrer par récurrence que toute solution qui consiste à prendre $\left\lfloor \frac{n}{\Delta+1} \right\rfloor$ sommets tous distants de Δ deux à deux dans S est forcément une Δ -dislocation valide.

D'après l'inégalité qui précède, cette solution sera donc optimale, et on a donc montré la formule voulue.

Remarque : L'unicité d'un ensemble de dislocation optimal est fautive en général (en fait dès que $n \bmod (\Delta + 1) < \Delta$ pour la chaîne (resp. $(n - 1) \bmod (\Delta + 1) < \Delta$ pour le cycle), comme le montre l'exemple suivant, après avoir retiré un sommet du cycle :

Une 5-dislocation s'obtient en enlevant n'importe quel paire de sommets, une 4-dislocation en supprimant deux sommets diamétralement opposés par exemple, ..etc. Pour une 2-dislocation, il faut et il suffit de prendre un sommet sur trois après suppression d'un sommet quelconque, comme on le verra bientôt.

2.2 Cas de l'arbre complet

On va utiliser dans cette section l'algorithme décrit dans [GMP04] sur la dislocation des arbres.

On commence donc par montrer que celui-ci est correct (la linéarité du temps d'exécution étant claire en prenant pour opération élémentaire le traitement d'une feuille), après l'avoir rappelé brièvement :

Algorithme :

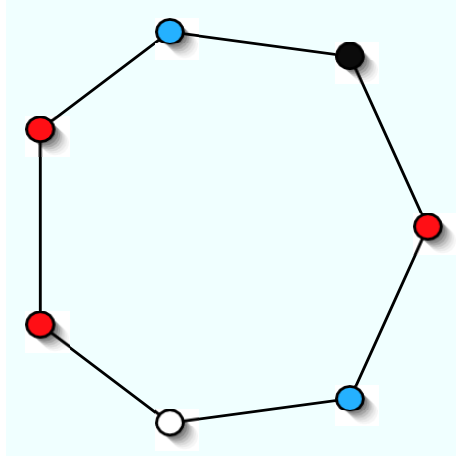


FIG. 6 – Δ – *dislocations* sur un cycle à 7 sommets.

1. Initialisation : donner un poids de 1 à chaque nœud, $S \leftarrow \emptyset$
2. Tant qu'il reste un nœud dans l'arbre, faire :
 - (a) Tant qu'il reste une feuille f de poids 1 dans l'arbre, faire :
 - i. Poids(pere(f)) \leftarrow poids(pere(f))+1
 - ii. Supprimer f de l'arbre
 - (b) Tant qu'il reste une feuille f de poids $> \Delta$, faire :
 - i. $S \leftarrow S \cup \{f\}$
 - ii. Supprimer f de l'arbre
3. Retourner S , $\rho = |S|$

Justification de l'optimalité :

On montre par récurrence finie qu'après k suppressions de nœuds dans l'arbre via l'algorithme, on a en fait choisit k sommets faisant partie d'une solution optimale de Δ -dislocation sur l'arbre.

Initialisation : La propriété est claire pour $k = 0$.

Hérédité : Soit $k \in [0, \dots, |S| - 1]$ où S désigne un ensemble de dislocation optimal, et supposons effectuées k suppressions de sommets.

Faisons tourner l'algorithme jusqu'au prochain ajout de feuille dans S (ce moment arrive, car lorsqu'on supprime une feuille pour l'ajouter dans S , les sous-arbres correspondant aux feuilles supprimées dans cette étape sont de cardinal au plus Δ par hypothèse de récurrence).

Notons alors f cette feuille, de poids $p > \Delta$, correspondant à un certain sous-arbre. Le poids de la racine anciennement supprimée du sous-arbre gauche est inférieur ou égal à Δ , de même pour celui de l'ancienne racine du sous-arbre droit.

Le sous-arbre de racine f est connexe et contient plus de $\Delta + 1$ sommets : un des ces sommets appartient alors nécessairement à S . D'après ce qui précède il suffit d'en prendre un : f . De plus en faisant ce choix on minimise la taille de l'arbre résultant (qui contient de toutes façons au moins le père de f et ses nœuds ascendants). Donc ce choix optimise effectivement le nombre de dislocation d'un arbre résultant, et on en déduit que f fait partie d'une solution optimale de Δ -dislocation sur l'arbre.

Notons n le nombre de sommets de l'arbre complet m -aire, h sa hauteur égale par définition à la longueur du plus grand chemin de la racine à une feuille, la hauteur d'un arbre réduit à sa racine valant 0 par convention. On a $h = \lfloor \log_m(n) \rfloor$.

Nombre de dislocation de l'arbre complet :

L'algorithme que nous venons de présenter, appliqué à un arbre binaire complet pour une 2-dislocation nous permet d'avoir une bonne idée de la forme des solutions sur les arbres complets m -aires. En effet, la solution donnée par l'algorithme est de la forme suivante :

Suivant cette solution, toute chaîne reliant une feuille et la racine l'arbre

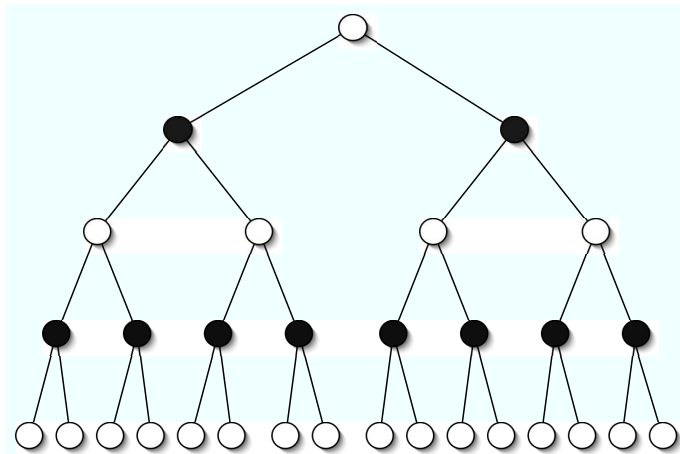


FIG. 7 – Une 2-dislocation sur un arbre binaire complet.

a subi une 1-dislocation. On peut en fait montrer qu'une solution optimale

donnée par l'algorithme en question a pour effet de diviser toutes les chaînes partant de la racine et arrivant aux feuilles selon une T -dislocation, où T est la distance à une feuille du premier noeud disloqué par l'algorithme. T est égal à $T_\Delta = \lceil ((m-1)\Delta + m)\log_m \rceil$. Cela revient à diviser l'arbre en "étages".

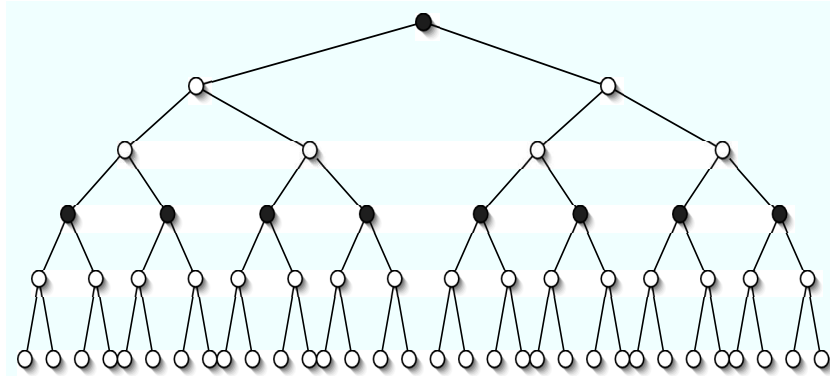


FIG. 8 – Une 3-dislocation sur un arbre binaire complet.

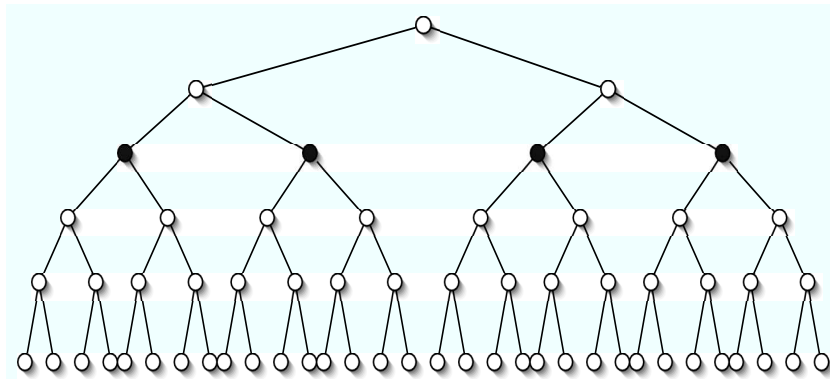


FIG. 9 – Une 7-dislocation sur un arbre binaire complet.

Intuitivement, on a obtenu ce résultat en remarquant que pour une 2-dislocation il faut mettre un étage sur deux dans l'ensemble solution optimal. Pour une 3-dislocation, un sur trois, puis encore un sur trois pour une 4-dislocation ..etc. Il semblait y avoir un rapport "presque logarithmique" entre Δ et T . On a alors montré une relation simple sur les T_Δ sur l'arbre binaire : $T_\Delta = 2(T_{\Delta-1} + 1)$, ce qui permet d'exprimer T en fonction de Δ , et

subissent une T -dislocation optimale, où T est la distance à une feuille du premier noeud sélectionné par l'algorithme.

Nombre optimal de dislocation

Puisqu'il est entendu que l'algorithme que nous appliquons est optimal sur les arbres, on va pouvoir donner le nombre de dislocation d'un arbre complet dans le cas général.

Notons n le nombre de sommets de l'arbre complet m -aire, h sa hauteur. Nous allons maintenant dénombrer les sommets de S lorsque l'algorithme termine. Il faut donc sommer sur tous les éléments appartenant aux T -dislocations des chaînes de longueur h . D'après ce qui précède sur l'étude des chaînes, on déduit :

$$\begin{aligned} \rho_{n,\Delta} &= \sum_{l=1}^{\lfloor h/T \rfloor} m^{h-Tl} \\ &= m^h \frac{m^{-T} - m^{-T(\lfloor h/T \rfloor + 1)}}{1 - m^{-T}} \\ &= \frac{m^h - m^{h \bmod (T\Delta)}}{m^{T\Delta} - 1} \end{aligned}$$

On a donc :

$$\boxed{\rho_{n,\Delta} = \frac{m^h - m^{h \bmod (T\Delta)}}{m^{T\Delta} - 1}}$$

avec

$$T_\Delta = \lceil \log_m((m-1)\Delta + m) \rceil \quad \text{et} \quad h = \lfloor \log_m(n) \rfloor$$

2.3 Autres arbres

Bien sûr on ne peut pas donner de formule générale pour un arbre, ni même pour un arbre localement complet car le nombre de Δ -dislocation dépend fortement de la structure de l'arbre. Par exemple :

Cependant, on peut utiliser un peu le même type de raisonnement que pour la chaîne pour régler le cas du peigne m -aire :

On voit que si $m \geq \Delta$, l'algorithme va donner un poids plus grand que $\Delta + 1$ à chaque noeud intérieur, qui feront donc tous partie de l'ensemble optimal de dislocation. Or les noeuds intérieurs forment clairement un ensemble de

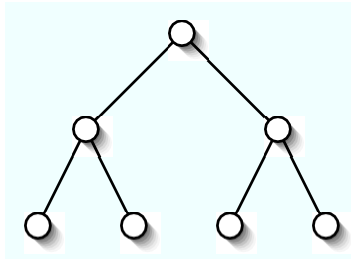


FIG. 11 – Arbre binaire complet, 7 sommets, où $\rho_1 = 2$

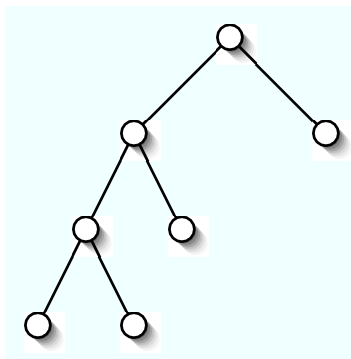


FIG. 12 – Un peigne, 7 sommets, où $\rho_1 = 3$.

Δ -dislocation quel que soit Δ , car toutes les parties connexes résultantes de leur suppression sont constituées d'un unique sommet.

Dans ce cas on a donc $\rho_{h,\Delta} = \lfloor \log_m(n) \rfloor = h$.

L'étude du second cas est quelque peu plus complexe, et nécessite les mêmes outils que pour l'arbre complet.

Notons v_1, \dots, v_h les noeuds internes de la racine vers les feuilles les plus éloignées de la racine (si $h = 0$, il n'y a pas de noeuds internes).

Tout d'abord on remarque qu'une application de l'algorithme dans laquelle on supprimerait systématiquement les noeuds internes mettrait des poids de $m + 1, m, \dots, m$ respectivement sur v_h, v_{h-1}, \dots, v_1 (récurrence immédiate).

On commence donc par remonter les m feuilles les plus éloignées de la racine, ce qui donne un poids de $m + 1$ sur v_h , qui remonte à son tour vers v_{h-1} .

Ensuite, on remonte les $m - 1$ autres fils de v_{h-1} , ce qui donne un poids de $2m + 1$ sur v_{h-1} .

Si $2m + 1 > \Delta$, on met v_{h-1} dans S et on continue ; sinon, on remonte encore jusqu'à v_{h-2} qui porte alors un poids de $3m + 1$..etc. On s'arrête quand on trouve k tel que $km + 1 > \Delta$.

Par division euclidienne :

$$\exists! k \in \left[2, \dots, \left\lfloor \frac{n}{m} \right\rfloor \right] \quad / \quad (k-1)m + 1 \leq \Delta < km + 1$$

Intuitivement, on enlèvera "à peu près" 1 nœud interne sur k pour la Δ -dislocation optimale du peigne. Plus exactement on retire d'abord le k^{ime} nœud interne le plus éloigné de la racine comme montré précédemment, puis on est ramené à un peigne m -aire de hauteur $h - k$, dont on aurait supprimé une des feuilles les plus éloignées de la racine. On peut faire le même raisonnement sur ce nouveau peigne, noté P'_{h-k} , associé au nombre de dislocation $\rho'_{h-k,\Delta}$.

Etat actuel de la formule de récurrence :

$$\rho_{h,\Delta} = \begin{cases} 1 + \rho'_{h-k,\Delta} = 1 + \rho'_{h - \left(1 + \left\lfloor \frac{\Delta-1}{m} \right\rfloor\right), \Delta} & \text{si } \Delta < n \\ 0 & \text{si } \Delta \geq n \end{cases}$$

Si $h \leq k + 1$,

$$\exists! k' \in \left[2, \dots, \left\lfloor \frac{n - (km + 1)}{m} \right\rfloor \right] \quad / \quad (k' - 1)m \leq \Delta < k'm$$

Sinon, on a clairement $\rho_{h,\Delta} = 1 + 0 = 1$.

Plaçons-nous dans le premier cas ; on a alors la formule générale de récurrence suivante, k' ne dépendant que de m et Δ :

$$\begin{aligned}\rho'_{h-k,\Delta} &= 1 + \rho'_{h-k-k',\Delta} \\ \rho'_{h-k,\Delta} &= 1 + \rho'_{h-k-(1+\lfloor \frac{\Delta}{m} \rfloor),\Delta}\end{aligned}$$

Une résolution immédiate donne

$$\begin{aligned}\rho'_{h-k,\Delta} &= \left\lfloor \frac{h-k}{1+\lfloor \frac{\Delta}{m} \rfloor} \right\rfloor + \rho'_{(h-k) \bmod (1+\lfloor \frac{\Delta}{m} \rfloor),\Delta} \\ &= \left\lfloor \frac{h-k}{1+\lfloor \frac{\Delta}{m} \rfloor} \right\rfloor,\end{aligned}$$

car les peignes P'_i n'ont que $m-1$ feuilles sur leur nœud interne le plus éloigné de la racine, ce qui implique $\rho'_{(h-k) \bmod (1+\lfloor \frac{\Delta}{m} \rfloor),\Delta} \leq \rho'_{\lfloor \frac{\Delta}{m} \rfloor,\Delta}$

Or un arbre de hauteur $\lfloor \frac{\Delta}{m} \rfloor$ a moins de $m \frac{\Delta}{m} = \Delta$ nœuds, donc est déjà Δ -disloqué.

Donc pour $\Delta > m$ et $h > 0$, avec $k = 1 + \lfloor \frac{\Delta-1}{m} \rfloor$:

$$\rho_{h,\Delta} = 1 + \left\lfloor \frac{h-k}{1+\lfloor \frac{\Delta}{m} \rfloor} \right\rfloor$$

Bien sûr si $h = 0$, $\rho_{h,\Delta} = 0$ pour tout Δ .

Conclusion :

$$\forall h \in \mathbb{N}, \forall m \in [2, +\infty], \forall \Delta \in \mathbb{N}^*,$$

$$\rho_{h,\Delta} = \begin{cases} 0 & \text{si } h = 0 \text{ ou } \Delta \geq n \\ h & \text{si } \Delta \leq m \\ 1 + \left\lfloor \frac{h-1-\lfloor \frac{\Delta-1}{m} \rfloor}{1+\lfloor \frac{\Delta}{m} \rfloor} \right\rfloor & \text{si } \Delta > m, \end{cases}$$

On vérifie que dans le cas $m = 1$ on retrouve la formule pour la chaîne : dans ce cas $h = n + 1$, et on obtient $\frac{n+1}{\Delta+1}$.

3 Recherche de cas extrémaux

Indépendamment de la difficulté du problème, il est intéressant de connaître les structures de graphes qui nécessitent le moins (resp. le plus) de sommets dans les ensembles de dislocation (idéalement pour tout Δ , sinon on peut faire l'étude pour quelques valeurs de Δ particulières). En effet ce critère semble pertinent en vue d'une application.

3.1 Graphes faciles à disloquer

Tout d'abord, il nous faut préciser la notion de difficulté de dislocation. Trois définitions viennent naturellement :

- Les Δ -dislocations avec Δ petit sont plus intéressantes que les autres, par exemple dans un réseau où il est nécessaire qu'un nœud communique toujours avec au moins deux autres :

$$\mu_p(G) = \frac{m}{n} \sum_{\Delta=1}^n (n - \Delta) \rho_{n,\Delta}$$

- Les Δ -dislocations avec Δ grand sont plus intéressantes que les autres, par exemple dans un réseau qui doit rester avec des parties connexes maximales :

$$\mu_g(G) = \frac{m}{n} \sum_{\Delta=1}^n \Delta \rho_{n,\Delta}$$

- On souhaite juste que G soit globalement résistant aux dislocations, et on choisit donc simplement :

$$\mu_s(G) = \frac{m}{n} \sum_{\Delta=1}^n \rho_{n,\Delta}$$

Le terme en $\frac{m}{n}$ représente en quelque sorte la densité du graphe en nombre d'arêtes par rapport au nombre de sommets.

Étant donnée une mesure μ , on dit que G est plus facile (resp. plus difficile) à disloquer que G' si $\mu(G) \leq \mu(G')$ (resp. $\mu(G) \geq \mu(G')$). Comparer directement des graphes pour ces mesures risque d'être délicat,

compte-tenu des définitions des formules : il faudrait pouvoir connaître suffisamment finement les nombres de dislocation d'un graphe quelconque. Mais on peut remarquer qu'une condition suffisante pour qu'un graphe G soit plus facile (resp. plus difficile) à disloquer qu'un graphe G' est que tous ses nombres de Δ -dislocation soient inférieurs (resp. supérieurs) à leurs homologues dans G' . On utilisera cette propriété par la suite.

Commençons par les graphes connexes ; on pourra ensuite éventuellement appliquer l'étude de ce cas aux graphes non connexes.

3.1.1 Cas connexe

L'étoile est clairement le graphe connexe le plus simple à disloquer parmi ceux à n sommets. En effet pour tout $\Delta < n$ on a $|S| = 1$, et tout autre graphe connexe à n sommets vérifie bien sûr $|S| \geq 1$.

Partons donc de l'étoile à n sommets et $n-1$ arêtes, et supposons que l'on dispose de plus d'arêtes (disons, pour fixer les idées, un nombre $k < n-1$ en surplus par rapport à $n-1$), il semble naturel de créer un début d'étoile sur un des sommets extrémaux avec ses voisins ordonnés ; ainsi $\rho_{n,\Delta} = 1$, pour $\Delta \geq k+1$, et $\rho_{n,\Delta} = 2$ sinon.

On voit s'amorcer une caractérisation des "pseudo-étoiles" avec un nombre arbitraire d'arêtes.

Cas général :

Soit $n \in \mathbb{N}$, soit $m \in [0, \dots, \frac{n(n-1)}{2}]$.

On construit le graphe suivant, appelé $E_{n,m}$:

1. Un sommet v_0 est pris comme référence, il sera le centre de l'"étoile" ; les autres sommets sont numérotés de 1 à $n-1$
2. Initialisation : $i \leftarrow 0$
3. Tant qu'il reste des arêtes, faire :
 - (a) Pour j allant de $i+1$ à $n-1$, créer l'arête (v_i, v_j)
 - (b) $i \leftarrow i+1$

On décompose m de la façon suivante : $m = (n-1) + (n-2) + (n-3) + \dots + (n-k) + r$ où $r < (n-k-1)$.

Par construction de $E_{n,m}$, on sait alors que $n-k$ sommets sont les centres de "pseudo-étoiles", et qu'il y a r arêtes en surplus : $(v_{k+1}, v_{k+2}), \dots, (v_{k+1}, v_{k+r+1})$.

Il est alors très facile de trouver le nombre de Δ -dislocation dans ce graphe, pour tout Δ :

$\forall \Delta \in \mathbb{N}^*$,

$$\rho_{E,\Delta} = \begin{cases} k+1 & \text{si } \Delta \leq r \\ k & \text{si } \Delta \in [r+1, n-k] \\ k-1 & \text{si } \Delta = n-(k-1) \\ k-2 & \text{si } \Delta = n-(k-2) \\ \dots & \\ 1 & \text{si } \Delta = n-1 \\ 0 & \text{si } \Delta \geq n \end{cases}$$

Ce qui peut être résumé en :

$$\rho_{E,\Delta} = \begin{cases} k+1 & \text{si } \Delta \leq r \\ k & \text{si } \Delta \in [r+1, n-k] \\ (n-\Delta)^+ & \text{si } \Delta > n-k \end{cases}$$

Cherchons à ce stade à déterminer k (qui peut être vu comme un indice de la grandeur de m par rapport à n).

Par définition, $n-k$ est l'unique entier α qui vérifie

$$\sum_{\alpha}^{n-1} i \leq m < \sum_{\alpha-1}^{n-1} i$$

Soit :

$$\frac{n(n-1)}{2} - \frac{\alpha(\alpha-1)}{2} \leq m < \frac{n(n-1)}{2} - \frac{(\alpha-1)(\alpha-2)}{2}$$

$$(\alpha-1)(\alpha-2) < n(n-1) - 2m \leq \alpha(\alpha-1)$$

On obtient deux inéquations quadratiques en $\beta = \alpha - 1$, que l'on résout séparément.

$$\begin{cases} \beta^2 - \beta + (2m - n(n-1)) < 0 \\ \beta^2 + \beta + (2m - n(n-1)) \geq 0 \end{cases}$$

Pour satisfaire les deux inégalités, on doit choisir β dans l'intervalle

$$\left[\frac{-1 + \sqrt{1 + 4(n(n-1) - 2m)}}{2}, \frac{1 + \sqrt{1 + 4(n(n-1) - 2m)}}{2} \right[$$

D'où, comme $k = n - \alpha = n - \beta - 1$:

$$k = \begin{cases} n - 1 - \frac{-1 + \sqrt{1 + 4(n(n-1) - 2m)}}{2} & \text{si } \sqrt{1 + 4(n(n-1) - 2m)} \text{ est un entier impair} \\ n - 1 - \left\lfloor \frac{1 + \sqrt{1 + 4(n(n-1) - 2m)}}{2} \right\rfloor & \text{sinon} \end{cases}$$

On sent donc d'après la formule précédente que $E_{n,m}$ est simple à disloquer, car si m est de l'ordre de n , k est très petit devant n . Donnons pour terminer les mesures du graphe $E_{n,m}$ (après simplifications)

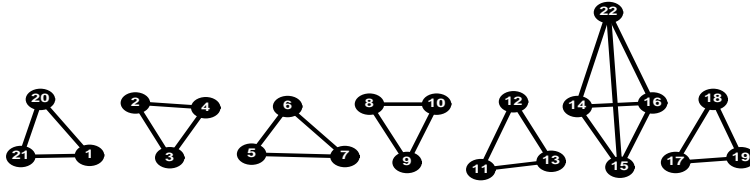
$$\begin{aligned} \mu_p(E_{n,m}) &= \frac{m}{n} \sum_{\Delta=1}^n (n - \Delta) \rho_{n,\Delta} \\ &= \frac{mr}{n} (n - k - 1) + m \frac{n - k - r}{n} (n - k) + \frac{m}{n} \sum_{\Delta=n-k+1}^n (n - \Delta)^2 \\ &= m \frac{(n - k)^2}{n} - \frac{m}{n} \left(\frac{k(k-1)(2k-1)}{6} + r \right) \end{aligned}$$

$$\begin{aligned} \mu_g(E_{n,m}) &= \frac{m}{n} \sum_{\Delta=1}^n \Delta \rho_{n,\Delta} \\ &= \frac{mr}{n} (k + 1) + m \frac{n - k - r}{n} k + \frac{m}{n} \sum_{\Delta=n-k+1}^n \Delta (n - \Delta) \\ &= m \frac{k(n - k)}{n} + m \frac{3k^2(n + 1) - k(3n + 1) - 2k^3 + r}{6n} \end{aligned}$$

$$\begin{aligned} \mu_s(E_{n,m}) &= \frac{m}{n} \sum_{\Delta=1}^n \rho_{n,\Delta} \\ &= \frac{m}{n} (r + n - k - r) + \sum_{\Delta=n-k+1}^n \{n - \Delta\} \\ &= m \frac{n - k}{n} + m \frac{k(k-1)}{2n} \\ &\simeq O(m) \text{ s'il y a peu d'arêtes : } \rho_{\Delta, \text{moy}} \text{ proche de } 1 \end{aligned}$$

On remarque qu'en général $E_{n,m}$ n'est pas le plus facile à disloquer pour un Δ fixé, comme le montre l'exemple suivant avec $\Delta = 3$; en revanche, les

$E_{n,m}$ ont l'avantage d'être très faciles à construire, et sont probablement les structures pour lesquelles les trois mesures proposées sont minimales. Ceci étant, une démonstration semble hors de portée de cet article.



3.1.2 Cas disconnexe

S'inspirant du cas précédent, on imagine que des étoiles à p sommets et $p - 1$ arêtes déconnectées constitueront des graphes faciles à disloquer. Or quelques problèmes se posent : une fois construit un certain nombre d'étoiles, s'il reste des arêtes, vaut-il mieux les ajouter dans une étoile particulière ou bien en parallèle dans chaque étoile ? Comment choisir p ?

On va plutôt chercher de l'inspiration dans le dernier exemple du paragraphe précédent, dont on généralise l'idée :

Soient donc $n, m \in \mathbb{N}$ avec $m \leq \frac{n(n-1)}{2}$.

On commence par la construction d'un graphe complet d'ordre Δ (K_Δ), puis s'il reste moins de Δ sommets on inter-connecte ces derniers aléatoirement ; si on a construit un graphe complet et qu'il reste des arêtes, on choisit un sommet dans le dernier graphe construit que l'on relie à ceux du K_Δ de départ ; si après avoir crée toutes les liaisons possibles il reste des arêtes, on continue avec un autre sommet ..etc.

S'il reste plus de Δ sommets, on commence la construction d'un second K_Δ que l'on prend pour nouveau graphe de départ ..etc.

Plus formellement : on effectue la division euclidienne de n par Δ : $n = k\Delta + r$, $r < \Delta$, puis on groupe les sommets en k paquets de Δ sommets, plus un à r sommets. Tant qu'il reste des arêtes, on construit des K_Δ avec les k premiers paquets, puis on interconnecte aléatoirement le dernier paquet à r sommets. S'il reste des arêtes (et tant qu'il en reste), on connecte ensuite les sommets du dernier paquet à tous ceux du k^{eme} paquet, puis à tous ceux du $(k-1)^{eme}$ paquet ..etc. S'il reste des arêtes à la fin de cette opération, on recommence en connectant de la même façon le k^{eme} paquet au $(k-1)^{eme}$, jusqu'à obtenir K_n dans le cas extrême.

Sur un tel graphe, on obtient les formules suivantes pour $r = 0$:

$$\rho_{n,m,\Delta} = \begin{cases} 0 & \text{si } m \leq k \frac{\Delta(\Delta-1)}{2} \\ 1 & \text{si } k \frac{\Delta(\Delta-1)}{2} < m \leq (2k-1) \frac{\Delta(\Delta-1)}{2} \\ 2 & \text{si } (2k-1) \frac{\Delta(\Delta-1)}{2} < m \leq (3k-3) \frac{\Delta(\Delta-1)}{2} \\ \dots & \\ p & \text{si } (pk - \frac{p(p-1)}{2}) \frac{\Delta(\Delta-1)}{2} < m \leq ((p+1)k - \frac{p(p+1)}{2}) \\ \dots & \end{cases}$$

Soit, si $m > k \frac{\Delta(\Delta-1)}{2}$:

$$\left(\rho_{n,m,\Delta} k - \frac{\rho_{n,m,\Delta}(\rho_{n,m,\Delta} - 1)}{2} \right) \frac{\Delta(\Delta-1)}{2} < m \leq \left((\rho_{n,m,\Delta} + 1)k + \frac{\rho_{n,m,\Delta}(\rho_{n,m,\Delta} + 1)}{2} \right) \frac{\Delta(\Delta-1)}{2}$$

Double inégalité de degré 2 que l'on résout pour trouver finalement que $\rho_{n,m,\Delta}$ est l'unique entier dans l'intervalle $\left[-\frac{3}{2} + \sqrt{\frac{1}{4} + \frac{4m}{\Delta-1}}, -\frac{1}{2} + \sqrt{\frac{1}{4} + \frac{4m}{\Delta-1}} \right]$.

Cas $r > 0$:

$$\rho_{n,m,\Delta} = \left\{ \begin{array}{ll} 0 & \text{si } m \leq k \frac{\Delta(\Delta-1)}{2} + \frac{r(r-1)}{2} \\ 1 & \text{si } k \frac{\Delta(\Delta-1)}{2} + \frac{r(r-1)}{2} < m \leq 2k \frac{\Delta(\Delta-1)}{2} + \frac{r(r-1)}{2} \\ 2 & \text{si } 2k \frac{\Delta(\Delta-1)}{2} + \frac{r(r-1)}{2} < m \leq 3k \frac{\Delta(\Delta-1)}{2} + \frac{r(r-1)}{2} \\ \dots & \\ r-1 & \text{si } (r-1)k \frac{\Delta(\Delta-1)}{2} + \frac{r(r-1)}{2} < m \leq rk \frac{\Delta(\Delta-1)}{2} + \frac{r(r-1)}{2} \\ r & \text{si } rk \frac{\Delta(\Delta-1)}{2} + \frac{r(r-1)}{2} < m \leq (rk+k-1) \frac{\Delta(\Delta-1)}{2} + \frac{r(r-1)}{2} \\ r+1 & \text{si } (rk+k-1) \frac{\Delta(\Delta-1)}{2} + \frac{r(r-1)}{2} < m \leq (rk+2k-3) \frac{\Delta(\Delta-1)}{2} + \frac{r(r-1)}{2} \\ \dots & \\ p & \text{si } (rk+(p-r)k - \frac{(p-r)(p-r+1)}{2}) \frac{\Delta(\Delta-1)}{2} < m \leq (rk+(p-r+1)k - \frac{(p-r+1)(p-r+2)}{2}) \frac{\Delta(\Delta-1)}{2} \\ \dots & \end{array} \right.$$

On remarque que l'on peut souvent construire un graphe plus facile à disloquer que dans le cas connexe.

De plus quand la construction disconnexe arrive à un graphe connexe (si le nombre d'arêtes le permet), on obtient de nouvelles structures connexes faciles à disloquer ; on devrait alors pouvoir montrer que celles-ci sont globalement plus difficiles à disloquer que les pseudo-étoiles précédemment construites.

3.2 Chaîne et cycle : difficiles à disloquer ?

La chaîne est un arbre à n sommets (et donc $n-1$ arêtes). En fait imposer à un graphe d'avoir moins d'arêtes que de sommets et d'être connexe revient à dire que c'est un arbre : en effet si ce graphe possède un cycle, chaque ajout de sommet à partir de ce cycle se fera nécessairement avec un ajout d'arête ; on ne pourra donc pas faire baisser le nombre d'arêtes par rapport au nombre de sommets.

Il suffit donc de montrer que la chaîne est l'arbre le plus difficile à disloquer parmi les arbres à n sommets.

Or ce dernier résultat est évident en examinant ce que fait l'algorithme de dislocation d'arbres : à chaque ajout d'un sommet dans S , le sous-arbre de racine v est de poids $\geq \Delta$, donc on ajoute moins d'un sommet sur $\Delta+1$ dans S . Or pour la chaîne on ajoute exactement un sommet sur $\Delta+1$ dans S . D'ù le résultat.

Théorème :

La chaîne est l'arbre à n sommets le plus difficile à disloquer

On montre maintenant que le cycle est le graphe connexe à n sommets et n arêtes le plus difficile à disloquer, pour tout Δ .

Tout d'abord, on a la formule suivante pour la Δ -dislocation du cycle C_n déduite directement de celle pour la chaîne :

$$\rho_{C_n, \Delta} = 1 + \left\lfloor \frac{n-1}{\Delta+1} \right\rfloor$$

Soit G un graphe connexe à n sommets et n arêtes.

H_n : Pour tout Δ dans \mathbb{N}^* , pour tout graphe connexe G à n sommets et n arêtes, $\rho_{G, \Delta} \leq \rho_{C_n, \Delta}$

Démonstration :

Soit $n \in \mathbb{N}$. Soit G un graphe connexe à n sommets et n arêtes. On va montrer H_n en deux étapes :

1. On montre que G possède un unique cycle
2. On montre en utilisant une variante de l'algorithme sur les arbres que G est plus facile à disloquer que le cycle

1— On commence par constater que G contient toujours un cycle, car tout graphe à n sommets et n arêtes s'obtient à partir du cycle par déplacement d'arêtes. La preuve correcte se fait donc par récurrence (l'initialisation étant immédiate) : supposons qu'après un certain nombre de changement d'arêtes, G possède le cycle $(v_1, v_2, \dots, v_k = v_1)$. Soit alors (u, v) une arête de G que l'on déplace. Si (u, v) n'est pas une arête du cycle, la propriété reste vraie au rang suivant. Si en revanche $(u, v) = (v_i, v_{i+1})$, il faut regarder de plus près la structure de G après suppression de (u, v) , mais avant son rajout :

Certains sommets parmi ceux de l'ancien cycle sont les racines d'arborescences (car G avait un unique cycle par hypothèse de récurrence), et les autres sommets ne sont que de degré 2. On en déduit que G est à présent un arbre, et donc le rajout de l'arête (u, v) aura pour effet de créer un unique cycle (propriété connue).

D'où le résultat annoncé.

2- G est ainsi constitué d'un unique cycle $(v_1, v_2, \dots, v_k = v_1)$, dont certains sommets sont racines d'arborescences, les autres étant simplement de degré 2. La résolution algorithmique est alors claire : il suffit d'appliquer l'algorithme fonctionnant sur les arbres à chaque arborescence enracinée, en remontant jusqu'à la racine qui se retrouve non pas supprimée comme si on avait un vrai arbre, mais affectée d'un poids compris entre 1 et $m\Delta$ où m est le nombre de descendants de la racine. Bien sûr les nœuds ayant ainsi un poids plus grand que $\Delta + 1$ appartiennent à S , l'ensemble optimal de dislocation. Il nous reste plus qu'à disloquer optimalement un cycle "avec poids" ou bien une réunion disjointe de chaînes "avec poids" dans le cas où une des racines a un poids plus grand que $\Delta + 1$.

Il faut faire attention car on ne se ramène pas facilement au second cas lorsque l'on est dans le premier en supprimant un sommet de plus haut poids, étant donné l'absence de symétrie du problème.

Tout d'abord on remarque que le nombre de nœuds dans l'ensemble de dislocation à la fin de l'application de l'algorithme de dislocation d'arbres (et donc avant de supprimer les sommets de poids trop grand dans le cycle obtenu) est majoré par $\lfloor \frac{n-P}{\Delta+1} \rfloor$ où P est le poids total du cycle, d'après le paragraphe démontrant que la chaîne est l'arbre le plus difficile à disloquer. Il suffit donc de montrer qu'on ajoutera ensuite moins de $1 + \lfloor \frac{P-1}{\Delta+1} \rfloor$ sommets dans S , étant donné l'inégalité $\lfloor \frac{a+b}{c} \rfloor \geq \lfloor \frac{a}{c} \rfloor + \lfloor \frac{b}{c} \rfloor$.

Ce dernier résultat se montre facilement en choisissant d'abord un sommet de plus haut poids, puis en appliquant l'algorithme de dislocation d'une chaîne qui marque un sommet si et seulement si le poids total P' de ses prédécesseurs est $\leq \Delta$, et que son poids ajouté à P' est $> \Delta$: lorsqu'on marque un sommet, on fait diminuer le poids total d'au moins $\Delta + 1$. Étant donné que le poids total disponible alors est $\leq P - 1$, on a démontré ce qu'on voulait en recollant les morceaux.

Théorème :

Le cycle est le graphe connexe à n sommets et n arêtes le plus difficile à disloquer

Qu'en est-il lorsque l'on impose plus à G d'être connexe ?

G est alors constitué de parties connexes disjointes, chacune ayant p sommets et q arêtes tels que la somme des p (resp. q) donne n . On observe que si une des sous-parties connexes a plus d'arêtes que de sommets, alors il y a $a - s$ sommets isolés dans le graphe, où a est le nombre d'arêtes de la composante connexe et s son nombre de sommets. On peut donc séparer les composantes

connexes en deux : celles qui ont un unique sommet, et celles à plus d'un sommet.

On affine cette classification en séparant les composantes connexes à plus d'un sommet en trois catégories : celles qui ont p sommets et $p - 1$ arêtes (des arbres, donc), celles à p sommets et p arêtes (on se ramène alors au cas précédent), et les autres (qui ont trop d'arêtes). Dans les composantes connexes qui ont trop d'arêtes on peut faire la majoration grossière du nombre de dislocation par $q - p + 1 + \lfloor \frac{p-1}{\Delta+1} \rfloor$, et dans les autres on obtient respectivement les majorations par $\lfloor \frac{p}{\Delta+1} \rfloor$ et $1 + \lfloor \frac{p-1}{\Delta+1} \rfloor$.

Notons $U = \{G_k\}_{k \in [1, \dots, p]}$ l'ensemble des sous-graphes connexes qui sont des arbres, $V = \{G'_k\}_{k \in [1, \dots, p']}$ l'ensemble des sous-graphes connexes avec autant de sommets que d'arêtes, $W = \{G''_k\}_{k \in [1, \dots, p']}$ l'ensemble des sous-graphes connexes avec un surplus d'arêtes et enfin $\{v_k\}_{k \in [1, \dots, q]}$ l'ensemble des sommets isolés. On a la relation

$$\sum_{G \in W} (m_G - n_G) = q,$$

avec les notations évidentes. On obtient donc :

$$\rho_{n, \Delta} \leq \sum_{G \in U} \lfloor \frac{n_G}{\Delta + 1} \rfloor + \sum_{G \in V} \left\{ 1 + \lfloor \frac{n_G - 1}{\Delta + 1} \rfloor \right\} + q + \sum_{G \in W} \left\{ 1 + \lfloor \frac{n_G - 1}{\Delta + 1} \rfloor \right\}$$

On voit maintenant que la majoration par $q - p + 1 + \lfloor \frac{p-1}{\Delta+1} \rfloor$ était trop grossière, car si l'on dispose de $\frac{n(n-1)}{2}$ sommets, et que l'on construit un graphe complet d'ordre n cotoyant $\frac{n(n-3)}{2}$ sommets isolés, on aboutit à $\rho_{\frac{n(n-1)}{2}, \Delta} \leq \alpha + \frac{n(n-3)}{2}$ où $\alpha \geq 0$. Or pour $\Delta = n$, on devrait avoir la majoration $\rho_{\frac{n(n-1)}{2}, \Delta} \leq 1 + \frac{n-1}{2}$.

Finalement, on ne peut pas donner de démonstration générale si l'on n'est pas capables de majorer finement les nombres de dislocation dans un graphe connexe quelconque, tâche qui semble délicate.

Donnons en conclusion les mesures approchées de la chaîne et du cycle, pour n "grand" :

Chaîne :

$$\begin{aligned}
\mu_p(CH_n) &= \frac{n-1}{n} \sum_{\Delta=1}^n (n-\Delta) \rho_{n,\Delta} \\
&= \frac{n-1}{n} \sum_{\Delta=1}^n (n-\Delta) \left\lfloor \frac{n}{\Delta+1} \right\rfloor \\
&\simeq n^2 \ln(n) - n^2 + n \ln(n) \\
&= n^2 (\ln(n) - 1) + n \ln(n)
\end{aligned}$$

$$\begin{aligned}
\mu_g(CH_n) &= \frac{n-1}{n} \sum_{\Delta=1}^n \Delta \left\lfloor \frac{n}{\Delta+1} \right\rfloor \\
&\simeq n^2 - n \ln(n)
\end{aligned}$$

$$\begin{aligned}
\mu_s(CH_n) &= \frac{n-1}{n} \sum_{\Delta=1}^n \left\lfloor \frac{n}{\Delta+1} \right\rfloor \\
&\simeq n \ln(n)
\end{aligned}$$

Cycle :

$$\begin{aligned}
\mu_p(CH_n) &= \frac{n-1}{n} \sum_{\Delta=1}^n (n-\Delta) \rho_{n,\Delta} \\
&= \frac{n-1}{n} \sum_{\Delta=1}^n (n-\Delta) \left(1 + \left\lfloor \frac{n-1}{\Delta+1} \right\rfloor \right) \\
&\simeq n^2 (\ln(n) - 1) + n \ln(n) + \frac{n+1}{2}
\end{aligned}$$

$$\begin{aligned}
\mu_g(CH_n) &= \frac{n-1}{n} \sum_{\Delta=1}^n \Delta \left(1 + \left\lfloor \frac{n-1}{\Delta+1} \right\rfloor \right) \\
&\simeq n^2 - n \ln(n) + \frac{n+1}{2}
\end{aligned}$$

$$\begin{aligned}\mu_s(CH_n) &= \frac{n-1}{n} \sum_{\Delta=1}^n \left(1 + \left\lfloor \frac{n-1}{\Delta+1} \right\rfloor\right) \\ &\simeq n(\ln(n) + 1)\end{aligned}$$

Bien sûr le cycle et la chaîne sont asymptotiquement de même difficulté de dislocation.

3.3 Une généralisation du cycle

En s'inspirant du cas précédent, on a pensé que les puissances de cycles seraient les graphes les plus difficiles à disloquer parmi tous ceux ayant le même nombre de sommets et d'arêtes. Définissons d'abord ce qu'est une puissance de cycle dans le cas où tous les sommets sont de degré pair, à n sommets :

Soit $p \in 2\mathbb{N}$, $p = 2k$.

Si $p = 0$, le graphe est un nuage de points sans arêtes. Sinon, on commence par construire le cycle à n sommets, puis si les degrés des nœuds sont encore trop petits, on crée les arêtes $(v_i, v_{i+2 \bmod n})$; si les degrés sont encore trop petits (ils valent maintenant 4), on crée les arêtes $(v_i, v_{i+3 \bmod n})$..etc.

Soit $\Delta \in \mathbb{N}^*$.

Observons tout d'abord qu'une solution pour la Δ -dislocation s'obtient en choisissant une chaîne (chaîne au sens du plus grand cycle) de longueur $\Delta + k$, puis en découpant le plus grand cycle suivant cette chaîne en $\lfloor \frac{n}{\Delta+k} \rfloor$ chaînes consécutives de longueur $\Delta + k$. On prend les k premiers sommets de chaque chaîne ainsi définie, et il nous reste une chaîne de longueur $\Delta + n \bmod (\Delta + k)$, que l'on disloque facilement avec $\min \{k, n \bmod (\Delta + k)\}$ sommets. Les sommets "coincés" entre deux paquets de k sommets de S ne communiquent pas avec les autres sommets, donc on a bien une Δ -dislocation.

On obtient ainsi un majorant de $\rho_{n,p,\Delta} : k \lfloor \frac{n}{\Delta+k} \rfloor + \min \{k, n \bmod (\Delta + k)\}$. On va montrer que ce majorant est en fait optimal.

Pour cela, commençons par montrer que pour toute chaîne de longueur $\Delta + k$ aura au moins k sommets dans S , où S est un ensemble de dislocation optimal.

Soit donc $C = (v_1, \dots, v_{\Delta+k})$ une chaîne de longueur $\Delta + k$. On montre que pour toute paire de sommets il y a au moins k chemins disjoints au sens des sommets les reliant.

Soient v_i, v_j deux sommets distincts de C , avec $i < j$.

Il suffit de faire des pas de longueur k en partant de v_i vers la droite, puis lorsqu'on dépasse v_j ou que l'on ne peut plus avancer on rejoint v_j par une arête. On recommence par un pas de longueur 1 vers la droite au départ de v_i , puis à nouveau des pas de longueur k ..etc. lorsqu'on arrive à α tel que $i + \alpha = j$, on effectue d'abord un pas en arrière puis à nouveau des pas de longueur k vers v_j . Ensuite, 2 pas en arrière puis des pas de longueur k jusqu'à v_j comme précédemment ..etc.

On en déduit que dans chaque chaîne de longueur $\Delta + k$ il faut supprimer au moins k sommets. Le résultat annoncé arrive alors par récurrence sur $n \bmod (\Delta + k)$, en prenant pour hypothèse de récurrence le schéma de solution suivant :

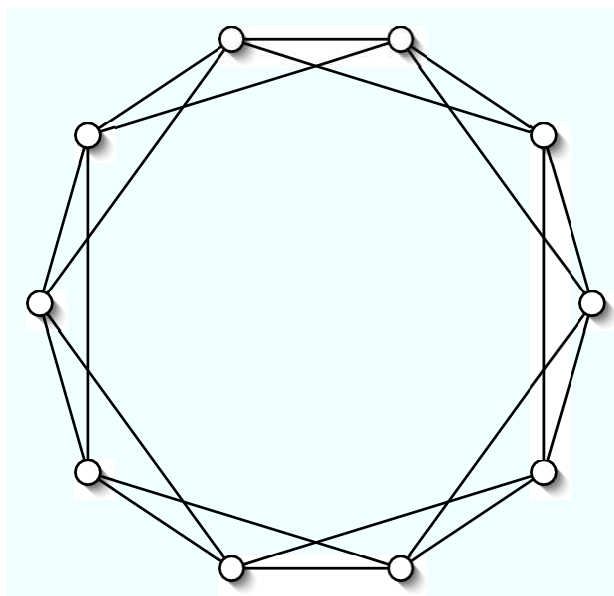


FIG. 13 – Un exemple de puissance de cycle. (A 10 sommets, 4-régulier)

Si $n \bmod (\Delta + k) = 0$, on a trouvé une solution atteignant la borne inférieure ci-dessus (en fait égale à la borne supérieure dans ce cas).
 Supposons $n \bmod (\Delta + k) = m$, et que la borne supérieure soit atteinte dans le cas $n \bmod (\Delta + k) = m - 1$.
 Si $k \leq m - 1$ c'est clair ; sinon, on remarque qu'il faut et qu'il suffit d'ajouter un sommet dans S .

Conclusion :

$$\rho_{n,p,\Delta} = k \left\lfloor \frac{n}{\Delta+k} \right\rfloor + \min \{k, n \bmod (\Delta+k)\}$$

On peut maintenant se demander si tous les graphes dont tous les sommets sont de même degré "se ressemblent". c'est faux avec le contre-exemple suivant qui n'est bien sûr pas isomorphe à une puissance (ou pseudo-puissance) de cycle déjà construite :

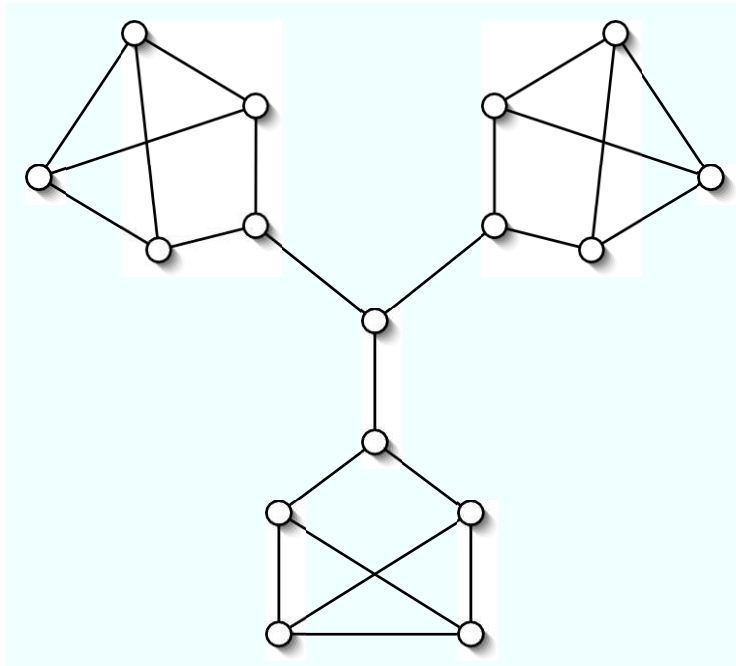


FIG. 14 – Un contre-exemple : tous les sommets ont même degré

Or ce dernier exemple n'est que 1-connexe car il suffit d'enlever une des arêtes centrales pour déconnecter le graphe. En revanche on voit facilement que les p -puissances (ou pseudo-puissances) de cycle sont p connexes. C'est à dire que ces graphes ont un nombre de connexité maximal (celui-ci ne pouvant excéder le degré maximal d'un nœud, ici p). Il doit y avoir un lien entre les notions de Δ -dislocation et de k -connexité.

On pourrait essayer alors de montrer que les puissances de cycles ainsi définies sont les plus difficiles à disloquer dans leur classe de graphes de même nombre de sommets et d'arêtes.

On a pensé à l'idée suivante : on part du sommet de G de plus bas degré

(en supposant que G ne soit pas régulier ; sinon, voir l'exemple plus bas). On construit alors le sous-graphe induit par lui et ses voisins immédiats : si ce voisinage est de cardinal $\leq \Delta$, on continue en choisissant un des voisins de degré minimal.

Le problème est que l'on ne sait pas disloquer optimalement un sous-graphe par rapport à la structure globale ; l'approche par récurrence (le cas du cycle étant réglé) aurait peut-être plus de succès, mais nous n'avons pas trouvé de preuve.

A défaut de pouvoir donner une démonstration, donnons des approximations des mesures d'une puissance de cycle à n sommets, p -régulier avec $p = 2k$, en supposant n "grand" :

$$\begin{aligned}
\mu_p(P_{n,p}) &= \frac{nk}{n} \sum_{\Delta=1}^n (n - \Delta) \rho_{n,\Delta} \\
&= k \sum_{\Delta=1}^n (n - \Delta) \left\{ k \left\lfloor \frac{n}{\Delta + k} \right\rfloor + \min \{k, n \bmod (\Delta + k)\} \right\} \\
&\simeq k^2 \sum_{\Delta=1}^n (n - \Delta) \left\lfloor \frac{n}{\Delta + k} \right\rfloor \\
&\simeq k^2 n^2 \sum_{\Delta=1}^n \frac{1}{\Delta + k} - k^2 n \sum_{\Delta=1}^n \frac{\Delta}{\Delta + k} \\
&\simeq k^2 n^2 (\ln(n + k) - \ln(k)) + k^3 n (\ln(n + k) - \ln(k)) - k^2 n^2 \\
&\simeq k^2 n^2 (\ln(n + k) - \ln(k) - 1) + k^3 n (\ln(n + k) - \ln(k))
\end{aligned}$$

$$\begin{aligned}
\mu_g(P_{n,p}) &= k \sum_{\Delta=1}^n \Delta \left\{ k \left\lfloor \frac{n}{\Delta + k} \right\rfloor + \min \{k, n \bmod (\Delta + k)\} \right\} \\
&\simeq k^2 \sum_{\Delta=1}^n \Delta \left\lfloor \frac{n}{\Delta + k} \right\rfloor \\
&= k^2 n (n - \ln(n + k)) + \ln(k) \\
&= k^2 n^2 - k^2 (\ln(n + k) - \ln(k))
\end{aligned}$$

$$\begin{aligned}
\mu_s(P_{n,p}) &= k \sum_{\Delta=1}^n \left\{ k \left\lfloor \frac{n}{\Delta+k} \right\rfloor + \min \{k, n \bmod (\Delta+k)\} \right\} \\
&\simeq k^2 \sum_{\Delta=1}^n \left\lfloor \frac{n}{\Delta+k} \right\rfloor \\
&= k^2 n (\ln(n+k) - \ln(k))
\end{aligned}$$

Regardons le cas des "pseudo puissances de cycles", où tous les sommets sont cette fois de degré p impair (nécessairement alors n est pair). Cela revient à prendre la puissance de cycle avec tous les sommets de degré $p+1$, puis à supprimer une arête sur deux en parcourant le plus grand cycle. On n'a alors plus la symétrie du problème, ce qui rend difficile a priori l'obtention de solutions périodiques élégantes; toujours est-il que sur de petits exemples résolus à l'aide des algorithmes, on voit apparaître une structure régulière des solutions, en groupant encore les sommets comme dans le cas précédent.

Ce pourrait être l'objet d'une étude ultérieure.

3.4 Comparaison à une autre classe de graphes

Maintenant que l'on a vu qu'il est intéressant d'avoir un graphe dont tous les sommets sont de même degré (graphe *régulier*), on peut essayer d'en construire d'autres.

Soit $n, p \in \mathbb{N}$. On construit un graphe composé de p sous-graphes complets d'ordre n , connectés chacun à leurs voisins de droite et de gauche (en numérotant les K_n) de la façon suivante : chaque sommet v_k est connecté à son homologue dans le K_n voisin à droite, et dans le K_n voisin à gauche (en numérotant les sommets de chaque K_n).

Chaque sommet est alors de degré $n+1$, et on peut essayer de trouver des formules sur ce type de graphe.

Commençons par deux K_n connectés. On voit que lorsque l'on enlève un sommet, il subsiste un K_n cotoyant un K_{n-1} , un seul des sommets du K_n n'ayant pas d'homologue dans l'autre graphe complet, puis deux si on enlève un sommet du K_{n-1} obtenu qui devient alors K_{n-2} ..etc. Par récurrence immédiate, après avoir retiré k sommets il subsiste deux graphes complets, d'ordre total $n-k$.

On voit alors déjà que pour une Δ -dislocation où $\Delta > n$, il faut et il suffit de retirer $(2n - \Delta)^+$ sommets quelconques au graphe (appelé CK_n). Si

$\Delta < \lceil \frac{n}{2} \rceil$, encore d'après ce qui précède il faut retirer au moins $2(n - \Delta)$ sommets. Or on peut ne pas en enlever plus, en suivant la stratégie suivante : on enlève $n - \Delta$ sommets quelconques du K_n de gauche, puis on supprime dans le K_n de droite en priorité les sommets qui sont reliés à un de l'autre graphe complet (il y en a exactement Δ , or par hypothèse $\Delta \leq n - \Delta$). On obtient ainsi deux parties connexes disjointes de cardinal Δ .

Le cas intéressant est lorsque $\lceil \frac{n}{2} \rceil \leq \Delta \leq n$.

On va alors montrer que le nombre de Δ -dislocation est toujours n (ce qui donne un intérêt à ces graphes).

Il suffit en fait de prendre $n - \Delta$ sommets quelconques dans un des K_n , puis de supprimer les Δ sommets de l'autre K_n qui sont connectés à un sommet du premier K_n . On obtient clairement une Δ -dislocation, car $n - \Delta \leq \Delta$.

On ne peut pas faire mieux, car ces nombres de dislocation doivent être bornés inférieurement (par monotonie de $\rho_{n,\Delta}$) par le plus grand nombre de dislocation pour le cas $\Delta > n$, qui est $2n - n - 1 = n - 1$, et supérieurement par le plus petit nombre de dislocation du cas $\Delta < \lceil \frac{n}{2} \rceil$, qui est $2n - 2\lceil \frac{n}{2} \rceil + 2 \leq n$. Or une solution avec $n - 1$ sommets est impossible, car alors les deux graphes communiqueraient toujours, et il subsisterait plus de $n + 1$ sommets connectés.

D'où au final les nombres de dislocation suivants :

$$\rho_{CK,\Delta} = \begin{cases} 2(n - \Delta) & \text{si } \Delta < \lceil \frac{n}{2} \rceil \\ n & \text{si } \lceil \frac{n}{2} \rceil \leq \Delta \leq n \\ (2n - \Delta)^+ & \text{si } \Delta > n \end{cases}$$

Donnons pour terminer les mesures de ces graphes (après simplifications)

$$\begin{aligned} \mu_p(CK_n) &= \frac{n + 2\frac{n(n-1)}{2}}{2n} \sum_{\Delta=1}^{2n} (2n - \Delta) \rho_{n,\Delta} \\ &= \frac{n}{2} \left(\sum_{\Delta=1}^{\lceil \frac{n}{2} \rceil - 1} 2(n - \Delta)(2n - \Delta) + \sum_{\Delta=\lceil \frac{n}{2} \rceil}^n n(2n - \Delta) + \sum_{\Delta=n+1}^{2n} (2n - \Delta)^2 \right) \\ &= \frac{n}{12} \left(18n^2 \left(\lceil \frac{n}{2} \rceil - 1 \right) - 9n \lceil \frac{n}{2} \rceil \left(\lceil \frac{n}{2} \rceil - 1 \right) + 2 \lceil \frac{n}{2} \rceil^3 - 3 \lceil \frac{n}{2} \rceil^2 + \lceil \frac{n}{2} \rceil \right) \end{aligned}$$

$$\begin{aligned}
\mu_g(P_{n,p}) &= \frac{n}{2} \left(\sum_{\Delta=1}^{\lfloor \frac{n}{2} \rfloor - 1} 2\Delta(n - \Delta) + \sum_{\Delta=\lfloor \frac{n}{2} \rfloor}^n \Delta n + \sum_{\Delta=n+1}^{2n} \Delta(2n - \Delta) \right) \\
&= \frac{n}{12} \left(7n^3 + 6n \lfloor \frac{n}{2} \rfloor \left(\lfloor \frac{n}{2} \rfloor - 1 \right) - 4 \lfloor \frac{n}{2} \rfloor^3 + 3 \lfloor \frac{n}{2} \rfloor^2 + \lfloor \frac{n}{2} \rfloor - n \right)
\end{aligned}$$

$$\begin{aligned}
\mu_s(P_{n,p}) &= \frac{n}{2} \left(\sum_{\Delta=1}^{\lfloor \frac{n}{2} \rfloor - 1} 2(n - \Delta) + \sum_{\Delta=\lfloor \frac{n}{2} \rfloor}^n n + \sum_{\Delta=n+1}^{2n} (2n - \Delta) \right) \\
&= \frac{n}{4} \left(3n^2 + 2n \lfloor \frac{n}{2} \rfloor - 3n - 2 \lfloor \frac{n}{2} \rfloor \left(\lfloor \frac{n}{2} \rfloor - 1 \right) \right)
\end{aligned}$$

La généralisation à p graphes complets connectés semble abordable et intéressante, aussi nous-y attardons-nous un instant :

Comme précédemment, on commence (par symétrie) par un sommet quelconque, puis ayant obtenu $p - 1$ graphes complets d'ordre n et un d'ordre $n - 1$, on continue en prenant un sommet soit dans un des K_n , soit dans le K_{n-1} ..etc.

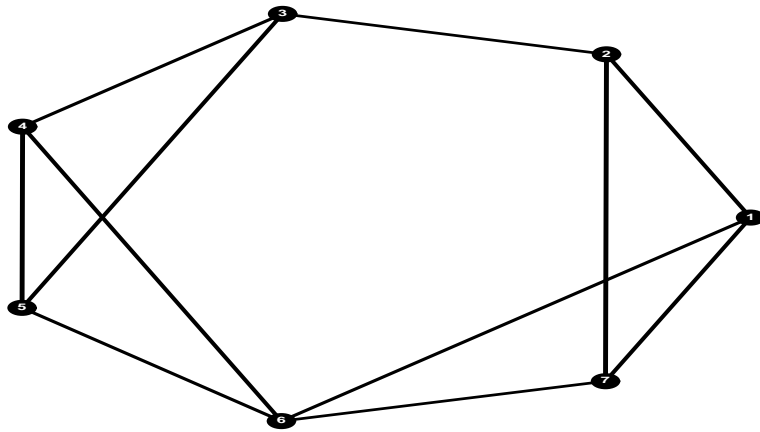
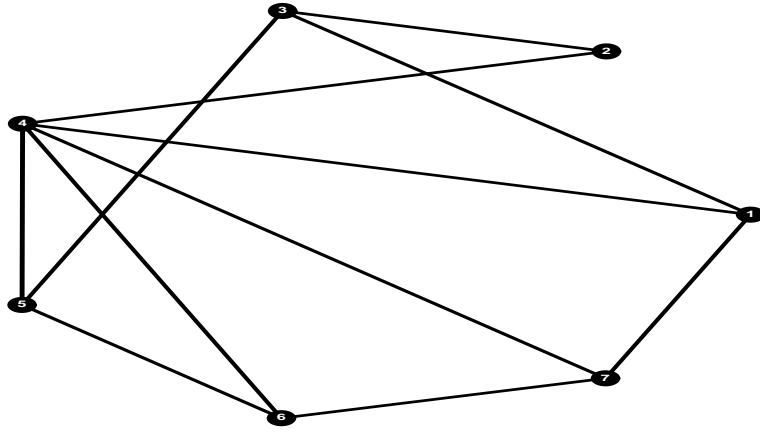
Il resterait à faire l'étude complète en suivant un peu le même schéma *conjecture-formules-preuve*.

3.5 Conclusion

On a montré que lorsque n divise $2m$ les graphes les plus difficiles à disloquer sont les puissances de cycles. Qu'en est-t-il lorsque n et m sont quelconques ? Une première idée consiste à construire la puissance de cycle s'approchant le plus de la structure voulue (on résout l'équation $\frac{nk}{2} = m'$ pour $m' = \lfloor \frac{m}{n} \rfloor$ si n est impair, $m' = \lfloor m/\frac{n}{2} \rfloor$ si n est pair), puis à ajouter des arêtes de manière à augmenter le plus possible les nombres de dislocation. Ce n'est pas toujours optimal, comme le montre l'exemple suivant à 7 sommets et 11 arêtes,

qui est plus difficile à 3-disloquer que le début de puissance de cycle construit ici :

On en conclut qu'en pratique avec $n \ll m$, on peut se contenter de l'approximation par la puissance de cycle, et que les cas où n et m sont du même ordre de grandeur doivent être étudiés au cas par cas, faute de structure générale des graphes difficiles à disloquer.



4 Applications

4.1 Stoppage de la propagation d'un virus

Supposons qu'un virus informatique soit en train de se propager à travers un réseau d'ordinateurs, et que l'on ait pour tâche de limiter la contamination des postes tout en maintenant le plus de stations de travail actives. Il suffit en fait de trouver une Δ -dislocation optimale du graphe formé par les ordinateurs, où Δ est choisit assez petit pour que la contamination d'un des postes d'une partie connexe affecte un voisinage réduit.

Si l'on ne peut pas localiser le virus (par exemple s'il n'est pas encore actif, et indétectable par les antivirus), alors on garantit avec une bonne probabilité que le plus de groupes de Δ ordinateurs possibles sont sains.

4.2 L'invasion au jeu de go

Au go on a la notion de vie et de mort, un groupe de pierres étant "vivant" lorsqu'il possède au moins deux espaces vides (appelés "yeux") ou peut s'en créer, mort dans le cas contraire. Un cas intéressant est de regarder quand un des deux camps peut envahir le territoire adverse lorsque celui-ci est trop grand : il faut pouvoir former deux yeux inconditionnellement, donc posséder un certain espace vital.

Un espace vital étant représenté par un groupe d'intersections, on construit le graphe dont les sommets sont les espaces vitaux à l'intérieur d'un territoire, deux sommets étant reliés par une arête si les intersections correspondantes se chevauchent.

En défense on cherche alors à placer le moins de pierres possible sur les espaces définis par les sommets, tout en limitant au maximum la taille d'une partie connexe résultante. On retrouve bien un problème de dislocation, où Δ doit être choisi le plus grand possible parmi les valeurs qui vérifient la contrainte suivante : avec moins de Δ espaces vitaux différents se chevauchant dans une zone, une invasion y est impossible.

5 Algorithmes

Nous avons d'abord implémenté une classe *Grappe* qui gère toutes les manipulations dont on a besoin sur les graphes à l'aide de la matrice d'adjacence : principalement la mise à jour des classes de connexité du graphe

(obtenue par un parcours récursif des lignes et colonnes de la matrice), la suppression et la restitution de sommets.

5.1 Stratégies utilisées, algorithmes probabilistes

Nous avons commencé par écrire des algorithmes gloutons pour lesquels il a fallu choisir des critères de sélection des sommets ; voici ceux qui ont été retenus (on prend à chaque fois un sommet dans la plus grande classe de connexité) :

- Choisir un sommet de plus haut degré
- Choisir un sommet tel qu'après sa suppression la partie connexe minimale soit de cardinal maximal
- Choisir un sommet tel qu'après sa suppression la partie connexe maximale soit de cardinal minimal

L'efficacité de ces algorithmes est variable, mais en général assez mauvaise lorsque n croît.

Des choix étant souvent possibles malgré les différents critères utilisés, on obtient de meilleurs résultats en relançant simplement les algorithmes plusieurs fois et en gardant le meilleur résultat ; le nombre d'exécutions d'un algorithme est déterminé en fonction du nombre de sommets du graphe par une fonction rampe, constante à gauche et à droite d'un certain segment.

On obtient des résultats souvent meilleurs, mais toujours loin de l'optimal dans certains cas.

Voici un extrait de code correspondant à la première stratégie :

```
max=-1 ;
choix d'un élément dont la classe de connexité est la plus grande possible
(de cardinal > delta)
for (int i=g.n-1 ; i>=0 ; i-) dejaTraite[i]=false ;
for (int i=g.n-1 ; i>=0 ; i-)
  if (!dejaTraite[i] && g.indices_connex[i]) {
    compt=0 ;
    for (int j=g.n-1 ; j>=0 ; j-)
      if (g.cl_connex[i][j]) {
        compt++ ;
        dejaTraite[j]=true ;
      }
    if (compt>max) {
```

```

    max=compt ;
    temp.clear() ;
    for (int j=g.n-1 ; j>=0 ; j-)
        if (g.cl_connex[i][j]) temp.push_back(j) ;
    }
    else if (compt>=max) {
        for (int j=g.n-1 ; j>=0 ; j-)
            if (g.cl_connex[i][j]) temp.push_back(j) ;
        }
    }
}
// recherche des sommets de degré maximum
max=-1 ;
for (int i=(int)temp.size()-1 ; i>=0 ; i-) {
    compt=0 ;
    for (int j=g.n-1 ; j>=0 ; j-)
        if (g.tab[temp[i]][j]) compt++ ;
    if (compt>max) {
        max=compt ;
        tabMax.clear() ;
        tabMax.push_back(temp[i]) ;
    }
    else if (compt>=max) tabMax.push_back(temp[i]) ;
}

```

5.1.1 Une heuristique plus efficace

Avec les deux derniers critères mentionnés on est en mesure de donner heuristiquement une note à chaque sommet ; on a choisit de lui donner la note correspondant au cardinal de la partie connexe maximale résultant de la suppression du sommet si cette suppression change le nombre de parties connexes, 0 sinon ; ce critère semblait être le plus intéressant en se basant sur le cas du cycle (difficile à disloquer). On généralise cette idée à la notation d'un sous-ensemble de sommets, potentiellement solution : sa note est égale au cardinal maximum d'une partie connexe résultante (idéalement, celui-ci est plus petit que Δ), auquel on ajoute des termes tenant compte de la dispersion des cardinaux des parties connexes autour de la moyenne (après suppression du sommet), de la taille de la partie connexe maximale résultante, et du nombre de partie connexes résultantes.

Code correspondant à la notation d'un essai de solution :

```

somme=0 ;

```

```

max=-1 ;
nbClasses=0 ;
somme des cardinaux des parties connexes, calcul du cardinal maximal
d'une partie connexe
for (int u=g.n-1 ; u>=0 ; u-)
  if (g.indices_connex[u]) {
    nbClasses++ ;
    compteur[u]=0 ;
    for (int v=g.n-1 ; v>=0 ; v-)
      if (g.cl_connex[u][v]) {
        compteur[u]++ ;
        somme++ ;
      }
    if (compteur[u]>max) max=compteur[u] ;
  }
vérification que le nombre de classes de connexite augmente, puis notation
de la solution en fonction du resultat
if (nbClasses>old_nbC) {
  compt=g.n ;
  for (int j=g.n-1 ; j>=0 ; j-)
    if (g.indices_connex[j]) {
      nbClasses=0 ;
      for (int k=g.n-1 ; k>=0 ; k-)
        if (g.cl_connex[j][k]) nbClasses++ ;
      if (nbClasses<compt) compt=nbClasses ;
    }
  eval=-(float)compt/((float)(3*g.n)) ;
}
else eval=0 ;
calcul des ecarts a la moyenne
moy=(float)somme/nbClasses ;
dispersion=0 ;
for (int u=g.n-1 ; u>=0 ; u-)
  if (g.indices_connex[u])
    dispersion+=(compteur[u]-moy)*(compteur[u]-moy) ;
// on en deduit l'ecart type empirique, puis l'evaluation : < 1 car max > 0
// pour p < g.n, et ecarttype ≤ maximum(moy, max - moy)
if (moy>=max-moy)
  eval+=max+(float)(somme-max)/somme+sqrt(dispersion)/(3*moy) ;
else eval+=max+(float)g.m/((float)(3*g.n*g.n)-(float)(somme-
max)/((float)(3*somme)+sqrt(dispersion)/(3*(0.1+max-moy))) ;

```

L'algorithme de recuit simulé avec recherche tabou implémenté fonctionne alors de la façon suivante :

1. Initialisation : on choisit un essai avec des sommets de plus haut degrés
2. Si cet essai convient, stop
3. Tant qu'il reste du temps, faire :
 - (a) Tant que l'on n'a pas énuméré toutes les permutations de $[1, \dots, n]$ faire :
 - i. Choisir une transposition de $[1, \dots, n]$ qui permute un sommet dans l'essai en cours et un qui n'y est pas
 - ii. Si cet essai fait partie des trois derniers tentés, revenir en *i*)
 - iii. Noter cet essai avec la note N ; si N convient, stop
 - iv. l'ajouter à la liste des voisins de l'essai en cours
 - (b) Sélectionner un voisin de l'essai courant avec une probabilité liée à sa note et à un bonus si le sommet ajouté a déjà été souvent présent dans un essai

5.1.2 L'algorithme exact

On se contente de tester toutes les possibilités, en testant en premier les sommets de plus haut degré. On regarde d'abord si G n'est pas déjà disloqué, puis on essaye toutes les possibilités en enlevant 1 sommet, puis 2 sommets..etc jusqu'à obtenir une solution, ,nécessairement optimale.

5.2 Expérimentations et limitations

Nous avons tester les algorithmes présentés sur plusieurs familles de graphes différentes, construites automatiquement. Les performances varient suivant le type de graphe :

Sur des graphes aléatoires, les trois types d'algorithmes renvoient des résultats similaires, l'algorithme suivant la troisième heuristique mentionnée (choisir un sommet tel qu'après sa suppression la partie connexe maximale soit de cardinal minimal) étant un peu moins bon.

Lorsqu'on passe à des graphes complets connectés entre eux comme décrit en 3.4, on constate la même chose.

En revanche sur les cycles, l'algorithme suivant la troisième stratégie est asymptotiquement meilleur, comme on le verra bientôt ; ce n'est plus vrai

pour les puissances de cycle en général. De plus le recuit simulé est plus efficace que les autres algorithmes pour des Δ élevés.

Sur des toiles d'araignée, les performances sont à nouveau comparables.

L'algorithme exact fonctionne bien pour des graphes à moins de 14, 15 sommets, et sur certains graphes sans trop d'arêtes ; au-delà il faudrait attendre longtemps, la complexité au pire étant en $n!$.

Pour l'algorithme choisissant un sommet de degré maximum sans regarder ce qu'on obtient après suppression du sommet, on monte jusqu'à 3000 sommets sur un graphes aléatoire en moins de 2 minutes. Les autres algorithmes sont plus lents (précisément car ils basent leur recherche sur la structure obtenue après suppression éventuelle d'un sommet).

Le recuit simulé est le moins rapide, il faut attendre une bonne dizaine de minutes pour le voir résoudre un problème de dislocation à 60 sommets ; en revanche, on garantit une complexité polynômiale pour des résultats satisfaisants.

Quant à lui, l'algorithme exact a permis des conjectures démontrées ensuite ; il a ensuite passé le relais aux heuristiques, car celles-ci donnent de très bons résultats tant que le nombre de sommets n'excède pas 30, 40.

6 Complexité

6.1 APX et NP-complétude

On sait que le problème DISLOCATION est *NP*-complet d'après l'article [GSM04].

On peut en fait donner un algorithme d'approximation qui est toujours à un facteur multiplicatif $\Delta + 1$ de la solution optimale : il suffit de chercher une partie connexe à $\Delta + 1$ éléments ; celle-ci contient forcément un sommet présent dans l'ensemble de dislocation, donc pour être sûr de ne pas se tromper on les retire tous. On obtient ainsi un sous-graphe du graphe obtenu en choisissant un sommet optimal, ce qui garantit le facteur $\Delta + 1$.

Plus formellement :

1. Initialisation : $\rho = 0$
2. Tant que G possède une partie connexe P de cardinal $\Delta + 1$, faire :
 - (a) Retirer tous les sommets de P à G

(b) $\rho \leftarrow \rho + \Delta + 1$

3. Retourner ρ

6.2 Un algorithme d'approximation efficace ?

On aimerait bien que les stratégies utilisées soient à un facteur multiplicatif constant de l'optimal, ou encore mieux à un facteur additif (éventuellement non constant, mais en $o(\rho)$) de cet optimal. Hélas, on va montrer que toutes les idées utilisées peuvent être arbitrairement mauvaises :

1) Choisir un sommet de plus haut degré dans une classe de connexité maximale :

Soit $n \in \mathbb{N}^*$. Soient $n - 1$ graphes complets d'ordre n : K^1, \dots, K^{n-1} . Dans chacun de ces graphes, on choisit un sommet quelconque : v^i dans K^i . On crée ensuite un sommet v n'appartenant à aucun graphe déjà construit, que l'on relie à chaque sommet v^i .

v est ainsi de degré $n-1$, et tous les v_i sont de degré n . Pour une n -dislocation, l'algorithme va donc choisir un des v_i , puis sa suppression ne modifiant pas la structure locale au voisinage des autres v_i et laissant une unique classe de connexité, il va continuer jusqu'à enlever $n - 1$ sommets v_i . Or il suffisait bien sûr d'enlever v . On est donc à un facteur multiplicatif $n - 1$ de l'optimal.

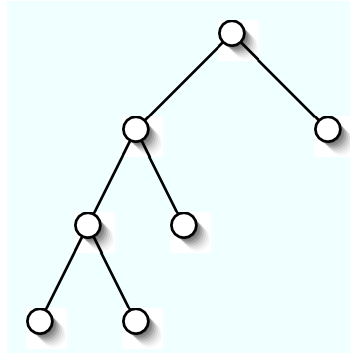


FIG. 15 – Exemple avec $n = 4$

2) On raffine l'algorithme précédent en choisissant en plus un sommet qui maximise une partie connexe minimale en essayant de créer le plus de parties connexes possible. Cela résout donc le cas précédent.

En revanche, sur des puissances de cycle où tous les sommets sont de degré au moins 4 l'algorithme commence par prendre un sommet sur k (si $p = 2k$,

notations déjà rencontrées) dans le pire cas, puis il recommence tant que les degrés des nœuds sont ≥ 4 . D'où le nombre approché de sommets suivant dans le pire cas :

$$\left\lfloor \frac{n}{k} \right\rfloor + \left\lfloor \frac{n - \left\lfloor \frac{n}{k} \right\rfloor}{k-1} \right\rfloor + \dots,$$

asymptotiquement aussi mauvais que l'on veut.

3) Le même contre-exemple fonctionne. Par contre, cet algorithme est asymptotiquement efficace sur les cycles et chaînes, comme on le montre ici : Si l'on cherche à maximiser la plus petite partie connexe, on va découper la chaîne (ou le cycle, c'est pareil) en 2, puis en 4, puis en 8 ..etc. On remarque alors que pour couper la chaîne en 2 il faut 1 sommet, puis pour la couper en 4, 3 sommets, puis pour la couper en 8, 7 sommets ..etc.

On encadre alors Δ comme suit :

$$\frac{n - 2^p - 1}{2^p} \leq \Delta < \frac{n - 2^{p-1} - 1}{2^{p-1}},$$

pour en déduire p de l'ordre de $\log_2\left(\frac{n}{\Delta}\right)$, et donc le nombre de sommets à retirer est asymptotiquement du même ordre de grandeur que $\frac{n}{\Delta}$, qui est précisément la solution optimale asymptotique.

6.3 Conclusion

Ce qui précède nous a donné l'idée de chercher des classes de graphes sur lesquelles nos algorithmes seraient optimaux, ou au moins efficaces. On trouve parmi eux les étoiles itérées présentées en section 3.1.1 où l'algorithme prenant un sommet de plus haut degré est optimal.

Plus généralement, sur des graphes assez "étoilés", nos algorithmes sont très proches de l'optimal. Ceci correspond à des graphes faciles à disloquer, donc les résultats sont cohérents.

On peut aussi rechercher des classes de graphes sur lesquelles un algorithme polynomial existe; des idées d'arbres dans lesquels les nœuds sont remplacés par des cycles impairs bornés en utilisant un algorithme proche de celui sur les arbres, ou encore une suite de polygones bornés (indépendamment de n) connexe, telle que deux polygones consécutifs n'aient pas d'arêtes en commun.

Ceci pourrait constituer une prolongation du travail effectué.

Bibliographie :

[GOL94] **Solomon W. Golomb :**

Polyominoes-Puzzles, Patterns, Problems, and Packings.

Princeton University Press, Princeton, (1994)

[BOS98] **Robert A. Bosch :**

A Pentomino Exclusion Problem

Mathematical Programming Newsletter, Optima 60, 3, (1998)

[GP98] **Sylvain Gravier , Charles Payan :**

On The Pentomino Exclusion Problem

Discrete Comput. Geom.26, No.3, 375-386 (2001)

[GMP04] **Sylvain Gravier, Julien Moncel, Charles Payan :**

A Generalization of the Pentomino Exclusion Problem : Dislocation of Graphs

[ZER04] **Janez Žerovnik :**

Deriving Formulas for the Pentomino exclusion problem