



HAL
open science

Quelques propriétés topologiques des graphes et applications à Internet et aux réseaux

Mauricio Abel Gomez Soto

► **To cite this version:**

Mauricio Abel Gomez Soto. Quelques propriétés topologiques des graphes et applications à Internet et aux réseaux. Réseaux et télécommunications [cs.NI]. Paris Diderot University, 2011. Français. NNT : . tel-01259904

HAL Id: tel-01259904

<https://theses.hal.science/tel-01259904>

Submitted on 21 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Pour l'obtention du grade de

DOCTEUR DE L'UNIVERSITÉ

PARIS DIDEROT - PARIS VII

ÉCOLE DOCTORALE DE SCIENCES MATHÉMATIQUES DE PARIS CENTRE

SPECIALITÉ : INFORMATIQUE

Présentée par

Mauricio Abel Soto Gómez

Quelques propriétés topologiques des graphes et applications à Internet et aux réseaux

Directeur de thèse : **M. Laurent Viennot**

Co-direction : **M. Fabien de Montgolfier**

Soutenue le 02 décembre 2011

JURY

M. Nicolas Hanusse	Directeur de recherche, CNRS-LABRI	Rapporteur
M. Sébastien Tixeuil	Professeur, UPMC	Rapporteur
M. Laurent Viennot	Directeur de recherche, INRIA-LIAFA	Directeur
M. Jean-Loup Guillaume	Maître de Conférence, UPMC	Examineur
M. Michel Habib	Professeur, Université Paris Diderot	Examineur
M. Fabien de Montgolfier	Maître de conférence, Université Paris Diderot	co-Directeur

A mi familia.

Remerciement

Avant de terminer ma thèse, je voudrais remercier les nombreuses personnes qui ont pris part à ce projet et sans lesquelles il n'aurait pas été possible.

En premier lieu, mes plus sincères remerciements à Laurent et Fabien pour m'avoir permis de travailler avec eux pendant ces quatre années. Ils ont su m'orienter grâce à leurs questions pertinentes et leurs réponses justes. Leurs qualités professionnelles et personnelles m'ont aidé dans un cadre plus large que ce projet. Pour leur confiance, leur disponibilité et leur patience, merci infiniment.

Je remercie Nicolas Hanusse et Sébastien Tixeuil, qui ont accepté d'être les rapporteurs de cette thèse. Je suis très honoré du temps et de l'intérêt qu'ils ont porté à mon travail. Vos différents points de vue m'offrent des pistes d'explorations futures et vos commentaires représentent un complément enrichissant. Merci également à Michel Habib et Jean-Loup Guillaume d'avoir accepté de faire partie du jury.

Merci aux membres du LIAFA et plus généralement de l'UFR Informatique de Paris 7 de m'avoir accueilli dans son ambiance de travail agréable et encourageante.

Je tiens à remercier le personnel du laboratoire pour avoir aménagé les conditions idéales pour développer mon travail de recherche.

Je remercie particulièrement Noëlle Delgado pour son aide avec les petits et grands problèmes administratifs.

Je souhaite aussi remercier mes collègues stagiaires et doctorants avec qui j'ai partagé de très bons moments : Anh, Thach, Denis, Cezara, Xavier et Hervé.

Je remercie à CONICYT et l'ambassade de France au Chili qui m'ont honoré avec une bourse d'études doctorales que m'a permis le développement de ce travail. Merci ensuite à Marco Kiwi, qui m'a fait découvrir le monde de la recherche et qui a été la flamme initiale de ce projet. Merci également à Adi Rosen pour avoir encadré mon mémoire de Master, et à Miklos Santha pour son soutien dans les moments difficiles à mon arrivée à Paris.

Merci à mes encadrants non-informaticien : mes amis, pour sa guide, leurs conseils et leur soutien : Feña, Papelucho, Claudia, Lucho, Mario, Carolina, Emilita, Caroline (merci pour tes corrections), Jenny, Edu, Cathy, Betsa, Cony, Pierre, Sophie, Charlotte et Rodrigo. Je veux aussi remercier mes amis de l'autre côté de l'Atlantique et des Andes : Miguel, Mota et Gino.

J'adresse mes vifs remerciements à Hanz et Marie Schütte, qui m'ont aidé de manière désintéressée et pour qui j'ai une très haute estime.

Agradecimientos especiales merece Annina, quien con amor y preocupación me ha acompañado en todo momento y que me ha enseñado más de lo podría haber imaginado.

Finalmente agradezco a mi familia, quienes han sido la pieza fundamental en este proceso y cuyo orgullo es motivo suficiente de cualquier esfuerzo. A mis madre, Julia, por su amor incondicional y por vencer sus miedos para estar conmigo ; a mi padre Abel, por ser fuente de fortaleza inagotable y a mi hermana Fabiola por su dulce atención y apoyo.

Merci a tous.

Paris, décembre 2011.

Table des matières

I	Introduction	1
	Introduction	1
II	Graphes et Hyperbolicité	7
1	Graphes et Espaces Métriques	8
2	Définition de la δ -hyperbolicité et propriétés basiques	9
3	δ -hyperbolicité et triangles δ -fins	12
4	Quelques exemples	16
5	Hyperbolicité et propriétés de graphes	17
5.1	Hyperbolicité et arbres d'approximation	17
5.2	Hyperbolicité du graphe métrique	17
5.3	δ -hyperbolicité et cordalité	18
6	Décomposition des graphes et hyperbolicité	22
6.1	Décomposition modulaire	22
6.2	Split-décomposition	26
6.3	δ -hyperbolicité et séparateurs	29
7	Calcul pratique de l'hyperbolicité	31
7.1	Calcul approché	31
7.2	Heuristiques pour le calcul exact	32

Table des matières

III	Mesure de l'arborescence d'Internet	33
1	Travaux similaires	34
2	Le graphe d'Internet	35
2.1	Les sources de données	37
2.2	Évolution des graphes des AS	40
3	Largeur Arborescente (<i>treewidth</i>)	41
3.1	Définition	41
3.2	Largeur arborescente du graphe des AS	43
3.3	Largeur Arborescente du graphe de routeurs	43
3.4	Une conséquence : le cœur du réseau	44
4	Hyperbolicité	46
4.1	Hyperbolicité dans le graphe d'Internet	46
5	Arborescence des modèles d'Internet	47
6	Conclusions	48
IV	Graphes et modularité	53
1	Définition et propriétés basiques	54
1.1	Définition	54
1.2	Premières propriétés	56
2	Quelques graphes de modularité zéro	57
2.1	Étoiles	57
2.2	Cliques	58
3	Graphes de modularité asymptotique un	58
3.1	Graphes décomposables	58
3.2	Grilles	59
3.3	Tores multidimensionnels	60
3.4	Hypercubes	60
4	Graphes de degré moyen borné	61
4.1	Arbres de degré maximal borné	61
4.2	Graphes de degré moyen constant	63
4.3	Graphes et loi de puissance	64

5	Graphe des boules unitaires dans espaces de croissance bornée	64
5.1	Modularité dans les espaces métriques	65
5.2	Espaces de croissance bornée	65
5.3	R -nets	65
5.4	Modularité d'un R -net	66
6	Simulations en espace euclidien	67
V Réseaux et files d'attente avec modèle d'adversaire		73
1	Description du modèle	74
2	Travaux Similaires	78
3	Résultats de stabilité	79
3.1	Stabilité des graphes orientés acycliques	80
3.2	Stabilité des cycles orientés	82
3.3	Caractérisation des réseaux stables	85
4	Résultat d'instabilité	87
5	Modèle fluide	93
Perspectives		95
Annexe		97
1	Modèle fluide pour le files d'attente contre un adversaire	97
1.1	Définitions et notations	97
1.2	Equations du modèle discret	98
1.3	Équations du modèle fluide	99
1.4	Du modèle discret au modèle fluide	100
1.5	Résultat principal	101
Table des figures		103
Références bibliographiques		105

Table des matières

Chapitre I

Introduction

Internet, depuis sa création, soulève de nombreuses problématiques. Son étude, en tant que phénomène social et technologique, embrasse plusieurs domaines de recherche. La démocratisation des moyens de communication se traduit par une évolution explosive d'Internet. Cette expansion s'accompagne d'un grand éventail de difficultés de natures hétérogènes. Les travaux présentés dans cette thèse se consacrent à l'analyse de différentes questions théoriques dans les réseaux de communication et leur applicabilité dans le cas d'Internet.

L'objet d'étude des travaux présentés ici tourne principalement autour des *graphes*. La théorie des graphes offre un cadre riche pour l'analyse des propriétés des réseaux de communication. La quantification des propriétés structurelles d'un graphe peut être déterminée à partir de différents paramètres, comme par exemple la largeur arborescente.

Internet peut être vu comme un empilement de réseaux dans une composition hiérarchique. Cette organisation impose l'analyse du réseau à différents niveaux. Par exemple, on peut considérer Internet comme un ensemble de routeurs reliés entre eux. Dans ce cas, le graphe associé contient un sommet pour chaque routeur et une arête existe entre deux sommets (routeurs) s'ils sont reliés physiquement. D'autre part, les routeurs sont organisés en Systèmes Autonomes ou AS. Un Système Autonome est un ensemble de routeurs sous la même administration qui partagent une politique de routage commune. À la date de publication de cette thèse, il existe environ quarante mille Systèmes Autonomes. Le réseau de connexions entre les AS correspond à la structure de plus haut niveau d'Internet. Dans le graphe associé, chaque sommet représente un Système Autonome, et deux sommets sont reliés s'il existe un lien entre deux routeurs des deux Systèmes Autonomes correspondants.

Dans cette thèse, trois problématiques relativement indépendantes sont abordées.

La première concerne la structure d'Internet. Nous nous intéressons particulièrement à deux paramètres de graphe : *la largeur arborescente (treewidth) et l'hyperbolicité*. Ces deux paramètres mesurent la proximité d'un graphe avec un arbre sur deux points de vue différents et nous intéressent pour différentes raisons. D'un côté, il existe

des travaux suggérant que l'hyperbolicité d'Internet est faible. Par exemple, Shavitt et Tankel [ST04, ST08] ont montré que les graphes d'Internet peuvent être plongés avec une meilleure précision dans un espace hyperbolique plutôt que dans un espace euclidien de même dimension. D'autre part, la structure hiérarchique d'Internet se rapproche dans son design d'une structure arborescente. C'est ce que suggère le travail de Ramasubramanian *et al.* [RMK⁺09] concernant les distances dans Internet. Outre l'hyperbolicité, nous nous intéressons donc à la treewidth qui est une autre mesure de proximité avec la structure d'arbre. Nous en arrivons donc à la question suivante : quelles sont la treewidth et l'hyperbolicité d'Internet ?

La largeur arborescente, introduite par Robertson et Seymour dans [RS86], mesure le niveau de connectivité globale d'un graphe. Dans les arbres, la suppression d'un sommet déconnecte l'arbre en deux composantes connexes. A cet égard, on peut dire qu'un graphe est proche d'un arbre si deux sommets quelconques du graphe peuvent être déconnectés en enlevant une petite quantité de sommets du graphe. Ceci donne une idée intuitive de ce que mesure la treewidth. L'hyperbolicité, d'autre part, détermine la proximité de la métrique du graphe à celle d'un arbre. La notion d'hyperbolicité a été introduite par Gromov [GRO87] dans le cadre des groupes géométriques mais elle s'applique aussi en théorie des graphes. Un graphe a une faible hyperbolicité s'il est possible de construire un arbre et une correspondance entre les sommets du graphe et ceux de l'arbre de sorte que les distances dans le graphe sont bien approchées par les distances dans l'arbre.

La question de la treewidth et de l'hyperbolicité d'Internet est d'autant plus intéressante qu'elle a des implications algorithmiques. En effet, une faible treewidth garantit l'existence d'algorithmes polynomiaux (ou même linéaires) pour de nombreux problèmes NP-difficiles (voir par exemple [KT05, AP89A, BLW87, GK03]). D'autre part, la treewidth est une mesure de la connectivité globale du réseau [BAI97]. À l'inverse, une treewidth élevée est donc le signe d'une bonne connectivité du réseau. D'un autre côté, une faible hyperbolicité permet de mettre en place des algorithmes de routage compact [CDE⁺] ou des schémas d'étiquetage pour approximer les distances [GL05A].

La deuxième problématique abordée dans cette thèse concerne la recherche des communautés dans un réseau et correspond au problème de *partitionnement* ou de *clustering*. Une mesure classique de la qualité d'un partitionnement est donnée par la *modularité* [GN02, NG04]. La modularité a été principalement utilisée sur des cas pratiques. Ainsi, pour mieux appréhender cette notion nous nous demandons ce que vaut modularité de familles classiques de graphes (comme les grilles ou les hypercubes). Nous nous sommes particulièrement intéressés à l'évolution de la modularité lorsque la taille du graphe croît. L'intérêt de cette question est notamment de caractériser les bons clusterings du point de vue de la modularité. Le partitionnement est un problème commun aux graphes et aux espaces métriques et différentes approches ont été développées des deux côtés. Une question vient alors naturellement : peut-on faire un lien entre le clustering de graphe et le clustering de métrique ?

Finalement, la dernière problématique abordée ici concerne les flux de paquets dans un réseau de communication. Il existe une approche combinatoire des réseaux de

files d'attente qui revient à considérer des politiques d'ordonnement au niveau des routeurs et à analyser leur comportement face à un adversaire qui injecte des paquets dans le réseau et qui décide des routes qu'ils doivent suivre [BKR⁺96, BKR⁺01]. Des résultats de stabilité peuvent alors être obtenus par des politiques simples d'ordonnement. Une généralisation naturelle consiste à considérer différentes classes de paquets et l'existence de délais de reconfiguration lorsque un routeur veut se mettre à traiter les paquets d'une autre classe que celle en cours. Peut-on généraliser les résultats de stabilité lorsqu'il y a différentes classes de paquets et des délais associés? Quels sont les réseaux où il existe des stratégies pour gérer un tel scénario?

Préliminaires Avant de donner un plan plus détaillé, nous donnons les définitions des paramètres principaux de cette thèse pour permettre une meilleure compréhension des résultats obtenus ici.

Un graphe G est défini par un couple (V, E) où V est l'ensemble de sommets et E est un ensemble de paires de sommets, les arêtes. La distance $d(u, v)$ entre deux sommets u et v est la longueur d'un plus court chemin de u à v .

La **largeur arborescente** ou treewidth [RS86] mesure à quel point la structure d'un graphe est proche de celle d'un arbre. Cette comparaison est faite en considérant des séparateurs du graphe. Sa définition est basée sur la notion de *décomposition arborescente*. Étant donné un graphe $G = (V, E)$ une décomposition arborescente de G est une paire (T, \mathcal{S}) où $T = (I, F)$ est un arbre et $\mathcal{S} = \{V_i \subseteq V \mid i \in I\}$ est une famille de sous-ensembles de l'ensemble des sommets V (on note que la famille \mathcal{S} est indexée par les sommets de T). La paire (T, \mathcal{S}) doit satisfaire trois propriétés :

- **Couverture des sommets** : chaque sommet $u \in V$ appartient à au moins un élément de la famille \mathcal{S} .
- **Couverture des arêtes** : pour chaque arête $e \in E$, il existe au moins un élément de \mathcal{S} contenant les extrémités de l'arête e .
- **Cohérence** : pour chaque sommet $u \in V$ l'ensemble des éléments de \mathcal{S} qui contiennent u induit un sous-arbre connexe de l'arbre T .

Chacun des éléments $V_i \in \mathcal{S}$ induit un séparateur du graphe G . La largeur d'une décomposition arborescente (T, \mathcal{S}) est définie comme le cardinal maximum des éléments dans \mathcal{S} moins un. La largeur arborescente de G est la largeur minimale obtenue sur toutes les décompositions arborescentes de G . Au niveau algorithmique une telle décomposition peut permettre de se concentrer sur la structure locale induite par les ensembles de \mathcal{S} , ce qui est la stratégie principale dans la conception d'algorithmes polynomiaux pour des problèmes NP-difficiles ([AP89B, BLW87, KHK02, TEL97] entre autres) qui se résolvent en temps exponentiel en la largeur arborescente, mais polynomial en le nombre de sommets. Ces algorithmes sont usuellement obtenus via l'utilisation de programmation dynamique [BOD88]. Plus généralement, il a été montré par Courcelle *et al.* [COU01] que tout problème définissable en logique monadique du second ordre (MSOL) est soluble en temps polynomial pour les graphes de treewidth bornée.

L'**hyperbolicité** [GRO87] se concentre sur la métrique des espaces. Elle mesure la déviation additive de la métrique par rapport à celle d'un arbre et elle est définie

à partir d'une condition locale. Plus précisément, soit $G = (V, E)$ un graphe. Étant donné quatre sommets du graphe x, y, u, v , on calcule la somme des distances pour tous les matchings du quadruplet : $d(x, y) + d(u, v)$, $d(x, u) + d(y, v)$, $d(x, v) + d(y, u)$. On calcule la demi-différence entre les deux valeurs les plus grandes parmi ces trois quantités. L'hyperbolicité du graphe G est définie comme le maximum de cette demi-différence parmi tous les quadruplets du graphe. Tout graphe d'hyperbolicité δ peut être plongé dans un arbre de sorte que la différence entre la distance de deux sommets dans le graphe et ces mêmes sommets dans l'arbre est au plus δ [GRO87, GDLHeds.90]. Plusieurs problèmes associés aux distance sont solubles plus efficacement dans la classe des graphes d'hyperbolicité bornée comme par exemple le calcul approché du diamètre et du rayon [CDE⁺08], la couverture du graphe par des boules [CE07A], la construction d'un schéma d'étiquetage de distances [GL05B] ou le routage compact [CDE⁺].

Plan détaillé et contributions Le chapitre II traite de l'hyperbolicité. Nous commençons par donner les différentes définitions de l'hyperbolicité et les relations qui existent entre elles. Un résultat original présenté dans cette partie donne une amélioration de la borne liant les notions de δ -hyperbolicité et celle de triangles δ -fins. D'autre part, on exhibe aussi un exemple sur lequel cette inégalité est presque atteinte. On décrit ensuite les valeurs d'hyperbolicité pour différentes familles de graphes, et la relation entre l'hyperbolicité et d'autres paramètres comme la cordalité. Même si la définition de l'hyperbolicité permet d'obtenir un algorithme polynomial naïf en $O(|V|^4)$, nous examinons certaines propriétés qui permettent de réduire le temps de calcul de ce paramètre. On s'intéresse à des stratégies de type "diviser pour régner", en montrant que l'hyperbolicité est compatible avec différentes techniques de décomposition de graphes comme la décomposition modulaire ou la split-décomposition. On donne aussi une relation entre les valeurs de l'hyperbolicité dans une décomposition par des séparateurs quelconques d'un graphe. Finalement, on présente une série de propriétés qui permettent de réduire le temps de calcul de l'hyperbolicité.

Le chapitre III présente les valeurs des paramètres de largeur arborescente et de l'hyperbolicité pour les graphes d'Internet. On commence le chapitre par un panorama, certainement pas exhaustif, des résultats existants par rapport aux propriétés topologiques d'Internet. À ce stade, une description technique s'impose : on décrit des notions sur la politique de routage utilisée par Internet et de quelle façon elle nous permettent d'inférer la topologie interne du réseau. On détaille ensuite les différents dépôts publics de traces utilisées dans ce travail. S'ensuit la description de la largeur arborescente et les valeurs obtenues pour les graphes d'Internet. Un premier résultat permet d'affirmer que, indépendamment de la source et de la période de temps, la valeur de la largeur arborescente est élevée. De plus, elle est comparable à celle d'une grille de même taille. Ce résultat nous permet d'inférer l'existence d'un cœur bien connecté au sein d'Internet. On poursuit avec les mesures d'hyperbolicité dans les graphes d'Internet. Ces graphes, de par leur petit diamètre, ne peuvent avoir une valeur d'hyperbolicité élevée. Ainsi, on s'intéresse à son évolution en fonction de la distance entre les sommets considérés. Une seconde conclusion s'impose : l'hyperbolicité d'Internet est faible. Ceci a des conséquences importantes sur l'algorithmique d'Inter-

net : plusieurs problèmes ont des solutions plus rapides que dans le cas général. On finit le chapitre par l'étude de différents modèles classiques de génération de graphes pour modéliser la topologie d'Internet. En effet, vérifier que ces modèles produisent des graphes avec des valeurs similaires à celles d'Internet permet de les valider. Un dernier résultat est observé : les modèles qui prennent seulement en compte la distribution des degrés capturent mieux les propriétés arborescentes d'Internet. Une grande partie des résultats du chapitre III est publiée dans [DMSV11C].

Le chapitre IV aborde le domaine de la modularité et du calcul d'un partitionnement dans le cadre théorique des graphes. Il débute avec la définition de la modularité et expose quelques propriétés basiques. Ces propriétés nous permettent d'introduire la notion de modularité asymptotique d'une famille de graphes. On montre que les familles des cliques et des étoiles ont une modularité asymptotique égale à zéro (la plus basse possible). Ensuite, on introduit le concept de graphe décomposable, ce qui nous permet d'obtenir les résultats suivants : (1) la famille des tores d -dimensionnels et celle des hypercubes d -dimensionnels ont une modularité asymptotique égale à un (la plus haute possible), (2) les arbres ont une modularité asymptotique égale à un si le degré maximal est $o(\sqrt[5]{|V|})$ et (3) graphes avec degré moyen d ont une modularité asymptotique supérieure ou égale à $\frac{2}{d}$ si le degré maximal Δ est $o(\sqrt[5]{d^3|V|})$. Enfin, on établit une relation entre le clustering d'espaces métrique et la modularité de graphes dans le cas des espaces métriques de croissance bornée en montrant que la modularité est bornée intérieurement par une fonction qui dépend seulement du paramètre de croissance de l'espace. Ces résultats montrent que la modularité n'indique pas toujours l'existence d'une structure bien partitionnée lorsque la structure du graphe est régulière. Ces résultats ont fait l'objet des publications [DMSV11A, DMSV11B].

Le chapitre V généralise la théorie des files d'attente avec un modèle d'adversaire dans le cadre multi-classe avec des coûts de délai. Le chapitre commence avec une description formelle du modèle introduit. Puis, on donne un état de l'art qui contextualise les contributions de cette thèse. On montre la stabilité de familles classiques de graphes, notamment les graphes orientés acycliques et le cycle orienté. Ces résultats se basent sur l'utilisation de **protocoles sensibles** introduits par Dai et Jennings [DJ04] pour le modèle multi-classe stochastique. La stratégie des protocoles sensibles est d'accumuler les paquets dans la file d'attente avant de changer de classe de paquet servie, ce qui évite un coût excessif en délai. On montre aussi l'importance de ce type de protocoles en présentant un exemple de réseau instable pour une adaptation naïve des protocoles classiques de routage. Une analyse du modèle fluide pour ce modèle est proposé en appendice. Ces résultats, obtenus au Chili dans l'équipe du laboratoire d'ingénierie mathématique, ont été publiés dans [KST09].

Publications

- [KST09] M. Kiwi, M.Soto et C.Thraves. *Adversarial queuing theory with setups*. Theoretical Computer Science. 410 :670–687, Mai 2009.
- [DMSV11B] F. de Montgolfier, M. Soto et L. Viennot. *Clustering de métrique et clustering de graphe*. In 13es Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (ALGOTEL), pages 1–4, Cap Estérel, 2011.
- [DMSV11C] F. de Montgolfier, M. Soto et L. Viennot. *Treewidth and hyperbolicity of the Internet*. In 10th IEEE International Symposium on Network Computing and Applications (IEEE NCA), pages 1–8, Boston, 2011.
- [DMSV11A] F. de Montgolfier, M. Soto et L. Viennot. *Asymptotic modularity of some graph classes*. In 22nd International Symposium on Algorithms and Computation (ISAAC), Yokohama, 2011.

Chapitre II

Graphes et Hyperbolicité

La notion de δ -hyperbolicité a été introduite par Mikhaïl Gromov dans le cadre de la théorie des groupes géométriques. La théorie développée par Gromov capte, à partir d'une unique condition métrique locale, les propriétés essentielles des espaces de courbure négative en donnant un cadre adapté pour l'étude de nombreux problèmes.

Dans le cas particulier des graphes, la condition de δ -hyperbolicité peut-être entendue comme une mesure de proximité entre la métrique du graphe et celle d'un arbre. Le structure d'un arbre est associée à une faible complexité algorithmique puisque les arbres peuvent être décomposés facilement dans des parties relativement indépendantes. Sur les graphes à faible hyperbolicité, ayant une structure arborescente, certains problèmes difficiles dans le cas général se résolvent plus facilement. Sous cet angle, une hyperbolicité bornée peut être aussi interprétée comme un signe de manipulabilité. Par exemple, le calcul du diamètre dans un graphe général $G = (V, E)$ a une complexité $O(|V||E|)$. Dans le cas d'un arbre, il suffit de faire deux parcours en largeur ($O(|E|)$) pour obtenir cette valeur. Dans [CDE⁺08] les auteurs montrent que la même procédure dans un graphe δ -hyperbolique donne le diamètre avec une erreur additive bornée par 2δ . D'autres problèmes étudiés dans la littérature sont le calcul du rayon [CDE⁺08], le recouvrement par une quantité minimum de boules de taille $R + \delta$ [CE07A] et le calcul de graphes de recouvrement (spanner) [CDE⁺]. Deux problèmes importants pour le routage dans les graphes d'hyperbolicité bornée sont les travaux de Gavaille et Ly [GL05B] et Chepoi *et al.* [CDE⁺]. Dans [GL05B] les auteurs construisent un schéma d'étiquetage pour les distances avec étirement additif de $O(\log |V|)$ et avec des étiquettes de taille $O(\log^2 |V|)$. D'autre part [CDE⁺] s'intéresse aux schémas de routage pour les graphes δ -hyperbolique, ils proposent un schéma avec un étirement additif $O(\delta \log |V|)$ et des étiquettes de taille $O(\delta \log^2 |V|)$.

Dans ce chapitre nous nous intéressons à l'hyperbolicité et sa relation avec d'autres paramètres des graphes. Nous commençons par la définition de différentes notions de graphes δ -hyperboliques et les relations entre eux. Notamment, nous montrons une relation serrée entre l'hyperbolicité et les triangles δ -fins (proposition II.20). Nous nous intéressons au comportement de l'hyperbolicité face à la décomposition des graphes, comme la décomposition modulaire et la split-décomposition. Ce qui nous

permettre de borner l'hyperbolicité de certaines familles de graphes (théorème II.44). Nous reprenons dans ce chapitre de nombreux résultats déjà connus pour lesquels nous donnons de preuves simples. Nous finissons le chapitre avec certaines propriétés qui permettent la construction d'heuristiques pour le calcul de l'hyperbolicité.

1 Graphes et Espaces Métriques

Définition II.1 (Espace métrique). Soit X un ensemble non vide. Une *distance* sur X est une fonction définie de $X \times X$ vers \mathbb{R}^+ avec les propriétés suivantes :

1. *Symétrie* : $\forall x, y \in X, d(x, y) = d(y, x)$;
2. *Séparation* : $\forall x, y \in X, d(x, y) = 0 \Leftrightarrow x = y$;
3. *Inégalité triangulaire* : $\forall x, y, z \in X, d(x, z) \leq d(x, y) + d(y, z)$.

La paire (X, d) est appelée un *espace métrique*.

Notation. Lorsque le contexte est clair, on notera par xy la distance entre deux éléments x et y d'un espace métrique.

Définition II.2. Soit (X, d) un espace métrique. La distance entre un élément $x \in X$ et un sous-ensemble $Y \subseteq X$ noté par $d(x, Y)$ est l'infimum de l'ensemble $\{d(x, y), y \in Y\}$. La distance entre deux sous-ensembles Y et Z est définie par $d(X, Y) = \inf_{x \in X} \{d(x, Y)\}$.

Définition II.3 (r -voisinage). Soit S un sous-ensemble d'un espace métrique X et r un nombre réel non-négatif. Un r -voisinage de S est l'ensemble de tous les éléments de X qui sont à une distance inférieure ou égale à r d'un élément de S . Autrement dit, le r -voisinage de S est l'ensemble $\{x \in X, d(x, S) \leq r\}$

Sur certains espaces métriques, la notion de distance est associée à celle de plus courts chemins entre les différents éléments de l'espace.

Définition II.4. Dans un espace métrique (X, d) , une *géodésique* reliant $x, y \in X$ est une fonction continue (courbe) $\rho : [a, b] \subset \mathbb{R} \rightarrow X$ satisfaisant $x = \rho(a), y = \rho(b)$ et pour chaque $s, t \in [a, b], d(\rho(s), \rho(t)) = |s - t|$. Par abus de notation, on peut noter une géodésique par $[x, y]$. Un espace métrique est *géodésique* si pour toute paire d'éléments dans l'espace il existe une géodésique qui les relie.

Dans un graphe connexe G pondéré sur les arêtes, on peut munir l'ensemble de ses sommets $V(G)$ d'une distance naturelle : pour toute paire $u, v \in V(G)$, la distance $d_G(u, v)$ entre eux correspond à la plus petite somme des poids dans un chemin entre u et v . Si toutes les arêtes ont un poids unitaire cette distance correspond à la métrique de plus courts chemins entre les sommets. Il est facile de vérifier que cette définition satisfait les propriétés d'une distance. L'espace métrique $(V(G), d_G)$ n'est pas un espace géodésique. Néanmoins, une définition analogue peut être fournie dans ce cas.

Définition II.5. Étant donné un graphe $G = (V, E)$, un *intervalle* entre deux sommets $u, v \in V(G)$, dénoté par $I(u, v)$, est l'ensemble des sommets qui se trouvent sur un plus court chemin entre u et v .

Définition II.6. Deux espaces métriques X, Y sont *isométriques* si il existe un fonction $f : Y \rightarrow X$ satisfaisant $d_Y(y_1, y_2) = d_X(f(y_1), f(y_2))$ pour tout $y_1, y_2 \in Y$. Si $Y \subseteq X$ et f est la fonction identité nous disons que Y est un *sous-ensemble isométrique* de X .

Finalement, il est possible et utile de plonger de façon isométrique un graphe dans une version continue et géodésique du graphe [BH99, p. 7]. La construction de cette version est obtenue en remplaçant chaque arête du graphe par un intervalle de longueur unitaire et en utilisant la distance des plus courts chemins. Plus formellement,

Définition II.7 (Graphe Métrique). Soit $G = (V, E)$ un graphe. Pour chaque arête du graphe, on choisit une orientation arbitraire. Cette orientation permet de définir les fonctions $\partial_0, \partial_1 : E(G) \rightarrow V(G)$ qui correspondent respectivement aux têtes et queues des arêtes orientées. Le *graphe métrique* associé à G est l'espace géodésique (X_G, \tilde{d}_G) défini par l'espace quotient $E(G) \times [0, 1]$ et par la relation $(e, i) \sim (e', i')$ si $\partial_i(e) = \partial_{i'}(e')$ et $i, i' \in \{0, 1\}$. De cette façon, un élément (e, t) appartenant à l'arête e est à distance t de $\partial_0(e)$ et à distance $(1 - t)$ de $\partial_1(e)$. Alors, la distance entre deux éléments (e, t) et (e', t') correspond à la longueur du chemin le plus court parmi les quatre chemins qui utilisent les extrémités des arêtes e et e' , plus formellement :

$$\tilde{d}_G((e, t), (e', t')) = \begin{cases} |t - t'| & \text{si } e = e' \\ \min_{i, i' \in \{0, 1\}} \{d_G(\partial_i(e), \partial_{i'}(e')) + i + (-1)^i t + i' + (-1)^{i'} t'\} & \text{si } e \neq e'. \end{cases}$$

On verra par la suite qu'il existe une relation étroite entre les valeurs de l'hyperbolicité d'un graphe et son graphe métrique.

2 Définition de la δ -hyperbolicité et propriétés basiques

La définition de la δ -hyperbolicité dérive de la relation des distances entre les quadruplets (sous-ensembles de taille quatre) d'un espace métrique.

Définition II.8 (δ -hyperbolicité). Soit (X, d) un espace métrique et $\delta \geq 0$ un nombre réel, on dit que (X, d) est *δ -hyperbolique* si pour tout quadruplet $u, v, x, y \in X$ la différence entre la distance la plus grande et la distance médiane parmi $uv + xy$, $ux + vy$ et $uy + xv$ est bornée supérieurement par 2δ . L'hyperbolicité de l'espace X est définie par $\delta(X) := \inf\{\bar{\delta} : X \text{ est } \bar{\delta} - \text{hyperbolique}\}$.

Un graphe G est δ -hyperbolique si l'espace métrique $(V(G), d_G)$ est δ -hyperbolique.

Remarque II.9. Dans certains travaux, la définition d'hyperbolicité varie légèrement. La différence entre les deux sommes les plus grandes de la définition antérieure est bornée par δ au lieu de 2δ .

On peut obtenir une définition équivalente de la δ -hyperbolicité à partir du produit de Gromov.

Définition II.10 (Produit de Gromov,[BH99]). Soit (X, d) un espace métrique. Pour un élément donné $u \in X$, le produit de Gromov entre x et y par rapport à u est

$$(x|y)_u = (xu + yu - xy)/2.$$

Proposition II.11 ([BH99]). *L'espace métrique (X, d) est δ -hyperbolique si et seulement si pour tout quadruplet $x, y, u, v \in X$, $(x|y)_u \geq \min\{(x|v)_u, (y|v)_u\} - \delta$.*

Démonstration. Montrons ici que cette condition est équivalente à celle de δ -hyperbolicité. Étant donné un quadruplet x, y, u, v , les inégalités suivantes sont équivalentes :

$$\begin{aligned} (x|y)_u &\geq \min\{(x|v)_u, (y|v)_u\} - \delta \\ xu + yu - xy &\geq \min\{xu + vu - xv, yu + vu - vy\} - 2\delta \\ -xu - yu + xy &\leq -vu + \max\{-xu + xv, -yu + vy\} + 2\delta \\ xy + vu &\leq xu + yu + \max\{-xu + xv, -yu + vy\} + 2\delta \\ xy + vu - \max\{xv + yu, vy + xu\} &\leq 2\delta. \end{aligned}$$

Cette dernière expression est toujours vraie si $xy + vu$ n'est pas la somme la plus grande parmi $xy + vu, xv + yu, vy + xu$. Dans le cas contraire, la condition est équivalente à celle de δ -hyperbolicité. \square

Définition II.12. Étant donné un quadruplet $x, y, u, v \in X$, les valeurs la plus grande, médiane et la plus petite parmi $(uv + xy)/2, (ux + vy)/2, (uy + xv)/2$ seront toujours notées par $\delta_1(x, y, u, v), \delta_2(x, y, u, v)$ et $\delta_3(x, y, u, v)$ respectivement. La différence entre $\delta_1(x, y, u, v)$ et $\delta_2(x, y, u, v)$ est notée $\delta(x, y, u, v)$, et donc $\delta(X) = \sup_{x,y,u,v \in X} \delta(x, y, u, v)$.

Un quadruplet x, y, u, v est *maximal* si le maximum de la fonction δ est atteint sur ce quadruplet : $\delta(x, y, u, v) = \delta(X)$.

Pour chaque quadruplet, les distances entre les éléments peuvent être graphiquement représentées sur le plan. La figure II.1 montre un exemple ayant $xv + yu \leq xu + yv \leq xy + uv$. La différence entre $\delta_1(x, y, u, v)$ et $\delta_2(x, y, u, v)$ est ξ . D'autre part η correspond à la différence entre $\delta_1(x, y, u, v)$ et $\delta_3(x, y, u, v)$. Un espace est δ -hyperbolique si pour tout quadruplet $\xi \leq \delta$.

Une propriété de décomposition importante provient du fait que l'hyperbolicité de tout graphe est égale au maximum de la valeur de l'hyperbolicité parmi ses composantes bi-connexes. Bien que simple, on omet la démonstration puisqu'on traite des généralisations de cette décomposition dans la section 6.

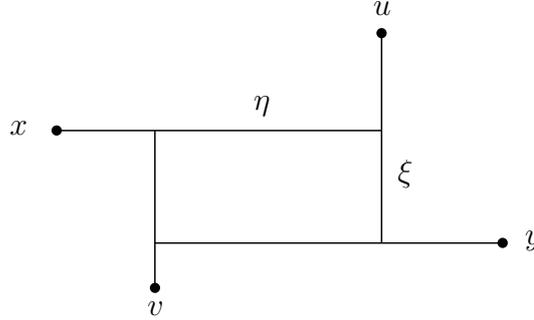


Figure II.1 – Représentation graphique de la condition d'hyperbolicité pour $xv + yu \leq xu + yv \leq xy + uv$.

Les propriétés suivantes montrent le comportement de la fonction δ sur les quadruplets d'un graphe et seront utiles par la suite.

Proposition II.13. *Soit (X, d) un espace métrique :*

1. Si Y est un sous-ensemble isométrique de X alors $\delta(Y) \leq \delta(X)$.
2. $\forall x, y, u, v, w \in X, \delta(x, y, u, v) \leq \delta(x, y, u, w) + vw$.
3. $\forall x, y, u, v \in X, \delta(x, y, u, v) \leq \min_{w, z \in \{x, y, u, v\}} wz$.
4. $\forall x, y, u, v, w \in X, \delta(x, y, u, v) \leq \delta_1(x, y, u, v)/2$.
5. $\delta(X) \leq \text{diam}(X)/2$.

Démonstration. Quitte à renommer les sommets, on suppose que $2\delta_1(x, y, u, w) = xy + uv$ et $2\delta_2(x, y, u, v) = xu + vy$.

1. C'est un résultat direct de la définition de δ -hyperbolicité.
2. $2\delta(x, y, u, v) = 2\delta_1(x, y, u, v) - 2\delta_2(x, y, u, v) = xy + uv - (xu + vy)$. Par inégalité triangulaire $2\delta(x, y, u, v) \leq xy + uv + vw - (xu + vy - vw) \leq 2\delta(x, y, u, w) + 2vw$.
3. Étant donné un quadruplet x, y, u, v et une paire dans ce quadruplet, disons x, y . On a $\delta(x, x, u, v) = 0$, en utilisant (2), $\delta(x, y, u, v) \leq \delta(x, x, u, v) + xy \leq xy$.
4. En utilisant l'inégalité triangulaire $2xy \leq xu + uy + xv + vy$ et $2uv \leq ux + xv + uy + yv$, alors $2\delta_1(x, y, u, v) = xy + uv \leq xu + vy + xv + uy = 2\delta_2(x, y, u, v) + 2\delta_3(x, y, u, v) \leq 4\delta_2(x, y, u, v)$, donc $\delta_1(x, y, u, v) \leq \delta_2(x, y, u, v)/2$. Ainsi, $\delta(x, y, u, v) = \delta_1(x, y, u, v) - \delta_2(x, y, u, v) \leq \delta_1(x, y, u, v) - \delta_1(x, y, u, v)/2 = \delta_1(x, y, u, v)/2$.
5. Soit x, y, u, v un quadruplet maximal ($\delta(x, y, u, v) = \delta(X)$). D'après (4) on a $\delta(X) \leq \delta_1(x, y, u, v)/2 = (xy + uv)/2 \leq \max\{xy, uv\}/2 \leq \text{diam}(G)/2$.

□

Remarque II.14. La proposition II.13(1) peut être paraphrasée en termes de graphes : si H est un sous-graphe isométrique du graphe G , alors $\delta(H) \leq \delta(G)$.

3 δ -hyperbolicité et triangles δ -fins

Dans les espaces métriques géodésiques, il est possible donner une définition équivalente d'hyperbolicité à partir de la notion des triangles δ -fins. Cette définition a été élaborée par Eliyahu Rips selon Gromov [GRO87].

Soit (X, d) un espace métrique géodésique et x, y, z trois éléments du X . Un *triangle isométrique* entre x, y et z est l'union de trois géodésiques $[x, y] \cup [y, z] \cup [z, x]$ que l'on note par $\Delta(x, y, z)$.

Considérons le point m_z sur la géodésique $[x, y]$ placé à une distance $(y|z)_x$ de x , selon la définition du produit de Gromov, m_z est à une distance $(x|z)_y$ de y . De manière analogue on peut définir m_x et m_y . Le *trépied* $T_{\Delta(x,y,z)}$ associé au triangle géodésique $\Delta(x, y, z)$ est l'étoile avec trois segments de longueur $(y|z)_x$, $(x|z)_y$ et $(x|y)_z$ (figure II.3). Il existe une unique fonction isométrique $\phi : \Delta(x, y, z) \rightarrow T_{\Delta(x,y,z)}$ telle que les images de x, y, z sont placées dans les extrémités du trépied et les images des points m_x, m_y et m_z sont égales au point central du trépied. Tout autre point du trépied est l'image de deux points du triangle $\Delta(x, y, z)$.

Définition II.15 (Triangle δ -mince, δ -fin). Soient x, y, z trois éléments dans un espace métrique X et $\Delta(x, y, z)$ un triangle isométrique.

Le triangle $\Delta(x, y, z)$ est *δ -fin* si chaque côté du triangle est inclus dans l'union du δ -voisinage des deux autres côtés du triangle (figure II.2).

Si pour toute paire $a, b \in \Delta(x, y, z)$ telle que $\phi(a) = \phi(b)$ on a $d(a, b) \leq \delta$ on dit alors que le triangle géodésique est *δ -mince* (figure II.3).

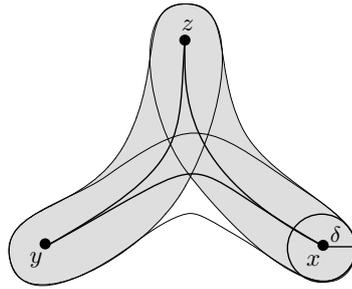


Figure II.2 – Triangle géodésique δ -fin. La géodésique $[x, y]$ est incluse dans le δ -voisinage de $[x, z] \cup [y, z]$

Un espace métrique est δ -fin si tous ses triangles géodésiques sont δ -fins. De manière analogue, un espace métrique est δ -mince si tout triangle géodésique est δ -mince. Évidemment, tout espace δ -mince est aussi δ -fin mais la relation inverse n'est pas toujours vraie.

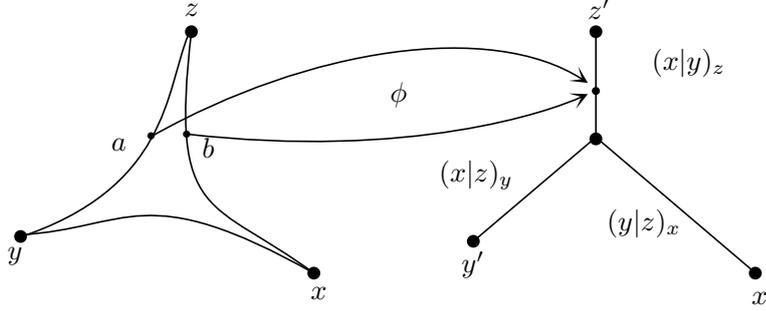


Figure II.3 – Triangle géodésique δ -mince et son trépied associé. La condition de δ -fin garantit que si $\phi(a) = \phi(b)$ alors $d(a, b) \leq \delta$.

La notion de triangle δ -fin peut être adaptée pour le cas des graphes. Pour $x, y, z \in V(G)$, le triangle entre x, y et z est l'union des intervalles $I(x, y), I(y, z)$ et $I(z, x)$. Ce triangle est δ -fin si tout sommet dans un intervalle est dans le δ -voisinage de l'union des deux autres intervalles. Un graphe est δ -fin, si pour tout triplet, le triangle entre les éléments du triplet est δ -fin.

La proposition suivante témoigne de la relation étroite qui existe entre les espaces δ -fins, δ -mince et leur hyperbolicité.

Proposition II.16 ([GDLHeds.90, ABC⁺90, BH99, GRO87]). *Un espace géodésique δ -hyperbolique est 4δ -fin et 4δ -mince. Réciproquement, un espace géodésique δ -mince est 2δ -hyperbolique et un espace géodésique δ -fin est 8δ -hyperbolique.*

Dans le cadre de graphes, Wu et Zhang [WZ09] montrent une meilleure valeur de l'hyperbolicité dans les graphes avec triangles δ -fins.

Proposition II.17 ([WZ09], Lemme 37). *Un graphe avec triangles δ -fins est 3δ -hyperbolique.*

Nous allons prouver de meilleures bornes dans la proposition II.20. Mais auparavant, intéressons-nous au lemme suivant :

Lemme II.18 ([WZ11, Lemme 8]). *Soit x, y, u, v un quadruplet dans un espace métrique géodésique (X, d) tel que $2\delta_1(x, y, u, v) = xy + uv$ et $2\delta_2(x, y, u, v) = xu + yv$. Alors*

$$\delta(x, y, u, v) \leq \min \{d([x, u], [v, y]), d([x, v], [u, y])\}$$

Démonstration. Soit $\varepsilon > 0$ il existe $a \in [x, u], b \in [v, y], c \in [x, v]$ et $d \in [x, y]$ tels que $ab \leq d([x, u], [v, y]) + \varepsilon$ et $cd \leq d([x, v], [u, y]) + \varepsilon$. Par l'inégalité triangulaire,

$$2\delta_1(x, y, u, v) \leq xa + ab + by + ua + ab + bv = xu + yv + 2ab = 2\delta_2(x, y, u, v) + 2ab.$$

De manière analogue

$$2\delta_1(x, y, u, v) \leq xc + cd + dy + du + cd + cv = xv + yu + 2cd = 2\delta_3(x, y, u, v) + 2cd.$$

Ainsi, $\forall \varepsilon > 0, \delta(x, y, u, v) \leq \min \{ab, cd\} \leq \min \{d([x, u], [v, y]), d([x, v], [u, y])\} + \varepsilon$, ce qui conclut la preuve. \square

Remarque II.19. Le lemme II.18 peut être réécrit dans le cadre des graphes : soit G un graphe et $x, y, u, v \in V(G)$ un quadruplet tel que $2\delta_1(x, y, u, v) = xy + uv$ et $2\delta_2(x, y, u, v) = xu + yv$. Alors $\delta(x, y, u, v) \leq \min \{d(I(x, u), I(v, y)), d(I(x, v), I(u, y))\}$.

Proposition II.20.

1. Si (X, d) est un espace métrique géodésique δ -fin alors il est 2δ -hyperbolique.
2. Tout graphe δ -fin est $(2\delta + \frac{1}{2})$ -hyperbolique.

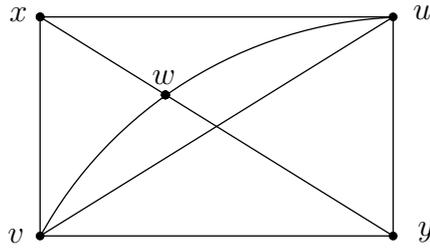


Figure II.4 – Quadruplet de la proposition II.20.

Démonstration. 1. Soit X un espace géodésique δ -fin et x, y, u, v un quadruplet maximal satisfaisant $2\delta_1(x, y, u, v) = xy + uv$ et $2\delta_2(x, y, u, v) = xu + yv$. Soit w le point sur $[x, y]$ à distance $xv - (x|y)_v$ de x (et donc à distance $yv - (x|y)_v$ de y).

Les triangles $\Delta(x, v, y)$ et $\Delta(x, u, y)$ sont δ -fins, donc $\min \{d(w, [x, v]), d(w, [v, y])\} \leq \delta$ et $\min \{d(w, [x, u]), d(w, [u, y])\} \leq \delta$. Ainsi, d'après le lemme II.18 on a $\delta(x, y, u, w) \leq \delta$ et $\delta(x, y, v, w) \leq \delta$.

D'autre part $xy + uw \geq xu + wy = xu + vy - (x|y)_v = \delta_2(x, y, u, v) - (x|y)_v \geq \delta_3(x, y, u, v) - (x|y)_v = xw + uy$, donc

$$2\delta(x, y, u, w) = xy + uw - (xu + vy) + (x|y)_v.$$

De la même manière, $xy + vw \geq xv + wy = xw + vy = xv + vy - (x|y)_v$, alors

$$2\delta(x, y, v, w) = xy + vw - (xv + wy) = vw + xw - xv = vw - (x|y)_v.$$

Donc,

$$\begin{aligned}
 2\delta &\geq \delta(x, y, u, w) + \delta(x, y, v, w) \\
 &= (xy + uw - (xu + vy) + (x|y)_v)/2 + (vw - (x|y)_v)/2 \\
 &\geq (xy + uv - (xu + vy))/2 \\
 &\geq \delta(x, y, u, v)
 \end{aligned}$$

2. La démonstration dans le cadre des graphes se déroule de façon similaire. On considère un quadruplet maximal avec les mêmes hypothèses que dans (1). Le sommet w est choisi sur l'intervalle $I(x, y)$ à distance $xv - \lceil (x|y)_v \rceil$ de x et à distance $yv - \lfloor (x|y)_v \rfloor$ de y . Des arguments identiques au cas des espaces géodésiques permettent de déduire que

$$\begin{aligned}
 2\delta(x, y, u, w) &= xy + uw - (xu + vy) + \lfloor (x|y)_v \rfloor \\
 2\delta(x, y, v, w) &= vw + xw - xv = vw - \lceil (x|y)_v \rceil
 \end{aligned}$$

Donc,

$$\begin{aligned}
 2\delta &\geq \delta(x, y, u, w) + \delta(x, y, v, w) \\
 &\geq (xy + uw + vw - (xu + vy))/2 + (\lfloor (x|y)_v \rfloor - \lceil (x|y)_v \rceil)/2 \\
 &\geq \delta(x, y, u, v) - \frac{1}{2}
 \end{aligned}$$

□

Remarque II.21. Pour tout $n > 0$ on peut construire un graphe n -fin avec une hyperbolicité plus grande ou égale à $2n$. La figure II.5 montre la construction d'un tel graphe : il est facile de voir que tous les triangles sont n -fins mais, en choisissant le quadruplet de points extrêmes, on peut atteindre une hyperbolicité égale à $2n$.

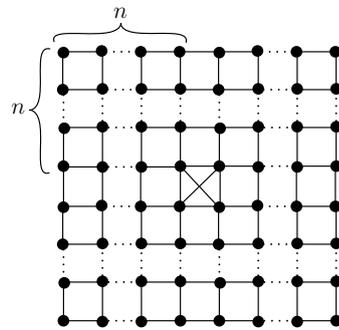


Figure II.5 – Un grille plus un croisillon : un graphe n -fin et $2n$ -hyperbolique.

4 Quelques exemples

Graphes 0-hyperboliques.

A partir de la définition il est facile de vérifier que la clique est 0-hyperbolique. Tout arbre T est aussi 0-hyperbolique ; pour le prouver il faut remarquer que, pour tout quadruplet¹ $x, y, u, v \in V(T)$, soit il existe un sommet w tel que chaque élément du quadruplet est dans une composante connexe différente de $T - w$, soit il existe une arête $e \in E(T)$ tel que chaque composant connexe de $T - e$ contient exactement deux éléments du quadruplet. Dans les deux cas $\delta_1(x, y, u, v) = \delta_2(x, y, u, v)$. Plus généralement, les graphes 0-hyperboliques sont exactement les graphes de blocs [BM86, DMS97, HOW79], c'est-à-dire les graphes où les composants bi-connexes sont des cliques. Pour le prouver, on considère une composante bi-connexe quelconque ; s'il existe une paire de sommets non-voisins, alors il existe deux chemins disjoints entre eux dans lesquels on peut trouver un quadruplet qui rend l'hyperbolicité strictement positive.

Cycles.

Le cycle C_3 , en tant que clique, est 0-hyperbolique. Pour tout $n > 3$, tout quadruplet dans le cycle C_n a deux éléments placés à distance plus petite ou égale à $\lfloor \frac{n}{4} \rfloor$. Ainsi, par la proposition II.13(3), tout cycle C_n est $\lfloor \frac{n}{4} \rfloor$ -hyperbolique. D'autre part, si pour un quadruplet $x, y, u, v \in V(C_n)$ toute paire d'éléments sont à une distance plus grande ou égale à $\lfloor \frac{n}{4} \rfloor$, on a $\delta(x, y, u, v) = \lfloor \frac{n}{4} \rfloor - 1$ quand $n \equiv 1 \pmod{4}$; sinon $\delta(x, y, u, v) = \lfloor \frac{n}{4} \rfloor$. Ainsi [KM02],

$$\delta(C_n) = \begin{cases} \lfloor \frac{n}{4} \rfloor & \text{si } n \equiv 0, 2, 3 \pmod{4} \\ \lfloor \frac{n}{4} \rfloor - \frac{1}{2} & \text{si } n \equiv 1 \pmod{4}. \end{cases}$$

Grilles.

Soit $G_{n \times m}$ la grille avec m lignes et n colonnes. Sans perte de généralité, on suppose que $m < n$. Soit $x, y, u, v \in V(G_{n \times m})$, si deux éléments sont dans la même colonne d'après la proposition II.13.(4) on a $\delta(x, y, u, v) \leq m$; on suppose donc que tous les éléments sont placés dans des colonnes différentes de $G_{n \times m}$. Soit C^* la deuxième colonne, de gauche à droite, qui contient un élément du quadruplet. Alors, tout chemin entre un élément placé de chaque côté de C^* doit le traverser. Ainsi, à partir du lemme II.18 on a $\delta(x, y, u, v) \leq m$. D'autre part, le quadruplet contenant les sommets dans les "coins" de la grille atteint la borne supérieure. On conclut que $\delta(G_{n \times m}) = \min \{n, m\}$.

1. Nous supposons que les éléments du quadruplet sont différents.

Graphes de diamètre borné

Si un graphe G a un diamètre plus petit ou égal à deux, alors à partir de la propriété (5) on conclut que G est 1-hyperbolique. De plus, l'hyperbolicité est exactement égale à un si et seulement si, il existe un quadruplet x, y, u, v tel que $\delta_1(x, y, u, v) = 4$ et $\delta_2(x, y, u, v) = 2$; donc x, y, u, v forment un cycle de taille quatre. Une application directe peut être effectuée pour la famille de cographes qui ont un diamètre borné par deux. Ce cas particulier a été prouvé dans [WZ11][Corollaire 25] par l'analyse de la cordalité.

5 Hyperbolicité et propriétés de graphes

5.1 Hyperbolicité et arbres d'approximation

La condition de 0-hyperbolicité est aussi connue comme la *condition des quatre points*. Cette condition caractérise les *métriques d'arbre* qui correspondent aux espaces métriques qui peuvent être plongés dans un arbre sans distorsion des distances. Autrement dit, l'espace (X, d) a une métrique d'arbre s'il existe une fonction de l'espace métrique vers un arbre (pondéré) T , $\phi : X \rightarrow T$ tel que pour toute paire $x, y \in X$ la distance entre x et y est égal à la distance entre ses images dans l'arbre : $d(x, y) = d_T(\phi(x), \phi(y))$.

Pour le cas des espaces δ -hyperboliques, un résultat similaire de Gromov donne une approximation des distances par une métrique d'arbre :

Théorème II.22 ([GRO87], [GDLHeds.90]). *Étant donné un espace δ -hyperbolique fini (X, d) avec $|X| = n$ et un sommet racine $s \in X$, il existe un arbre pondéré T et une fonction $\phi : X \rightarrow T$ tels que $d(s, x) = d_T(\phi(s), \phi(x))$ pour tout $x \in X$ et $d(x, y) - 2\delta \log_2(n) \leq d_T(\phi(x), \phi(y)) \leq d(x, y)$ pour toute paire $x, y \in X$. La construction de la fonction ϕ et de l'arbre T peut être faite en temps $O(n^2)$.*

Inversement, si pour un espace métrique (X, d) il existe une fonction $\phi : X \rightarrow T$ dans un arbre T , tel que, pour chaque paire $x, y \in X$ les distances varient d'au plus δ , alors X est 2δ -hyperbolique. Pour le vérifier, il suffit de noter que pour tout quadruplet $x, y, u, v \in X$, $\delta_1(x, y, u, v) \leq \delta_1(\phi(x), \phi(y), \phi(u), \phi(v)) + 2\delta$ et $\delta_2(x, y, u, v) \geq \delta_2(\phi(x), \phi(y), \phi(u), \phi(v)) - 2\delta$.

La discussion antérieure nous montre la relation qui existe entre l'hyperbolicité et les métriques arborescentes.

5.2 Hyperbolicité du graphe métrique

La proposition suivante montre la relation entre les valeurs de l'hyperbolicité entre un graphe et son graphe métrique associé (défini dans la section 1).

Proposition II.23 ([NOG09]). *Pour tout graphe G , $\delta(G) \leq \delta(X_G) \leq \delta(G) + 1$.*

Démonstration. La première inégalité découle de la proposition II.13(1). Pour la deuxième inégalité on considère un quadruplet maximal $x, y, u, v \in X_G$. On montrera qu'il est possible de choisir deux paires x', y' et $u', v' \in V(G)$ de sorte que $\delta(x, y, u, v) \leq \delta(x', y', u, v) + \frac{1}{2} \leq \delta(x', y', u', v') + 1$.

Sans perte de généralité, on peut supposer que $\delta_1(x, y, u, v) \geq \delta_1(\tilde{x}, \tilde{y}, \tilde{u}, \tilde{v})$ pour tout autre quadruplet maximal. Quitte à renommer les sommets, on a $\delta_1(x, y, u, v) = xy + uv$ et $\delta_2(x, y, u, v) = xu + vy$. Soient $e_x = (x_1, x_2)$ et $e_y = (y_1, y_2)$ les arêtes contenant respectivement x et y et tels que $xy = xx_1 + x_1y_1 + y_1y$. Du fait que xy est un plus court chemin entre x et y , on a $|x_2y_2 - x_1y_1| \leq 1$ et $xy \leq \frac{x_1y_1 + x_2y_2 + 2}{2}$. Trois cas sont possibles :

1. $[x_2y_2 = x_1y_1 - 1]$. On a $xx_1 + x_1y_1 + y_1y = xy \leq \frac{x_1y_1 + x_2y_2 + 2}{2} \leq x_1y_1 + \frac{1}{2}$. Donc $xx_1 + y_1y \leq \frac{1}{2}$. On définit $x' = x_1$ et $y' = y_1$. D'après la proposition II.13.(2), on a $\delta(x, y, u, v) \leq \delta(x', y', u, v) + 1/2$.
2. $[x_2y_2 = x_1y_1 + 1]$. Comme dans le cas (1) $xy = xx_1 + x_1y_1 + y_1y \leq \frac{x_1y_1 + x_2y_2 + 2}{2} \leq x_1y_1 + \frac{3}{2}$, donc $xx_1 + yy_1 \leq \frac{3}{2}$. Du fait que $\delta_1(x, y, u, v)$ est maximal on a $xx_1 + yy_1 \leq \frac{3}{2}$; autrement on peut déplacer x ou y en augmentant $\delta_1(x, y, u, v)$ ce qui contredit l'élection du quadruplet. D'autre part, $xx_2 + yy_2 = 2 - xx_1 + yy_1 = 1/2$. Ainsi, on peut définir $x' = x_2$ et $y' = y_2$ que, d'après la proposition II.13.(2), satisfait $\delta(x, y, u, v) \leq \delta(x', y', u, v) + 1/2$.
3. $[x_2y_2 = x_1y_1]$. De manière analogue, on a $xy = xx_1 + x_1y_1 + y_1y \leq \frac{x_1y_1 + x_2y_2 + 2}{2} \leq x_1y_1 + 1$. On suppose que $xx_1 + yy_1 = 1$, autrement en déplaçant x ou y dans ses arêtes respectives, on peut augmenter la valeur de $\delta_1(x, y, u, v)$ sans changer la valeur de $\delta(x, y, u, v)$. On choisit $x' = x_1$ et $y' = y_2$. Ainsi $2\delta_1(x', y', u, v) = 2\delta_1(x, y, u, v)$ et $2\delta_2(x, y, u, v) \geq 2\delta_2(x', y', u, v) - 1$, donc $\delta(x, y, u, v) \leq \delta(x', y', u, v) + 1/2$.

De façon symétrique, on peut choisir $u', v' \in V(G)$, de sorte que $\delta(X_G) = \delta(x, y, u, v) \leq \delta(x', y', u', v') + 1 \leq \delta(G) + 1$. □

La figure II.6 montre un exemple où la borne est atteinte : $\delta(K_4) = 0$ et $\delta(X_{K_4}) = 1$.

5.3 δ -hyperbolicité et cordalité

On rappelle que pour $k > 0$, un graphe est k -cordal s'il ne contient pas de cycles induits de taille plus grande que k . De plus, la cordalité d'un graphe G notée par $lc(G)$ est la valeur minimale de k telle que G est k -cordal.

Un graphe ayant une faible hyperbolicité ne peut pas avoir une grande cordalité. Cette relation, conceptuellement cohérente avec le fait que l'hyperbolicité mesure la

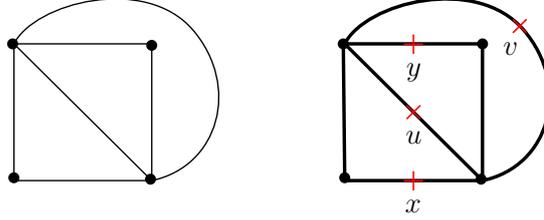


Figure II.6 – À gauche K_4 , à droite son graphe métrique X_{K_4} et un quadruplet maximal.

proximité de la métrique du graphe à celle d'un arbre, a été étudiée dans la littérature. Par exemple, un graphe $\frac{1}{2}$ -hyperbolique ne peut pas contenir le cycle C_4 comme sous-graphe induit [KM02]. Dans ce travail Koolen et Moulton donnent une caractérisation partielle des graphes $\frac{1}{2}$ -hyperboliques. Un graphe est *ponté*² s'il ne contient aucun cycle isométrique de taille strictement plus grande que trois [SC83, FJ87]. Il faut remarquer qu'un graphe ponté peut avoir une hyperbolicité non bornée [KM02][p. 684].

Théorème II.24. [KM02] *Un graphe fini ponté est $\frac{1}{2}$ -hyperbolique si et seulement s'il ne contient pas $H_1, H_2, E_1, E_2, G_1, G_2$ (figure II.7) comme sous-graphe isométrique.*

Une caractérisation complète des graphes $\frac{1}{2}$ -hyperboliques a été obtenue par Bandelt et Chepoi [BC03] en utilisant une condition plus forte que celle de [SC83, FJ87].

Définition II.25. Soit C un sous-graphe d'un graphe G . Si C est un cycle, un *pont* entre deux sommets $x, y \in V(C)$ est un chemin dans G qui relie x et y de longueur plus petite que la distance entre x et y dans le cycle.

Un cycle C est *bien ponté*³ si pour tout sommet $x \in V(C)$, soit il existe un pont entre x et un autre sommet du cycle, soit les deux voisins de x dans C sont voisins.

Il existe deux définitions équivalentes pour les graphes ayant des cycles bien pontés.

Proposition II.26 ([SC83, FJ87]). *Étant donné un graphe G , les propositions suivantes sont équivalentes*

- *Tout cycle de G est bien ponté.*
- *Toutes les boules dans G sont convexes.*
- *G ne contient aucun cycle induit C_n avec $n > 5$ et pour toute paire $x, y \in V(G)$ tous les sommets situés dans un plus court chemin entre x et y et voisins de x forment une clique.*

Théorème II.27. [BC03] *Un graphe est $\frac{1}{2}$ -hyperbolique si et seulement si tout cycle C_n avec $n \neq 5$ dans G est bien ponté et G ne contient pas $H_1, H_2, G_1, G_2, E_1, E_2$ (figure II.7) comme sous-graphe isométrique.*

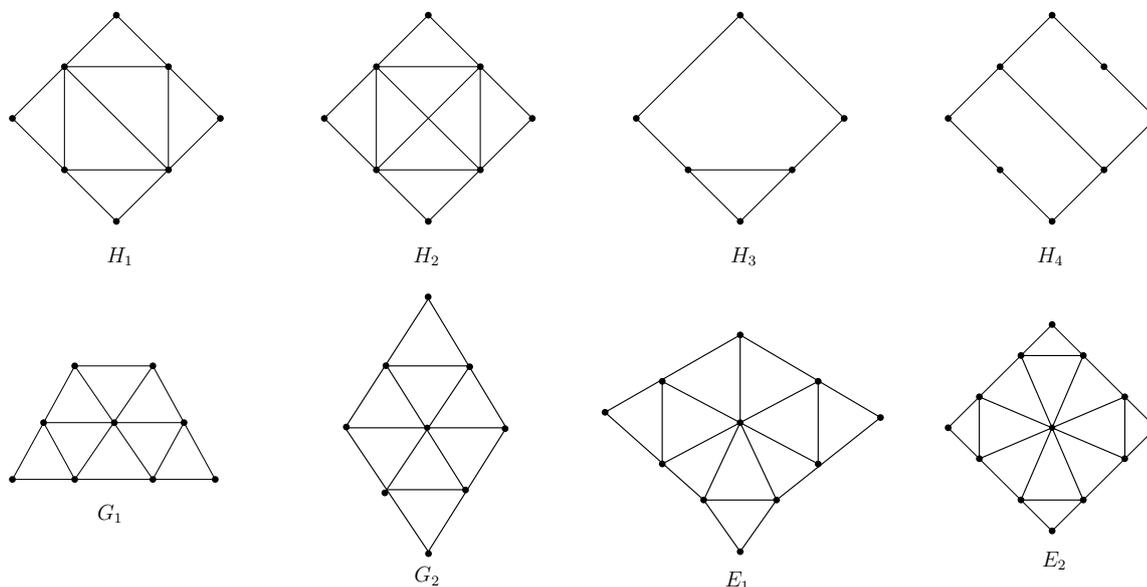


Figure II.7 – Graphes interdits 1-hyperboliques. Tous les graphes ont une hyperbolicité égale à un et $lc(H_1)=lc(H_2)=3$, $lc(H_3)=lc(H_4)=5$, $lc(G_1)=lc(G_4)=6$, $lc(E_1)=7$, $lc(E_2)=8$.

Brinkmann et Koolen montrent dans [BK01] que tout graphe triangulé (3-cordal) est 1-hyperbolique, de plus l’hyperbolicité d’un graphe triangulé est exactement égale à un si et seulement s’il contient comme sous-graphe induit un des graphes H_1, H_2 montré dans la figure II.7.

Le résultat suivant, obtenu indépendamment par Yaokun Wu avec Chengpeng Zhang et Damien Noguès, établit une relation entre l’hyperbolicité et la cordalité d’un graphe :

Théorème II.28. [WZ11, NOG09] *Pour tout graphe G tel que $lc(G) \geq 4$,*

$$\delta(G) \leq \frac{\lfloor \frac{lc(G)}{2} \rfloor}{2}.$$

Afin de mettre en évidence les propriétés qui interviennent dans ce résultat nous allons fournir une démonstration dans ce qui suit. Auparavant, montrons quelques conséquences de cette relation.

Bien qu’il existe des graphes (cycles) pour lesquels la borne du théorème II.28 est atteinte, dans le cas général ces deux paramètres peuvent avoir un rapport non borné. Par exemple, pour une roue R de taille n (un cycle C_n plus un sommet universel), $lc(R)/\delta(R) = 2n$.

2. Traduction de l’anglais, “bridged”

3. traduction de l’anglais, “well-bridged”

Corollaire II.29. [WZ11]

- Un graphe 4-cordal est 1-hyperbolique et son hyperbolicité est égale à un si et seulement s'il contient un des graphes C_4, H_1, H_2 comme sous-graphe isométrique.
- Un graphe 5-cordal est 1-hyperbolique et son hyperbolicité est égale à un si et seulement s'il contient un des graphes C_4, H_1, H_2, H_3, H_4 comme sous-graphe isométrique.

Démonstration. [du théorème II.28] Soit x, y, u, v un quadruplet maximal tel que $\delta_1(x, y, u, v) = xy + uv$, $\delta_2(x, y, u, v) = xu + yv$ et qui minimise $\delta_2(x, y, u, v)$. Si $I(x, u) \cap I(y, v) \neq \emptyset$ ou $I(x, v) \cap I(y, u) \neq \emptyset$ d'après le lemme II.18, $\delta(x, y, u, v) = 0$. D'autre part, s'il existe un sommet $z \in I(x, u) \cap I(x, v)$ alors $\delta(x, y, u, v) = \delta(z, y, u, v)$ et $\delta_2(x, y, u, v) > \delta_2(z, y, u, v)$ ce qui contredit le choix du quadruplet x, y, u, v . Ainsi, on suppose que $I(x, u) \cap I(x, v) = I(y, u) \cap I(y, v) = I(x, u) \cap I(y, v) = I(x, v) \cap I(y, u) = \emptyset$, c'est-à-dire que l'union des plus courts chemins entre $C = I(x, u) \cup I(u, y) \cup I(y, v) \cup I(v, x)$ forme un cycle.

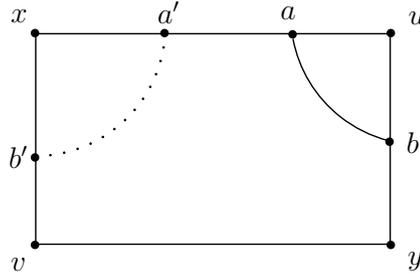


Figure II.8 – Un quadruplet du théorème II.28

Si C ne contient aucune corde, on a $\delta(G) \leq \min \{xu, uy, yv, vx\} \leq \left\lfloor \frac{lc(G)}{4} \right\rfloor$.

Soit ab une corde de C , si $a \in I(x, u)$ et $b \in I(y, v)$, ou $a \in I(x, v)$ et $b \in I(y, u)$ alors d'après le lemme II.18, on a $\delta(G) \leq 1$. Donc, on suppose que $a \in I(x, u)$ et $b \in I(y, u)$ (dans les autres cas, l'analyse est analogue). On considère aussi que la corde $I(a, b)$ maximise la distance $au + bu$.

D'autre part, comme la corde relie les deux plus courts chemins, on a $|au - bu| \leq 1$. Si $au = bu + 1$, on conclut que $\delta(x, y, u, v) = \delta(x, y, b, v)$ et que $\delta_2(x, y, u, v) > \delta_2(x, y, b, v)$ ce qui contredit le choix du quadruplet (le cas $bu = au + 1$ est similaire). On suppose donc que $au = bu$.

De plus, supposons qu'il existe une deuxième corde $a'b'$ dans le cycle $C' = I(x, a) \cup ab \cup I(b, y) \cup I(y, v) \cup I(v, x)$, par exemple $a' \in I(x, u)$ et $b' \in I(x, v)$ avec $a'x = b'x$, alors on a $\delta(b', a, u, v) = \delta(x, y, u, v)$ et $\delta_2(b', a, u, v) < \delta_2(x, y, u, v)$ ce qui contredit aussi le choix du quadruplet x, y, u, v (les autres cas sont similaires). Ainsi, le cycle C' n'a pas de corde et donc $|C'| \leq lc(G)$.

Finalement,

$$\begin{aligned}
 2\delta(G) &= 2\delta(x, y, u, v) \\
 &= xy + uv - (xu + yv) \\
 &\leq \frac{|C'|}{2} + \frac{|C|}{2} - (xu + yv) \\
 &\leq \frac{|C'|}{2} + \frac{2\delta_2(x, y, u, v) + 2\delta_3(x, y, u, v)}{2} - (xu + yv) \\
 &\leq \frac{|C'|}{2}.
 \end{aligned}$$

Ainsi $2\delta \leq \lfloor \frac{lc(G)}{2} \rfloor$ puisque 2δ est un nombre naturel. \square

6 Décomposition des graphes et hyperbolicité

Dans cette section, on analyse la variation de la valeur de l'hyperbolicité pour certaines décompositions classiques de graphes. On verra que lorsqu'un graphe possède une structure modulaire, il est possible de simplifier le graphe sans modifier la valeur de l'hyperbolicité. Du point de vue algorithmique, ces simplifications permettent d'appliquer des stratégies du type "diviser pour régner" afin de concevoir des algorithmes plus efficaces pour le calcul de l'hyperbolicité d'un graphe.

Pour les résultats suivants, nous avons besoin du concept de compactage qui correspond au remplacement d'un sous-ensemble $S \subseteq G$ du graphe avec un nouveau sommet relié à tous les voisins de S dans $G \setminus S$.

Définition II.30. Le *compactage* d'un sous-ensemble $S \subseteq V(G)$ avec $s \notin V(G)$ est le graphe noté par G_S^s , tel que

- $V(G_S^s) = (V(G) \setminus S) \uplus \{s\}$,
- $E(G_S^s) = \{(x, y) : (x, y) \in E(G) \wedge \{x, y\} \in V(G) \setminus S\} \cup \{(s, x) : \exists y \in S, (y, x) \in E(G)\}$.

Si $\mathcal{P} = \{S_1, \dots, S_k\}$ est une famille de sous-ensembles de $V(G)$, le *graphe quotient* $G_{/\mathcal{P}}$ correspond au compactage successif de tous les sous-ensembles dans \mathcal{P} , c'est-à-dire $G_{/\mathcal{P}} = (G_{S_1}^{s_1}) \cdots S_k^{s_k}$.

6.1 Décomposition modulaire

La décomposition modulaire est l'un des outils les plus importants de la théorie des graphes. Introduite par Tibor Gallai en 1967 [GAL67], la décomposition modulaire a trouvé des applications dans différentes problématiques comme la reconnaissance des familles de graphes [LHHM99, PLE71], le dessin des graphes [PV06], la bio-informatique [GKBC04] ou comme outil fondamental de résultats théoriques [LOV72] entre autres.

La question algorithmique de trouver une décomposition modulaire a été largement étudiée dans la littérature. En effet, plusieurs algorithmes ont été proposés avant d’obtenir une complexité linéaire [CH94, MS94]. Malheureusement, ce type d’algorithmes utilise des concepts sophistiqués et la preuve de leur complexité est très fine. Dans [TCHP08], les auteurs proposent un algorithme linéaire plus simple.

Définition II.31. Un *module* dans un graphe $G = (V, E)$ est un ensemble $M \subseteq V$ de sommets de G qui sont indistinguables par les sommets extérieurs. Plus formellement, pour tout sommet $x \in V \setminus M$, soit tous les sommets du module sont adjacents ($M \subseteq N(x)$), soit aucun sommet du module n’est adjacent ($M \cap N(x) = \emptyset$).

Pour tout graphe G , les ensembles $V(G)$ et $\{x\}$, $x \in V(G)$ sont les *modules triviaux* de G . Un graphe est *premier par module* s’il est indécomposable par module, autrement dit, si ses uniques modules sont triviaux. Les propriétés suivantes sont classiques et faciles à prouver à partir de la définition.

Proposition II.32. Soit $S \subseteq V(G)$ un sous-ensemble de sommets, si M est un module de G alors $S \cap M$ est un module du graphe induit $G[S]$. Réciproquement, si S est un module alors tout module du graphe induit $G[S]$ est aussi module de G .

Définition II.33. Deux modules M_1, M_2 dans un graphe G se *chevauchent* si $M_1 \cap M_2$, $M_2 \setminus M_1$ et $M_1 \setminus M_2$ sont non vides.

Proposition II.34. Si M_1 et M_2 sont deux modules disjoints alors soit ils sont complètement adjacents ($\forall x \in M_1, M_2 \subseteq N(x)$); soit ils ne sont pas reliés ($\forall x \in M_1, M_2 \cap N(x) = \emptyset$).

Cependant, il faut dire que dans la mesure où la taille d’un graphe augmente, la probabilité qu’il soit premier augmente aussi. En effet, la probabilité qu’un graphe de taille n soit premier tend vers 1 avec n . Néanmoins, “ce résultat décevant doit être tempéré par le fait que l’activité humaine produit en général des graphes largement décomposables” [DM03].

Parmi les modules d’un graphe, les modules forts sont un cas important :

Définition II.35. Un module est *fort*, s’il ne chevauche aucun autre module. Un module est *maximal* s’il n’est pas contenu dans un autre module, sauf $V(G)$.

Théorème II.36. (Théorème de Décomposition Modulaire)[CHM81, GAL67] Soit G un graphe. Un et seulement un des cas suivants est vrai

1. G est non-connexe (cas parallèle). Ses modules forts maximaux sont ses composantes connexes et le graphe quotient est un stable.
2. \bar{G} est non-connexe (cas série). Les modules forts maximaux de G sont les composantes connexes de \bar{G} et le graphe quotient est une clique.
3. G et \bar{G} sont connexes (cas premier). Tout module maximal de G est fort et le graphe quotient est premier.

Le Théorème de Décomposition Modulaire garantit, pour chaque graphe G , l'existence d'une unique partition modulaire \mathcal{P} en modules forts maximaux. De plus, chaque module fort de G est module dans un unique élément $M_i \in \mathcal{P}$. Cette relation permet la construction d'un arbre d'inclusion des modules forts de G .

Définition II.37. L'arbre de décomposition modulaire d'un graphe $G = (V, E)$ est l'arbre \mathcal{T}_G tel que :

- la racine est $V(G)$,
- les feuilles correspondent aux sommets de G ,
- chaque nœud intérieur de l'arbre correspond à un module fort de G et
- le père d'un nœud M de l'arbre est le nœud associé au plus petit module contenant le module associé à M .

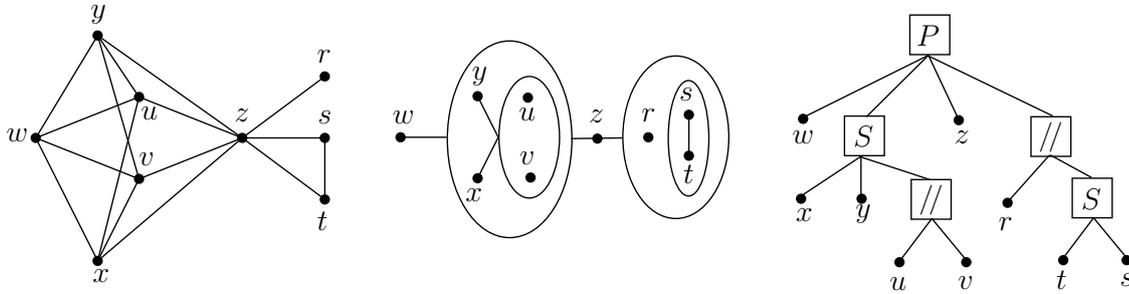


Figure II.9 – Décomposition modulaire en modules forts, son graphe quotient et l'arbre de décomposition modulaire. Les nœud internes son étiquetés comme premier (P), série (S) ou parallèle ($//$).

Remarque II.38. Soient G un graphe connexe et $G_{/\mathcal{P}}$ le graphe quotient par rapport à une partition \mathcal{P} des modules de G . Pour chaque paire $x, y \in V(G)$ soient $M_x, M_y \in \mathcal{P}$ les modules tels que $x \in M_x$ et $y \in M_y$, on a soit $d_G(x, y) = d_{G_{/\mathcal{P}}}(M_x, M_y)$ (si $M_x \neq M_y$), soit $d_G(x, y) \leq 2$ (sinon).

Théorème II.39. Soient G un graphe connexe et \mathcal{P} une partition modulaire, alors

$$\delta(G) \leq \max \{ \delta(G_{/\mathcal{P}}), 1 \}$$

Démonstration. Si G est premier, le résultat est évident, alors on suppose que G est non premier. Considérons un quadruplet maximal x, y, u, v . Soient $M_x, M_y, M_u, M_v \in \mathcal{P}$ les modules qui contiennent respectivement x, y, u, v , de plus M^* dénote un module qui contient autant de sommets de l'ensemble $\{x, y, u, v\}$ que tout autre module dans \mathcal{P} . On suppose aussi que deux sommets dans le quadruplet ne sont jamais adjacents, sinon $\delta(x, y, u, v) \leq 1$.

Si $|M^* \cap \{x, y, u, v\}| = 1$ alors chaque sommet est dans un module différent ; et donc, d'après II.38, on a $\delta_G(x, y, u, v) = \delta_{G_{/\mathcal{P}}}(M_x, M_y, M_u, M_v)$.

Si $|M^* \cap \{x, y, u, v\}| = 2$ sans perte de généralité, on suppose que $\{x, y\} \in M^*$. À partir de la remarque II.38 $xu = yu$ et $xv = yv$ et donc $xu + yv = yu + xv$. Alors, $xu + yv = xv + yu$, de plus $xy + uv \leq xy + uy + yv \leq 2 + uy + xv$ donc $2\delta(x, y, u, v) = 2$.

Si $M^* \cap \{x, y, u, v\} = \{x, y, u\}$ alors $xv = yv = uv = d(M^*, M_v)$, de plus il existe $v' \in V(G)$ tel que $xv' = yv' = uv' = 1$. Comme aucune paire du quadruplet n'est adjacente, alors $xy = xu = xv = 2$. Ainsi, $\delta(x, y, u, v) = 0$.

Finalement, si tous les sommets du quadruplet sont dans M^* , alors la distance entre chaque paire dans le quadruplet est égale à deux et donc $\delta(x, y, u, v) = 0$. \square

Remarque II.40. D'après la remarque II.38 on sait qu'il existe un sous-graphe isométrique $H \subseteq G$ isomorphe à $G_{/\mathcal{P}}$, de plus si \mathcal{P} est la partition en modules forts maximaux alors $G_{/\mathcal{P}}$ est soit premier, soit une clique ou un stable. Ainsi,

$$\delta(G) \leq \max \{ \delta(P) : P \text{ sous-graphe premier isométrique de } G \} \cup \{1/2\}.$$

Exemple II.41. Le graphe quotient de la figure II.9 est $G_{/\mathcal{P}} = P_4$ qui est 0-hyperbolique, donc le théorème II.39 garantit que le graphe est 1-hyperbolique. D'autre part, il est facile de vérifier que le graphe contient un cycle à quatre sommets comme sous-graphe induit (isométrique). Ce qui nous amène à conclure que l'hyperbolicité du graphe est égale à un.

Le théorème II.39 nous permet de donner une borne de l'hyperbolicité pour la famille des graphes où la structure des quotients premiers est connue. Par exemple, la famille des graphes complètement décomposables via décomposition modulaire correspond exactement aux cographes. Ainsi, tout cographe est 1-hyperbolique, comme nous l'avions déjà remarqué dans la section 2. La famille des cographes correspond exactement aux graphes qui n'ont pas le chemin à quatre sommets P_4 comme sous-graphe induit. Une généralisation des cographes est la famille de graphes P_4 -sparse introduite par Hoàng dans [HON85].

Définition II.42 (P_4 -sparse). Un graphe G est P_4 -sparse si tout sous-ensemble de sommets de taille cinq contient au plus un chemin P_4 comme sous-graphe induit.

Les graphes premiers P_4 -sparse peuvent être bien caractérisés :

Définition II.43 (araignées). Un graphe G est une *araignée fine* si l'ensemble des sommets du graphe peut être partitionné en deux sous-ensembles $V(G) = S \uplus K$ avec S un stable et K une clique, de façon que $|K| = |S|$ ou $|K| = |S| + 1$. De plus, chaque sommet $s \in S$ a une unique voisin dans K et chaque sommet dans K a, au plus, un voisin dans S . Une *araignée épaisse* est le complément d'une araignée fine.

Dans [HON85, JO92] les auteurs montrent que les graphes premiers P_4 -sparse sont des araignées. Clairement, les araignées sont des graphes 1-hyperboliques. Par conséquent,

Théorème II.44. *Les graphes P_4 -sparse sont 1-hyperboliques.*

6.2 Split-décomposition

La décomposition par split, ou 1-join composition, a été introduite par Cunningham [CE80, CUN82]. Un split peut être vu comme une généralisation de la décomposition modulaire, dans le sens où tout module est aussi un split. Comme la décomposition modulaire, la split décomposition est un outil avec un large spectre d'applications [RAO08].

Définition II.45 ([CUN82]). Étant donné un graphe connexe G , un *split* de G est une partition X_1, X_2 de $V(G)$ avec $|X_1| \geq 2$, $|X_2| \geq 2$ et telle que pour toute paire $x \in X_1 \cap \mathcal{N}(X_2)$ et $y \in X_2 \cap \mathcal{N}(X_1)$, on a $(x, y) \in E(G)$. On utilise la notation (V_1, V_2, V_3, V_4) où $X_1 = V_1 \cup V_2$, $X_2 = V_3 \cup V_4$, $V_2 = \mathcal{N}(X_2) \cap X_1$, $V_3 = \mathcal{N}(X_1) \cap X_2$ (voir figure II.10 gauche).

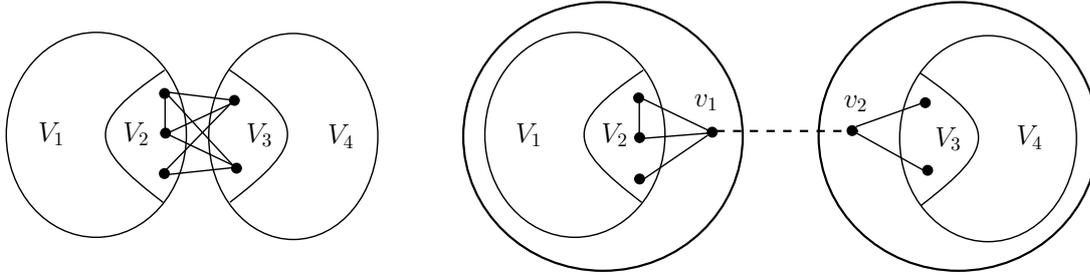


Figure II.10 – Un split $X_1 = V_1 \cup V_2$, $X_2 = V_3 \cup V_4$ et ses deux compactages $G_{X_2}^{v_1}$, $G_{X_1}^{v_2}$.

Si V_4 est vide, alors on dit que V_3 est un module du graphe. Pour tout graphe G , chaque singleton $\{v\}$, $v \in V(G)$ forme le split $(\emptyset, \{v\}, \mathcal{N}(v), V(G) \setminus \mathcal{N}[v])$: ce sont les splits triviaux. On dit qu'un graphe est *premier par splits* s'il ne contient aucun split non trivial. Inversement, un graphe est *dégénéré par splits* si tout sous-graphe induit de quatre ou plus sommets n'est jamais premier. Les graphes dégénérés sont les cliques (K_n) et les étoiles $(K_{1,n})$.

Une façon classique [LAN01] de représenter un split (V_1, V_2, V_3, V_4) est la création d'un graphe avec deux sommets et une unique arête. Chaque sommet est associé aux compactages $G_{X_2}^{v_1}$, $G_{X_1}^{v_2}$ et l'arête entre eux est associée aux (nouveaux) sommets v_1 et v_2 appelés *sommets marqueurs* (figure II.10 droite).

La décomposition récursive par split sur les composantes $G_{X_2}^{v_1}$ et $G_{X_1}^{v_2}$ permet d'obtenir un arbre qui, dans chaque sommet, contient les *composantes du split* et dont chaque arête représente un split du graphe. Cette décomposition n'est pas unique du fait qu'un graphe peut avoir une quantité exponentielle de splits différents (par exemple un graphe dégénéré). Néanmoins, Cunningham montre dans [CUN82] qu'il existe une représentation simple (linéaire) de tous les splits d'un graphe.

Théorème II.46. [CUN82] *Pour chaque graphe connexe G , il existe un unique arbre T appelé l'arbre de split tel que :*

1. l'ensemble des feuilles est égale à l'ensemble des sommets de G ;
2. tout sommet de T est associé, soit à une clique, soit à un graphe premier, soit à un graphe dégénéré ;
3. une partition X_1, X_2 de $V(G)$ est un split si et seulement si elle est associée, soit à une arête de T , soit à une partition des arêtes d'un sommet de T contenant un graphe dégénéré ;
4. T a une quantité minimale de sommets.

La décomposition associée à l'arbre de split est obtenue en faisant une partition récursive du graphe par splits où à chaque fois on choisit un split où $V_1 \cup V_2$ est maximal (par rapport à l'inclusion). Au niveau algorithmique, la décomposition par split du théorème II.46 et son arbre peuvent être obtenus en temps linéaires [CDMR09, DAH00].

Remarque II.47. Tous les graphes associés aux sommets de T sont des sous-graphes isométriques de G .

La figure II.11 montre l'exemple d'un graphe avec son arbre de décomposition.

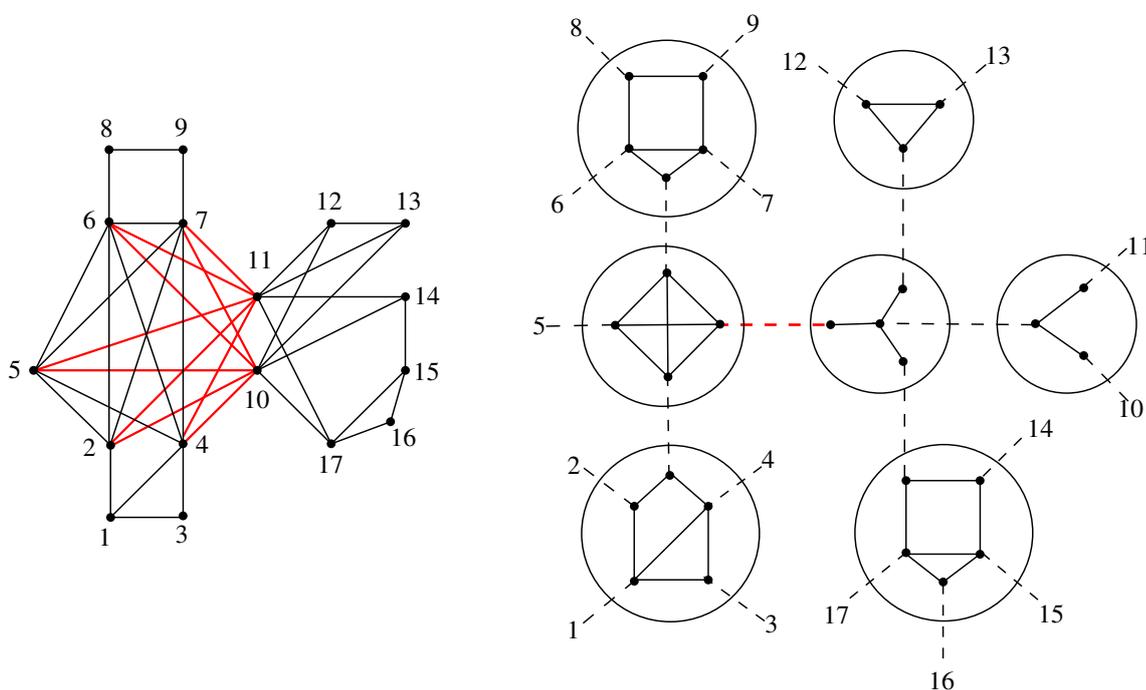


Figure II.11 – Un graphe et son arbre de splits.

La proposition suivante donne une borne de l'hyperbolicité par rapport à ses composantes de split.

Théorème II.48. *Si X_1, X_2 est un split d'un graphe connexe G alors,*

$$\max\{\delta(G_{X_2}^{v_1}), \delta(G_{X_1}^{v_2})\} \leq \delta(G) \leq \max\{\delta(G_{X_2}^{v_1}), \delta(G_{X_1}^{v_2}), 1\}.$$

Démonstration. La première inégalité est directe à partir de la remarque II.47 et la proposition II.13(1).

Soit x, y, u, v un quadruplet maximal. On suppose que $|X_1 \cap \{x, y, u, v\}| \geq |X_2 \cap \{x, y, u, v\}|$. Trois cas sont possibles :

1. Si tous les sommets du quadruplet sont dans X_1 alors le résultat est immédiat.
2. Si $|X_1 \cap \{x, y, u, v\}| = 3$ on suppose que $x, y, u \in X_1$ et que $v \in X_2$. Comme X_1, X_2 forment un split, alors pour $z \in \{x, y, u\}$ on a $zv = zv_1 + v_2v - 1$. Ainsi, $2\delta(x, y, u, v) = 2\delta_1(x, y, u, v) - 2\delta_2(x, y, u, v) = (2\delta_1(x, y, u, v_1) + v_2v - 1) - (2\delta_2(x, y, u, v_1) + v_2v - 1) = 2\delta(x, y, u, v_1)$.
3. Si $|X_1 \cap \{x, y, u, v\}| = 2$ on suppose que $x, y \in X_1$ et $u, v \in X_2$. On a $xu + yv = xv + yu = xv_1 + v_2u - 1 + yv_1 + v_2v - 1$, d'autre part $xy + uv \leq xv_1 + v_1y + uv_2 + v_2v$, donc $2\delta(G) \leq 2$. De plus, $2\delta(G) = 2$ si et seulement si x, y, u, v forment un cycle C_4 comme sous-graphe induit.

□

Le théorème II.46 garantit que toutes les composantes de split sont, soit des graphes dégénérés (qui ont hyperbolicité égale à zéro), soit premiers. Donc, d'après la proposition II.48 on a :

Corollaire II.49. *Pour tout graphe connexe G ,*

$$\delta(G) \leq \max\{\{\delta(H), H \subseteq G \text{ et } H \text{ premier par split}\} \cup \{1\}\}$$

La propriété ci-dessus nous permet de borner l'hyperbolicité lorsque les graphes induits premiers sont connus. La famille des graphes distance héréditaires en est un exemple.

Définition II.50. Un graphe est *distance héréditaire* si tout sous-graphe induit est isométrique.

Un graphe est distance héréditaire si et seulement si il est complètement décomposable par split [HM90], c'est-à-dire si toutes les composantes de split dans son split arbre sont dégénérées. Ainsi,

Corollaire II.51 ([BC08, WZ11]). *Tout graphe héréditaire est 1-hyperbolique, en plus son hyperbolicité est 1 si et seulement s'il contient le cycle C_4 comme graphe induit.*

Dans la section suivante, nous nous intéressons aux séparateurs dans un cas plus général.

6.3 δ -hyperbolicité et séparateurs

Dans les sections précédentes, nous avons vu comment calculer l'hyperbolicité à partir d'une décomposition du graphe. Malheureusement, ce type de décomposition n'est pas utile si le graphe est premier. Par la suite, nous donnerons des heuristiques qui peuvent être utiles lorsqu'une décomposition par splits (modules) n'existe pas.

La proposition suivante donne une relation entre l'hyperbolicité du graphe et l'hyperbolicité de ses composantes connexes générées via un séparateur isométrique (figure II.12).

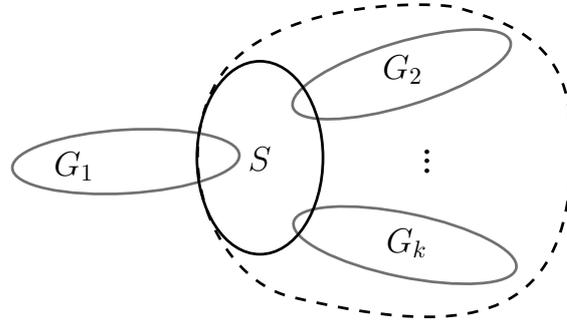


Figure II.12 – Séparateur isométrique et ses composantes bi-connexes.

Proposition II.52. *Étant donné un graphe G et un sous-graphe isométrique S , on dénote par G_1, \dots, G_k les composantes connexes de $G - S$. Alors,*

$$\max_{1 \leq i \leq k} \{\delta(G_i \cup S)\} \leq \delta(G) \leq \max \left\{ \max_{1 \leq i \leq k} \{\delta(G_i \cup S)\} + \text{diam}_G(S)/2, \text{diam}_G(S) \right\}$$

Démonstration. Comme S est un sous-graphe isométrique, alors tout sous-graphe $G_i \cup S$ avec $i \in \{1, \dots, k\}$ est aussi isométrique. D'après la remarque II.14, on a la première inégalité.

Pour la deuxième inégalité, soit $x, y, u, v \in V(G)$ un quadruplet maximal avec $\delta_1(x, y, u, v) = xy + uv$ et $\delta_2(x, y, u, v) = xu + vy$. Sans perte de généralité, on assume que $|(G_i \cup S) \cap \{x, y, u, v\}| \leq |(G_1 \cup S) \cap \{x, y, u, v\}|$ pour tout $i \in \{1, \dots, k\}$. Autrement dit, l'ensemble G_1 contient autant d'éléments du quadruplet que toutes les autres composantes connexes.

Quitte à renommer les sommets, quatre scénarios sont possibles :

1. $|(G_i \cup S) \cap \{x, y, u, v\}| = 4$.

de manière évidente $\delta(u, v, x, y) \leq \delta(G_1 \cup S)$.

Par la suite, pour tout sommet $x \in V(G)$ on dénote par \tilde{x} la projection sur S du sommet x , c'est-à-dire $\tilde{x} \in S$ et $d(x, S) = x\tilde{x}$.

2. $|(G_i \cup S) \cap \{x, y, u, v\}| = 3.$

Sans perte de généralité on assume que $y \notin G_1 \cup S$. Du fait que S est un séparateur, on a pour tout sommet $z \in G_1 \cup S$ que le plus court chemin entre z et y doit traverser S , donc $z\tilde{z} + y\tilde{y} \leq zy$. Donc, $z\tilde{y} \leq z\tilde{z} + \tilde{z}\tilde{y} \leq zy - y\tilde{y} + \tilde{z}\tilde{y} \leq zy - y\tilde{y} + \text{diam}(S)$, donc

$$z\tilde{y} \leq zy - y\tilde{y} + \text{diam}(S).$$

En utilisant l'inégalité ci-dessus pour u et v , on obtient :

$$\begin{aligned} v\tilde{y} + ux &\leq vy + ux + \text{diam}_G(S) - y\tilde{y} \\ u\tilde{y} + vx &\leq uy + vx + \text{diam}_G(S) - y\tilde{y} \end{aligned}$$

Et donc :

$$\begin{aligned} \max\{v\tilde{y} + ux, u\tilde{y} + vx\} &\leq \max\{vy + ux, uy + vx\} + \text{diam}_G(S) - y\tilde{y} \\ 2\delta_2(u, v, x, \tilde{y}) &\leq 2\delta_2(u, v, x, y) + \text{diam}_G(S) - y\tilde{y} \\ 2\delta_2(u, v, x, \tilde{y}) - \text{diam}_G(S) + y\tilde{y} &\leq 2\delta_2(u, v, x, y). \end{aligned}$$

D'autre part,

$$2\delta_1(u, v, x, y) = xy + uv \leq x\tilde{y} + y\tilde{y} + uv \leq 2\delta_1(u, v, x, \tilde{y}) + y\tilde{y}.$$

Ainsi,

$$\begin{aligned} 2\delta(u, v, x, y) &= \delta_1(x, y, u, v) - 2\delta_2(u, v, x, y) \\ &\leq (\delta_1(u, v, x, \tilde{y}) + y\tilde{y}) - (\delta_2(u, v, x, \tilde{y}) - \text{diam}(S) + y\tilde{y}) \\ &\leq \delta(u, v, x, \tilde{y}) + \text{diam}(S) \\ &\leq 2\delta(G \cup S) + \text{diam}(S). \end{aligned}$$

3. $|(G_i \cup S) \cap \{x, y, u, v\}| = 2.$

Comme S est un séparateur, alors indépendamment de la paire dans $G_i \cup S$, deux paires de chemins associées aux distances $\delta_1(x, y, u, v)$, $\delta_2(x, y, u, v)$ et $\delta_3(x, y, u, v)$ doivent traverser S . Donc, soit $(I(x, y) \cap S \neq \emptyset$ et $I(u, v) \cap S \neq \emptyset)$, soit $(I(x, u) \cap S \neq \emptyset$ et $I(y, v) \cap S \neq \emptyset)$. D'après la remarque II.19 on conclut que

$$\delta(u, v, x, y) \leq \min\{d(I(x, u), I(v, y)), d(I(x, v), I(u, y))\} \leq \text{diam}(S).$$

4. $|(G_i \cup S) \cap \{x, y, u, v\}| = 1.$

Par le choix de G_1 , chaque sommet du quadruplet u, v, x, y se retrouve dans des composants connexes différentes. Tout chemin entre les paires du quadruplet doivent traverser S . Ainsi, nous pouvons conclure à partir de la remarque II.19 que $\delta(x, y, u, v) \leq \text{diam}(S)$.

□

Remarque II.53. On peut utiliser des séparateurs qui ne sont pas nécessairement des sous-graphes connexes mais un sous-ensemble quelconque. Néanmoins, dans ce cas, il faut utiliser la métrique du graphe original plutôt que la métrique induite par les sous-graphes.

La proposition II.52 avec la proposition II.13(5) nous permettent d'obtenir une borne supérieure de l'hyperbolicité en terme de diamètre.

Corollaire II.54. Soit S un séparateur du graphe G et G_1, \dots, G_k les composantes connexes de $G \setminus S$, alors

$$\delta(G) \leq \max \left\{ \max_{i \in \{1, \dots, k\}} \{\text{diam}(G_i)\}, \text{diam}(S) \right\}$$

7 Calcul pratique de l'hyperbolicité

Dans cette section, nous nous intéressons à la question du calcul de l'hyperbolicité. Il est clair, à partir de sa définition, que déterminer l'hyperbolicité d'un graphe G est un problème de temps polynomial en fonction du nombre de sommets. Plus précisément, un calcul exhaustif parmi tous les quadruplets du graphe permet d'obtenir l'hyperbolicité en temps $O(|V(G)|^4)$.

Dans ce chapitre nous avons montré l'existence des diverses décompositions qui permettent de réduire la taille des instances en respectant la valeur de l'hyperbolicité. Notamment, la décomposition par split est la décomposition la plus générale (par rapport aux composantes bi-connexes et à la décomposition modulaire) qui peut être obtenue en temps linéaire [CDMR09].

Nous montrerons une série de propriétés qui permettent d'éviter le calcul sur chaque quadruplet du graphe.

7.1 Calcul approché

On s'intéresse à la propriété suivante qui a été prouvé par Gromov dans [GRO87],

Proposition II.55. [GRO87] Soient G un graphe et $x^* \in V(G)$ alors on a

$$\delta(G) \leq 2 \max_{y, u, v \in V(G)} \delta(x^*, y, u, v)$$

.

Cette approximation permet de borner l'hyperbolicité en fixant un sommet x^* et en calculant la valeur de $2\delta(x^*, y, u, v)$ pour tout triplet $y, u, v \in V(G)$. Il existe alors un algorithme $O(V(G)^3)$ pour un calcul approché de l'hyperbolicité avec un facteur multiplicatif égal à deux.

7.2 Heuristiques pour le calcul exact

Paires éloignées

Une stratégie différente consiste à identifier des sous-ensembles de quadruplets maximaux afin de réduire l'espace dans lequel on cherche un maximum de la fonction δ .

On dit que une paire $(x, y) \in V(G)$ est *éloignée* si pour tout sommet $z \in V(G)$ $xy \geq \max\{xz, yz\}$.

Proposition II.56. [NOG09] *Pour tout graphe G il existe un quadruplet $x, y, u, v \in V(G)$ tel que $\delta_1(x, y, u, v) = xy + uv$ et les paires $(x, y), (u, v)$ sont éloignées.*

La démonstration de cette proposition peut être obtenue en considérant un quadruplet maximal x, y, u, v qui maximise la fonction $\delta_1(x, y, u, v) = xy + uv$. Si (x, y) n'est pas une paire éloignée alors il existe $x' \in V(G)$ (ou $y' \in V(G)$) tel que $x' \in \mathcal{N}(x)$ et $x'y > xy$. Ainsi, $\delta(x', y, u, v) = \delta(x, y, u, v)$ et $\delta_1(x', y, u, v) > \delta_1(x, y, u, v)$ ce qui contredit le choix du quadruplet x, y, u, v .

Le calcul de toutes les paires éloignées peut être fait par un parcours en profondeur qui commence sur chaque sommet du graphe. Ceci se fait en temps $O(|V(G)||E(G)|) = O(|V(G)|^3)$. On calcule donc la fonction δ seulement dans les quadruplets x, y, u, v où (x, y) et (u, v) sont des paires alignées.

Un graphe peut avoir une quantité quadratique (en fonction du nombre de sommets) de paires éloignées, ce qui ramène à un algorithme en $O(|V(G)|^4)$. Néanmoins, dans le cas général, des calculs expérimentaux [NOG09] montrent que cette stratégie représente une réduction du temps pratique.

Chapitre III

Mesure de l'arborescence d'Internet

Ce chapitre porte sur la topologie d'Internet. Comprendre la structure d'Internet couvre plusieurs aspects. Le premier est de comprendre le processus de construction d'Internet, un autre consiste à concevoir des modèles de simulation fidèles, un troisième est de mesurer certaines propriétés. Nous nous sommes surtout consacrés à ce dernier point. Nous nous intéressons notamment au caractère arborescent d'Internet via l'analyse de deux paramètres en rapport avec l'arborescence d'un graphe.

Les deux paramètres ici étudiés sont : **la largeur arborescente et l'hyperbolicité**. Ces deux paramètres évaluent dans quelle mesure un graphe est comparable à un arbre par rapport à la connectivité et à la métrique (chapitre II), respectivement.

Pour une plus grande fiabilité par rapport aux données, nous analysons ces paramètres sur les graphes obtenus de sources et de stratégies différentes.

La motivation de ce problème est claire : la connaissance de la structure d'un réseau conduit à la conception d'algorithmes de communications plus adaptés et plus rapides. Autrement dit, notre approche a d'abord été motivée par l'idée de trouver les propriétés qui rendent Internet manipulable algorithmiquement. En effet, de nombreux problèmes difficiles ont des algorithmes rapides lorsque le graphe a une largeur arborescente ou une hyperbolicité faible.

Comme nous avons vu dans le chapitre II, l'hyperbolicité est liée à la *métrique* d'un graphe : un graphe a une hyperbolicité faible si les distances entre ses sommets se comportent comme dans un arbre. Le choix de ce paramètre est double. D'abord, des propriétés arborescentes ont déjà été remarquées sur Internet [ST08, RMK⁺09, NS09], ce qui fait d'Internet un bon candidat pour avoir une faible hyperbolicité. En deuxième lieu, comme nous avons vu au chapitre II, une hyperbolicité petite a des conséquences algorithmiques sur des problèmes importants dans tout réseau de communication.

Notre deuxième paramètre, la largeur arborescente ou *treewidth* est en rapport avec la *connectivité* d'un graphe. En effet, un arbre peut être facilement déconnecté en supprimant *un* sommet interne. À cet égard, la connectivité peut, à titre d'exemple

introductif, être mesurée par le nombre de sommets qu'on doit supprimer afin de déconnecter le graphe. Il semblerait alors naturel de mesurer la connectivité comme le nombre de sommets que nous avons besoin de supprimer afin de déconnecter une paire de noeuds dans le graphe. Étant donné qu'un sommet avec un faible degré peut apparaître isolé par la suppression de ses voisins, cette mesure ne rend pas compte de la connectivité globale du graphe. Le paramètre de largeur arborescente est mieux adapté dans la mesure où il est lié à l'existence de *séparateurs équilibrés*. En deuxième lieu, une largeur arborescente faible permet la conception d'algorithmes polynomiaux (ou linéaires) pour de nombreux problèmes NP-difficiles [BOD88].

Nous commençons le chapitre avec une description des travaux similaires et leur relation avec notre étude. La section 2 décrit les outils utilisés pour obtenir les différents graphes et les sources de traces utilisées. La section 3 donne la définition de la largeur arborescente et sa relation avec la connectivité. Des bornes supérieures et inférieures à la largeur arborescente γ sont présentées. La section 4 traite de l'hyperbolicité et de sa mesure sur nos graphes. Finalement, dans la section 5 nous nous tournons vers l'analyse de différents modèles classiques de génération de graphes pour modéliser la topologie d'Internet, nous mesurons les deux paramètres pour quelques modèles classiques.

Les résultats obtenus dans ce chapitre ont fait objet de la publication [DMSV11C].

1 Travaux similaires

La topologie d'Internet a été étudiée par exemple par Pansiot et Grad [PG98] au niveau des routeurs et Govindan et Reddy [GR97] au niveau des AS. De nombreux travaux ont abordé la distribution des degrés aux deux niveaux. La distribution est à aile lourde et peut être modélisée avec des lois de puissance [FFF99] ou la distribution de Weibull [Bc01, CCG⁺02, SMWA04].

Broido et claffy étudient la *connectivité* d'Internet [Bc01]. Ils travaillent notamment sur le comportement de la composante fortement connexe à l'égard de la suppression de sommets. La largeur arborescente, est un paramètre qui mesure la connectivité globale du graphe.

Une vaste littérature est dédiée à la compréhension de l'espace des délais dans Internet où les arêtes entre les routeurs ont un poids proportionnel au temps requis pour le transit d'un paquet entre les routeurs. Cet espace des délais peut être plongé plus au mois bien dans un espace euclidien et permet de construire un système de coordonnées virtuelles [NZ02, DCKM04]. Shavitt et Tankel [ST08] ont observé qu'on obtient de meilleurs résultats lorsque le plongement est fait dans un espace hyperbolique. Ramasubramanian et al. introduisent une version modifiée qui semble être bien adaptée aux délais dans Internet [RMK⁺09]. Cette propriété peut être utilisée pour prédire le temps de latence via le plongement dans un arbre. D'autres notions de dimension ont été testées sur Internet telles que la dimension fractale [AK08] ou la dimension doublante [FLV08].

La notion originale d'hyperbolicité définie par Gromov[GRO87], basée sur une condition sur quatre points, n'avait pas encore été mesurée sur Internet à notre connaissance. Narayan et Sanjeev [NS09] ont étudié une définition alternative basée sur les triangles δ -fins qui est mesurée sur les données de Rocketfuel [SMWA04]. Nos résultats, qui sont du même type, valent pour la définition originale de Gromov.

La découverte de l'*aile lourde* de la distribution des degrés a soulevé la question de la conception de modèles adéquats pour Internet. Un premier but de la modélisation est de comprendre la cause de cette distribution. L'*attachement préférentiel* a été proposé par Barabási et Albert [BA99] pour expliquer la structure d'Internet ou des réseaux sociaux. En ce qui concerne Internet, une explication possible est l'optimisation de plusieurs critères avec des compromis [FKP02]. Une autre raison de modéliser la topologie d'Internet est de générer des réseaux avec des propriétés similaires pour de besoins de simulations. Étonnamment les graphes aléatoires générés à partir de la distribution des degrés semblent être appropriés pour bien modéliser les propriétés structurelles [TGJ⁺02]. Notre travail accrédite ce point par rapport aux deux paramètres mesurés.

2 Le graphe d'Internet

Internet peut être décrit, grosso modo, comme un réseau de réseaux, organisés de façon hiérarchique. Sa structure peut donc être étudiée à différents niveaux. Le niveau le plus bas correspond à la connexion entre les routeurs, à savoir, les machines qui interviennent dans le routage de l'information. Chaque routeur appartient à une organisation (fournisseur d'accès à Internet, université, etc.) responsable de sa politique de routage. Un *Système Autonome* ou AS¹ est un ensemble de routeurs sous l'administration d'une même organisation et qui partage une même politique de routage. Le graphe des connexions entre les AS constitue le réseau de plus haut niveau, qui forme la structure principale d'Internet. Cela se traduit en trois types de graphes qui sont intéressants pour nos mesures : le graphe des AS qui modélise les connexions entre les systèmes autonomes, le graphe des routeurs qui modélise les connexions de tous les routeurs d'Internet et finalement les graphes intra-AS qui modélisent les connexions entre routeurs à l'intérieur d'un même AS.

L'obtention d'une version complète et fidèle des graphes d'Internet est un problème bien identifié. Les connexions entre les AS ainsi que la topologie interne de la plupart des AS, ne sont pas publiques et leur obtention est une branche de recherche en soi. Nous utilisons des sources de traces différentes qui se servent des stratégies diverses pour découvrir les connexions entre les routeurs. Cependant, les stratégies peuvent être classifiées en deux grands groupes : *l'exploration passive* et *l'exploration active*. L'exploration passive utilise les données des tables de routage entre AS (tables BGP²)

1. l'acronyme en anglais de Autonomous System
2. Border Gateway Protocol

Chapitre III. Mesure de l'arborescence d'Internet

sans intervenir sur le réseau, alors que l'*exploration active* se sert de paquets sondes qui parcourent le réseau en explorant les connexions.

Les techniques de sondages et les graphes ainsi obtenus ont des problèmes associés aux méthodes de mesure, comme l'incomplétude ou des imprécisions [WAD09, OPW⁺10]. Néanmoins, un des paramètres (treewidth) que nous étudions donne des bornes valides même en cas de manque d'arêtes ou sommets.

Traceroute : Traceroute est un exemple d'exploration active. Il correspond à un outil de diagnostic de réseau qui découvre la route suivie par un paquet vers une destination. Le principe basique de ce type d'exploration est l'envoi de plusieurs paquets sondes depuis le routeur moniteur vers la même destination. Les paquets contiennent dans leur en-têtes des valeurs de *durée de vie* (TTL³) de plus en plus grands. Chaque fois que le paquet arrive vers une interface d'un routeur, sa durée de vie est décrémenté de un. Si la valeur de la durée de vie atteint zéro le paquet n'est pas transmis et un message d'erreur est envoyé vers la source. Pour obtenir l'information du routeur qui envoie le message d'erreur, un type spécial de paquet est utilisé. En général il s'agit de paquets de type UDP⁴ ou ICMP⁵ avec une demande de réponse (ECHO request). De cette manière, ils est possible d'inférer un lien entre deux interfaces consécutives (plus précisément entre les deux routeurs possédant ces interfaces).

```
tracert biblio.math.jussieu.fr
tracert to biblio.math.jussieu.fr (134.157.100.3), 30 hops max, 40 byte packets
 1 gw.dim.uchile.cl (146.83.7.1) 0.278 ms 0.287 ms 0.274 ms
 2 172.17.32.1 (172.17.32.1) 1.348 ms 1.710 ms 1.962 ms
 3 cisco-dti.cec.uchile.cl (200.9.98.130) 1.559 ms 1.796 ms 2.017 ms
 4 172.16.38.1 (172.16.38.1) 3.989 ms 4.036 ms 4.025 ms
 5 200.89.75.37 (200.89.75.37) 4.005 ms 4.092 ms 3.981 ms
 6 rci2-uchile.reuna.cl (146.83.242.25) 4.142 ms 4.039 ms 4.005 ms
 7 reuna-cl-sant.core.redclara.net (200.0.204.141) 4.101 ms 4.481 ms 4.712 ms
 8 saopaulo-santiago.core.redclara.net (200.0.204.38) 53.259 ms 53.241 ms 53.230 ms
 9 RedCLARA.rt1.mad.es.geant2.net (62.40.124.137) 291.246 ms 291.467 ms 291.329 ms
10 so-4-1-0.rt1.par.fr.geant2.net (62.40.112.118) 307.726 ms 308.380 ms 308.364 ms
11 renater-gw.rt1.par.fr.geant2.net (62.40.124.70) 311.907 ms 311.896 ms 311.884 ms
12 te1-1-paris1-rtr-021.noc.renater.fr (193.51.189.38) 309.027 ms 309.008 ms 308.746 ms
13 * * *
14 rap-vl165-te3-2-jussieu-rtr-021.noc.renater.fr (193.51.181.101) 307.888 ms 324.131 ms
322.731 ms
15 site-6.01-jussieu.rap.prd.fr (195.221.127.182) 307.927 ms 308.137 ms 308.487 ms
16 r-jussieu2.reseau.jussieu.fr (134.157.254.14) 308.512 ms 308.438 ms 308.426 ms
17 134.157.100.3 (134.157.100.3) 308.584 ms 307.828 ms 307.464 ms
```

Figure III.1 – Exemple de traceroute

La figure III.1 montre la sortie de l'outil `tracert` depuis un hôte chilien (eu-ler.dim.uchile.cl 146.83.7.8) vers une destination en France (biblio.math.jussieu.fr). Par exemple, nous trouvons que la route suivie par le paquet utilise le lien (transatlan-

3. "Time to live"

4. "User Datagram Protocol"

5. "Internet Control Message Protocol"

tique) qui relie saopaulo-santiago.core.redclara.net et RedCLARA.rt1.mad.es.geant2.net entre les étapes huit et neuf respectivement.

L'utilisation de traceroute pose plusieurs difficultés. Par exemple, le chemin suivi par les différents paquets sondes peuvent être différents ou l'adresse de l'interface du routeur dans le message ICMP n'est pas l'interface utilisée par le paquet sonde. De plus, plusieurs routeurs ne répondent pas aux requêtes ICMP (comme par exemple à l'étape 13 de la figure III.1). Cependant, un problème clé reste le plus difficile est la *résolution d'alias* qui consiste à identifier les différentes interfaces (qui ont une chacune une adresse IP différente) de chaque routeur.

Tables BGP BGP⁶ est le protocole avec lequel les informations de routage sont échangées entre les Systèmes Autonomes. Il est la base de la politique de routage dans Internet. Les routeurs à l'intérieur d'un réseau et reliés avec des routeurs en dehors correspondent aux *routeurs de frontière*⁷. Ils jouent un rôle fondamental lorsqu'un paquet doit être envoyé vers une destination en dehors de l'AS.

Dans le protocole BGP, les routeurs de frontière gardent une table (table BGP) contenant une liste avec les destinations vers lesquelles un paquet peut être envoyé avec la suite des AS que le paquet doit traverser pour atteindre la destination. Il serait impossible de gérer une liste exhaustive avec toutes les adresse IP avec un délai acceptable. Les destinations qui partagent un même chemin peuvent être agrégées par préfixe.

La figure III.2 montre une partie d'une table BGP correspondant à un routeur de frontière. Nous voyons que pour atteindre le Système Autonome responsable des adresses IP qui partagent les premiers 16 bits avec 2.0.0.0 (le bloc 2.0.0.0/16), la route la plus courte passe par les AS 3292 et 12654. Nous pouvons en déduire qu'il existe un lien entre ces deux AS. De plus, nous pouvons aussi conclure que toutes les adresses dans ce bloc sont gérés par l'AS 12654.

Les routeurs de frontière appartenant à des Systèmes Autonomes voisins échangent des messages pour actualiser leur tables.

Les tables BGP de certains routeurs sont publiques. Le projet *Route View* [ROU] de l'université d'Oregon publie depuis 2001, et de manière périodique, les tables BGP de différents routeurs en divers points du réseau.

2.1 Les sources de données

Internet peut être étudié à différents échelles. Dans ce travail nous nous intéressons à trois types de réseaux :

- la topologie globale des routeurs,
- la topologie des AS et

6. "Border Gateway Protocol"

7. "border gateway"

Chapitre III. Mesure de l'arborescence d'Internet

```
BGP table version is 334348919, local router ID is 128.223.51.108
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network          Next Hop          Metric LocPrf Weight Path
r> 0.0.0.0         95.140.80.253      0
* 2.0.0.0/16       74.40.7.35         0
*                  72.36.126.8        904
*                  64.57.28.241       0
*>                 195.215.109.252    0
*                  218.189.6.129      0
*                  207.246.129.1      0
*                  200.160.127.238    0
*                  195.215.109.254    65469
*                  216.218.252.164    0
...
* 4.79.22.0/23     74.40.7.35         0
*                  200.160.127.238    0
*                  95.140.80.253      0
*>                 134.222.87.4       0
* 4.79.181.0/24    74.40.7.35         0
*                  72.36.126.8        0
*                  207.246.129.1      0
*                  195.215.109.252    0
*                  207.246.129.2      0
*                  200.160.127.238    0
*                  216.218.252.164    0
*                  195.215.109.254    75499
*                  202.147.0.142     10003
*>                 89.149.178.10      10
...
```

Figure III.2 – Exemple d'une table BGP.

– la topologie interne de certains AS.

Pour chaque graphe, nous utilisons différentes sources de données afin de dépasser les difficultés associées aux techniques utilisées. Le réseau étant très dynamique, nous analysons des instantanés obtenus à différents moments. Par la suite, on détaillera les sources pour chaque type de graphe.

Graphes des routeurs

Rocketfuel : Le Projet Rocketfuel [SMWA04] a été développé au sein de l'université de Washington. Il utilise `traceroute` comme outil d'exploration. Le projet est focalisé sur la topologie interne de certains Systèmes Autonomes. Les données ont été obtenues avec l'aide de 294 serveurs publics entre le 27/12/2001 et le 01/02/2002. La résolution d'alias et la relation routeur-AS est faite à partir des tables BGP et du système de nommage DNS.

MRINFO : Le projet *mrinfo* (actuellement le projet MERLIN⁸) dépend de l'Université Catholique de Louvain et de l'Université de Strasbourg [MdSD⁺09, PMDB09].

8. [HTTP://SVNET.U-STRASBG.FR/MRINFO/INDEX.HTML](http://svnet.u-strasbg.fr/mrinfo/index.html)

L'exploration est faite via l'outil multicast `mrinfo` en utilisant le protocole IGMP⁹. Les paquets sondes contiennent un message IGMP ASK NEIGHBORS, le routeur IPv4 multicast destination répond à ce message avec la liste de toutes les adresses IP de ses interfaces. À partir d'une unique source placée à l'Université de Strasbourg, une application récursive de cette procédure permet l'exploration du réseau. Cette méthode évite d'avoir à utiliser un processus de résolution d'alias. Cependant, seuls les routeurs supportant IGMP répondent à ce type de messages (les routeurs du mbone en particulier), ce qui limite la puissance de l'exploration.

CAIDA : Dans le cadre du projet "*Macroscopic Topology Project*", CAIDA¹⁰ exécute, depuis 1998, de façon journalière des explorations d'Internet. Sa première architecture d'exploration, *skitter*, a été remplacée par sa nouvelle version, *Archipelago (Ark)*, en septembre 2007. Les moniteurs sont divisés en trois groupes ayant chacun 18 - 19 routeurs. Chaque groupe a une liste de destinations contenant au mois une adresse IPv4 dans chaque ensemble /24 de l'espace des adresses valides. Les moniteurs font des traceroutes. Après résolution d'alias on obtient un graphe de routeurs : ITDK¹¹. Nous utilisons deux groupes des données pour le graphe de routeurs :

- ITDK-200304 [CAI03], correspondant aux explorations faites avec *skitter* entre le 21 avril et le 8 mai 2003.
- ITDK-201007 [CAI], explorations faites par *Ark* entre le 11 et 26 juillet 2007.

Graphes des AS

CAIDA : Comme dans le cas des routeurs, CAIDA obtient un graphe au niveau des AS. Le graphe est obtenu en utilisant les données de traceroute du projet "*Macroscopic Topology Project*" en associant à chaque adresse IP l'AS auquel l'adresse appartient. La procédure est faite à partir des tables de routage BGP du projet Route View [ROU] (pour plus de détails voir [HDFC10]). Des instantanés sont obtenus plusieurs fois par mois [HHA⁺09].

AQUALAB : Le projet AQUALAB [CCP⁺09] utilise traceroute pour découvrir les liens entre les interfaces des routeurs. La nouveauté de ce projet réside dans la liste des moniteurs et destinations qui correspondent aux noeuds d'un réseau pair-à-pair. Entre décembre 2007 et septembre 2008, des paquets sondes ont été envoyés depuis 992.000 clients appartenant à 3.700 AS différents. Par la suite, les adresses IP ont été associées aux AS correspondants par l'utilisation des tables BGP du projet Route View.

9. "Internet Group Management Protocol"

10. "Cooperative Association for Internet Data Analysis" : [HTTP://WWW.CAIDA.ORG/](http://www.caida.org/)

11. "Internet Data Kit"

Chapitre III. Mesure de l'arborescence d'Internet

			Graphe			Plus grande composante bi-connecte		
			$ V $	Degré moyen	β	$ V $	Degré moyen	β
AS	AQUALAB 12/2007-	09/2008	31847	9,00	2,18	25341	10,80	2,18
	CAIDA	12/2010	29797	5,31	2,16	17559	7,49	2,16
	DIMES	12/2010	29542	6,84	2,12	21296	8,72	2,12
	UCLA	12/2010	37450	6,65	2,14	25271	8,73	2,14
Routeurs	CAIDA router	04/2003	192.244	6,36	(2,93)	132,367	8,17	(2,96)
	CAIDA router	07/2010	3.360.982	2,93		1.644.761	3,84	2,18
	mrinfo	09/2008	8.636	2,72	3,26	1705	3,60	3,71
AS interne	1221 Telstra (Australia)		2669	2,*38	2,45	246	6,07	2,46
	1239 Sprintlink (US)		7337	2,70	2,37	1054	6,70	2,77
	1755 Ebone (Europe)		295	3,68	2,86	178	4,76	3,24
	2914 Verio (US)		4670	3,26	2,59	1644	5,54	2,76
	3257 Tiscali (Europe)		411	3,18	2,77	166	4,81	2,97
	3356 Level3 (US)		1620	8,32	2,39	729	16,03	2,43
	3967 Exodus (US)		375	4,53	2,85	254	5,33	3,22
	4755 VSNL (India)		41	3,32	2,29	22	3,91	2,33
	7018 AT&T (US)		9430	2,48	2,65	1199	5,54	2,89

Tableau III.1 – Caractéristiques basiques des données utilisées.

DIMES : Le projet DIMES [SS05] utilise une exploration active depuis environ 1000 moniteurs volontaires. Chaque semaine un instantané est généré à partir des explorations et l'association aux AS via les tables BGP.

UCLA Le projet Internet Topology Collection¹² [MRWK03] dépend de l'Université de Californie à Los Angeles (UCLA). Il réunit différentes sources d'exploration passive (Route Views, RIPE-RIS, Abilene, CERNET BGP View) et active (Packet Clearing House, UCR, traceroute.org, Route Server Wiki). Depuis janvier 2004, il actualise ses instantanés quotidiennement.

Le tableau III.1 montre, pour chaque source, un exemple d'instantané avec des statistiques basiques : sa taille, son degré moyen et l'exposant β de loi de puissance sur la distribution des degrés. Ces paramètres sont mesurés pour le graphe complet et pour sa composante bi-connecte la plus grande, la raison de ce choix est algorithmique et il sera détaillée par la suite. L'exposant des lois de puissance est obtenu via une régression linéaire sur la fonction de répartition complémentaire de la distribution des degrés.

2.2 Évolution des graphes des AS

La figure III.3, montre l'évolution du graphe des AS dans l'année 2010. Chaque point correspond à un mois d'observations agrégées. Nous voyons que dans toutes les sources la taille du graphe augmente lentement. Les effets de l'agrégation peuvent être vus dans la figure III.4. Elle montre le nombre des AS et le nombre d'arêtes découvertes entre le début de l'année et la fin du mois en abscisse. Nous remarquons que tandis que la quantité des AS ($|V|$) est relativement constante, le nombre des arêtes

12. [HTTP://IRL.CS.UCLA.EDU/TOPOLOGY/](http://irl.cs.ucla.edu/topology/)

III.3 Largeur Arborescente (*treewidth*)

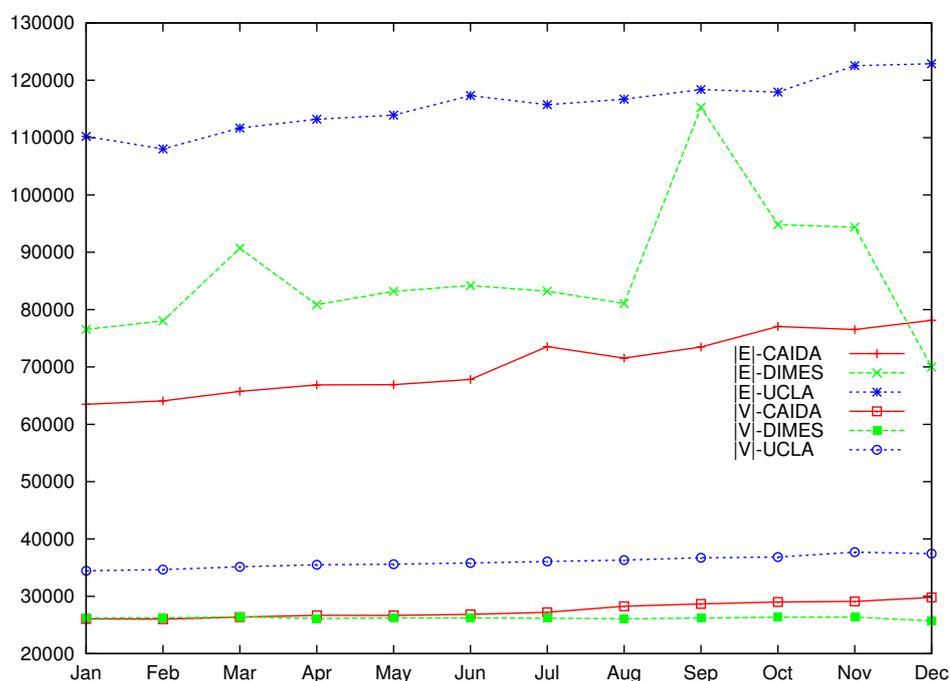


Figure III.3 – Taille des instantanés mensuels pour l’année 2010

($|E|$) découvertes augmente rapidement. Ce comportement témoigne le caractère très dynamique du réseau avec plusieurs actualisations des accords de trafic entre AS. Un comportement similaire a été décrit pour le graphe des routeurs dans [MOVL09] pour l’exploration active depuis un moniteur vers la même liste de destinations.

3 Largeur Arborescente (*treewidth*)

3.1 Définition

Une décomposition arborescente d’un graphe $G = (V, E)$ est composée d’un arbre $T = (I, F)$, et d’une famille de sous-ensembles de V $\{V_i \subseteq V | i \in I\}$. On appelle ces sous-ensembles les “sacs” de la décomposition. L’arbre T et l’ensemble des sacs $\{V_i | i \in I\}$ doivent satisfaire les conditions suivantes :

- Chaque sommet du graphe G appartient à au moins un sac V_i .
- Pour chaque arête e de G , il existe au moins un sac V_i contenant ses extrémités.
- La collection des sacs qui contiennent un sommet donné induit un sous-arbre connexe de T .

La *largeur* d’une décomposition arborescente $(T, \{V_i\})$ est la taille du plus grand sac moins un : $\text{largeur}(T, \{V_i\}) = \max_i |V_i| - 1$

La **largeur arborescente** du graphe G est la plus petite largeur possible parmi toutes

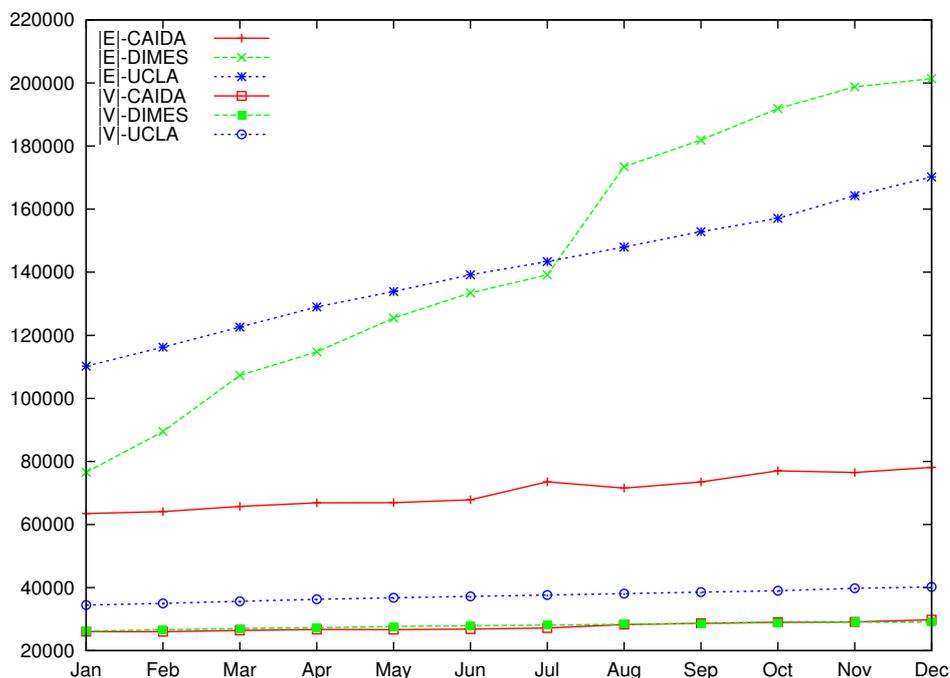


Figure III.4 – Graphes des AS agrégés depuis Janvier 2010.

les décompositions arborescentes possibles de G . Un graphe connexe est de largeur arborescente 1 si et seulement s'il est un arbre. On remarque qu'une décomposition arborescente de G induit de façon naturelle une décomposition arborescente de tout sous-graphe H de G . On a donc $\text{largeur}(H) \leq \text{largeur}(G)$. En particulier, une borne inférieure de la largeur de H est aussi une borne valable pour G . Cette propriété nous permet garantir des bornes inférieures même avec des instantanés incomplets d'Internet.

La largeur arborescente peut être vue comme une mesure de *connectivité* globale d'un graphe. Par exemple, la clique de n sommets a largeur $n - 1$. Un graphe peu dense (peu d'arêtes) peut avoir une largeur élevée (c'est le cas par exemple d'une grille carrée à n sommets, qui a une largeur \sqrt{n} et $O(n)$ arêtes) donc on mesure bien une propriété topologique et non purement de densité.

Mesurer la largeur arborescente d'un graphe est un problème NP-dur. Cependant il existe des heuristiques pour obtenir des bornes supérieures et inférieures¹³.

13. On a écrit des programmes basés sur ces de l'équipe de Bodlaender disponibles sur [HTTP://WWW.TREEWIDTH.COM/](http://www.treewidth.com/)

3.2 Largeur arborescente du graphe des AS

Nous calculons des bornes inférieures de la largeur arborescente pour les instantanés des graphes des AS pour chaque source et leur évolution dans le temps. Nous utilisons les graphes agrégés du mois de décembre pour chaque année (sauf pour 2011 où on utilise le mois de février). On ajoute deux graphes : INTER et UNION correspondant à l'union des sources (tous les liens) et leur intersection (lien trouvés par toutes les sources).

La figure III.5 montre les résultats de notre calcul. Comme nous pouvons espérer, les instantanés avec une densité plus élevée (tableau III.1) ont une borne plus grande pour la largeur arborescente. Le fait le plus remarquable est que la largeur arborescente est très élevée pour toutes les instantanés, en montrant un grand niveau de connexions dans les graphes. En effet, les valeurs obtenues sont proches de la racine carrée du nombre de sommets (tableau III.2), ce qui correspond à la largeur arborescente d'une grille carrée ayant le même nombre de sommets.

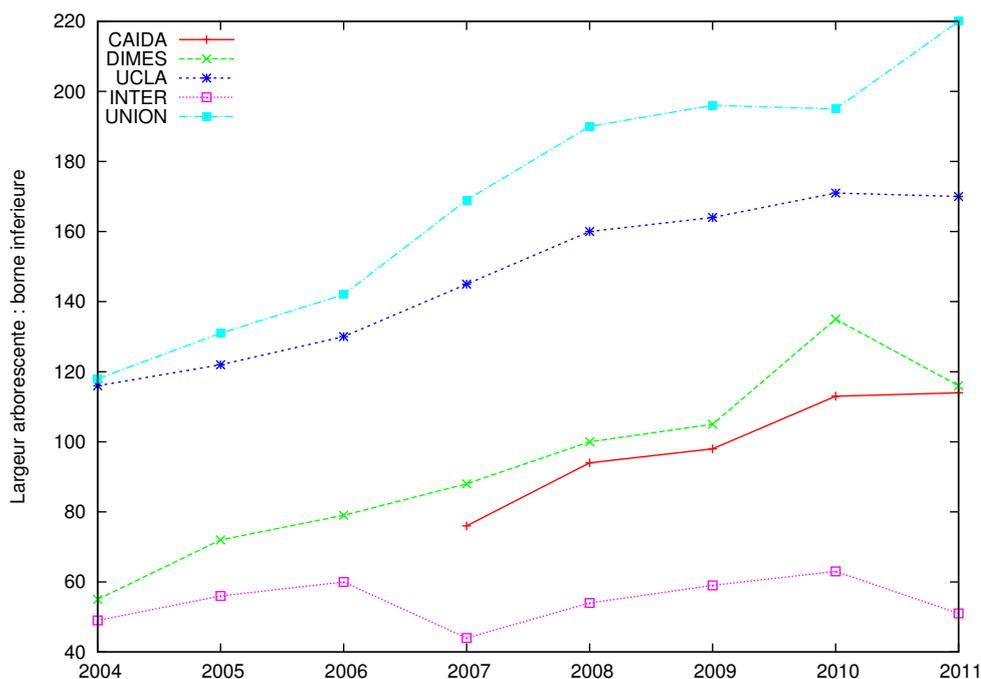


Figure III.5 – Évolution de la largeur arborescente des graphes des AS.

3.3 Largeur Arborescente du graphe de routeurs

La figure III.6 et le tableau III.3 montrent les bornes calculées pour les graphes des routeurs et les instantanés de la topologie interne des AS.

Graphe		tw	$\sqrt{ V }$
AQUALAB	12/2007-09/2008	236	178
CAIDA	12/2010	113	162
DIMES	12/2010	135	156
UCLA	12/2010	171	177
UNION	12/2010	195	179
INTERS.	12/2010	63	146

Tableau III.2 – Bornes de la largeur arborescente pour les graphes des AS.

Graphe	Borne inférieure	Borne supérieure	Grille
router level 03/2004	372	-	363
mrinfo (router) 09/2008	11	48	41
1221 Telstra (Austr.)	9	10	15
1239 Sprintlink (US)	29	55	32
1755 Ebone (Europe)	7	8	13
2914 Verio (US)	24	35	40
3257 Tiscali (Europe)	11	14	27
3356 Level3 (US)	54	137	11
3967 Exodus (US)	8	11	15
4755 VSNL (India)	4	4	4
7018 AT&T (US)	18	26	34

Tableau III.3 – Bornes de la largeur arborescente pour les graphes des routeurs

Comme dans le cas des graphes des AS, il est intéressant de remarquer que pour presque tous nos graphes, la largeur arborescente est proche de celle d'une grille carrée avec le même nombre de sommets. Cela montre la forte connectivité d'Internet. L'unique exception est l'AS3356 (Level3) : on suppose que cela est dû au déploiement de MPLS (*Multiprotocol Label Switching*) sur cet AS.

3.4 Une conséquence : le cœur du réseau

La largeur arborescente est un paramètre qui capture la connectivité globale d'un graphe dans la mesure où elle donne la quantité de sommets qu'il faut supprimer afin de réduire la connectivité de manière significative. Plus précisément, un sous-ensemble S de sommets est k -lié si pour tout sous-ensemble X contenant moins de k sommets il existe une composante connexe de $G - X$ qui contient au moins la moitié de S . En d'autres termes, la suppression de moins de k sommets ne peut pas déconnecter "beaucoup" S . La *ligature* d'un graphe est le plus grand k tel qu'il existe un ensemble k -lié.

III.3 Largeur Arborescente (*treewidth*)

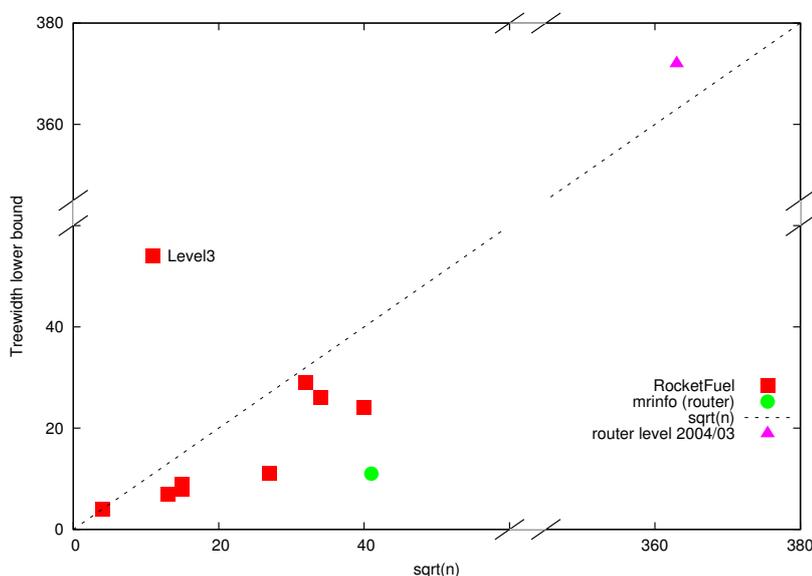


Figure III.6 – Bornes inférieures de la largeur arborescente pour les graphes des routeurs en comparaison avec \sqrt{n} .

L'ensemble S précédemment défini peut être vu comme un *cœur* du graphe, qu'on ne peut pas déconnecter globalement par la suppression de moins de k sommets : plus de la moitié du cœur reste connexe. La ligature est une mesure de la tolérance aux pannes du cœur.

La largeur arborescente et la ligature sont liées par le théorème suivant :

Théorème III.1. [Bai97] *Pour tout graphe G ,*

$$ligature(G) \leq tw(G) + 1 \leq 2 \cdot ligature(G).$$

Nous avons choisi d'estimer la valeur de la largeur arborescente au lieu de la ligature parce qu'il existe des heuristiques efficaces pour obtenir des bornes de la largeur arborescente. Une autre raison de notre choix est que les bornes ainsi obtenues sont valables pour Internet, puisque nos instantanés sont des sous-graphes du graphe d'Internet.

Le tableau III.2 donne les bornes inférieures pour la largeur arborescente (et donc pour la ligature). Par exemple on peut déduire que la ligature du graphe des AS selon UCLA est au moins 86, autrement dit, qu'il existe un cœur des AS qui reste globalement connexe même si 86 AS tombent en panne. Cela donne aussi une mesure de tolérance aux pannes d'Internet.

4 Hyperbolicité

Comme pour la largeur arborescente, la δ -hyperbolicité mesure la ressemblance entre un graphe et un arbre. Les arbres ont hyperbolicité $\delta = 0$, et pour les graphes triangulés, qui ont une structure arborescente, $\delta = 1$. Cependant nos deux notions d'arborescence ne sont pas liées : une grande clique a une largeur arborescente élevée mais est 0-hyperbolique ; inversement dans un grand cycle la largeur arborescente est 2 mais l'hyperbolicité est grande. Un cas différent est celui de la grille à $n \times m$ sommets, pour laquelle la δ -hyperbolicité et la largeur ont la même valeur ($\min(n, m)$) et qui est loin d'être un arbre pour ces deux notions.

Contrairement au cas de la largeur arborescente, mesurer l'hyperbolicité est un problème polynomial. Mais une implémentation naïve $O(n^4)$ est impraticable pour nos graphes ; nous utilisons des heuristiques plus rapides mais exactes.

4.1 Hyperbolicité dans le graphe d'Internet

Bien que l'hyperbolicité soit une mesure globale (maximum calculé sur tous les quadruplets de sommets), on s'intéresse à sa distribution : on calcule pour chaque quadruplet la valeur de δ associée et on obtient ainsi une distribution sur l'ensemble des quadruplets. En premier lieu, la figure III.7 nous montre que la distribution des quadruplets est stable dans le temps indépendamment de la source.

De plus, pour presque tous les quadruplets, la valeur de l'hyperbolicité est 0 ou 1, puis décroît fortement (Figure III.8). Tous nos graphes se comportent ainsi, sauf le graphe de minfo qui semble avoir une décroissance plus faible aboutissant à un maximum plus élevé. Souvenons nous que cet outil peut seulement détecter des routeurs *multicast* qui peuvent avoir une sous-structure bien différente d'Internet.

Les remarques ci-dessus impliquent que les plus courts chemins entre routeurs (ou entre AS) peuvent être plongés sur un arbre avec forte probabilité et erreur faible. Cela permet l'application de schémas de routage efficaces [CDE⁺]. Krioukov et al. [KFY04, KCFB07] ont déjà travaillé sur de tels schémas.

Nous inférons que l'hyperbolicité des graphes d'Internet est très faible. Une raison possible pour expliquer ce fait est le faible diamètre du graphe ou l'existence de plusieurs paires à petite distance dans les quadruplets (propriété II.13(2)). La figure III.9 montre que la valeur moyenne de l'hyperbolicité ne change pas lorsque les éléments des quadruplets s'éloignent. Un tel comportement est consistant avec une structure arborescente. Par contre, une grille à une comportement complètement différent : les valeurs de l'hyperbolicité augmentent de façon linéaire avec les distances.

5 Arborescence des modèles d'Internet

Nous comparons nos résultats pour le graphe des AS avec différents modèles de génération classiques du graphe d'Internet. Notre motivation est de connaître la pertinence de ces modèles vis-à-vis de paramètres d'arborescence. Nous considérons, comme graphe de référence, un des instantanés mensuels de CAIDA que nous comparons avec les modèles suivants :

- Graphes d'Erdős-Rényi (ER) ayant le même nombre de sommets et degré moyen que le graphe des AS.
- Graphe aléatoire tiré avec une distribution de degré qui suit une loi de puissance de la même valeur ($\beta = 2.19$) que le graphe des AS. Nous utilisons l'algorithme de Latapy et al [VL05].
- Graphe aléatoire (nommé "AS Degree Dist") tiré avec *exactement* la même distribution des degrés que le graphe des AS [VL05].
- Modèle Barabási-Albert [BA99] (BA), graphe construit par *attachement préférentiel* [BA99].

Le tableau III.4 montre les valeurs de la largeur arborescente. Nous voyons que les valeurs des bornes inférieures sont légèrement plus élevées, ce qui suggère que la largeur arborescente d'Internet est plus petite que celle d'un graphe aléatoire.

D'autre part, la distribution de l'hyperbolicité fait apparaître une vraie différence entre les modèles. La figure III.10 montre la proportion des valeurs de l'hyperbolicité ainsi que l'hyperbolicité moyenne en fonction de la plus petite distance entre les quadruplets.

De façon remarquable, les modèles les plus ajustés sont ces basés sur la distribution de degrés. Par contre, les modèles de BA et de ER ont une structure éloignée d'un arbre de point de vue métrique.

Notre conclusion coïncide avec celle de Tangmunarunkit et al. [TGJ⁺02] : du point de vue de l'arborescence, de simples modèles basés sur les distributions de degrés semblent mieux modéliser «la réalité» que des modèles ad-hoc plus complexes, ce qui est a priori surprenant. Cette distribution semble donc être une propriété liée à la distribution des degrés avec une loi de puissance.

Graphe	Degré moyen	Degré maximum	β	Hyp.	tw
CAIDA AS	6.31	1,815	2.19	2.0	$\in [82, 473]$
Erdős-Rényi	6.34	18	-	2.5	≥ 135
Barabási	6.00	283	2.92	2.0	≥ 130
AS degree dist.	6.31	1,815	2.19	1.5	≥ 110
Power Law	8.97	1,507	2.19	1.5	≥ 150

Tableau III.4 – Paramètres des graphes de modèles.

6 Conclusions

Les mesures présentées ici nous permettent conclure que, d'une part, la largeur arborescente d'Internet est élevée : tous les instantanés ont une largeur arborescente comparable à une grille carrée avec le même nombre de sommets (une grille est un exemple classique de graphe de faible degré et de grande largeur arborescente). De ce point de vue, Internet est loin d'être un arbre : il est beaucoup mieux connecté. Ce résultat donne une estimation de la connectivité d'Internet : il existe un cœur qui ne peut pas être globalement déconnecté en supprimant moins de $\frac{tw}{2}$ AS (où tw dénote la valeur de la largeur arborescente). Ce résultat est valable indépendamment de l'exactitude des mesures du graphe : une vision plus précise peut seulement révéler une borne inférieure plus élevée sur la mesure de la connectivité.

Notre deuxième conclusion est que l'hyperbolicité d'Internet est très faible, ce qui confirme des résultats précédents [ST08, RMK⁺09, NS09]. De ce point de vue, Internet est proche d'un arbre et loin d'une grille.

Finalement, nous comparons nos mesures avec celles obtenues pour des graphes aléatoires générés à partir de modèles classiques d'Internet. Ces modèles présentent une largeur arborescente similaire et une proportions plus grande des quadruplets avec une valeur d'hyperbolicité plus élevée. Les modèles les mieux adaptés sont basés sur la distribution des degrés. Étonnamment, la structure d'Internet semble être représentée assez fidèlement par un tel modèle. Ce fait a été déjà observé [TGJ⁺02]. La concordance est particulièrement frappante pour l'hyperbolicité. Cette observation contredit l'interprétation donnée par des travaux précédents [ST08, RMK⁺09, NS09] indiquant que les faibles valeurs de l'hyperbolicité sont une propriété particulière d'Internet. Au contraire, cette propriété semble être vérifiée a.f.p. par les graphes aléatoires, en particulier ceux de distribution de degrés similaires.

III.6 Conclusions

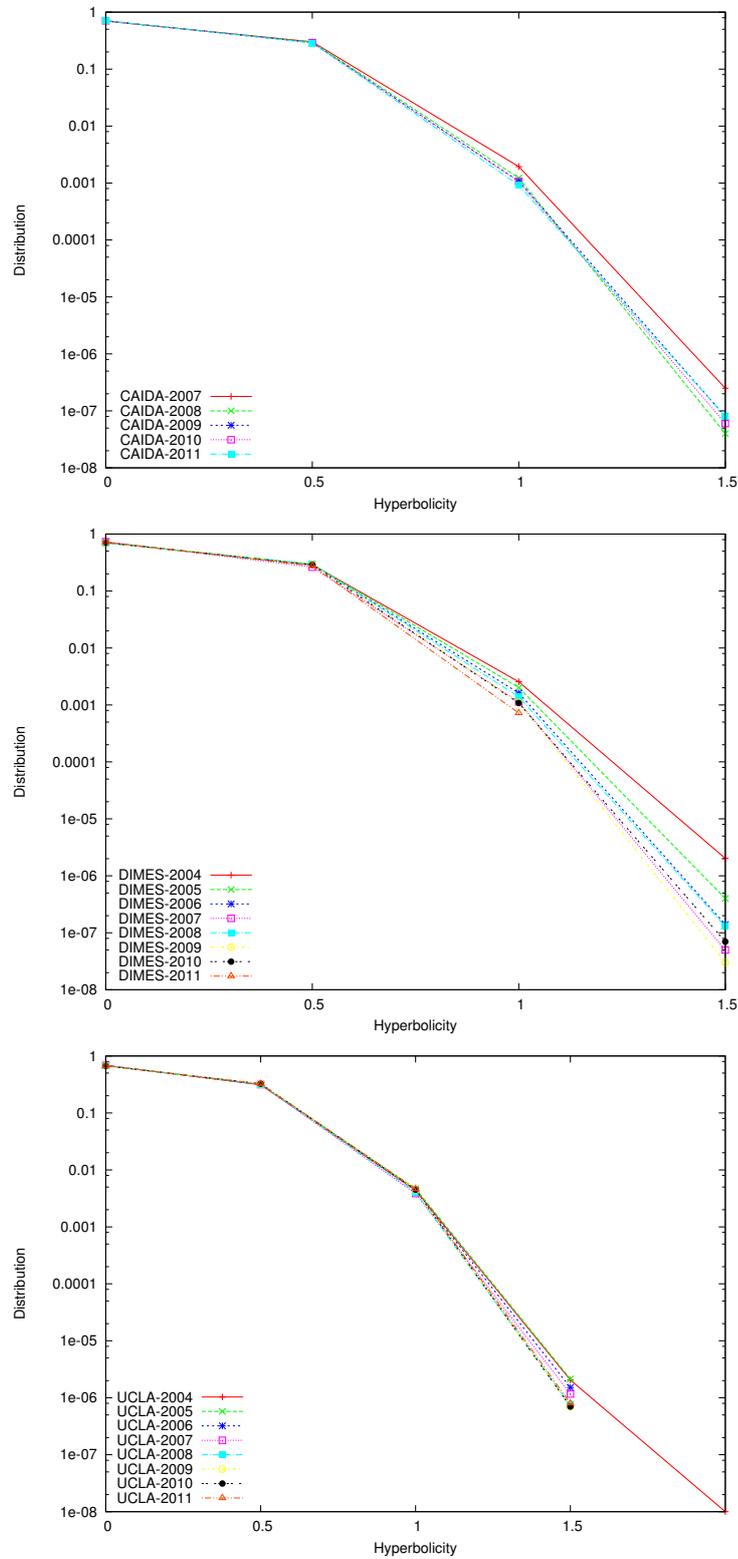


Figure III.7 – Variation de la distribution de l’hyperbolicité pour les graphes des AS (CAIDA, DIMES and UCLA).

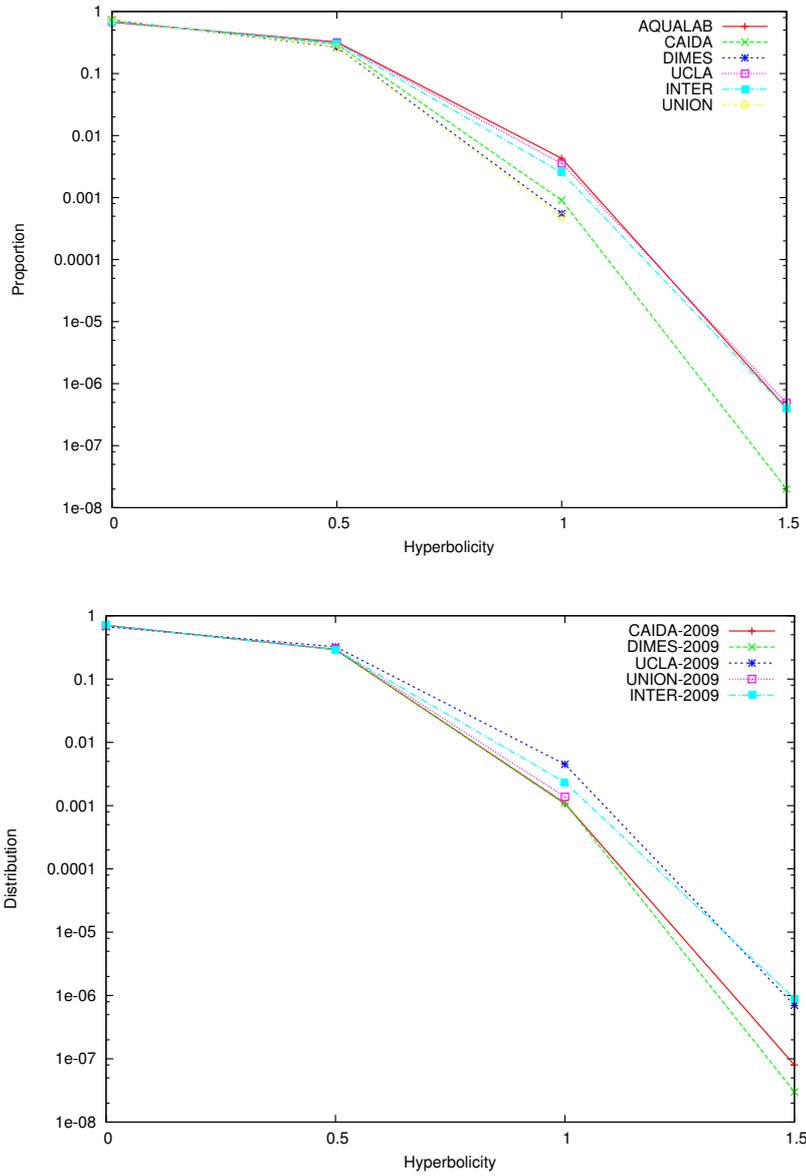


Figure III.8 – Distribution des valeurs de l'hyperbolicité sur tous les quadruplets (Haut : Agrégé par dix mois. Bas : Mensuelle 2009/12)

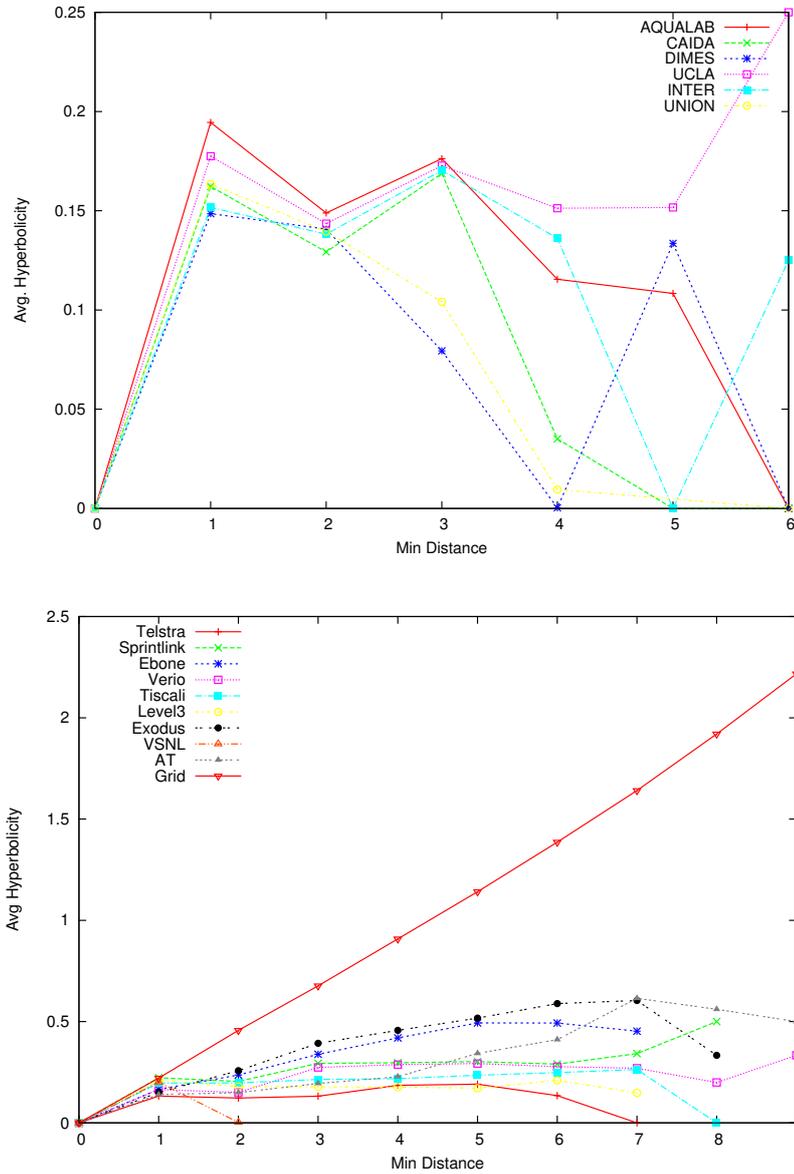


Figure III.9 – Hyperbolicité moyenne en fonction de la distance plus petite entre les éléments du quadruplet. Haut : AS graphe. Bas : graphe de routeurs.

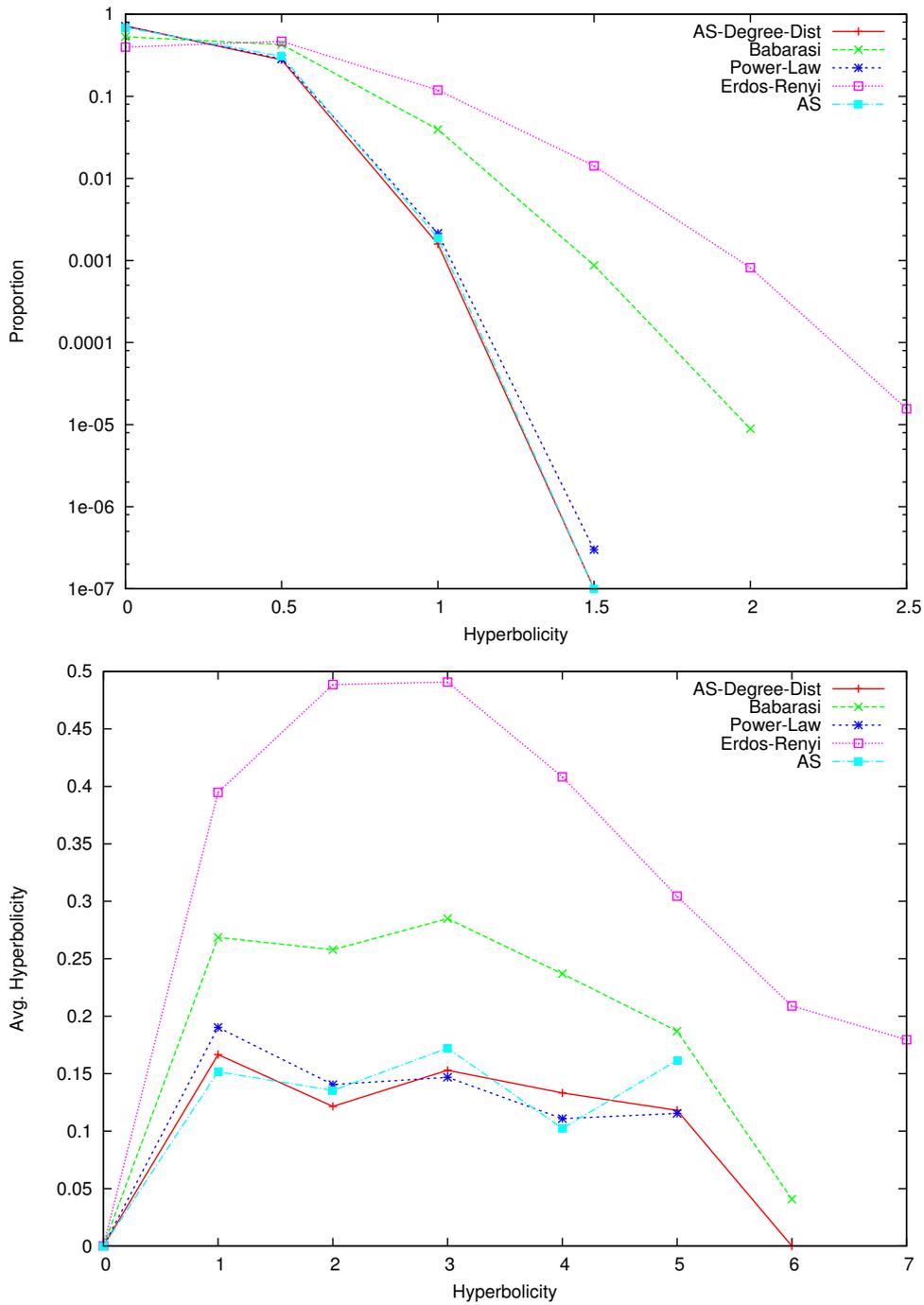


Figure III.10 – Modeles d'Internet. Haut : Distribution de l'hyperbolicité pour les quadruplets. Bas : Hyperbolicité moyenne en fonction de la plus petite distance du quadruplet.

Chapitre IV

Graphes et modularité

Dans ce chapitre on s'intéresse au problème du *partitionnement de données*. Étant donné un ensemble, un partitionnement consiste à trouver une partition en des sous-ensembles appelés *groupes* ou *clusters*, de sorte que les éléments dans un même groupe aient des caractéristiques similaires. La notion de similitude est généralement associée à celle de proximité entre les éléments. C'est pourquoi le partitionnement a lieu généralement dans un espace où on peut définir une distance ou métrique.

Il n'existe pas une définition unique d'un bon partitionnement ; généralement, elle dépend de l'application particulière [SCH07]. Néanmoins, la qualité du partitionnement peut être mesurée par la maximisation (ou la minimisation) d'un paramètre ou par une fonction objectif.

Ce type de problématique a des applications dans un grand éventail de disciplines, comme la biologie [WZ07, OBDJ07] ou la détection de communautés dans les réseaux sociaux [FB07, MRC05].

Le partitionnement a été bien étudié dans le cadre des espaces métriques où la ressemblance entre les éléments est mesurée par la distance entre eux. Dans le contexte de graphes, l'objectif du partitionnement est de trouver une partition dans des sous-graphes *denses*. Autrement dit, les sommets placés dans un même cluster sont bien reliés entre eux, tandis que peu d'arêtes relient les sommets dans des clusters différents (voir figure IV.1).

Parmi les nombreux paramètres associés au partitionnement des graphes, on s'intéresse ici à la *modularité*. Introduite par Newman et Girvan [GN02, NG04], la modularité a attiré beaucoup d'attention dans la littérature avec des applications dans différents types de réseaux [GA05, NEW06, OBDJ07, KA05]. Le calcul d'un partitionnement de modularité maximale est un problème NP-complet [BDG⁺08]. Il a été étudié comme un problème d'optimisation pour lequel différentes heuristiques ont été proposées [DA05, CNM04, GSPA04, BGLL08].

Dans ce chapitre, nous traiterons le problème de la modularité d'un partitionnement dans le cadre théorique des graphes. On montrera que pour plusieurs familles

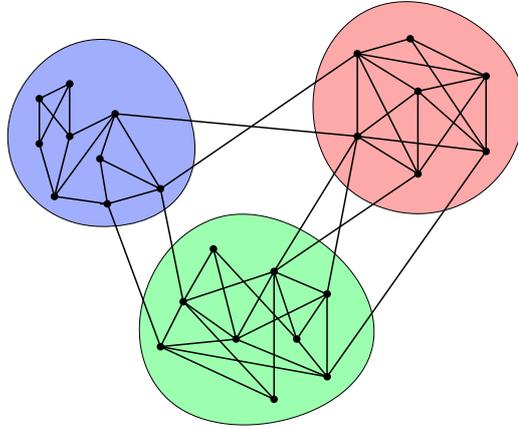


Figure IV.1 – Partitionnement d'un graphe

de graphes classiques, il est possible de trouver des partitionnements avec une valeur de modularité élevée quand la taille du graphe tend vers l'infini. Nous nous intéressons aussi à établir une relation entre les partitionnements dans les espaces métriques et dans les graphes. On trouve que dans un espace métrique de croissance bornée, il est possible de construire un partitionnement avec une valeur de modularité bornée inférieurement en fonction du paramètre de croissance de l'espace (théorème IV.28). Ces résultats montrent que la modularité n'est pas toujours indicative de l'existence de clusters prédéfinis existant, comme on verra plus loin.

Les résultats de ce chapitre ont fait objet des publications [DMSV11A, DMSV11B].

1 Définition et propriétés basiques

1.1 Définition

Afin de simplifier l'exposition, nous commençons par rappeler et introduire quelques notations. Étant donné un graphe $G = (V, E)$, la taille d'un sous-ensemble de sommets $S \subseteq V$ est notée par $|S|$. De plus, le *volume* de S , noté par $\text{vol}(S)$, est la somme des degrés des sommets de S , c'est-à-dire $\text{vol}(S) = \sum_{v \in S} \deg(v)$. D'autre part $|E(S)|$ note le nombre des arêtes internes du sous-graphe induit par S , c'est-à-dire les arêtes qui ont leur deux extrémités dans S .

Un *partitionnement* du graphe G correspond à fournir une partition de l'ensemble $V(G)$ en ensembles disjoints. Un *cluster* est un élément de la partition.

Définition IV.1 ([GN02, NG04]). Étant donné un graphe G et un partitionnement

IV.1 Définition et propriétés basiques

$\mathcal{C} = \{C_1, \dots, C_k\}$, la *modularité* de \mathcal{C} est définie par :

$$\mathcal{Q}(\mathcal{C}) = \sum_{i=1}^k \left[\frac{|E(C_i)|}{|E(G)|} - \frac{\text{vol}(C_i)^2}{\text{vol}(G)^2} \right] \quad (\text{IV.1})$$

La modularité du graphe G , noté $\mathcal{Q}(G)$, est la valeur maximale de la modularité que peut prendre un partitionnement de G .

Une propriété désirable d'un bon partitionnement est que la majorité des arêtes relie des sommets qui sont dans le même cluster. En effet, le terme de gauche de l'équation (IV.1) correspond à la proportion des arêtes internes du partitionnement. Ce terme est appelé la *couverture* du partitionnement [GAE04].

Le terme de droite correspond à l'espérance de la proportion d'arêtes internes du même partitionnement, dans un graphe aléatoire avec la même distribution des degrés. En effet, étant donné un partitionnement \mathcal{C} et un cluster $C \in \mathcal{C}$, considérons un graphe aléatoire où chaque sommet a le même degré que dans G . L'espérance du nombre d'arêtes internes de C est proportionnelle à $\sum_{u,v \in C} \deg(u) \deg(v)$. Ainsi, le nombre espéré des arêtes internes du partitionnement dans le graphe aléatoire est proportionnel à $\sum_{i=1}^k \sum_{u,v \in C_i} \deg(u) \deg(v) = \sum_{i=1}^k (\sum_{u \in C_i} \deg(u))^2 = \sum_{i=1}^k \text{vol}(C_i)^2$. La motivation de cette définition est détaillée dans [NEW06].

Pour maximiser le terme de gauche, on serait tenté de choisir un partitionnement avec le moins possible de clusters. Par contre, le terme de droite compense cette tendance puisqu'il se maximise par la création de partitionnements avec plusieurs clusters de petite taille.

Comme chacun des termes a une valeur entre zéro et un, pour tout partitionnement \mathcal{C} , la modularité $\mathcal{Q}(\mathcal{C})$ appartient à l'intervalle $[-1, 1]$. De plus, dans [BDG⁺08] les auteurs montrent que la modularité se trouve toujours dans $[-1/2, 1[$. Pour le prouver, il faut remarquer que la valeur minimum de la modularité est réalisée pour un partitionnement sans arêtes internes, pour lequel le terme de droite est maximum. De plus, d'après la propriété $(x + y)^2 \geq x^2 + y^2$, pour toute paire de nombres réels positifs, le maximum du terme de droite, est réalisé quand le partitionnement consiste en deux clusters. Ainsi, le pire cas vis-à-vis de la modularité, est la partition en deux stables d'un graphe biparti avec une valeur égale à $-\frac{1}{2}$. Finalement, le terme de droite n'est jamais nul, ce qui montre que la modularité de toute partition est strictement inférieure à un. Une seule exception réside dans le cas dégénéré d'un graphe sans arêtes qui par convention a une modularité unitaire. Néanmoins, nous ne considérons pas ce cas.

D'un point de vue algorithmique, le calcul de la modularité d'un partitionnement se fait en temps $O(|V(G)| + |E(G)|)$. Par contre, l'obtention d'un partitionnement de modularité maximal est un problème NP-complet [BDG⁺08].

1.2 Premières propriétés

Nous avons constaté que la modularité n'atteint jamais la valeur maximum. La proposition suivante donne une borne supérieure sur la modularité d'un graphe.

Proposition IV.2. *Soit G un graphe de degré maximal Δ , alors $\mathcal{Q}(G) < 1 - \frac{\Delta^2}{4m^2}$.*

Démonstration. Soit $v^* \in V(G)$ un sommet de degré maximal. Étant donné un partitionnement, on considère le cluster C qui contient le sommet v^* . Ainsi, le terme de droite est plus grand ou égal à $\frac{\Delta^2}{4m^2}$. De plus, si C contient aussi un voisin de v^* , alors le terme de droite est strictement plus grand que $\frac{\Delta^2}{4m^2}$; dans le cas contraire, la couverture du partitionnement est strictement plus petite que un et son volume est au moins Δ . Donc, tout partitionnement a une modularité strictement inférieure à $1 - \frac{\Delta^2}{4m^2}$. \square

La propriété suivante nous permet trouver une famille de partitionnements de modularité maximale,

Proposition IV.3 ([BDG⁺08]). *Il existe un partitionnement avec modularité maximale tel que tout cluster est connexe.*

Démonstration. Soit \mathcal{C} un partitionnement de modularité maximale contenant un cluster non connexe C . Nous allons prouver que la modularité ne décroît pas par la séparation du cluster C en un cluster par composante connexe induite.

Sans perte de généralité, on suppose $C = C_1 \uplus C_2$. On note par \mathcal{C}' le partitionnement qui consiste à remplacer le cluster C par ses composantes connexes, c'est-à-dire $\mathcal{C}' = \mathcal{C} \setminus \{C\} \cup \{C_1, C_2\}$. Ainsi, la couverture reste invariante, et donc $\mathcal{Q}(\mathcal{C}') - \mathcal{Q}(\mathcal{C}) = (\text{vol}(C)^2 - \text{vol}(C_1)^2 - \text{vol}(C_2)^2)/(4m^2) \geq 0$. \square

La proposition 1.2 permet borner ma modularité en fonction du degré maximum, nous pouvons aussi la borner en fonction du nombre de clusters :

Lemme IV.4. *Pour tout partitionnement \mathcal{C} avec k clusters, $\mathcal{Q}(\mathcal{C}) \leq 1 - \frac{1}{k}$.*

Démonstration. D'après la convexité de la fonction $f(x) = x^2$ sur l'ensemble des nombres réels, pour tout ensemble $\{x_1, \dots, x_k\} \subset \mathbb{R}$

$$\left(\frac{1}{k} \sum_{i=1}^{i=k} x_i \right)^2 \leq \frac{1}{k} \sum_{i=1}^{i=k} x_i^2. \quad (\text{IV.2})$$

Ainsi, $\sum_{i=1}^{i=k} \text{vol}(C_i)^2 \geq \frac{1}{k} \text{vol}(G)^2$. De plus, tout partitionnement a une couverture bornée par un, ce qui montre le résultat. \square

On remarque que l'inégalité du lemme peut être atteinte dans le cas d'un graphe contenant k cliques disjointes de la même taille et un partitionnement dans ses composantes connexes [BDG⁺08, Corollaire 6.4]. Plus généralement, tout graphe qui consiste en k copies disjointes du même sous-graphe, atteint la borne $1 - \frac{1}{k}$ pour le partitionnement contenant comme cluster chaque copie du sous-graphe.

La proposition montre qu'aucun graphe n'a modularité exactement égale à un. Néanmoins, il est possible de trouver une suite de graphes avec une modularité qui converge vers un, lorsque leur taille augmente. D'autre part, le lemme 1.2 montre que, pour construire un partitionnement ayant une modularité proche d'un, il doit avoir nécessairement une grande quantité de clusters tendant asymptotiquement vers l'infini. On peut définir alors la notion de modularité asymptotique :

Définition IV.5. Une famille de graphes a *modularité asymptotique* égale à $\ell \geq 0$, si pour toute suite de graphes de taille croissante dans la famille, la modularité de ces graphes converge vers ℓ .

Nous allons voir que certaines familles de graphes ont une modularité asymptotique égale à un, même quand les graphes dans la famille n'ont pas une structure clairement décomposable. Regardons tout d'abord quelques graphes de modularité nulle.

2 Quelques graphes de modularité zéro

On commence pour montrer deux familles avec une modularité (asymptotique) nulle.

2.1 Étoiles

Théorème IV.6. *L'étoile $K_{1,m}$ avec m rayons a modularité zéro.*

Démonstration. Étant donné un partitionnement de modularité maximale, on note par C_0 le cluster qui contient le sommet central de l'étoile avec a autres sommets. D'après la proposition IV.3 on peut supposer que tout autre cluster, sauf C_0 contient un unique sommet. Donc,

$$\mathcal{Q}(\mathcal{C}) = \sum_{i=1}^k \left[\frac{|E(C_i)|}{|E(G)|} - \frac{\text{vol}(C_i)^2}{\text{vol}(G)^2} \right] = \frac{a}{m} - \frac{(m+a)^2 + (m-a)}{4m^2} = \frac{(m-a)(a-m-1)}{4m^2}.$$

La modularité a un unique maximum pour $a = m$. Ainsi, C_0 est l'unique cluster du partitionnement avec une modularité nulle. \square

Corollaire IV.7. *La famille des arbres contient des graphes de modularité zéro.*

2.2 Cliques

Théorème IV.8. *La clique avec n sommets, K_n a modularité zéro.*

Démonstration. Un calcul simple [BDG⁺08], montre que pour un partitionnement $\mathcal{C} = \{C_1, \dots, C_k\}$ de la clique K_n .

$$\mathcal{Q}(\mathcal{C}) = -\frac{1}{n-1} + \frac{1}{n^2(n-1)} \sum_{i=1}^k |C_i|^2.$$

Le maximum est alors atteint par un partitionnement avec un unique cluster. □

3 Graphes de modularité asymptotique un

Pour que la modularité asymptotique d'une famille de graphes soit égale à un, deux conditions s'imposent. D'abord, par le Lemme 1.2, le nombre de clusters du partitionnement maximal doit croître vers l'infini. De plus, la couverture doit converger vers un, ou, de façon équivalente, la proportion des arêtes entre les clusters doit converger vers zéro. Ces conditions motivent la notion de *graphes décomposables*. Cette définition prend en compte les trois paramètres qui interviennent dans la modularité.

3.1 Graphes décomposables

Définition IV.9. Un graphe G est (k, c, e) -décomposable s'il existe un partitionnement en k clusters tel que

- chaque cluster a un volume borné supérieurement par $c \cdot \frac{\text{vol}(G)}{k}$ et,
- le nombre d'arêtes entre les clusters est borné supérieurement par $e \cdot |E(G)|$.

Intuitivement, un graphe est décomposable s'il peut être partitionné en k clusters de volumes similaire (à un facteur près), de sorte que la proportion d'arêtes entre les clusters reste bornée. Ces conditions permettent de garantir une borne pour la modularité d'un graphe décomposable :

Lemme IV.10. *Si un graphe G est (k, c, e) -décomposable, on a $\mathcal{Q}(G) \geq 1 - e - \frac{c^2}{k}$.*

Démonstration. Soit $\mathcal{C} = \{C_1, \dots, C_k\}$ un partitionnement qui rend G (k, c, e) -décomposable, donc

$$\mathcal{Q}(\mathcal{C}) = \sum_{i=1}^k \left(\frac{|E(C_i)|}{|E(G)|} - \frac{\text{vol}(C_i)^2}{\text{vol}(G)^2} \right) \geq \frac{(1-e)|E(G)|}{|E(G)|} - \sum_{i=1}^k \frac{(c \cdot \text{vol}(G)/k)^2}{\text{vol}(G)^2} \geq 1 - e - \frac{c^2}{k}.$$

□

IV.3 Graphes de modularité asymptotique un

Par la suite, on verra que plusieurs familles classiques de graphes sont décomposables.

3.2 Grilles

Soit G une grille carrée de taille $n = a \times a$. L'idée de la décomposition est de faire un pavage en sous-grilles de taille inférieure. Soit $b \leq a - 2$, on construit le partitionnement \mathcal{C}_b avec des sous-grilles de taille $n' = b \times b$ (sauf probablement pour la dernière ligne et la dernière colonne). Le partitionnement \mathcal{C}_b a une taille $k = \lceil a/b \rceil^2$ et chaque cluster a un volume inférieur à $4b^2$.

Lemme IV.11. *Une grille de taille $n = a \times a$ est $(k, 4, \frac{\sqrt{k}-1}{a-1})$ -décomposable pour tout $k = \lceil a/b \rceil^2$ avec $b < a - 2$.*

Démonstration. Considérons le partitionnement \mathcal{C}_b par les sous-grilles de taille au plus $n' = b \times b$. Chaque ligne (ou colonne) est coupée $\lceil a/b \rceil - 1$ fois. Donc, il existe $2a(\lceil a/b \rceil - 1)$ arêtes entre les clusters. Ainsi, $e = \frac{2a(\lceil a/b \rceil - 1)}{2a(a-1)} = \frac{\sqrt{k}-1}{a-1}$.

D'autre part, pour tout cluster C_i , $\text{vol}(C_i) \leq 4b^2$, $\text{vol}(G) \leq 4a^2$ et $b \leq a - 2$. Donc,

$$k \cdot \text{vol}(C_i) = \lceil a/b \rceil^2 \cdot 4b^2 \leq \frac{(a+b)^2}{b^2} \cdot 4b^2 \leq 4(a+b)^2 \leq 4(2(a-1))^2 \leq 16 \text{vol}(G)$$

□

Théorème IV.12. *Un grille carrée avec n sommets a modularité $1 - \mathcal{O}(n^{-1/3})$.*

Démonstration. D'après le Lemme IV.10,

$$\mathfrak{Q}(G) \geq 1 - \frac{\sqrt{k}-1}{a-1} - \frac{16^2}{k}.$$

En choisissant $b = \sqrt[3]{n}$, on a $b = a^{2/3}$ et $k = \lceil a/b \rceil^2 = \lceil a^{1/3} \rceil^2$. Ainsi,

$$\mathfrak{Q}(G) \geq 1 - \frac{\sqrt{\lceil a^{1/3} \rceil^2} - 1}{a-1} - \frac{16^2}{\lceil a^{1/3} \rceil^2} = 1 - \mathcal{O}(a^{-2/3}) = 1 - \mathcal{O}\left(\frac{1}{\sqrt[3]{n}}\right) \quad \square$$

Donc, la famille de grilles carrés a modularité asymptotique un.

3.3 Tores multidimensionnels

Étant donné un vecteur $\vec{p} \in \mathbb{N}^d$ on définit le *tore d -dimensionnel* $G_{\vec{p}}$ comme le produit cartésien $C_{p_1} \times \dots \times C_{p_d}$ où C_k note le cycle avec k sommets. Le tore $G_{\vec{p}}$ a $\prod_{i=1}^d p_i$ sommets et son volume est $2dn$.

Lemme IV.13. *Le tore d -dimensionnel est $(b, 2, \frac{2b}{dn^{1/d}})$ -décomposable.*

Démonstration. Soit p la dimension la plus grande du tore $p = \max_{i=1}^d p_i$, alors $p \geq n^{1/d}$. Le partitionnement \mathcal{C}_b est construit en coupant le tore en b tranches dans la dimension la plus grande. Plus formellement $C_i = \{x \in G_{\vec{p}}, x_p \in [i \cdot \lceil p/b \rceil, (i+1) \cdot \lceil p/b \rceil [\}$. Les sommets dans le tore avec une même coordonnée, forment un $(d-1)$ -hyperplan de taille n/p . Donc, le nombre d'arêtes sortant d'un cluster est égal ou inférieur à $2n/p$. Ainsi, $e \leq \frac{2bn/p}{dn} \leq \frac{2b}{dn^{1/d}}$.

D'autre part, chaque cluster C_i contient moins de $n/p \cdot \lceil p/b \rceil$ sommets. De plus, chaque sommet a $2d$ voisins et $b < p$, donc

$$\text{vol}(C_i) \leq 2d \cdot \left(\frac{n}{b} + \frac{n}{p} \right) \leq \frac{4dn}{b} \leq \frac{2}{b} \text{vol}(G)$$

□

Théorème IV.14. *Un tore d -dimensionnel avec n sommets a modularité $1 - \mathcal{O}(n^{-1/2d})$.*

Démonstration. D'après le Lemme IV.10, $\mathcal{Q}(G) \geq 1 - \frac{2b}{dn^{1/d}} - \frac{2^2}{b}$. Le maximum de cette fonction est atteint lorsque $\frac{2n^{-1/d}}{d} = \frac{4}{b^2}$, soit $b = \sqrt{2dn^{1/2d}}$. Ainsi, la modularité pour cette valeur est $\mathcal{Q}(\mathcal{C}_b) \geq 1 - \frac{4\sqrt{2}}{\sqrt{d}} n^{-1/2d} \geq 1 - \mathcal{O}(n^{-1/2d})$ □

3.4 Hypercubes

Un hypercube d -dimensionnel consiste dans un ensemble de $n = 2^d$ sommets où chaque sommet est identifié avec un nombre binaire de d chiffres. Deux sommets sont voisins si les nombres associés diffèrent d'exactly un bit. Le volume d'un d -hypercube est donc égal à $d2^d$.

Lemme IV.15. *Un hypercube d -dimensionnel est $(2^b, 1, \frac{b}{d})$ -décomposable.*

Démonstration. Pour $b < d$, le partitionnement \mathcal{C}_b par préfixes de longueur b , contient 2^b clusters. Chaque cluster est associé à un nombre a de longueur b . Le cluster C_a contient tous les sommets dont les nombres ont a comme préfixe.

Chaque sommet a d voisins, dont b sont à l'extérieur de son cluster et $d - b$ sont des voisins intérieurs, donc $e = b/d$. D'autre part, chaque cluster dans \mathcal{C}_b contient 2^{d-b} sommets, ainsi son volume est égal à $d2^{d-b}$. D'où, $\text{vol}(C_i)/\text{vol}(G) = 2^{-b} = \frac{1}{k}$ et donc $c = 1$. \square

Théorème IV.16. *Un hypercube d -dimensionnel a modularité $1 - \mathcal{O}\left(\frac{\log d}{d}\right) = 1 - \mathcal{O}\left(\frac{\log \log n}{\log n}\right)$.*

Démonstration. En utilisant le Lemme IV.10, $\mathcal{Q}(G) \geq 1 - \frac{b}{d} - \frac{1}{2^b}$. Cette fonction atteint son maximum pour $b = \log_2(d \ln(2))$ avec une valeur de modularité $\mathcal{Q}(G) \geq 1 - \frac{\log_2(d \ln(2))}{d} - \frac{1}{d \ln(2)} = 1 - \mathcal{O}\left(\frac{\log \log(n)}{\log(n)}\right)$. \square

4 Graphes de degré moyen borné

Dans cette section, nous étudions les graphes éparés, c'est-à-dire les familles de graphes avec degré moyen constant, indépendamment de la taille du graphe. On montrera que la modularité asymptotique dépend de ce degré maximum. La preuve est basée sur la construction d'un arbre couvrant, nous commençons donc avec ce cas.

4.1 Arbres de degré maximal borné

Étant donné un arbre T , la suppression de toute arête de l'arbre le coupe en deux composantes connexes. Une *arête centrale* e , est une arête qui maximise la plus petite composante connexe de $T - e$.

Définissons l'algorithme *décompose-glouton $_{\leq h}$* . L'algorithme reçoit comme argument une forêt F . Il s'arrête lorsque toutes les composantes connexes de F ont une taille inférieure ou égale à h . Tant que ce n'est pas le cas, il cherche une composante connexe de taille plus grande que h et il enlève son arête centrale e , et effectue $F := F - e$. Le partitionnement \mathcal{C}_h est le résultat d'une exécution de *décompose-glouton $_{\leq h}$* sur un arbre T .

Lemme IV.17. *Soit T un arbre avec degré maximum inférieur ou égal à Δ et $\mathcal{C}_h = \{C_1, \dots, C_k\}$ le partitionnement obtenu par l'algorithme *décompose-glouton $_{\leq h}$* . On a, pour tout $i \in \{1, \dots, k\}$, $\frac{h}{\Delta} \leq |C_i| \leq h$.*

Démonstration. Par la définition de l'algorithme *décompose-glouton $_{\leq h}$* tous les clusters ont une taille au plus h .

D'autre part, soient T_0 un arbre de taille n_0 et e son arête centrale. Nous notons par s la taille de la composante connexe la plus petite de $T_0 - e$, clairement $s \leq n_0/2$.

Chapitre IV. Graphes et modularité

Soit x l'extrémité de e dans la composante la plus grande de $T_0 - e$. Pour toute arête e' incidente à x , la composante de $T_0 - e'$ qui ne contient pas le sommet x , doit avoir une taille inférieure à s . Sinon, e ne serait pas une arête centrale, ce qui mène à une contradiction. Donc, comme le degré de x est inférieur à Δ , on a $n_0 \leq \Delta \cdot s + 1$, et donc $s \geq \frac{n_0 - 1}{\Delta} \geq \frac{h}{\Delta}$. \square

Le lemme IV.17 montre que les clusters du partitionnement \mathcal{C}_h ont la même taille à un facteur Δ près. De plus, comme la somme des tailles des clusters est égale à n , l'algorithme *décompose-glouton* $_{\leq h}$ partitionne l'arbre T en $\frac{n}{h} \leq k \leq \frac{\Delta n}{h}$ clusters.

Lemme IV.18. *Pour tout arbre T de degré maximum borné par Δ , il existe $k > 0$ tel que T est $(k, \Delta^2, \frac{k-1}{n-1})$ -décomposable.*

Démonstration. Tout partitionnement d'un arbre en k composantes connexes laisse $k - 1$ arêtes entre les clusters et donc $e = \frac{k-1}{n-1}$. De plus, tout sommet a un degré inférieur à Δ , donc $\text{vol}(C_i) \leq \Delta \cdot h \leq \Delta^2 \cdot \frac{n}{k} \leq \Delta^2 \cdot \frac{2(n-1)}{k} \leq \Delta^2 \cdot \frac{\text{vol}(T)}{k}$. \square

Théorème IV.19. *La famille des arbres de degré maximum borné par $\Delta = o(\sqrt[5]{n})$ a modularité asymptotique égale à un.*

Démonstration. Nous avons $\frac{n}{h} \leq k \leq \frac{\Delta n}{h}$. Le Lemme IV.10 garantit

$$\mathcal{Q}(T) \geq 1 - \frac{k-1}{n-1} - \frac{\Delta^4}{k} \geq 1 - \frac{(\Delta n/h) - 1}{n-1} - \frac{\Delta^4}{(n/h)} \geq 1 + \frac{1}{n-1} - \frac{\Delta n}{h(n-1)} - \frac{\Delta^4}{n} h$$

Pour maximiser cette expression, on prend la dérivée de l'expression en h : $\frac{\Delta n}{h^2(n-1)} - \frac{\Delta^4}{n} = 0$. Ce qui donne $h = \frac{n}{\Delta \sqrt{\Delta(n-1)}}$. Le partitionnement obtenu par l'algorithme *décompose-glouton* $_{\leq \frac{n}{\Delta \sqrt{\Delta(n-1)}}$ a une modularité,

$$\mathcal{Q}(T) \geq 1 + \frac{1}{n-1} - \frac{\Delta n}{(n-1)} \cdot \frac{\Delta \sqrt{\Delta(n-1)}}{n} - \frac{\Delta^4}{n} \cdot \frac{n}{\Delta \sqrt{\Delta(n-1)}} \geq 1 + \frac{1}{n-1} - \frac{2\Delta^{2.5}}{\sqrt{n-1}}.$$

Ainsi, si $\Delta = o(\sqrt[5]{n})$, alors on a $\frac{2\Delta^{2.5}}{\sqrt{n-1}} = o(1)$. \square

Remarque IV.20. Les arbres peuvent avoir une modularité strictement inférieure à un. Par exemple, les étoiles $K_{1,n}$ ont une modularité asymptotique nulle. De plus, étant donnée une famille d'arbres, s'il existe une constante $c > 0$ et une suite $\{T_i\}_{i \in \mathbb{N}}$ dans la famille tel que tout arbre T_i contient un sommet de degré cn ; alors d'après la proposition 1.2, $\mathcal{Q}(T_i) \leq 1 - \frac{cn^2}{4(n-1)^2} < 1 - \frac{c}{4} < 1$.

4.2 Graphes de degré moyen constant

Cette section généralise les résultats obtenus dans la section précédente aux graphes. Nous prouverons que si le degré moyen d'une famille de graphes est constant, alors la modularité asymptotique est bornée inférieurement à condition que le degré maximum soit borné par une fonction de la taille du graphe.

L'idée principale de la généralisation est l'utilisation du même algorithme *décompose-glouton*_{≤h} sur un arbre couvrant du graphe. Soit G un graphe de degré moyen égal à d et degré maximum Δ , nous prenons un arbre couvrant T de G . Clairement, le degré maximum de T est au plus Δ . Nous construisons un partitionnement, noté par \mathcal{C}_h , via l'algorithme *décompose-glouton*_{≤h}.

Le partitionnement \mathcal{C}_h satisfait le lemme IV.17, on a donc pour tout cluster $C \in \mathcal{C}_h$, $\frac{h}{\Delta} \leq |C| \leq h$ et donc, $\text{vol}(C) \leq \Delta|C| \leq \Delta h$. De plus, le nombre k de clusters satisfait $\frac{n}{h} \leq k \leq \frac{\Delta n}{h}$.

Lemme IV.21. *Étant donné un graphe G de degré maximum Δ et de degré moyen égal à d , il existe $k > 0$ tel que G est $(k, \frac{\Delta^2}{d}, 1 - \frac{2(n-k)}{dn})$ -décomposable.*

Démonstration. Parmi les arêtes de G , nous pouvons garantir seulement que les $n - k$ arêtes internes de la forêt obtenue à partir de T sont aussi internes dans le partitionnement de G . Ainsi, $e = 1 - \frac{n-k}{m} = 1 - \frac{2(n-k)}{dn}$. D'autre part, $k \frac{\text{vol}(C_i)}{\text{vol}(G)} \leq k \frac{\Delta h}{dn} \leq \frac{\Delta^2}{d}$. \square

Théorème IV.22. *La famille des graphes connexes de degré moyen d et de degré maximum borné par $\Delta = o(\sqrt[5]{d^3 n})$ a une modularité asymptotique supérieure ou égale à $\frac{2}{d}$.*

Démonstration. D'après le lemme IV.10,

$$\mathcal{Q}(G) \geq 1 - \left(1 - \frac{n-k}{nd}\right) - \left(\frac{\Delta^2}{d}\right)^2 \cdot \frac{1}{k} \geq \frac{2}{d} - \frac{2k}{dn} - \frac{\Delta^4}{d^2 k}.$$

Comme $\frac{1}{k} \leq \frac{h}{n}$ et $k \leq \frac{\Delta n}{h}$, on a :

$$\mathcal{Q}(G) \geq \frac{n}{m} - \frac{\Delta n}{hm} - \frac{\Delta^4 h}{nd^2}.$$

Le maximisation de la modularité en fonction de h satisfait la condition $\frac{2\Delta}{d} \frac{1}{h^2} - \frac{\Delta^4}{nd^2} = 0$. La solution pour cette équation est $h = \frac{\sqrt{2dn}}{\Delta^{1.5}}$. Nous construisons le partitionnement avec l'algorithme *décompose-glouton* _{$\frac{\sqrt{2dn}}{\Delta^{1.5}}$} , ce partitionnement a pour modularité :

$$\mathcal{Q}(G) \geq \frac{2}{d} - \frac{2\Delta}{d} \frac{\Delta^{1.5}}{\sqrt{2dn}} - \frac{\Delta^4}{nd^2} \frac{\sqrt{2dn}}{\Delta^{1.5}} \geq \frac{2}{d} - \frac{2^{1.5} \Delta^{2.5}}{d^{1.5} \sqrt{n}} \geq \frac{2}{d} - o(1)$$

puisque $\Delta = o(\sqrt[5]{d^3 n})$. \square

Remarque IV.23. Dans le cas de arbres, on a $d = 2(n - 1)/n$, ce qui converge vers 2. Théorème IV.19 correspond donc à un cas particulier du théorème IV.22.

4.3 Graphes et loi de puissance

Une famille de graphes suit une *loi de puissance* avec paramètre $\alpha > 0$ si pour tout $k > 0$, la proportion de sommets avec degré supérieur à k est $\mathcal{O}(k^{-\alpha})$. Nous prouverons que si le paramètre α est suffisamment grand, on peut garantir une borne inférieure pour la modularité asymptotique de la famille.

Lemme IV.24. *Dans une famille de graphes connexes suivant une loi de puissance de paramètre $\alpha > 5$ le degré maximum est borné par $\Delta = o(\sqrt[5]{n})$.*

Démonstration. Soit $k = n^\gamma$ pour $\gamma \in (\frac{1}{\alpha}, \frac{1}{5})$, alors la quantité de sommets de degré k est $n\mathcal{O}(k^{-\alpha}) = \mathcal{O}(n^{1-\alpha\gamma}) = o(1)$. Donc, pour n suffisamment grand les graphes de la famille ont degré maximum borné par $\Delta = o(\sqrt[5]{n})$. \square

Comme $o(\sqrt[5]{n}) = o(\sqrt[5]{cn})$ pour toute constante $c > 0$, nous obtenons :

Théorème IV.25. *Une famille de graphes avec une loi de puissance de paramètre $\alpha > 5$ et degré moyen d a modularité asymptotique supérieure ou égale à $\frac{2}{d}$.*

5 Graphe des boules unitaires dans espaces de croissance bornée

Nous étudions dans cette partie la modularité des graphes générés à partir d'un espace métrique. Cette construction se fait en utilisant le graphe de boules unitaires. Étant donné un ensemble de sommets, dans le graphe de boules unitaires, deux éléments sont voisins si la distance entre eux est plus petite que un. On remarque que les grilles et les tores peuvent être construits à partir d'un ensemble de sommets placés de façon régulière. Nous analyserons un cas plus général, quand les sommets n'ont pas une structure régulière rigide mais s'ils satisfont une condition plus faible : la croissance bornée.

Nous montrerons que la modularité du graphe des boules unitaires est bornée inférieurement par une constante qui dépend de la dimension de l'espace. Finalement on fait des expériences numériques qui confirment notre borne théorique.

5.1 Modularité dans les espaces métriques

Nous souhaiterions appliquer la définition de modularité à un clustering d'un espace métrique V . Étant donné un ensemble de points dans V et un rayon $L > 0$, nous définissons l'ensemble d'arêtes $E = \{uv \mid d(u, v) \leq L\}$. Nous appellerons *graphe de boules unitaires* ce graphe, car c'est aussi le graphe d'intersection des boules de centre $v \in V$ et de rayon $L/2$. Nous le noterons $G_L = (V, E)$. On peut donc maintenant parler de la "modularité d'un partitionnement de V " comme étant celle calculée dans le graphe G_L .

5.2 Espaces de croissance bornée

Dans le cas général, il est impossible de donner une borne pour la modularité : si les points sont distribués de façons quelconque dans l'espace, certains clusters auront beaucoup de sommets et d'autres peu. En effet, pour une dimensions assez grande il est possible de construire tout graphe. Une condition d'homogénéité, même assez lâche, fournit des clusters de taille plus homogène.

On note $B(u, r) \subset V$ (boule de centre u et de rayon r) les points de V situés à distance au plus r de u .

Définition IV.26 (Dimension de grille [KR02] aussi appelée de croissance bornée¹). L'espace V est de dimension de grille bornée par $\gamma > 0$ si, en doublant le rayon d'une boule, on en multiplie le volume par moins de γ :

$$\forall x \in V, \forall r > 0, |B(x, r)| \leq \gamma \cdot |B(x, r/2)|$$

On remarque qu'un espace euclidien de dimension d est de dimension de grille $\gamma = 2^d$ (pour la norme $\|\cdot\|_\infty$).

5.3 R -nets

Nous introduisons un outil algorithmique pour construire des partitionnements dans les espaces de croissance bornée. Un R -net est un sous-ensemble de points placés mutuellement à une distance supérieure à R et qui recouvre l'espace de façon homogène.

Définition IV.27 (R -net). Un sous-ensemble U d'un espace V est un R -net si

- $\forall u, u' \in U, d(u, u') > R$.
- $\forall v \in V \exists u \in U d(u, v) \leq R$.

A partir d'un R -net nous construisons son partitionnement associé. Pour tout $u_i \in U$ le cluster C_i est défini comme l'ensemble des points plus proches de u_i que de

1. bounded growth.

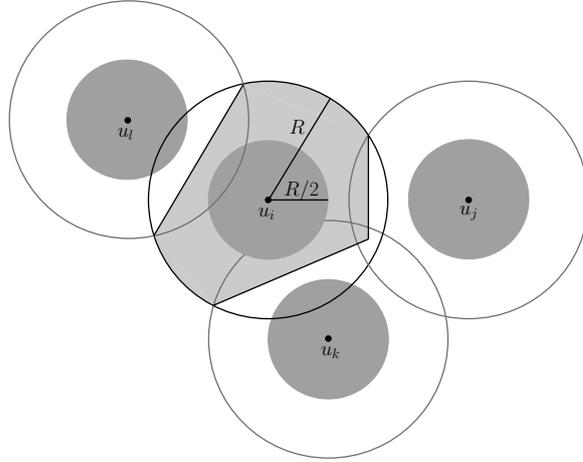


Figure IV.2 – Une R -net et son partitionnement associé

tout autre point du R -net (voir figure IV.2). Les cas d'égalité sont cassés de manière arbitraire. Un R -net peut être construit de façon gloutonne en temps $O(|V||U|)$. Dans la section suivante, nous montrons qu'un tel clustering a une modularité strictement positive et bornée inférieurement par une fonction de la dimension.

5.4 Modularité d'un R -net

Théorème IV.28. *Soit V un espace fini de n points, muni d'une métrique d , de dimension de grille bornée par γ , et $R \geq 0$ tel que pour tout $v \in V$ on a $|B(v, R/2)| > 1$ et $|B(v, R/2)| = o(\sqrt{n})$. Alors on a :*

$$\mathfrak{Q}(G_R) \geq \frac{1}{2\gamma^3} - o(1)$$

Tout espace métrique fini ayant une certaine dimension de grille γ , ce théorème s'applique pour tout rayon vérifiant la condition sur les tailles des boules. Cependant, la borne obtenue est d'autant plus élevée que la valeur de γ est faible.

Démonstration. Soit $U = \{u_1, \dots, u_k\}$ un R -net de V (il existe un algorithme glouton simple pour en produire). On construit le clustering \mathcal{C}_U qui contient un cluster pour chaque élément du R -net : $\mathcal{C}_U = \{C_i\}_{i \in \{1, \dots, k\}}$ avec

$$C_i = \left\{ v \in V \mid \forall j \neq i, d(v, u_i) \leq d(v, u_j) \right\}.$$

Nous avons par construction $B(u_i, R/2) \subseteq C_i \subseteq B(u_i, R)$. Posons $b_i = |B(u_i, R/2)|$. La définition de la dimension de grille nous donne : $b_i \leq |C_i| \leq \gamma \cdot b_i$. De plus, comme

IV.6 Simulations en espace euclidien

U est un R -net ses points sont mutuellement à distance au moins R . Les points de $B(u_i, R/2)$ sont donc tous reliés dans G_R et forment une clique incluse dans C_i . On a donc :

$$\mathcal{Q}(\mathcal{C}_U) = \sum_{i=1}^k \left[\frac{|E(C_i)|}{m} - \frac{(\sum_{v \in C_i} d_v)^2}{4m^2} \right] \geq \sum_{i=1}^k \left[\frac{b_i(b_i - 1)}{2m} - \frac{(\sum_{v \in C_i} d_v)^2}{4m^2} \right]$$

pour tout $v \in C_i$, on a $d_v = |B(v, R)| \leq |B(u_i, 2R)| \leq \gamma^2 b_i$, d'où :

$$\mathcal{Q}(\mathcal{C}_U) \geq \sum_{i=1}^k \left[\frac{b_i(b_i - 1)}{2m} - \frac{(|C_i| \gamma^2 b_i)^2}{4m^2} \right] \geq \sum_{i=1}^k \left[\frac{b_i(b_i - 1)}{2m} - \frac{\gamma^6 b_i^4}{4m^2} \right]$$

Or,

$$\sum_{i=1}^k b_i(b_i - 1) \leq 2m = \sum_{u \in V(G)} d_u \leq \sum_{i=1}^k |C_i| \gamma^2 b_i \leq \gamma^3 \sum_{i=1}^k b_i^2,$$

Et donc :

$$\mathcal{Q}(\mathcal{C}_U) \geq \frac{\sum_{i=1}^k b_i(b_i - 1)}{\gamma^3 \sum_{i=1}^k b_i^2} - \frac{\gamma^6 \sum_{i=1}^k b_i^4}{2m \sum_{i=1}^k b_i(b_i - 1)}$$

Posons $\bar{b} = \max_i \{b_i\}$. Comme $\frac{\sum_{i=1}^k b_i(b_i - 1)}{\sum_{i=1}^k b_i^2} = 1 - \frac{\sum_{i=1}^k b_i}{\sum_{i=1}^k b_i^2} \geq 1 - \frac{\sum_{i=1}^k b_i}{\sum_{i=1}^k 2b_i} \geq \frac{1}{2}$, on a :

$$\mathcal{Q}(\mathcal{C}_U) \geq \frac{1}{2\gamma^3} - \frac{\gamma^6 \bar{b}^2 \sum_{i=1}^k b_i^2}{2m \sum_{i=1}^k b_i(b_i - 1)} \geq \frac{1}{2\gamma^3} - \frac{\gamma^6 \bar{b}^2}{m}$$

Enfin, comme $\bar{b} \in o(\sqrt{n})$ et $m \geq \frac{n}{2}$, nous obtenons $\mathcal{Q}(\mathcal{C}_U) = \frac{1}{2\gamma^3} - o(1)$, donc

$$\mathcal{Q}(G_R) = \frac{1}{2\gamma^3} - o(1)$$

□

6 Simulations en espace euclidien

Nous exécutons des simulations numériques dans le but de tester le résultat du théorème IV.28. Les simulations montrent que l'on obtient en pratique une bien meilleure valeur que la borne de la section précédente dans le cas euclidien uniforme. Le choix de la géométrie euclidienne est dû à la nécessité d'avoir un \mathbb{R} -espace vectoriel pour comparer nos résultats avec un algorithme classique de partitionnement dans les espaces métriques vectoriels : k -means.

L'algorithme k -means essaie de diviser l'espace en k clusters de manière à minimiser la distance de chaque élément au barycentre du cluster auquel il appartient. Plus précisément, étant donné un ensemble de points X , k -means tente de construire un partitionnement C_1, \dots, C_k qui minimise la fonction $WCSS^2$ définie par

$$WCSS = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2,$$

où $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x_i$ est le barycentre du cluster C_i .

L'algorithme standard pour ce problème est l'algorithme de Lloyd. Cet algorithme commence avec un partitionnement C_1, \dots, C_k quelconque, et de manière itérative actualise les clusters jusqu'à la stabilisation de la façon suivante :

1. $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x_i$,
2. $C_i = \left\{ x \in X \mid \forall j \neq i, d(x, \mu_i) \leq d(x, \mu_j) \right\}$.

On pose le modèle suivant : n points sont tirés au hasard, uniformément et indépendamment, dans un hypercube de côté 1, et de dimension d . On pose $R < 1$ et on construit le R -net gloutonnement, puis le clustering au plus proche (en brisant arbitrairement les égalités). La figure IV.3 montre la modularité en fonction de R . Notons que les conditions du théorème IV.28 sur la valeur de R ne sont pas remplies pour R trop petit ou trop grand, ce qui peut amener à une modularité faible. Dans la preuve du théorème IV.28, nous avons posé l'égalité entre R , le rayon des clusters, et L , le rayon des boules de G_L . Cela permet de maximiser la taille de la clique $B(u_i, R/2)$ dont nous avons besoin. Il peut être intéressant de comparer ces deux paramètres indépendamment. Cette relation est montrée dans la figure IV.5, pour R fixé à 0,1. Le paramètre est $a = R/L$. Partout ailleurs que dans cette figure on a $R = L$.

Les différentes courbes font varier le nombre de points (500 ou 5000) et la dimension ($d = 2$ ou 3) de l'espace. Comme on voit, ces deux paramètres sont sans influence claire. Nous avons aussi utilisé un tore pour supprimer les effets de bord : les résultats sont quasiment identiques.

Nous avons comparé empiriquement avec le résultat de l'algorithme Lloyd pour les k -means, en l'initialisant de sorte que \mathcal{B}_0 soit un R -net. La troisième figure reprend la première, mais appliquée à \mathcal{C}_∞ (résultat de Lloyd), comparé à \mathcal{C}_0 (celui du R -net). Nous constatons que l'algorithme de Lloyd produit un clustering de modularité sensiblement égale : elle est quasi-invariante d'une itération à l'autre.

Remarquons que pour R trop petit le graphe des boules est quasiment sans arêtes. On obtient de plus énormément de clusters très petits (voir figure IV.4) et donc l'algorithme de Lloyd, pour k tellement grand, atteint un objectif presque nul. Le graphe devient connecté (avec forte probabilité) quand $n\pi R^2 \geq \ln n$, soit $R \geq 0,023$ pour $n = 5000$. Pour $R = 0,11$ on n'a plus que $k = 59$ clusters, valeur qui tombe à $k = 10$ pour $R = 0,31$ et $k = 3$ pour $R = 0,71$. Les courbes sont donc significatives (clustering non trivial) seulement quand R est environ entre 0,1 et 0,5. Le nombre de clusters varie alors en $1/\pi R^2$ (figure IV.4).

2. within-cluster sum of squares.

IV.6 Simulations en espace euclidien

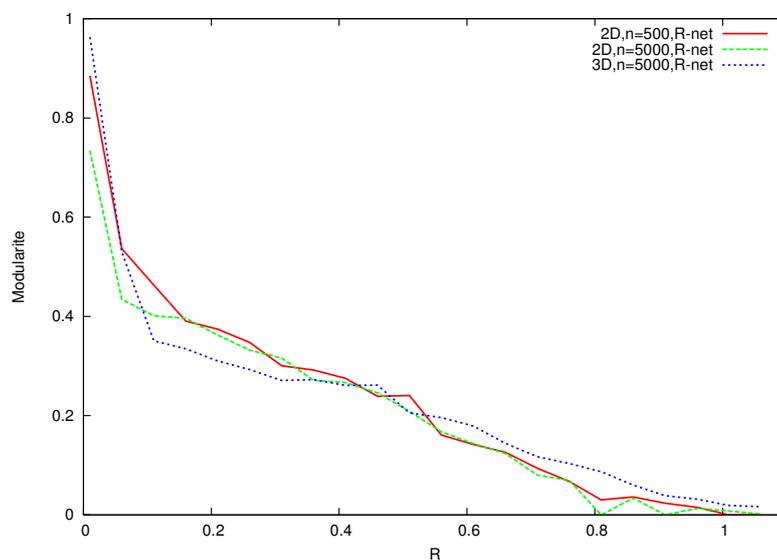


Figure IV.3 – Valeurs de la modularité en fonction du rayon du R -net

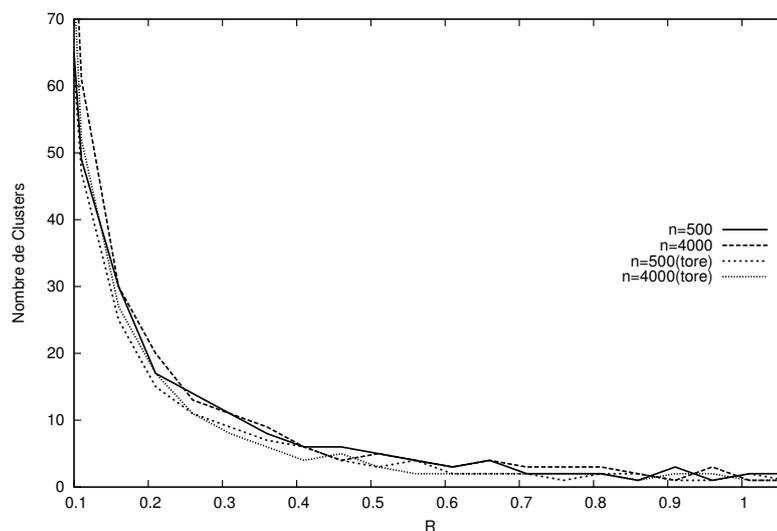


Figure IV.4 – Nombre de clusters en fonction du rayon du R -net

Enfin, la figure IV.7 montre que le R -net est déjà une bonne heuristique pour le calcul de k -means : utilisé pour initialiser Lloyd, ce dernier fait baisser WCSS mais pas énormément (pour $0,1 < R < 0,5$). En ce qui concerne le nombre de clusters, Lloyd ne vide quasiment jamais de cluster si on l'initialise avec le R -net, alors qu'une initialisation de \mathcal{C}_0 avec une partition aléatoire crée beaucoup de clusters vides dans \mathcal{C}_∞ .

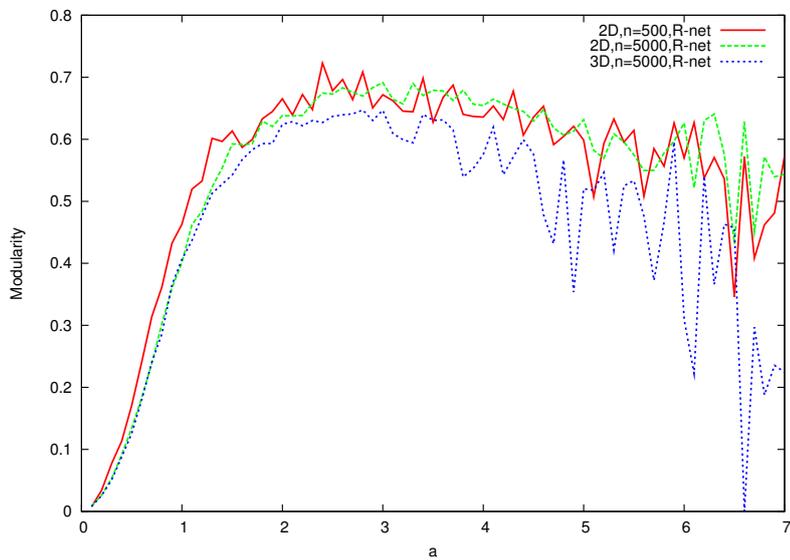


Figure IV.5 – Valeurs de la modularité pour des clusters de taille variable : $a = R/L$.

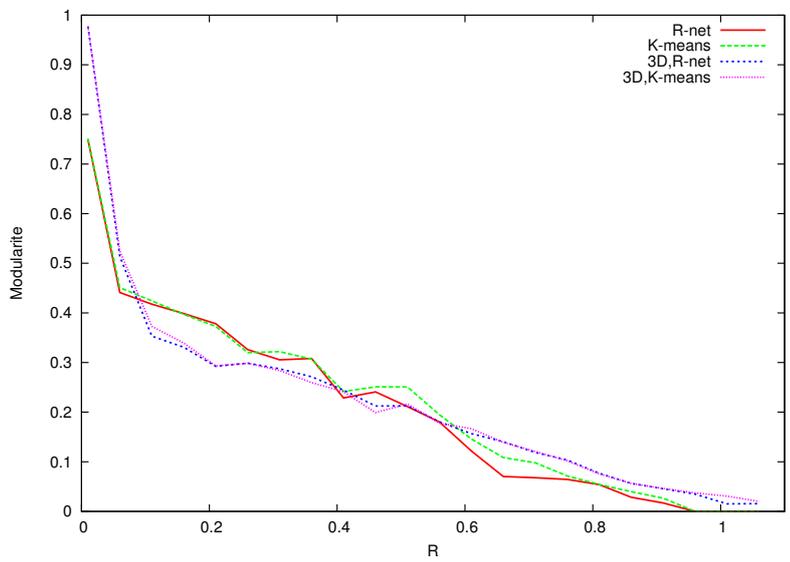


Figure IV.6 – Comparaison entre les valeurs de la modularité en utilisant l'algorithme glouton et K -means.

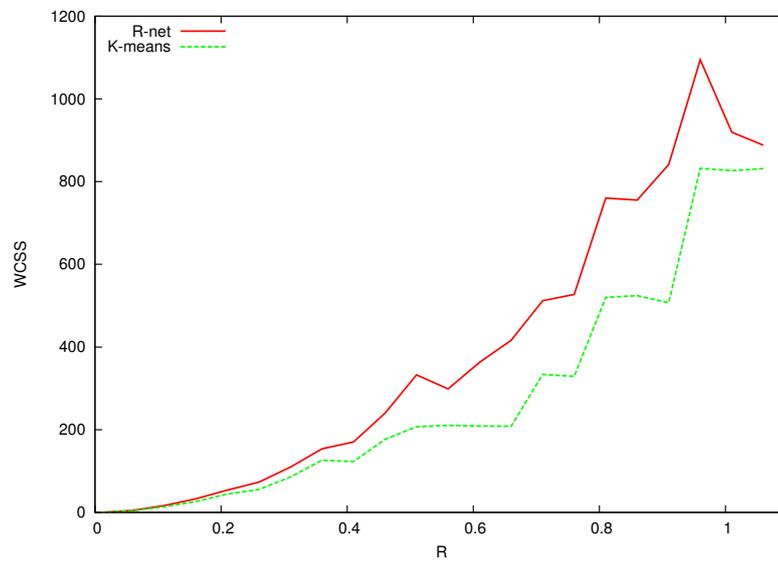


Figure IV.7 – Comparaison entre les valeurs de la somme de carrés au baricentre des clusters (fonction objective de K -means).

Chapitre V

Réseaux et files d'attente avec modèle d'adversaire

Les chapitres précédents portent sur les propriétés topologiques des graphes. Dans ce chapitre nous nous intéressons à la dynamique du réseau au niveau de l'ordonnement de la transmission des messages. Notre but est l'analyse des protocoles des files d'attente des routeurs et de leurs conséquences sur la stabilité globale du système.

La discipline mathématique par excellence pour l'étude de ce problème est *la théorie de files d'attentes*¹. La théorie classique impose des hypothèses rigides sur le processus d'arrivée et le temps de service sur chacune des files d'attente : dans le cas le plus fréquent, les arrivés suivent un processus de Poisson et le temps de service suit une distribution exponentielle [JAC63]. Une généralisation importante est proposée par Kelly [KEL79, KEL75] avec différentes classes de paquets ayant des temps de service différents pour chaque classe.

Les hypothèses faites par les modèles probabilistes ne sont pas toujours valables pour le routage dans Internet dans le cadre le plus général. Afin de surmonter les restrictions du modèle classique, une approche plus combinatoire a été proposée par Borodin et al. [BKR⁺96, BKR⁺01] : *les réseaux de files d'attente avec modèle d'adversaire*². Dans ce modèle, les injections de paquets dans le réseau ne correspondent pas à un processus stochastique mais sont faites par un **adversaire** et le temps de service est unitaire (un paquet prend une unité de temps pour traverser une arête). De plus un **protocole** établit l'ordre de passage dans chaque file d'attente. La dynamique peut alors être vue comme un jeu de l'adversaire contre le protocole afin de maintenir borné le nombre de paquets dans le réseau. Cette condition définit la **stabilité** du système.

Nous proposons ici une généralisation du modèle d'adversaire en considérant plusieurs classes de paquets. La nouveauté est l'introduction de délais lorsque la classe de pa-

1. Queueing Theory en anglais.

2. Adversarial Queueing Theory ou AQT

quets servie par une arête change. Dans ce nouveau cadre, nous étudions le comportement de **protocoles sensibles** introduit par Dai et Jennings [DJ04] pour le modèle multi-classe stochastique. Le principe des protocoles sensibles est d'éviter de changer vers une classe si la quantité de paquets dans la file d'attente ne dépasse pas une certaine borne.

Nous adaptons des arguments standard de stabilité pour le nouveau modèle avec plusieurs classes. Néanmoins nous présentons un contre-exemple qui montre que des adaptations naturelles de certains protocoles pour le cadre multi-classe ne conservent pas la stabilité du modèle original.

Un outil important pour l'analyse de la stabilité du modèle stochastique ainsi que le cas d'adversaire mono-classe, est l'utilisation d'un **modèle fluide** associé au réseau. Le modèle fluide est une version continue de la dynamique du réseau, il correspond à un système d'équations obtenues comme processus limite des relations entre les valeurs du processus discret. La stabilité du système discret peut être inférée à partir de la stabilité du modèle fluide associé tant pour le cadre stochastique multi-classe [DAI95] que pour le modèle avec adversaire [GAM98]. Cette relation nous pousse à étudier le modèle fluide pour le nouveau contexte multi-classe. Nous prouvons que la stabilité peut être aussi obtenu à partir du modèle fluide pour notre scénario.

Ce travail à été réalisé avec la collaboration de Marcos Kiwi et Christopher Thraves de l'Université du Chili. Les résultats font partie de la publication [KST09].

1 Description du modèle

Un *réseau de routage* est un graphe orienté avec une capacité unitaire sur chaque arête. Le temps est discrétisé. Pour chaque arête orientée, il existe une file d'attente dans son sommet d'origine que nous assumons de taille infinie. Un paquet est une entité unitaire de travail qui réside dans la file d'attente d'une arête à la fin de chaque pas de temps. Le paquet traverse le réseau depuis son injection à sa source jusqu'à sa destination où le paquet disparaît du réseau. Formellement, un *paquet* est un triplet (ID, P, t) où ID est un identifiant unique, P est un chemin sur le réseau à suivre pour le paquet et $t \in \mathbb{N}$ est le temps auquel le paquet a été injecté dans le réseau.

Un *adversaire* est une entité (processus) qui, à chaque pas de temps, génère l'ensemble des paquets injectées, en particulier le chemin que chaque paquet doit suivre pour arriver à sa destination.

La nouveauté de notre modèle est la possibilité de l'adversaire de déterminer une *classe* pour chaque paquet et pour chaque arête sur le chemin du paquet. Cette classe est prise dans un ensemble donné de classes \mathcal{S} . Ainsi, lorsqu'un paquet p est injecté, l'adversaire spécifie son chemin à parcourir $e_1 \dots e_\ell$ et une *fonction de traversée* $f_p : \{1, \dots, \ell\} \rightarrow \mathcal{S}$. La fonction de traversée capture le fait qu'un paquet peut demander différents types de services (classes) dans son parcours.

Si l'ensemble \mathcal{S} contient plus d'un élément, nous parlons d'un adversaire multi-classe. Dans ce cas, deux scénarios sont possibles : si la fonction de traversée est non-constante nous parlerons d'un adversaire variable multi-classe ; au contraire si la fonction f_p est constante et la classe de paquet est fixe depuis son injection, nous parlons d'un adversaire constant multi-classe. Bien que plus restreint, l'adversaire statique inclut plusieurs exemples importants, notamment le protocole de transmission ATM³ ou Mode de transfert asynchrone, où les paquets sont de taille contant et ils indiquent dans leur entête un type qui reste fixe au long de son parcours.

Un adversaire sans aucune restriction d'injection peut saturer le réseau facilement en injectant une grande quantité de paquets qui utilisent une même arête. Dans le modèle d'adversaire, les injection sont bornées selon une paire (r, b) où $b \geq 1$ est un entière et $0 < r \leq 1$. Un (r, b) -adversaire [BKR⁺01, AAF⁺01] satisfait pour chaque arête e et pour tout $t_1 < t_2$,

$$A_e(t_1, t_2) \leq r(t_2 - t_1) + b, \quad (\text{V.1})$$

où $A_e(t_1, t_2)$ dénote le nombre de paquets injectés par l'adversaire dans l'intervalle $[t_1, t_2)$ et dont le chemin utilise l'arête e .

La restriction (V.1) est appelée *la restriction de charge*. Elle exprime le fait qu'en moyenne, au plus r paquets utilisant la même arête sont injectés pour chaque pas de temps, sauf pour une quantité sporadique de paquets b . La classe des *adversaires à taux borné* est l'ensemble des (r, b) adversaires avec $r < 1$. Cette définition est motivée par le fait qu'un $(1, b)$ -adversaire avec $b > 0$ atteint facilement la instabilité dans le cadre multi-classe.

Si plusieurs paquets veulent traverser une même arête, un *protocole d'ordonnement de paquets* décide le paquet qui utilisera le lien. Dans notre modèle il existe un *délai* lorsque une arête change la classe de paquet qu'elle serve. Plus précisément, une arête est configuré pour servir une classe particulière de paquets. Le protocole peut décider de reconfigurer une arête pour changer la classe servie. Dans ce cas l'arête reste inactive (aucun paquet ne peut pas être envoyé) pendant un temps Δ . Formellement, si $D_{e,i}(t_1, t_2)$ dénote le nombre de paquets de classe i qui traversent l'arête e pendant l'intervalle $[t_1, t_2)$. Le respect du délais implique la restriction suivante :

$$D_{e,i}(t, t + 1) > 0 \implies \forall j \neq i, D_{e,j}(t + 1, t + 1 + \Delta) = 0. \quad (\text{V.2})$$

Une généralisation possible est l'utilisation de différents temps de délais en fonction du changement de classe. Ainsi, si pour une arête, le protocole décide de changer la classe de paquets servie de la classe i vers la classe j , alors aucun paquet ne peut traverser l'arête pendant $\Delta_{i,j}$ pas de temps. Dans ce cas général on suppose $\Delta_{k,k} = 0$. Par souci de simplicité nous nous plaçons dans le cas où tous les délais de changement sont égaux à Δ . Néanmoins, tous les résultats concernant à la stabilité de ce chapitre sont aussi valables si on définit Δ comme le maximum des délais.

3. Asynchronous Transfer Mode

Le *protocole de routage de paquets* décide, pour chaque arête du réseau et pour chaque pas de temps, quel paquet dans la file d'attente envoyer. Quelques exemples étudiés par [BKR⁺01] :

- Premier-Arrivé-Premier-Servi (First-In-First-Out) — les paquets sortent par l'arête dans l'ordre dans lequel ils son arrivés.
- Au-Plus-Proche (Nearest-To-Go) — le paquet plus proche de sa destination a la priorité.
- Au-Plus-Loin (Farthest-To-Go) — le paquet plus éloigné de sa destination a la priorité.
- Au-Plus-Récent (Shortest-In-System) — la priorité est donnée aux paquets les plus récemment injectés.
- Au-Plus-Ancien (Longest-In-System) — la priorité est donnée aux paquets les plus anciennement injectés.

Une *protocole de changement* décide, pour chaque arête du réseau et pour chaque pas de temps, s'il faut continuer à envoyer des paquets de la même classe ou changer de classe. Autrement dit, le protocole de routage de paquets établit la priorité pour les paquets d'une même classe pendant que le protocole de changement assigne la priorité entre les classes. Des exemples de protocoles de changement sont :

- Longueur-De-File (Largest-Queue) — prendre la classe la plus représentée dans la file (les égalités sont cassés de façon arbitraire).
- Tourniquet (Round-Robin) — les classes sont numérotées par $0, \dots, |\mathcal{S}|-1$. S'il n'est pas possible de continuer à envoyer des paquets de classe i , alors on passe à la classe $(i + 1) \bmod |\mathcal{S}|$.
- Premier-Arrivé-Premier-Servi (First-In-First-Out) — change pour la classe du premier paquet arrivé en file (les égalités sont cassés de façon arbitraire).
- Priorité (Priority) — les classes sont numérotées par $0, \dots, |\mathcal{S}|-1$. Le protocole choisit la classe ayant la valeur la plus petite.

Nous remarquons qu'une fois que le protocole de changement décide de changer de classe, il n'est pas possible de modifier de choix avant Δ pas de temps. Par exemple, une arête qui est restée inactive pour une période de moins de Δ pas, ne peut pas servir une classe quelconque. Un protocole de changement est *exhaustif*, si une fois qu'il a choisit d'envoyer une classe i dans une arête, alors il n'est pas possible de changer la classe servie tant que des paquets de de classe i restent dans la file d'attente. Dans le cas d'une unique file d'attente les protocoles exhaustifs permettent attendre la stabilité, mais pour un réseau quelconque ils peuvent mener à l'instabilité [DJ04]. Néanmoins nous ne considérons que des protocoles exhaustifs par la suite. Nous verrons qu'ils suffisent pour attendre la stabilité dans notre cadre.

Un protocole de changement est *insensible au type* si, pour un pas de temps quelconque, une renumérotation des classes laisse invariante l'élection du paquet à servir. Par exemple Longueur-De-File est un insensible au type alors que Tourniquet ne l'est pas. Un protocole de changement est *insensible au décalage* si, étant donné un pas de temps, tout paquet de classe i est renumérotés par $(i + 1) \bmod |\mathcal{S}|$, alors le protocole donne la même préférence qu'avant le décalage. Le protocole Premier-Arrivé-Premier-Servi est un protocole insensible au décalage alors que Priorité ne l'est pas.

Un *protocole d'ordonnancement* est un couple (P, S) , où P dénote un protocole

de routage de paquets et S correspond à un protocole de changement. Un protocole de routage est *glouton*⁴ si chaque arête qui possède au moins un paquet dans sa file d'attente est active. Cette notion coïncide avec les protocoles avec conservation de travail⁵ définis dans le cadre mono-classe.

Nous remarquons que le protocole de changement est pertinent même quand les délais sont nuls ($\Delta = 0$). En effet, la propriété d'exhaustivité oblige à envoyer tous les paquets d'une classe donnée avant de pouvoir changer la classe à servir.

Nous étudions la *stabilité* des protocoles d'ordonnancement exhaustifs sur différents réseaux quand les injections sont gérées par un adversaire.

Le *système* est défini par le quadruplet $(G, \Delta, (P, S); A)$ où G est un réseau de routage, $\Delta \geq 0$ est le délai du réseau, P est un protocole de routage de paquets, S est un protocole de changement et A est l'adversaire qui contrôle l'injection des paquets.

Remarque V.1. Dans le modèle mono-classe, il existe une équivalence entre les réseaux avec une configuration initiale vide et non-vide [AAF⁺01, Lemma 2.9] : pour tout réseau avec configuration initiale non-vide il est possible de construire un réseau et un adversaire qui commence avec les files d'attente vides et qui se comporte de façon analogue. Néanmoins, pour un protocole qui dépend de l'histoire des paquets (par exemple Au-Plus-Ancien) cette équivalence n'est pas claire, comme il a été signalé dans [CE07B]. Nous considérons le cas où les files d'attente ont une configuration arbitraire au début de la dynamique.

Définition V.2. Le système $(G, \Delta, (P, S); A)$ est *stable* si pour toute configuration initiale, il existe une constante $M \geq 0$ (qui dépend de la configuration initiale, de la taille du réseau et des paramètres Δ , P , S et A) telle que si les paquets sont injectés par l'adversaire A et s'ils sont routés selon le protocole d'ordonnancement (P, S) , le nombre de paquets dans toutes les files d'attente du réseau est borné par M .

Si le système $(G, \Delta, (P, S); A)$ est stable, nous disons que le réseau G avec délai Δ et protocole d'ordonnancement (P, S) est stable contre l'adversaire A . Si le système $(G, \Delta, (P, S); A)$ est stable pour tout graphe appartenant à une classe \mathcal{G} et pour tout délai Δ , nous disons que pour la classe de graphe \mathcal{G} le protocole (P, S) est stable contre l'adversaire A .

Nous nous intéressons à déterminer si un protocole d'ordonnancement est stable pour tout réseau (graphe) d'une classe \mathcal{G} et tout délai Δ contre tout adversaire de taux borné. De plus, étant donnée une famille de protocoles d'ordonnancement, nous essayons de déterminer les réseaux qui sont stables contre tout adversaire de taux borné.

Il est naturel que, pour atteindre la stabilité, les protocoles d'ordonnancement doivent éviter de changer la classe de paquets envoyés de façon trop fréquente. Nous étudions la notion de *protocoles sensibles*, une variation des protocoles de changement

4. greedy

5. work conserving packet scheduling protocols

introduite par Dai and Jennings [DJ04]. Un protocole est dit θ -sensible s'il sert en priorité les classes de paquets avec plus de θ paquets dans la file. Étant donné un protocole de changement S et un paramètre θ , S_θ dénote le protocole de changement qui, pour chaque arête, choisit la classe selon le protocole S mais restreint aux classes qui ont au moins θ paquets dans la file d'attente. Si aucune classe ne possède θ paquets ou plus, le protocole choisit de façon arbitraire. Remarquons qu'on obtient ainsi un protocole θ -sensible. Afin d'éviter au maximum les délais, nous utilisons des protocoles exhaustifs. Néanmoins, les résultats de stabilité peuvent être facilement généralisés pour le cas non exhaustif.

Définition V.3. Un réseau G est *stable pour le protocole d'ordonnement* (P, S) par rapport à une classe d'adversaire \mathcal{A} , si pour tout $\Delta \geq 0$ et tout $A \in \mathcal{A}$, il existe $\theta \geq 0$ (qui est une fonction de délai, de la taille du réseau, de la configuration initiale et de P, S et A) tel que le système $(G, \Delta, (P, S_\theta); A)$ est stable.

Définition V.4. Le réseau G est *universellement stable* par rapport à une classe d'adversaires \mathcal{A} , si G est stable pour tout protocole d'ordonnement glouton par rapport à \mathcal{A} .

Définition V.5. Le protocole d'ordonnement (P, S) est *universellement stable* pour \mathcal{G} par rapport à la classe d'adversaire \mathcal{A} , si pour tout réseau $G \in \mathcal{G}$, G est stable pour le protocole (P, S) par rapport à la classe d'adversaires \mathcal{A} .

Si aucune spécification n'est donnée, nous supposons que la classe d'adversaires \mathcal{A} correspond à la classe d'adversaires à taux borné

2 Travaux Similaires

La théorie de files d'attente introduit par Jackson [JAC63] pour le cas mono-classe a été largement étudié dans la littérature ([BOR86, FOS89, FOS91, MD94, CTK94, BF94, BFM96, DAI95] entre autres). Une des généralisations importantes est l'introduction de différentes classes de tâches développée par Kelly [KEL79, KEL75]. Dans ce cadre, les résultats dans [LU91, RS92, KS90] montrent que la condition de charge ne suffit pas pour garantir la stabilité du réseau (dans le contexte stochastique cette condition correspond à la relation entre les processus d'injection et de service). Par exemple, Bramson [BRA94A, BRA94B] montre que le protocole Premier-Arrivé-Premier-Servi est instable pour des processus avec injections arbitrairement petites. Remarquons le développement d'un modèle fluide pour prouver la stabilité des réseaux développés par Dai [DAI95] et Dai et May [DM95]. Cette technique a été aussi appliquée dans le cadre multi-classe stochastique avec délais par Dai et Jennings [DJ04].

Un premier modèle non stochastique a été étudié par Cruz [CRU91A, CRU91B] : le modèle de session permanentes ou "leaky-bucket", impose une restriction de charge d'injection sur tout chemin et peut être considéré comme une version restreinte du

modèle d'adversaire. Dans ces travaux, Cruz montre la stabilité des protocoles gloutons pour les réseaux acycliques. Dans ce scénario Tassioulas and Georgiadis [TG96] montrent la stabilité du cycle, tandis que Andrews [AND04] montre l'instabilité du Premier-Arrivé-Premier-Servi pour des taux d'injection arbitrairement petits.

Le modèle avec adversaire (mono-classe) a été introduit par Borodin et al. [BKR⁺96, BKR⁺01]. À partir de ce modèle plusieurs problèmes ont été étudiés pour le modèle d'adversaire, par exemple le problème de caractériser et reconnaître les réseaux universellement stables [AAF⁺01, ABS04, GOE01], l'analyse de la stabilité pour des réseaux de topologie variable [ABD⁺05, AKK08, BBS01], ou l'instabilité du protocole Premier-Arrivé-Premier-Servi [AND04, AAF⁺01, DKN⁺01, KNS01, LPSR04, BGL05]. Le modèle fluide a été étudié par Gamarnik [GAM98].

3 Résultats de stabilité

Dans cette section nous caractérisons les réseaux universellement stables contre la famille d'adversaires bornés. De façon similaire que dans le modèle d'adversaire mono-classe, ce résultat est basé sur la stabilité universelle des graphes orientés acycliques et des cycles orientés.

Les preuves de stabilité de graphes orientés acycliques et des cycles orientés reposent sur l'existence d'une fonction de potentiel similaire aux preuves pour le cadre mono-classe : [BKR⁺01, Théorème 1] et [AAF⁺01, Théorème 3.7]. L'argument de fonction de potentiel, introduit par Loynes et Walker [LOY62] permet borner le nombre de paquets dans le réseau par une fonction indépendante du temps (mais qui dépend du délai Δ) obtenant ainsi la stabilité.

De manière intuitive, si une arête contient une file d'attente avec une petite quantité de paquets, alors la classe de paquet servie n'est pas décisive. Au contraire, si une arête est congestionnée, tout protocole sensible glouton permet l'utilisation efficace de l'arête, en envoyant un paquet à chaque pas de temps, pendant une période suffisamment longue. Cet intervalle, borné inférieurement par θ , dépend du temps de délai, du taux d'injection r et du paramètre b .

Un protocole de changement sensible S_θ , pour une valeur θ suffisamment grande, garantit que le coût de délai est amorti par l'utilisation à pleine capacité des arêtes pendant une période de temps suffisamment longue après chaque changement de classe.

Auparavant, nous introduisons quelques notations. Soit G un graphe orienté, pour chaque arête e du graphe et pour chaque classe de paquet $i \in \mathcal{S}$, nous dénotons par $Q_{e,i}(t)$ le nombre de paquets de classe i dans la file d'attente de l'arête e au pas de temps t . De plus, $Q_e(t)$ dénote la quantité totale de paquets dans la file de l'arête e , $Q_e(t) = \sum_{i \in \mathcal{S}} Q_{e,i}(t)$. Soit $N_e(t)$ le nombre de paquets dans le réseau qui veulent traverser l'arête e dans le chemin vers leurs destination et qui ont été injecté avant le temps t (inclus). On dit que une arête $f = (u, v)$ entre dans une arête $e = (u', v')$ si $v = u'$. On dira aussi que f est une arête entrant de e (voir la figure V.1).

3.1 Stabilité des graphes orientés acycliques

La stabilité des graphes orientés acycliques pour le modèle d'adversaire mono-classe a été prouvé par Borodin et al. [BKR⁺01]. Nous généralisons ce résultat au modèle multi-classe avec délai. La preuve se déroule de façon similaire au cadre standard mono-classe. Nous considérons une fonction de potentiel qui compte le nombre de paquets qui veulent traverser une arête au temps t . Nous montrons que la fonction de potentiel reste bornée supérieurement. Cette borne dépend de la structure du graphe en amont de chaque arête.

Théorème V.6. *Pour tout graphe orienté acyclique G et tout délai $\Delta \geq 0$, tout protocole de routage glouton (P, S) , et pour tout (r, b) -adversaire A avec taux $r < 1$, il existe $\theta > 0$ tel que le système $(G, \Delta, (P, S_\theta); A)$ est stable.*

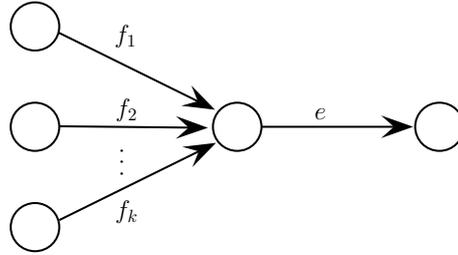


Figure V.1 – Arêtes entrants d'une arête e .

Démonstration. Soit θ suffisamment grand de sorte que

$$r(\theta + 2\Delta) + b \leq \theta. \tag{V.3}$$

Nous définissons la fonction $\psi(\cdot)$ de manière inductive : pour chaque arête e , nous dénotons f_1, \dots, f_k les arêtes entrantes de l'arête e , alors

$$\psi(e) = \max\{(|\mathcal{I}| + 1)\theta, Q_e(0)\} + \sum_{j=1}^k \psi(f_j).$$

Nous allons montrer que, à chaque instant $t = l(\theta + 2\Delta)$ avec $l \in \mathbb{N}$, le nombre de paquet qui veulent utiliser l'arête e est borné par $\psi(e)$, i.e. $N_e(t) \leq \psi(e)$. La preuve découle donc de ce résultat. En effet, la fonction $N_e(\cdot)$ peut augmenter sa valeur seulement par des injections de nouveaux paquets, donc $N_e(t') \leq N_e(t) + r(\theta + 2\Delta) + b$ pour tout $t \leq t' < t + (\theta + 2\Delta)$. Ainsi, nous avons une borne supérieure pour le nombre de paquets dans le réseau : $\sum_e (\psi(e) + r(\theta + 2\Delta) + b)$.

Une arête est dite *source* si elle ne possède aucune arête entrante. L'existence d'une telle arête est garantie du fait que le réseau est acyclique.

Lemme V.7. *Pour toute arête source et pour tout pas de temps $t = l(\theta + 2\Delta)$ avec $l \in \mathbb{N}$,*

$$Q_e(t) \leq \max\{(|\mathcal{I}| + 1)\theta, Q_e(0)\} = \psi(e). \quad (\text{V.4})$$

Démonstration du Lemme V.7. Soient e une arête source et $l \in \mathbb{N}$. Nous faisons la preuve par induction sur l . Clairement $Q_e(0) \leq \psi(e)$. Pour $l \geq 0$, soit $t = l(\theta + 2\Delta)$, alors $(l + 1)(\theta + 2\Delta) = t + (\theta + 2\Delta)$. Deux cas sont possibles :

- $Q_{e,i}(t) \geq \theta$ pour une classe $i \in \mathcal{I}$: dans ce cas, pour tout protocole sensible et glouton, le nombre de paquets z qui traverse e pendant l'intervalle $[t, t + \theta + 2\Delta)$ est au moins $\theta \geq r(\theta + 2\Delta) + b$. Donc,

$$Q_e(t + (\theta + 2\Delta)) \leq Q_e(t) + r(\theta + 2\Delta) + b - z \leq Q_e(t).$$

- $Q_{e,i}(t) < \theta$ pour toute classe $i \in \mathcal{I}$: clairement, $Q_e(t) \leq |\mathcal{I}|\theta$, alors par le choix de θ ,

$$Q_e(t + (\theta + 2\Delta)) \leq Q_e(t) + r(\theta + 2\Delta) + b \leq |\mathcal{I}|\theta + \theta.$$

□

Nous prouvons maintenant que $N_e(t) \leq \psi(e)$ pour toute arête e (pas nécessairement source) et $t = l(\theta + 2\Delta)$ où $l \in \mathbb{N}$. La preuve se fait par induction sur l et sur la profondeur p de l'arête, définie comme la distance maximale à une arête source. Autrement dit, nous considérons la distance maximale dans le graphe du sommet dans la queue d'un arête source vers la queue de l'arête e .

Le cas base de l'induction est pour un arête source. Dans ce cas $N_e(t) = Q_e(t)$ et $\psi(e) = \max\{(|\mathcal{I}| + 1)\theta, Q_e(0)\}$. Ainsi, d'après le lemme V.7, on a $N_e(t) \leq \psi(e)$.

Soit e une arête de G à profondeur $p > 0$. Soient f_1, \dots, f_k ses arêtes entrantes. Deux cas sont possibles,

- $Q_{e,i}(t) \geq \theta$ pour une classe $i \in \mathcal{I}$: pour ce cas, le nombre de paquets z qui traversent l'arête e pendant l'intervalle $[t, t + \theta + 2\Delta)$ est au moins $\theta \geq r(\theta + 2\Delta) + b$ car le protocole est sensible et glouton. Donc,

$$N_e(t + (\theta + 2\Delta)) \leq N_e(t) + r(\theta + 2\Delta) + b - z \leq N_e(t).$$

- $Q_{e,i}(t) < \theta$ pour toute classe $i \in \mathcal{I}$: clairement, $Q_e(t) \leq |\mathcal{I}|\theta$, à cause du choix de θ (équation (V.3)),

$$\begin{aligned} N_e(t + (\theta + 2\Delta)) &\leq Q_e(t) + r(\theta + 2\Delta) + b + \sum_{j=1}^k N_{f_j}(t) \\ &\leq (|\mathcal{I}| + 1)\theta + \sum_{j=1}^k N_{f_j}(t). \end{aligned}$$

D'autre part $N_e(0) = Q_e(0) + \sum_{j=1}^k N_{f_j}(0)$, par l'hypothèse d'induction, on conclut que pour $t = l(\theta + 2\Delta)$,

$$\begin{aligned} N_e(t) &\leq \max\{(|\mathcal{J}| + 1)\theta, Q_e(0)\} + \sum_{j=1}^k N_{f_j}(t) \\ &\leq \max\{(|\mathcal{J}| + 1)\theta, Q_e(0)\} + \sum_{j=1}^k \psi(f_j) \leq \psi(e). \end{aligned}$$

□

Corollaire V.8. *Les graphes orientés acycliques sont universellement stables.*

3.2 Stabilité des cycles orientés

Ce chapitre généralise le travail de Andrew et al. [AAF⁺01] où les auteurs montrent la stabilité des cycles orientés dans le modèle mono-classe.

Pour $n > 1$ soit R_n le cycle orienté avec n arêtes. Nous supposons que les arêtes et les sommets sont étiquetés par $0, 1, \dots, n - 1$. L'arête i est orientée depuis le sommet i vers le sommet $i + 1$. Toutes les opérations sont modulo n .

Énonçons-nous le résultat principal de la section :

Théorème V.9. *Soit $\Delta \geq 0$. Pour tout protocole glouton (P, S) et pour tout (r, b) adversaire A avec r strictement plus petit que 1, il existe $\theta > 0$ tel que $(R_n, \Delta, (P, S_\theta); A)$ est stable.*

Démonstration. Étant donnée l'arête e , soit $P_{e,i}(t)$ l'ensemble des paquets de classe $i \in \mathcal{J}$ qui doivent traverser l'arête e qui sont dans une file d'attente avec au moins θ paquets de classe i . Un paquet est compté autant de fois qu'il utilise l'arête. Nous notons $P_e(t)$ l'union, sur toutes les classes, de $P_{e,i}(t)$.

Pour $\varepsilon > 0$ tel que $r + \varepsilon < 1$, nous choisissons θ satisfaisant

$$r + \varepsilon \leq \frac{\theta}{\theta + 2\Delta}, \tag{V.5}$$

Par l'absurde, supposons que $(R_n, \Delta, (P, S_\theta); A)$ n'est pas stable. Alors, pour tout $Q' > 0$, il existe un jeu de l'adversaire A tel qu'une file d'attente s vérifie $|P_s(T)| > Q'$ pour un certain temps $T > 0$.

Soit $Q' > 0$ (nous choisirons sa valeur par la suite) et T' la première fois qu'une file d'attente satisfait $|P_s(T')| > Q'$. Nous définissons

$$Q = \max \left\{ |P_e(t)| : e \in \{0, \dots, n - 1\}, t \in [0, T') \right\}.$$

Pour $q \in P_s(t)$ nous notons par l_q la valeur j la plus grande telle que $q \in P_s(t-j)$ pour tout $j \in \{0, 1, \dots, l_q\}$. Soit p le paquet appartenant à $P_s(T')$ tel que $l_p = \max\{l_q : q \in P_s(T')\}$.

Soit $T_0 = T' - l_p$ et supposons, sans perte de généralité, que p est placé dans la file d'attente de l'arête 0 au temps T_0 . Autrement dit, dans l'intervalle $[T_0, T']$, le paquet p visite les files dans les arêtes $0, 1, \dots, s$.

Remarque V.10. D'après la définition de p , nous remarquons que pour chaque paquet p' dans la file d'attente de s au temps T' , il existe un pas de temps $t' \in [T_0, T']$ où le paquet p' est dans une file d'attente avec moins de θ paquets de sa classe.

Nous notons T_e le moment où le paquet p arrive dans la file de l'arête e . Alors, pour toute arête $e \in \{0, 1, \dots, s\}$ et $t \in [T_e, T_{e+1})$ nous avons que pour une classe i , $Q_{e,i}(t) \geq \theta$. Ainsi, pendant tous les sous-intervalle $[T_e, T_{e+1})$ de taille $\theta + 2\Delta$, au moins θ paquets doivent traverser e .

Considérons une arête e , $t \in [T_e, T_{e+1})$ et $T_0 \leq t' \leq t$. Pour un paquet dans $P_e(t)$ trois cas sont possibles :

- le paquet appartenait à $P_e(t')$,
- il a été injecté pendant l'intervalle $[t', t)$, ou
- il n'appartenait pas à $P_e(t')$ mais il appartient au temps t à une classe avec plus de θ paquets (par l'injection de paquets de sa classe ou par son arrivée à la file de e).

Les quantités de paquets dans ces trois cas peuvent être bornés par $|P_e(t')|$, $r(t - t') + b$, et $n|\mathcal{S}|\theta$ respectivement. Donc,

$$|P_e(t)| \leq |P_e(t')| + r(t - t') + b + n|\mathcal{S}|\theta - z, \quad (\text{V.6})$$

où z dénote le nombre de paquets qui traversent l'arête e pendant l'intervalle $[t', t)$.

Nous définissons une nouvelle fonction de potentiel suivante : pour $e \in \{0, \dots, n-1\}$, $t \geq T_0$ et ε satisfaisant (V.5) , soit

$$f(e, t) = \begin{cases} Q + e(b + n|\mathcal{S}|\theta), & \text{si } t = T_0, \\ Q - \varepsilon(t - T_0) + (e + 1)(b + n|\mathcal{S}|\theta), & \text{si } t > T_0. \end{cases}$$

Une paire (e, t) est *applicable* si une des conditions suivantes est satisfaite :

- $e \in \{0, 1, \dots, s\}$ et $t \in [T_0, T_{e+1}]$,
- ou $e > s$ et $t \in [T_0, T']$.

Lemme V.11. Pour toute paire (e, t) applicable, $|P_e(t)| \leq f(e, t)$

Démonstration du Lemme V.11. Nous nous intéressons au comportement de la fonction de potentiel. Il est facile de vérifier que pour toute arête e et tout temps $t \geq t' > T_0$,

$$f(e, t) = f(e, T_0) - \varepsilon(t - T_0) + b + n|\mathcal{S}|\theta, \quad (\text{V.7})$$

$$f(e, t) = f(e, t') - \varepsilon(t - t'). \quad (\text{V.8})$$

$$f(e, t) = f(e - 1, t) + b + n|\mathcal{S}|\theta. \quad (\text{V.9})$$

Pour $t = T_0$, d'après la définition de f et Q ,

$$|P_e(T_0)| \leq Q \leq Q + e(b + n|\mathcal{S}|\theta) \leq f(e, T_0).$$

Ainsi, l'affirmation est vraie pour (e, t) avec $t = T_0$.

Nous disons qu'une paire applicable (e, t) est *bonne* si $\forall t' \in [T_0, t), \exists i \in \mathcal{S}, Q_{e,i}(t') \geq \theta$ et *mauvaise* sinon. Par exemple toute paire applicable $(0, t)$ est bonne. En effet, comme $(0, t)$ est applicable alors t appartient à l'intervalle $[T_0, T_1]$ qui d'après la définition de p , contient plus de θ paquets de la classe du paquet $p : Q_{0,i}(t') \geq \theta$ pour $t' \in [T_0, t)$.

Nous prouvons l'affirmation pour les deux types de paires applicables. Soit (e, t) une paire applicable, alors

- Si (e, t) est bonne : par définition, pour tout intervalle de longueur $\theta + 2\Delta$, un protocole sensible permet qu'au moins θ paquets traversent l'arête e . Ainsi, d'après (V.6), (V.7), et le fait que $|P_e(T_0)| \leq Q \leq f(e, T_0)$ nous avons :

$$\begin{aligned} |P_e(t)| &\leq |P_e(T_0)| + r(t - T_0) + b + n|\mathcal{S}|\theta - (r + \varepsilon)(t - T_0) \\ &\leq f(e, T_0) - \varepsilon(t - T_0) + b + n|\mathcal{S}|\theta \\ &= f(e, t). \end{aligned}$$

- Si (e, t) est mauvaise : nous procédons par induction sur e (ordonné de 0 à n). Le cas de base est clair du fait que $(0, t)$ est applicable et bonne. Nous supposons que $|P_{e-1}(t)| \leq f(e-1, t)$ pour toute paire applicable $(e-1, t)$. Soit $T_0 \leq t' \leq t$ le temps le plus grand tel que $Q_{e,i}(t') < \theta$ pour toute classe $i \in \mathcal{S}$. De plus, $t' \neq T_0$ et $|P_e(t')| = |P_{e-1}(t')|$. D'après (V.6),

$$\begin{aligned} |P_e(t)| &\leq |P_e(t')| + r(t - t') + b + n|\mathcal{S}|\theta - (r + \varepsilon)(t - t') \\ &= |P_{e-1}(t')| - \varepsilon(t - t') + b + n|\mathcal{S}|\theta. \end{aligned}$$

La paire $(e-1, t')$ est applicable. En effet, comme (e, t) est applicable alors

- si $e > s$, alors $t' \leq t \leq T'$ et $(e-1, t')$ est applicable,
- si $e \in \{1, \dots, s\}$, alors $t' \leq t \leq T_{e+1}$. D'après la définition de p , pendant l'intervalle $[T_e, T_{e+1})$, il existe dans la file de e au moins θ paquets de la même classe que p . Ainsi, $t' < T_e$ et $(e-1, t')$ est applicable.

D'après (V.8) et (V.9),

$$\begin{aligned} |P_e(t)| &\leq f(e-1, t') - \varepsilon(t - t') + b + n|\mathcal{S}|\theta \\ &= f(e-1, t) + b + n|\mathcal{S}|\theta \\ &= f(e, t). \end{aligned}$$

Ce qui prouve le lemme V.11.

□

Nous continuons donc avec la preuve du théorème V.9. Par définition de T' , le fait que (s, T') est applicable et par définition de f , on a :

$$Q' < |P_s(T')| \leq f(s, T') = Q - \varepsilon(T' - T_0) + (s + 1)(b + n|\mathcal{S}|\theta).$$

D'après la définition de T' , $Q \leq Q'$, on a

$$\varepsilon(T' - T_0) < (n + 1)(b + n|\mathcal{S}|\theta).$$

D'autre part, considérons un paquet dans $P_s(T')$. Soit il était dans le système en T_0 , soit a été injecté pendant l'intervalle $[T_0, T']$. Le premier cas est de cardinal borné par $r(T' - T_0 + 1) + b$. Pour borner le deuxième cas, l'observation V.10 garantie que tout paquet dans ce cas sont restés dans une file avec mois de θ paquets de la même classe dans l'intervalle $[T_0, T']$. Alors, il existe au plus $n|\mathcal{S}|\theta(T' - T_0)$ paquets dans ce cas.

Donc,

$$Q' \leq |P_s(T')| \leq r(T' - T_0 + 1) + b + n|\mathcal{S}|\theta(T' - T_0).$$

Comme $\varepsilon(T' - T_0) < (n + 1)(b + n|\mathcal{S}|\theta)$, une contradiction apparaît en choisissant

$$Q' = (1/\varepsilon)(r + n|\mathcal{S}|\theta)(n + 1)(b + n|\mathcal{S}|\theta) + r + b.$$

□

Corollaire V.12. *Les cycles orientés sont universellement stables.*

3.3 Caractérisation des réseaux stables

Dans cette section nous caractérisons les réseaux universellement stables dans le modèle multi-classe. L'argument est une adaptation pour le cadre multi-classe de la preuve de Goel [GOE01] pour le modèle mono-classe.

Nous remarquons que tous les réseaux instables du modèle mono-classe sont aussi instables pour notre extension. Ainsi, tous les résultats d'instabilité de [ABS04] sont aussi valables pour le nouveau cadre.

Lemme V.13. *Soient G_1 et G_2 deux graphes orientés. Soit G' un graphe obtenu à partir de G_1 et G_2 en ajoutant des liens orientés depuis des sommets de G_1 vers des sommets de G_2 . Si G_1 et G_2 sont universellement stables alors G' est universellement stable.*

Démonstration. Considérons un réseau avec délai Δ et un (r, b) adversaire A' avec $r < 1$.

Comme G_1 est universellement stable contre l'adversaire induit par A' mais restreint à G_1 , alors il existe une constante C_1 (qui dépend du coût de délai, (r, b) , de la taille du réseau et de la configuration initiale) qui borne le nombre de paquets dans G_1 .

Les arêtes de G' qui connectent G_1 et G_2 sont appelés les *ponts* de G' . Soit \tilde{G} le graphe induit par G_1 avec les extrémités des ponts.

Considérons l'intervalle de temps $[t_1, t_2)$ et un (r, b) adversaire qui injecte les paquets dans \tilde{G} . Chaque paquet qui doit traverser un pont e pendant $[t_1, t_2)$, soit a été injecté dans \tilde{G} dans cet intervalle, soit était dans G_1 au début de l'intervalle, soit il était un parmi les C_e paquets dans la file de e à $t = 0$. Le nombre de paquets injectés est borné par $r(t_2 - t_1) + b$ et le nombre de paquets présents dans le réseau au début de l'intervalle est borné par $C_1 + C_e$. Ainsi, au plus $r(t_2 - t_1) + (b + C_1 + C_e)$ paquets doivent traverser une arête pont pendant l'intervalle $[t_1, t_2)$. D'après le théorème V.6, les graphes orientés acycliques sont universellement stables, donc la taille de files d'attente de chaque pont est bornée par une constante qui dépend de (r, b) , du coût de délai, de la taille du réseau \tilde{G} , de la configuration initiale et des paramètres du protocole d'ordonnancement. Nous concluons que les files d'attente de toute arête de \tilde{G} restent bornées indépendamment du temps.

Nous construisons un adversaire A_2 qui émule le comportement de A' sur G_2 et qui injecte tous les paquets qui ont destination en G_2 . Soit p un paquet injecté par A' . Si la source de p est dans G_2 , alors A_2 injecte un paquet avec le même chemin et au même temps que A' . Si la source de p est dans G_1 alors A_2 injecte un paquet avec le chemin de p restreint à G_2 mais le paquet est injecté au moment où p traverse une arête pont de G' .

Nous remarquons qu'un paquet injecté par A_2 qui doit traverser une arête de G_2 pendant l'intervalle $[t_1, t_2)$ soit a été injecté par A' dans \tilde{G} dans le même intervalle, soit était déjà dans \tilde{G} au début de l'intervalle. Il y a au plus $r(t_2 - t_1) + b$ paquets injectés pendant l'intervalle, et au plus \tilde{C} paquets dans les files d'attente de \tilde{G} . Ainsi, A_2 est un $(r, b + \tilde{C})$ adversaire. Du fait que G_2 est universellement stable, nous concluons qu'il existe une constante C_2 qui borne le nombre de paquets dans G_2 . Ainsi, le nombre de paquets dans G' reste borné par $\tilde{C} + C_2$ contre les injections de A' . \square

Une conséquence directe est

Théorème V.14. *Pour le modèle d'adversaire multi-classe, un réseau est universellement stable si et seulement si ses composantes fortement connexes sont universellement stables.*

Le théorème précédent permet de nous concentrer sur les graphes fortement connexes. De plus, si un graphe est fortement connexe et s'il n'est pas un cycle alors il contient un des graphes de la figure V.2 comme sous-graphe (pas nécessairement induit).

La caractérisation des graphes stables pour le cadre mono-classe a été prouvé par Álvarez, Blesa et Serna [ABS04]. Les auteurs donnent une collection de graphes instables, parmi lesquels se trouvent les graphes de la figure V.2 [ABS04, Theorem 3]. De plus, ils proposent deux opérations qui gardent l'instabilité d'un graphe :

1. *subdivision d'une arête* : étant donné une arête orientée $e = uv$, nous ajoutons un sommet w et nous remplaçons l'arête e par les deux nouvelles arêtes uw et wv .
2. *subdivision d'un cycle* : étant donné un 2-cycle (un cycle orienté qui a des arêtes dans les deux directions) nous ajoutons un nouveau sommet w et nous remplaçons les deux arêtes orientées $\{(u, v), (v, u)\}$ par les arêtes (u, w) , (w, u) , (v, w) et (w, v) .

Pour un graphe fortement connexe G , nous dénotons par $\varepsilon(G)$ les graphes qui peuvent être obtenus à partir de G par l'application consécutive des subdivisions (d'une arête ou d'un cycle) sur le graphe, nous remarquons que si G est instable alors tout graphe dans $\varepsilon(G)$ est aussi instable.

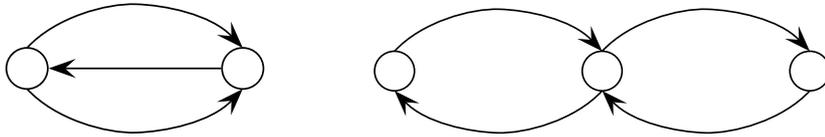


Figure V.2 – Graphes instables : à gauche U_1 , à droite U_2

Théorème V.15. *Pour le modèle d'adversaire multi-classe, un graphe est universellement stable s'il ne contient aucun graphe de $\varepsilon(U_1) \cup \varepsilon(U_2)$.*

Démonstration. Soit G un graphe universellement stable. D'après le théorème V.14 nous supposons que G est fortement connexe. Si G contient un graphe H appartenant $\varepsilon(U_1) \cup \varepsilon(U_2)$ alors il est instable, ce qui est une contradiction. Nous concluons que le graphe G est soit un cycle, soit un graphe acyclique qui sont universellement stables par le théorèmes V.6 et V.9 . \square

4 Résultat d'instabilité

Cette section montre que la stabilité ne s'étend pas directement du modèle avec une unique classe vers le modèle multi-classe. Plus précisément nous prouvons :

Théorème V.16. *Pour tout $0 < r < 1$, $\Delta \geq 0$, $\theta > 0$, et pour tout protocole d'ordonnancement (P, S) avec S un protocole de insensible au décalage, il existe un réseau G et un $(r, 1)$ adversaire borné A , tel que le système $(G, \Delta, (P, S_\theta); A)$ est instable.*

Nous utilisons une simplification du réseau présenté dans [BGL05] pour prouver l'instabilité de Premier-Arrivé-Premier-Servi pour des taux d'injection arbitrairement petits dans le modèle d'adversaire standard.

Le réseau du théorème V.16 est basé sur un sous-graphe basique appelé *gadget* qui permet de ralentir l'avancement des paquets vers leur destination. Le retard ainsi

créé à l'arrivée des paquets permet à l'adversaire d'injecter des paquets de façon qu'ils arrivent tous ensemble en un point spécifique du réseau. Pour la clarté des arguments, nous supposons $\Delta = 0$. Néanmoins, la preuve peut être facilement adaptée au cas général où $\Delta > 0$.

Les classes et ses étiquettes : Notre construction utilise au moins quatre classes de paquets. Du fait que S est insensible au décalage, il existe une bijection entre l'ensemble des classes \mathcal{S} et l'ensemble $\{0, 1, 2, 3\}$ telle que si pour un état donné du réseau, le protocole S se met à servir la classe j au lieu de la classe i , alors si on re-étiquette tout paquet i par $i + 1 \pmod{4}$, le protocole S se mettra à servir la classe $j + 1$ au lieu de $i + 1$. Par convenance, nous re-étiquetons les classe 2 et 3 par $\tilde{0}$ et $\tilde{1}$ respectivement.

Le réseau : La structure de base du réseau est appelée *gadget*. Un gadget contient deux sommets v_0 et v_1 , et trois paires d'arêtes (figure V.3) :

1. une paire d'arêtes d'entrée i_0 et i_1 ,
2. une paire d'arêtes de sortie o_0 et o_1 , et
3. deux paires de charge $e_0 = (v_0, v_1)$ et $e_1 = (v_1, v_0)$.

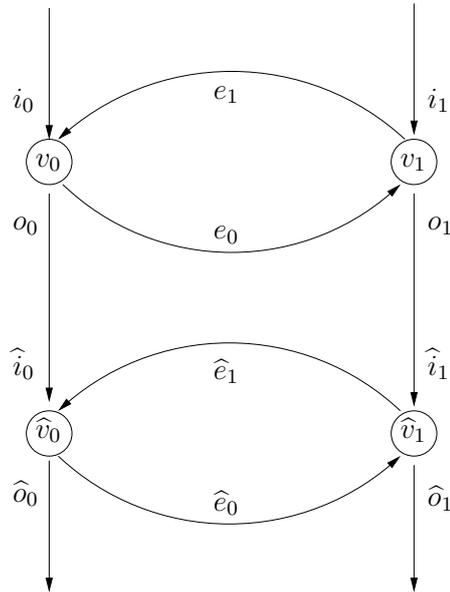


Figure V.3 – Deux gadgets enchaînés.

Nous disons que le gadget G_1 est enchaîné au gadget G_2 si les arêtes de sortie de G_1 correspondent aux arêtes d'entrée de G_2 . La figure V.3 montre un exemple de deux gadgets enchaînés. Un gadget peut être enchaîné à plusieurs gadgets et plusieurs gadgets peuvent être enchaînés à un gadget.

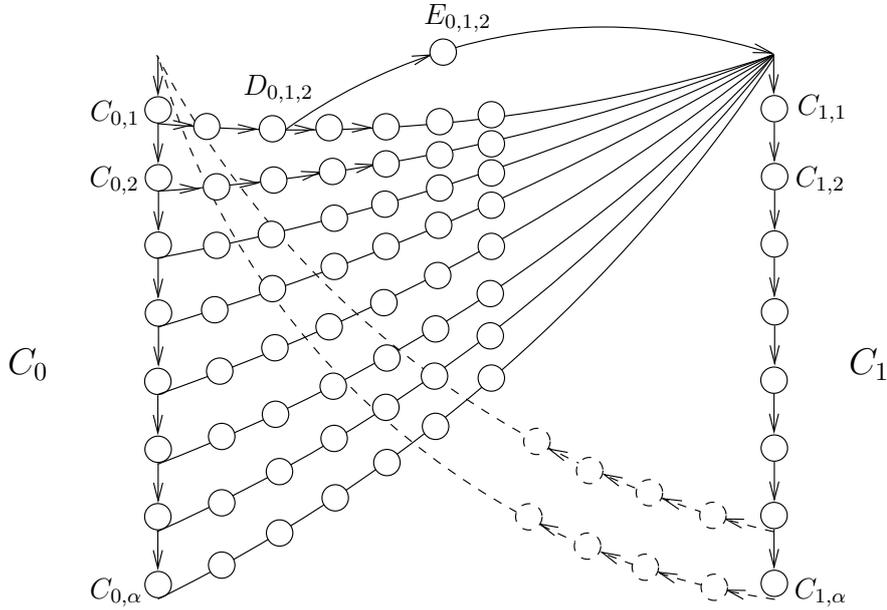


Figure V.4 – Réseau G .

Soit $n > 0$. Une *chaîne* de longueur n est une séquence H_1, H_2, \dots, H_n de gadgets enchaînés consécutivement. Nous le notons $C = \langle H_1, H_2, \dots, H_n \rangle$. Un *pont* de longueur m entre deux gadgets G_1 et G_2 correspond à une chaîne $B = \langle G_1, H_1, \dots, H_m, G_2 \rangle$.

Le réseau est composé de trois composantes principales : les colonnes, les connecteurs et les raccourcis. Il y a deux colonnes C_0 et C_1 , chacune est une chaîne de longueur α :

$$C_a = \langle C_{a,1}, C_{a,2}, \dots, C_{a,\alpha} \rangle.$$

Les connecteurs sont des chaînes de longueur β entre chaque gadget d'une colonne et le premier gadget de l'autre colonne. Donc, pour chaque gadget $C_{a,i}$ il existe un pont

$$\langle C_{a,i}, D_{a,i,1}, D_{a,i,2}, \dots, D_{a,i,\beta}, C_{a',1} \rangle.$$

La valeur de a' est 1 si a est égal à 0 et 0 si a est égal à 1. Les valeurs de α et β sont déterminées par la suite.

Finalement, les raccourcis sont des ponts de longueur 1 entre chaque gadget de chaque connecteur $D_{a,i,j}$ et le premier gadget de l'autre colonne $C_{a',1}$. Nous le notons par $E_{a,i,j}$ un tel raccourci. Une schéma du réseau est donnée dans la figure V.4.

L'adversaire : Nous disons qu'un paquet *traverse un gadget* s'il traverse les arêtes $i_a, e_a, e_{a'}, o_a$. Un paquet *traverse une chaîne* s'il traverse chacun des gadgets de la chaîne. Nous supposons que à l'instant initial il y a l paquets qui veulent traverser une

colonne, par exemple C_0 . Pendant que les paquets traversent le gadget $C_{0,i}$ l'adversaire injecte des paquets qui :

- utilisent une arête de charge du gadget,
- ensuite ils traversent le connecteur $\langle C_{0,i}, D_{0,i,1}, D_{0,i,2} \dots, D_{0,i,\beta}, C_{1,1} \rangle$ enchaîné au gadget $C_{0,i}$ vers $C_{0,1}$ jusqu'au
- ils prennent le raccourci $E_{0,i,j}$ (nous détaillons par la suite le choix de ce raccourci) et
- finalement ils traversent la colonne C_1 .

L'adversaire utilise les raccourcis de manière que les $l' > l$ paquets injectés arrivent simultanément vers les arêtes d'entrée du gadget $C_{1,1}$. Ainsi, nous nous trouvons dans un scénario symétrique à celui du début. Nous remplaçons donc l par l' et nous répétons le schéma d'injection de l'adversaire en accumulant un quantité plus grande de paquets. L'application consécutive de cette schéma d'injection produit l'instabilité du système.

Nous détaillons le schéma d'injection de l'adversaire. L'adversaire injecte quatre classes de paquets : 0, $\tilde{0}$, 1, et $\tilde{1}$. Les paquets de classe 0 et 1 traversent la colonne C_0 , tandis que les paquets de classe $\tilde{0}$ et $\tilde{1}$ traversent la colonne C_1 . De plus, les gadgets dans le connecteur $D_{0,i,j}$ (respectivement $D_{1,i,j}$) sont traversés seulement par des paquets de classe $\tilde{0}$ et $\tilde{1}$ (respectivement de classe 0 et 1).

Pour mieux décrire l'injection de paquets, nous introduisons le concept d'*activation d'un gadget* qui consiste en trois phases : la pre-condition, l'injection et la post-condition. Considérons une colonne a et son i -ème gadget $C_{a,i}$.

- Dans la pre-condition, nous supposons que pendant l pas de temps, il y a un paquet qui arrive à l'arête d'entrée et qui veut traverser le gadget $C_{a,i}$. Le paramètre l dépend du temps. De plus, ces paquets traversent la chaîne $\langle C_{a,i}, C_{a,i+1}, \dots, C_{a,\alpha} \rangle$.
- La phase d'injection du gadget $C_{a,i}$ dure l pas de temps. Elle commence avec l'arrivée du premier paquet de la pre-condition aux files d'arêtes de charge de $C_{a,i}$. Pendant les premier pas de temps de la phase d'injection, l'adversaire n'injecte aucun paquet. Pendant les $l^- = l - 1$ pas suivants l'adversaire injecte $lr/2$ paquets dans les sommets v_0 et v_1 du gadget $C_{a,i}$. Les paquets injectés dans le sommet v_k veulent traverser l'arête e_k , puis le connecteur $C_{a,i}$ jusqu'au raccourci $E_{a,i,j}$ (que nous décrirons par la suite) et finalement traverser la colonne $C_{a'}$. Les paquets injectés dans le sommet v_k sont de classe \tilde{k} si le gadget appartient à la colonne C_0 et de classe k si le gadget appartient à la colonne C_1 . Plus précisément, dans la colonne C_0 l'adversaire injecte des paquets de classe $\tilde{0}$ et $\tilde{1}$, tandis que dans la colonne C_1 l'adversaire injecte des paquets de classe 0 et 1. De plus, à la fin de la phase d'injection tous les paquets en la file d'attente dans la pre-condition ont déjà traversé au moins un arête de charge du gadget $C_{a,i}$.
- La post-condition du gadget $C_{a,i}$ est exactement la pre-condition du gadget $C_{a,i+1}$.

L'activation de gadgets dans les connecteurs et dans les raccourcis est définie de façon analogue, sauf que pendant la phase d'injection aucun paquet n'est injecté dans le gadget.

La période du temps entre l'arrivée et le départ du premier paquet qui traverse le gadget est appelée la *période d'activation* du gadget.

Le temps est divisé en étapes. Au début d'une étape paire s , nous garantirons que pendant $l = l(s)$ pas de temps consécutifs, un paquet de classe k qui traverse la colonne C_0 sera sur la file d'attente de l'arête d'entrée i_k du gadget $C_{0,1}$. Nous coordonnerons les paquets injectés pendant cette étape de façon à ce qu'au début de l'étape suivante $s + 1$, pendant au moins $2l$ pas de temps consécutifs, un paquet de classe \tilde{k} sera dans la file d'attente de l'arête d'entrée i_k de $C_{1,1}$. De plus, ces paquets veulent traverser la colonne C_1 .

Chaque étape est à son tour subdivisé en α sous-étapes. Pendant la sous-étape i , le gadget $C_{0,i}$ est actif et l'adversaire injecte $lr/2$ paquets de classe \tilde{k}' pour chaque $k \in \{0, 1\}$. Les routes suivies par les paquets injectés sont celles décrites ci-dessus.

La période d'activation de chaque gadget est donnée par :

Lemme V.17. *Si $lr^2/4 > \theta$, alors la période d'activation de chaque gadget d'une colonne est soit l , soit $l(1 + r/2)$ pas de temps. La période d'activation d'un gadget dans un connecteur ou dans un raccourci est $lr/2$ pas de temps. De plus, la période de la phase d'injection est l pas de temps et à chaque pas de temps au plus un gadget est dans sa phase d'injection.*

Démonstration. Sans perte de généralité, nous supposons que le système est dans une étape paire. Les injections sont faites donc dans la colonne C_0 . Considérons un gadget $C_{0,i}$. Au pas de temps juste avant la phase d'injection sur le gadget $C_{0,i}$, un paquet de classe k traverse l'arête d'entrée i_k et veut utiliser l'arête e_k . De plus, aucun paquet est injecté dans le premier pas de temps de la phase d'injection. Ainsi, seulement des paquets de classe k sont dans la file d'attente de e_k au début de la phase d'injection.

La définition de la pre-condition garantit que, pendant la phase d'injection, un paquet de classe k sera dans la file d'attente de l'arête e_k . Comme le protocole d'ordonnancement est glouton, un paquet de classe k traverse l'arête e_k . Le protocole continue à envoyer cette classe de paquets pendant les l pas de temps de la phase d'injection. Pendant les $l^- = l - 1$ derniers pas de temps de la phase d'injection, l'adversaire injecte $lr/2$ paquets de classe \tilde{k}' qui veulent utiliser l'arête e_k . À la fin de la phase d'injection, il ne reste aucun paquet de classe k dans la file d'attente de e_k . De plus, exactement l paquets de classe k' et $lr/2$ paquets de classe \tilde{k}' sont dans la file d'attente de e_k .

L'hypothèse garantit que $l \geq lr/2 \geq \theta$ et donc, le protocole de changement doit choisir pour chaque arête de charge, entre les paquets qui traversent la colonne (de classe k') et les paquets qui viennent d'être injectés (de classe \tilde{k}').

- Si le protocole de changement choisit de servir les nouveaux paquets de classe \tilde{k}' , alors la période d'activation dure $l + lr/2$ pas de temps, puisque les paquets qui traversent la colonne C_0 sont retenus par les nouveaux paquets.
- Si le protocole de changement choisit de servir la classe de paquets de classe k' qui traversent la colonne, alors la période d'activation dure l pas de temps.

Nous remarquons que dans les deux cas, le protocole de changement fait exactement le même choix pour les deux arêtes de charge, puisque le protocole est insensible au décalage.

L'analyse du temps d'activation des gadgets appartenant aux connecteurs ou aux raccourcis est analogue. Dans ces cas, $lr/2$ paquets traversent les gadgets et aucune injection est réalisée. De plus, d'après l'hypothèse $lr/2 \cdot r/2 > \theta$, la période d'activation dure $lr/2$ pas de temps. \square

Finalement, nous montrons que l'adversaire A ainsi décrit, est borné.

Lemme V.18. *L'adversaire A est un $(r, 1)$ adversaire.*

Démonstration. D'après le lemme V.17, pour chaque pas de temps, l'adversaire peut injecter des paquets sur un seul gadget. De plus, pour chaque intervalle de longueur l^- au plus $lr/2$ paquets sont injectés sur chacune des arêtes d'entrée (si $lr/2 > 1$). D'autre part, $lr < l^-r + 1$, donc les injections respectent la condition de charge. Ainsi, pour chaque intervalle de longueur $[t_1, t_2]$ l'adversaire n'injecte jamais plus que $r(t_2 - t_1) + 1$ paquets qui veulent utiliser une même arête. \square

Instabilité : Finalement, nous prouvons l'instabilité du système :

Démonstration du Theoreme V.16. D'après le lemme V.17 nous avons que le $(j+1)$ -ème gadget dans un connecteur $D_{a,i,j+1}$ est activé au plus $l + jlr/2 \leq l + \beta lr/2$ pas de temps après le gadget $C_{a,i}$ (l pas de temps pour que les nouveaux paquets commencent à traverser le gadget $C_{a,i}$ et $jlr/2$ pas de temps pour traverser les premiers j -gadgets du pont).

Nous devons garantir que :

- les ponts sont suffisamment longs de façon que les paquets qui traversent la colonne C_0 arrivent à la fin de la colonne avant que les nouveaux paquets injectés par l'adversaire traversent son connecteur, et
- le nombre de paquets injectés par l'adversaire est plus grand que l et qu'ils arrivent simultanément au gadget $C_{a',1}$.

La première condition peut être décrite par l'inégalité $l + \beta lr/2 \geq \alpha l(1 + r/2)$. En choisissant $\beta \geq \alpha(1 + 2/r)$ nous garantissons que les paquets injectés dans la colonne C_a n'arriveront pas à la colonne $C_{a'}$ avant que le dernier gadget $C_{a,\alpha}$ finisse sa période d'activation.

D'autre part, pour $\alpha \geq 4/r$, le nombre de paquets injectés par l'adversaire est $\alpha lr/2 \geq 2l$. Par l'utilisation adéquate des raccourcis, les nouveaux paquets peuvent arriver sans interruption aux arêtes d'entrée du gadget $C_{a',1}$.

En résumé, si $\beta \geq \alpha(1 + 2/r)$, $\alpha \geq 4/r$, et $l \geq 4\theta/r^2$, alors l'adversaire A est un $(r, 1)$ qui rend le système $(G, 0, (P, S_\theta); A)$ instable. \square

Remarque V.19. Une analyse plus soignée permet de montrer que dans le théorème V.16, la condition d'insensibilité au décalage peut être omise. En effet, pour le protocole Priorité la preuve est aussi valable.

5 Modèle fluide

Un outil important dans la théorie des files d'attente est l'utilisation du modèle fluide associé au système de routage. Le modèle fluide consiste en une réduction de la dynamique du réseau à une ensemble d'équations. Il correspond à une formulation analytique du système de routage. Les équations donnent un comportement asymptotique de la dynamique.

Nous présentons en Annexe la application de ce modèle pour le cadre multi-classe.

Perspectives

Les questions soulevées par cette thèse portent principalement sur l'hyperbolicité.

En effet, l'algorithme naïf en $O(n^4)$ devient vite impraticable pour de graphes de grande taille, d'où la nécessité d'écrire un algorithme en $o(n^4)$ pour le cas général. Les résultats de décomposition dans ce thèse sont un premier pas, mais un tel algorithme reste à écrire. Pour les graphes de terrains, il est probable que les différentes décompositions (en particulier par séparateurs de petit diamètre) réduisent considérablement la taille du problème.

Par ailleurs, nous avons constaté empiriquement dans le chapitre III que les graphes ayant leur degrés en loi de puissance (*power-law*) ont une hyperbolicité petite. La preuve formelle de cette question est souhaitable.

Nous concluons aussi dans le chapitre III l'existence d'un cœur bien connecté à l'intérieur d'Internet. Un travail envisageable est l'analyse détaillé du cœur et la mesure de son importance pour la connectivité globale.

Nous constatons que les quadruplets des graphes d'Internet ont, sauf une petite proportion, une valeur de l'hyperbolicité zero ou un. Une direction de recherche peut être l'étude des algorithmes de routage sur des graphe ayant hyperbolicité zero sauf dans une petite proportion des quadruplets.

D'autres questions ouvertes consistent à améliorer des bornes prouvées ici : l'inégalité liant triangles δ -fins et hyperbolicité des graphes est mérite d'être comblé (car on atteint l'égalité dans le cas des espaces géodésiques). De même la borne du chapitre IV en $1/\gamma^3$ liant la dimension de grille γ avec la modularité peut sans doute être améliorée.

Enfin, de bonnes propriétés structurelles pour caractériser les classes de graphe de modularité asymptotiquement un sont à trouver, en dehors des exemples ici abordés.

Annexe

1 Modèle fluide pour le files d'attente contre un adversaire

Le modèle fluide a été introduit par Rybko et Stolyar [RS92] pour le files d'attente stochastiques. De manière indépendante Dai [DAI95] et Stolyar [A.L95] ont établi une relation entre la stabilité du réseau de files d'attente et le modèle fluide associé.

Pour le modèle d'adversaire standard Gamarnik [GAM98, Theoreme 1] montrent comment la stabilité d'un système de routage peut être obtenue en considérant le modèle fluide. Nous généralisons ce résultat pour le cadre multi-classe.

1.1 Définitions et notations

Considérons le système de réseau $(G = (V, E), \Delta, (P, S); A)$. Soit \mathcal{R} l'ensemble des chemins simples dans le réseau G . Pour chaque chemin $P \in \mathcal{R}$ de taille $\ell(P)$, la séquence $e_0^P \dots e_{\ell(P)}^P$ correspond aux arêtes dans P . Nous notons par $A_{P,i}(t_1, t_2)$ le nombre de paquets de classe i injectés par l'adversaire dans l'intervalle $[t_1, t_2]$ qui doivent traverser le chemin P . De plus, notons $A_{e,i}(t_1, t_2)$ la somme de tous les paquets dans $A_{P,i}(t_1, t_2)$ où l'arête e appartient au chemin P : $A_{e,i}(t_1, t_2) = \sum_{P:e \in P} A_{P,i}(t_1, t_2)$.

Soit $B_{e,i}(t_1, t_2)$ le nombre de paquets de classe i qui arrivent à la file d'attente de e pendant l'intervalle $[t_1, t_2]$. Le nombre de paquets de classe i qui traversent l'arête e pendant l'intervalle $[t_1, t_2]$ et qui suivent le chemin P est noté $D_{P,e,i}(t_1, t_2)$.

Nous notons $Q_{e,i}(t)$ le nombre de paquets de classe i dans la file d'attente de l'arête e au temps t . Finalement, $S_e(t_1, t_2)$ et $Y_e(t_1, t_2)$ correspondent aux nombre de pas de temps pendant lesquels l'arête e est restée en temps de délai et oisive respectivement pendant l'intervalle $[t_1, t_2]$.

Pour l'intervalle de la forme $[0, t)$, nous utilisons une notation simplifiée avec une unique argument. Par exemple, $A_{P,i}(0, t)$ est noté $A_{P,i}(t)$. Nous utilisons la même

notation pour B , D , S , et Y . Finalement, si nous omettons un sous-indice, nous supposons que la fonction correspond à la somme sur toutes les valeur de l'indice manquant, comme par exemple $Q_e(t) = \sum_{i \in \mathcal{I}} Q_{e,i}(t)$.

1.2 Equations du modèle discret

Soit \mathcal{R}_e l'ensemble de chemins qui commencent avec l'arête e et $\mathcal{R}_{f,e}$ l'ensemble de chemins où l'arête e est utilisé juste après l'arête f (mais qui ne commencent pas nécessairement en f).

La dynamique d'injection par l'adversaire est décrite par les conditions de charge (V.1) et la restriction sur les délais (V.2).

Le nombre de paquets injectés jusqu'au temps t dans l'arête e est égal à $\sum_{P \in \mathcal{R}_e} A_{P,i}(t)$. De manière analogue, le nombre de paquets qui arrivent à la file d'attente de e depuis une autre arête dans le même intervalle de temps est égal à $\sum_{f \in T(e)} \sum_{P \in \mathcal{R}_{f,e}} D_{P,f,i}(t)$, où $T(e)$ est l'ensemble d'arêtes entrants de l'arête e . Ainsi, nous avons l'équation suivante pour toute classe $i \in \mathcal{I}$ et toute arête $e \in E(G)$:

$$B_{e,i}(t) = \sum_{P \in \mathcal{R}_e} A_{P,i}(t) + \sum_{f \in T(e)} \sum_{P \in \mathcal{R}_{f,e}} D_{P,f,i}(t), \quad (1)$$

De plus, le nombre de paquets de classe i dans la file d'attente de l'arête e correspond aux paquets présents dans la file au début de la dynamique, plus les paquets qui sont entés dans la file, moins les paquets qui ont traversé l'arête e . Formellement, pour chaque $i \in \mathcal{I}$ et $e \in E(G)$:

$$Q_{e,i}(t) = Q_{e,i}(0) + B_{e,i}(t) - D_{e,i}(t) \quad (2)$$

Finalement, nous avons la relation suivante :

$$S_e(t) + D_e(t) + Y_e(t) = t. \quad (3)$$

Considérons les processus discrètes : $\vec{A}(t) = (A_{P,i}(t))_{P,i}$, $\vec{B}(t) = (B_{e,i}(t))_{e,i}$, $\vec{Y}(t) = (Y_e(t))_e$, $\vec{Q}(t) = (Q_{e,i}(t))_{e,i}$, $\vec{D}(t) = (D_{P,e,i}(t))_{P,e,i}$ et $\vec{S}(t) = (S_{e,i}(t))_{e,i}$.

La séquence $(\vec{A}(t), \vec{B}(t), \vec{D}(t), \vec{Q}(t), \vec{S}(t), \vec{Y}(t))$ de processus vecteurs qui dépendent du temps est une *réalisation* du modèle discret s'il satisfait (V.1) - (V.2) et (1) - (3).

Pour le modèle multi-classe, la notion de protocole glouton peut être décrite comme l'absence de temps d'oisiveté quand une file d'attente contient un paquet. Formellement :

$$\forall t \in [t_1, t_2), Q_e(t) > 0 \implies Y_e(t_1) = Y_e(t_2),$$

1 Modèle fluide pour le files d'attente contre un adversaire

Définition .20 (Stabilité du modèle discret). Nous disons qu'une réalisation est *stable* si le nombre de paquets dans le réseau reste borné indépendamment du temps :

$$\exists M > 0, \quad \sup_t \sum_{e \in E} Q_e(t) < M.$$

Finalement, un réseau de routage multi-classe G est *universellement stable* contre un adversaire A , s'il est stable pour tout coût de délai Δ et pour tout protocole d'ordonnancement glouton.

1.3 Équations du modèle fluide

Le modèle fluide du réseau de routage correspond à une approximation continue du modèle discret. Nous associons à chaque fonction du modèle discret, une fonction continue avec des valeurs réelles. Cette fonction est croissante positive. Plus précisément, pour chaque arête $e \in E(G)$, pour chaque chemin du réseau P et pour chaque classe $i \in \mathcal{I}$, ils existe les fonctions : $\hat{A}_{P,i}(t)$, $\hat{B}_{e,i}(t)$, $\hat{D}_{P,e,i}(t)$, $\hat{Q}_{e,i}(t)$, $\hat{S}_{e,i}(t)$, et $\hat{Y}_e(t)$. Nous définissons aussi les vecteurs $\hat{A}(t)$, $\hat{B}(t)$, $\hat{D}(t)$, $\hat{Q}(t)$, $\hat{S}(t)$ et $\hat{Y}(t)$ de manière analogue au modèle discret.

La séquence $(\hat{A}(t), \hat{B}(t), \hat{D}(t), \hat{Q}(t), \hat{S}(t), \hat{Y}(t))$ est une *solution fluide* pour le modèle d'adversaire multi-classe si elle satisfait l'ensemble des équations suivantes pour chaque $i \in \mathcal{I}$, $P \in \mathcal{R}$, et $e \in E$:

$$\hat{A}_P(t) \leq r \cdot t, \tag{4}$$

$$\hat{B}_{e,i}(t) = \sum_{P \in \mathcal{R}_e} \hat{A}_{P,i}(t) + \sum_{f \in T(e)} \sum_{P \in \mathcal{R}_{f,e}} \hat{D}_{P,f,i}(t), \tag{5}$$

$$\hat{Q}_{e,i}(t) = \hat{Q}_{e,i}(0) + \hat{B}_{e,i}(t) - \hat{D}_{e,i}(t), \tag{6}$$

$$\hat{S}_e(t) + \hat{D}_e(t) + \hat{Y}_e(t) = t. \tag{7}$$

Une solution fluide est *gloutonne* si

$$\forall t \in [t_1, t_2], \quad \hat{Q}_e(t) > 0 \implies \hat{Y}_e(t_1) = \hat{Y}_e(t_2).$$

Définition .21 (Stabilité du modèle fluide). Une solution fluide $(\hat{A}(t), \hat{B}(t), \hat{D}(t), \hat{Q}(t), \hat{S}(t), \hat{Y}(t))$ est *stable* s'il existe un temps $\tau > 0$ où le réseau devient vide, autrement dit :

$$\exists \tau > 0, t \geq \tau \implies \sum_e \hat{Q}_e(t) = 0.$$

Un réseau est *globalement stable* pour le modèle fluide, s'il est stable pour toute solution fluide gloutonne où la taille des files d'attente au début de la dynamique est unitaire, i.e. $\sum_e \hat{Q}_e(0) = 1$.

1.4 Du modèle discret au modèle fluide

Pour énoncer le résultat principal de la section, nous devons établir la relation qui existe entre le modèle discret et le modèle de fluide d'une réalisation. Cette relation est faite via le résultat présenté par Gamarnik [GAM98] :

Proposition .22 (Proposition 1, [GAM98]). . *Étant donnée une réalisation du modèle discret $(\vec{A}(t), \vec{B}(t), \vec{D}(t), \vec{Q}(t), \vec{S}(t), \vec{Y}(t))$ et une suite croissante $t_1, t_2, \dots, t_k, \dots$. Alors, il existe une suite de nombres entiers $k_1, k_2, \dots, k_n, \dots$ avec les propriétés suivantes :*

– *Les limites*

$$\lim_{n \rightarrow \infty} \frac{A(t_{k_n} + tk_n) - A(t_{k_n})}{k_n}, \quad \lim_{n \rightarrow \infty} \frac{B(t_{k_n} + tk_n) - B(t_{k_n})}{k_n}, \quad \lim_{n \rightarrow \infty} \frac{D(t_{k_n} + tk_n) - D(t_{k_n})}{k_n},$$

$$\lim_{n \rightarrow \infty} \frac{Q(t_{k_n} + tk_n)}{k_n}, \quad \lim_{n \rightarrow \infty} \frac{S(t_{k_n} + tk_n) - S(t_{k_n})}{k_n}, \quad \lim_{n \rightarrow \infty} \frac{Y(t_{k_n} + tk_n) - Y(t_{k_n})}{k_n},$$

existent pour chaque $t \in \mathbb{R}$. Les limites, notés par $\hat{A}(t)$, $\hat{B}(t)$, $\hat{D}(t)$, $\hat{Q}(t)$, $\hat{S}(t)$ et $\hat{Y}(t)$ sont appelées les limites de fluide¹.

– *Le processus $(\hat{A}(t), \hat{B}(t), \hat{D}(t), \hat{Q}(t), \hat{S}(t), \hat{Y}(t))$ est une solution du modèle fluide. De plus, si la réalisation discrète est gloutonne, alors la solution fluide associée est aussi gloutonne.*

Éléments de démonstration. D'abord, l'existence des limites est prouvée pour les nombres rationnels non-négatifs. Pour un rationnel $q \in \mathcal{Q}$ fixe, il est facile de vérifier que les suites $\frac{Q(t_{k_n} + tk_n)}{k_n}$, $\frac{A(t_{k_n} + tk_n) - A(t_{k_n})}{k_n}$, $\frac{B(t_{k_n} + tk_n) - B(t_{k_n})}{k_n}$, $\frac{D(t_{k_n} + tk_n) - D(t_{k_n})}{k_n}$, $\frac{S(t_{k_n} + tk_n) - S(t_{k_n})}{k_n}$ et $\frac{Y(t_{k_n} + tk_n) - Y(t_{k_n})}{k_n}$ sont bornés supérieurement et donc convergents. Un argument de sous-suite diagonale permet d'obtenir une sous-suite $k_1, k_2, \dots, k_n, \dots$ qui rend les suites ci-dessus convergentes pour toute nombre rationnel. De plus, comme le processus d'origine satisfait les équations (V.1) - (V.2) et (1) - (3), alors les suites ont la propriété de Lipschitz et donc ont une unique extension sur les nombres réels. Il est facile de vérifier que les limites correspondent à une solution fluide.

Nous prouvons que la solution fluide définie est gloutonne. Par l'absurde, supposons que $\hat{Q}_e(t) > 0$ pour tout $t \in [t_1, t_2]$. Comme la fonction $\hat{Q}_e(\cdot)$ est continue, car il est Lipschitz, sur l'intervalle (compact) $[t_1, t_2]$, il existe $\gamma > 0$ tel que $\hat{Q}_e(t) > \gamma$ pour tout $t \in [t_1, t_2]$. D'autre part $\hat{Q}_e(t) = \lim_{n \rightarrow \infty} Q_e(t_{k_n} + tk_n)/k_n$, donc pour n suffisamment grand, $Q_e(t_{k_n} + tk_n) > \gamma k_n$ pour tout $t \in [t_1, t_2]$. Comme la réalisation est gloutonne, alors $Y_e(t_{k_n} + t_1 k_n) = Y_e(t_{k_n} + t_2 k_n)$ pour n suffisamment grand, et donc $\hat{Y}_e(t_1) = \hat{Y}_e(t_2)$. \square

1. Par soucis de simplicité nous utilisons les mêmes notations que les processus de la section 1.3 auxquelles elles s'identifient.

1.5 Résultat principal

Nous avons les éléments pour énoncer le résultat principal de la section :

Théorème .23. *Soit G un réseau tel que le modèle de fluide associé est globalement stable, alors le réseau G est universellement stable.*

Ce résultat est une extension du résultat de Gamarnik [GAM98, Theoreme 1] pour le cadre mono-classe. La preuve découle du lemme suivante :

Lemme .24 ([GAM98], Lemme 3). *Soit $(\vec{A}(t), \vec{B}(t), \vec{D}(t), \vec{Q}(t), \vec{S}(t), \vec{Y}(t))$ une réalisation instable et $\tau > 0$. Alors il existe une suite croissante $t_1, t_2, \dots, t_k, \dots$ telle que pour tout $k \geq |Q(0)|$ on a $k \leq |Q(t_k)| \leq k + C$ et $|Q(t_k + \tau k)| \geq k$ où C est une constante qui dépend de la taille du réseau et des paramètres de l'adversaire.*

Démonstration du théorème .23. La preuve se fait par contradiction. Nous supposons que le réseau G n'est pas universellement stable. Nous allons construire une solution fluide associée qui n'est pas stable avec $\sum_e \hat{Q}_e(0) = 1$ ce qui montre que le modèle de fluide associé n'est pas globalement stable.

Soit $(A(t), B(t), D(t), Q(t), S(t), Y(t))$ une réalisation instable du réseau G contre un adversaire borné. Considérons nous la suite du lemme .24 : $t_1, t_2, \dots, t_k, \dots$. Supposons que la solution fluide associée à cette réalisation est stable avec $\tau > 0$ le temps où le réseau devient vide pour la solution fluide.

Pour toute sous-suite de $t_1, t_2, \dots, t_k, \dots$ on a que :

$$|\hat{Q}(0)| = \lim_{n \rightarrow \infty} \frac{Q(t_{k_n})}{k_n} \leq \lim_{n \rightarrow \infty} \frac{k_n + C}{k_n} = 1.$$

De plus, $|Q(t_{k_n})| \geq k_n$, alors $|\hat{Q}(0)| \geq 1$ et donc $|\hat{Q}(0)| = 1$.

D'autre part,

$$|\hat{Q}(\tau)| = \lim_{n \rightarrow \infty} \frac{Q(t_{k_n} + \tau k_n)}{k_n} \geq \frac{k_n}{k_n} \geq 1,$$

ce qui est une contradiction avec la stabilité du modèle fluide. □

Table des figures

II.1	Représentation graphique de la condition d'hyperbolicité pour $xv + yu \leq xu + yv \leq xy + uv$	11
II.2	Triangle géodésique δ -fin. La géodésique $[x, y]$ est incluse dans le δ -voisinage de $[x, z] \cup [y, z]$	12
II.3	Triangle géodésique δ -mince et son trépied associé. La condition de δ -fin garantit que si $\phi(a) = \phi(b)$ alors $d(a, b) \leq \delta$	13
II.4	Quadruplet de la proposition II.20.	14
II.5	Un grille plus un croisillon : un graphe n -fin et $2n$ -hyperbolique.	15
II.6	À gauche K_4 , à droite son graphe métrique X_{K_4} et un quadruplet maximal.	19
II.7	Graphes interdits 1-hyperboliques. Tous les graphes ont une hyperbolicité égale à un et $lc(H_1)=lc(H_2)=3, lc(H_3)=lc(H_4)=5, lc(G_1)=lc(G_4)=6, lc(E_1)=7, lc(E_2)=8$	20
II.8	Un quadruplet du théorème II.28	21
II.9	Décomposition modulaire en modules forts, son graphe quotient et l'arbre de décomposition modulaire. Les nœud internes son étiquetés comme premier (P), série (S) ou parallèle ($//$).	24
II.10	Un split $X_1 = V_1 \cup V_2, X_2 = V_3 \cup V_4$ et ses deux compactages $G_{X_2}^{v_1}, G_{X_1}^{v_2}$	26
II.11	Un graphe et son arbre de splits.	27
II.12	Séparateur isométrique et ses composantes bi-connexes.	29
III.1	Exemple de traceroute	36

Table des figures

III.2	Exemple d'une table BGP.	38
III.3	Taille des instantanés mensuels pour l'année 2010	41
III.4	Graphes des AS agrégés depuis Janvier 2010.	42
III.5	Évolution de la largeur arborescente des graphes des AS.	43
III.6	Bornes inférieures de la largeur arborescente pour les graphes des routeurs en comparaison avec \sqrt{n}	45
III.7	Variation de la distribution de l'hyperbolicité pour les graphes des AS (CAIDA, DIMES and UCLA).	49
III.8	Distribution des valeurs de l'hyperbolicité sur tous les quadruplets (Haut : Agrégé par dix mois. Bas : Mensuelle 2009/12)	50
III.9	Hyperbolicité moyenne en fonction de la distance plus petite entre les éléments du quadruplet. Haut : AS graphe. Bas : graphe de routeurs.	51
III.10	Modeles d'Internet. Haut : Distribution de l'hyperbolicité pour les quadruplets. Bas : Hyperbolicité moyenne en fonction de la plus petite distance du quadruplet.	52
IV.1	Partitionnement d'un graphe	54
IV.2	Une R -net et son partitionnement associé	66
IV.3	Valeurs de la modularité en fonction du rayon du R -net	69
IV.4	Nombre de clusters en fonction du rayon du R -net	69
IV.5	Valeurs de la modularité pour des clusters de taille variable : $a = R/L$	70
IV.6	Comparaison entre les valeurs de la modularité en utilisant l'algorithme glouton et K -means.	70
IV.7	Comparaison entre les valeurs de la somme de carrés au baricentre des clusters (fonction objective de K -means).	71
V.1	Arêtes entrants d'une arête e	80
V.2	Graphes instables : à gauche U_1 , à droite U_2	87
V.3	Deux gadgets enchaînés.	88
V.4	Réseau G	89

Références bibliographiques

- [AAF⁺01] Matthew Andrews, Baruch Awerbuch, Antonio Fernández, Tom Leighton, Zhiyong Liu, and Jon Kleinberg. Universal-stability results and performance bounds for greedy contention-resolution protocols. *J. of the ACM*, 48 :39–69, January 2001. 75, 77, 79, 82
- [ABC⁺90] J M Alonso, T Brady, D Cooper, V Ferlini, M Lustig, M Mihalik, M Shapiro, and H Short. Notes on word hyperbolic groups. in group theory from a geometrical viewpoint, 1990. 13
- [ABD⁺05] C. Álvarez, M. Blesa, J. Díaz, M. Serna, and A. Fernández. Adversarial models for priority-based networks. *Networks*, 45 :23–35, January 2005. 79
- [ABS04] C. Álvarez, M. Blesa, and M. Serna. A characterization of universal stability for directed graphs in the adversarial queueing model. *SIAM Journal on Computing*, 34(1) :41–66, October 2004. 79, 85, 86
- [AK08] Bruno D. Abrahao and Robert D. Kleinberg. On the internet delay space dimensionality. In *Internet Measurement Conference*, pages 157–168, 2008. 34
- [AKK08] Elliot Anshelevich, David Kempe, and Jon Kleinberg. Stability of load balancing algorithms in dynamic adversarial systems. *SIAM J. Comput.*, 37(5) :1656–1673, 2008. 79
- [A.L95] A.L.Stolyar. On the stability of multiclass queueing networks : A relaxed sufficient condition via limiting fluid processes. *Markov Processes and Related Fields*, 1 :491–512, 1995. 97
- [And04] Matthew Andrews. Instability of fifo in session-oriented networks. *Journal of Algorithms*, 50 :232–245, February 2004. 79

Références bibliographiques

- [AP89a] Stefan Arnborg and Andrzej Proskurowski. Linear time algorithms for np-hard problems restricted to partial k-trees. *Discrete Applied Mathematics*, 23(1) :11 – 24, 1989. 2
- [AP89b] Stefan Arnborg and Andrzej Proskurowski. Linear time algorithms for np-hard problems restricted to partial k-trees. *Discrete Appl. Math.*, 23(1) :11–24, 1989. 3
- [BA99] A. L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439), 1999. 35, 47
- [Bai97] R. A. Bailey. Cambridge University Press, 1997. 2, 45
- [BBS01] P. Berenbrink, A. Brinkmann, and C. Scheideler. Simple routing strategies for adversarial systems. In *FOCS '01 : Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, page 158, Washington, DC, USA, 2001. IEEE Computer Society. 79
- [Bc01] A. Broido and kc claffy. Internet topology : Connectivity of IP graphs. In *Proceedings of SPIE International Symposium on Convergence of IT and Communication*, 2001. 34
- [BC03] H.J. Bandelt and V. Chepoi. 1-Hyperbolic graphs. *SIAM journal on discrete mathematics*, 16(2) :323–334, 2003. 19
- [BC08] H. J. Bandelt and Victor Chepoi. Metric graph theory and geometry : a survey, *Surveys on Discrete and Computational Geometry : Twenty Years Later. Contemporary Mathematics*, 453 :49–86, 2008. 28
- [BDG⁺08] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20 :172–188, 2008. 53, 55, 56, 57, 58
- [BF94] F. Baccelli and S.G. Foss. Ergodicity of jackson-type queueing networks. *Queueing systems : Theory and Applications*, 17 :5–72, 1994. 78
- [BFM96] F. Baccelli, S.G. Foss, and J. Mairesse. Stationary ergodic jackson networks : results and counter-examples. *Stochastic Networks*, pages 281–307, 1996. 78
- [BGL05] Rajat Bhattacharjee, Ashish Goel, and Zvi Lotker. Instability of fifo at arbitrarily low rates in the adversarial queueing model. *SIAM J. Comput.*, 34 :318–332, February 2005. 79, 87
- [BGLL08] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics : Theory and Experiment*, 2008. 53

- [BH99] M.R. Bridson and A. Haefliger. *Metric spaces of non-positive curvature*. Springer Verlag, 1999. 9, 10, 13
- [BK01] Gunnar Brinkmann and Jack H. Koolen. On the hyperbolicity of chordal graphs. *Annals of Combinatorics*, 5 :61–69, 2001. 10.1007/s00026-001-8007-7. 20
- [BKR⁺96] Allan Borodin, Jon M. Kleinberg, Prabhakar Raghavan, Madhu Sudan, and David P. Williamson. Adversarial queueing theory. In *In Proc. 28th ACM STOC*, pages 376–385, 1996. 3, 73, 79
- [BKR⁺01] Allan Borodin, Jon Kleinberg, Prabhakar Raghavan, Madhu Sudan, and David P. Williamson. Adversarial queueing theory. *J. of the ACM*, 48 :13–38, January 2001. 3, 73, 75, 76, 79, 80
- [BLW87] M.W Bern, E.L Lawler, and A.L Wong. Linear-time computation of optimal subgraphs of decomposable graphs. *Journal of Algorithms*, 8(2) :216 – 235, 1987. 2, 3
- [BM86] Hans-Jürgen Bandelt and Henry Martyn Mulder. Distance-hereditary graphs. *Journal of Combinatorial Theory, Series B*, 41(2) :182 – 208, 1986. 16
- [Bod88] Hans L. Bodlaender. Dynamic programming on graphs with bounded treewidth. In *ICALP*, pages 105–118, 1988. 3, 34
- [Bor86] A.A. Borovkov. Limit theorems for queueing networks. *Theory of Probability and its Applications*, 31 :413–427, 1986. 78
- [Bra94a] M. Bramson. Instability of FIFO queueing networks. *Ann. Appl. Prob.*, 4 :414–431, 1994. 78
- [Bra94b] M. Bramson. Instability of FIFO queueing networks. *Ann. Appl. Prob.*, 4 :693–718, 1994. 78
- [CAI] The internet topology data kit - 2010-07. 39
- [CAI03] Caida’s internet topology data kit #0304, 2003. 39
- [CCG⁺02] Qian Chen, Hyunseok Chang, Ramesh Govindan, Sugih Jamin, Scott Shenker, and Walter Willinger. The origin of power-laws in internet topologies revisited. In *IEEE INFOCOM*, 2002. 34
- [CCP⁺09] Kai Chen, David Choffnes, Rahul Potharaju, Yan Chen, Fabian Bustamante, Dan Pei, and Yao Zhao. Where the sidewalk ends : extending the internet as graph using traceroutes from p2p users. In *CoNEXT*, pages 217–228, 2009. 39

Références bibliographiques

- [CDE⁺] Victor Chepoi, Feodor Dragan, Bertrand Estellon, Michel Habib, Yann Vaxès, and Yang Xiang. Additive spanners and distance and routing labeling schemes for hyperbolic graphs. *Algorithmica*, pages 1–20. 10.1007/s00453-010-9478-x. 2, 4, 7, 46
- [CDE⁺⁰⁸] Victor Chepoi, Feodor F. Dragan, Bertrand Estellon, Michel Habib, and Yann Vaxès. Diameters, centers, and approximating trees of delta-hyperbolic geodesic spaces and graphs. In *Symposium on Computational Geometry*, pages 59–68, 2008. 4, 7
- [CdMR09] Pierre Charbit, Fabien de Montgolfier, and Mathieu Raffinot. A simple linear time split decomposition algorithm of undirected graphs. *CoRR*, abs/0902.1700, 2009. 27, 31
- [CE80] William H. Cunningham and Jack Edmonds. A combinatorial decomposition theory. *Canadian Journal of Mathematics*, 32(3) :734–765, 1980. 26
- [CE07a] Victor Chepoi and Bertrand Estellon. Packing and covering δ -hyperbolic spaces by balls. In Moses Charikar, Klaus Jansen, Omer Reingold, and José D. P. Rolim, editors, *APPROX-RANDOM*, volume 4627 of *Lecture Notes in Computer Science*, pages 59–73. Springer, 2007. 4, 7
- [CE07b] Vicent Cholvi and Juan Echagüe. Stability of fifo networks under adversarial models : State of the art. *Comput. Netw.*, 51 :4460–4474, October 2007. 77
- [CH94] Alain Cournier and Michel Habib. A new linear algorithm for modular decomposition. In Sophie Tison, editor, *Trees in Algebra and Programming - CAAP'94*, volume 787 of *Lecture Notes in Computer Science*, pages 68–84. Springer Berlin / Heidelberg, 1994. 23
- [CHM81] M. Chein, M. Habib, and M. C. Maurer. Partitive hypergraphs. *Discrete Mathematics*, 37(1) :35 – 50, 1981. 23
- [CNM04] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70 :066111, 2004. 53
- [Cou01] Makowsky J.A. Rotics U. Courcelle, B. On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. *Discrete Applied Mathematics*, 108(1-2) :23–52, 2001. cited By (since 1996) 70. 3
- [Cru91a] R. L. Cruz. A calculus for network delay. Part I. *IEEE Trans. Inf. Theory*, 37 :114–131, Jan 1991. 78

- [Cru91b] R. L. Cruz. A calculus for network delay. Part II. *IEEE Trans. Inf. Theory*, 37 :132–141, Jan 1991. 78
- [CTK94] C.S. Chang, J.A. Thomas, and S.H. Kiang. On the stability of open networks ; a unified approach by stochastic dominance. *Queueing systems : Theory and Applications*, 15 :239–260, 1994. 78
- [Cun82] William H. Cunningham. Decomposition of directed graphs. *SIAM Journal on Algebraic and Discrete Methods*, 3(2) :214–228, 1982. 26
- [DA05] Jordi Duch and Alex Arenas. Community detection in complex networks using extremal optimization. *Phys. Rev. E*, 72(2) :027104, Aug 2005. 53
- [Dah00] Elias Dahlhaus. Parallel algorithms for hierarchical clustering and applications to split decomposition and parity graph recognition. *J. Algorithms*, 36(2) :205–240, 2000. 27
- [Dai95] J. G. Dai. On positive harris recurrence of multiclass queueing networks : A unified approach via fluid limit models. *Annals of Applied Probability*, 5 :49–77, 1995. 74, 78, 97
- [DCKM04] Frank Dabek, Russ Cox, M. Frans Kaashoek, and Robert Morris. Vivaldi : a decentralized network coordinate system. In *ACM SIGCOMM*, 2004. 34
- [DJ04] J. G. Dai and O. B. Jennings. Stabilizing queueing networks with setups. *Math. Oper. Res.*, 29 :891–922, November 2004. 5, 74, 76, 78
- [DKN⁺01] Josep Díaz, Dimitrios Koukopoulos, Sotiris Nikolettseas, Maria Serna, Paul Spirakis, and Dimitrios M. Thilikos. Stability and non-stability of the fifo protocol. In *SPAA '01 : Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures*, pages 48–52, New York, NY, USA, 2001. ACM. 79
- [DM95] J.G. Dai and S.P. Meyn. Stability and convergence of moments for networks and their fluid models. *IEEE Trans. Aut. Cont.*, 40 :1889–1904, 1995. 78
- [dM03] Fabien de Montgolfier. *Décomposition modulaire de graphes*. PhD thesis, Université Montpellier II, LIRMM, 2003. 23
- [DMS97] Andreas Dress, Vincent Moulton, and Michael Steel. Trees, taxonomy, and strongly compatible multi-state characters. *Advances in Applied Mathematics*, 19(1) :1 – 30, 1997. 16
- [dMSV11a] Fabien de Montgolfier, Mauricio Soto, and Laurent Viennot. Asymptotic modularity of some graph classes. In *22nd International Symposium on Algorithms and Computation (ISAAC)*, pages 1–10, Yokohama, 2011. 5, 6, 54

Références bibliographiques

- [dMSV11b] Fabien de Montgolfier, Mauricio Soto, and Laurent Viennot. Clustering de métrique et clustering de graphe. In *13es Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (ALGOTEL)*, pages 1–4, Cap Estérel, 2011. 5, 6, 54
- [dMSV11c] Fabien de Montgolfier, Mauricio Soto, and Laurent Viennot. Treewidth and hyperbolicity of the internet. In *10th IEEE International Symposium on Network Computing and Applications (IEEE NCA)*, pages 1–8, Boston, 2011. 5, 6, 34
- [FB07] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104 :36–41, 2007. 53
- [FFF99] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM*, pages 251–262, 1999. 34
- [FJ87] Martin Farber and Robert E. Jamison. On local convexity in graphs. *Discrete Mathematics*, 66(3) :231 – 247, 1987. 19
- [FKP02] Alex Fabrikant, Elias Koutsoupias, and Christos H. Papadimitriou. Heuristically optimized trade-offs : A new paradigm for power laws in the internet. In *ICALP*, pages 110–122, 2002. 35
- [FLV08] Pierre Fraigniaud, Emmanuelle Lebhar, and Laurent Viennot. The in-frametric model for the internet. In *IEEE INFOCOM*, pages 1085–1093, Phoenix, 2008. 34
- [Fos89] S.G. Foss. Some properties of open queueing networks. *Problems of Information Transmission*, 25 :241–248, 1989. 78
- [Fos91] S.G. Foss. Ergodicity of queueing networks. *Siberian Mathematical Journal*, 32 :184–203, 1991. 78
- [GA05] Roger Guimera and Luis A. Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028) :895–900, 2005. 53
- [Gae04] Marco Gaertler. Clustering. In Ulrik Brandes and Thomas Erlebach, editors, *Network Analysis*, volume 3418 of *Lecture Notes in Computer Science*, pages 178–215. Springer, 2004. 55
- [Gal67] T. Gallai. Transitiv orientierbare graphen. *Acta Mathematica Hungarica*, 18 :25–66, 1967. 10.1007/BF02020961. 22, 23
- [Gam98] David Gamarnik. Stability of adversarial queues via fluid models. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, FOCS '98, pages 60–70, Washington, DC, USA, 1998. IEEE Computer Society. 74, 79, 97, 100, 101

- [GdlHeds.90] E. Ghys and P. de la Harpe (eds.). *Sur les Groupes Hyperboliques d'après Mikhael Gromov*. Number 83 in Progr. Math. Birkhäuser, Boston, MA, 1990. (Papers from the Swiss Seminar on Hyperbolic Groups held in Bern, 1988). 4, 13, 17
- [GK03] Michael U. Gerber and Daniel Kobler. Algorithms for vertex-partitioning problems on graphs with fixed clique-width. *Theoretical Computer Science*, 299(1-3) :719 – 734, 2003. 2
- [GKBC04] Julien Gagneur, Roland Krause, Tewis Bouwmeester, and Georg Casari. Modular decomposition of protein-protein interaction networks. *Genome Biology*, 5 :1–12, 2004. 10.1186/gb-2004-5-8-r57. 22
- [GL05a] Cyril Gavoille and Olivier Ly. Distance labeling in hyperbolic graphs. In Xiaotie Deng and Ding-Zhu Du, editors, *Algorithms and Computation*, volume 3827 of *Lecture Notes in Computer Science*, pages 1071–1079. Springer Berlin / Heidelberg, 2005. 2
- [GL05b] Cyril Gavoille and Olivier Ly. Distance labeling in hyperbolic graphs. In Xiaotie Deng and Ding-Zhu Du, editors, *ISAAC*, volume 3827 of *Lecture Notes in Computer Science*, pages 1071–1079. Springer, 2005. 4, 7
- [GN02] M. Girvan and M.E.J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12) :7821, 2002. 2, 53, 54
- [Goe01] A. Goel. Stability of networks and protocols in the adversarial queueing model for packet routing. *Networks*, 37(4) :219–224, 2001. 79, 85
- [GR97] Ramesh Govindan and Anoop Reddy. An analysis of internet inter-domain topology and route stability. In *IEEE INFOCOM*, pages 850–857, 1997. 34
- [Gro87] M. Gromov. Hyperbolic groups. In *Essay in group theory*(S.M. Gerste ed.), number 8 in MSRI Series, pages 75–263. 1987. 2, 3, 4, 12, 13, 17, 31, 35
- [GSPA04] Roger Guimera, Marta Sales-Pardo, and Luis A. Nunes Amaral. Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E*, 70 :025101, 2004. 53
- [HDFC10] Bradley Huffaker, Amogh Dhamdhere, Marina Fomenkov, and Kc Claffy. Toward topology dualism : improving the accuracy of as annotations for routers. In *Proceedings of the 11th international conference on Passive and active measurement*, PAM'10, pages 101–110, Berlin, Heidelberg, 2010. Springer-Verlag. 39

Références bibliographiques

- [HHA⁺09] Young Hyun, Bradley Huffaker, Dan Andersen, Emile Aben, Matthew Luckie, kc claffy, and Colleen Shanno. The ipv4 routed /24 as links datase, 2009. [HTTP://WWW.CAIDA.ORG/DATA/ACTIVE/IPV4_ROUTED_TOPOLOGY_ASLINKS_DATASET.XML](http://www.caida.org/data/active/ipv4_routed_topology_aslinks_dataset.xml). 39
- [HM90] Peter L. Hammer and Frédéric Maffray. Completely separable graphs. *Discrete Applied Mathematics*, 27(1-2) :85 – 99, 1990. 28
- [Hon85] C. T. Honàg. A class of perfect graphs. Master’s thesis, School of Computer Science, McGill University, Montreal, 1985. 25
- [How79] Edward Howorka. On metric properties of certain clique graphs. *Journal of Combinatorial Theory, Series B*, 27(1) :67 – 74, 1979. 16
- [Jac63] James R. Jackson. Jobshop-Like Queueing Systems. *Manage. Sci.*, 10(1) :131–142, 1963. 73, 78
- [JO92] B. Jamison and S. Olariu. A tree representation for p4-sparse graphs. *Discrete Applied Mathematics*, 35(2) :115 – 129, 1992. 25
- [KA05] N. Kashtan and U. Alon. Spontaneous evolution of modularity and network motifs. *Proceedings of the National Academy of Sciences of the United States of America*, 102(39) :13773, 2005. 53
- [KCFB07] Dmitri V. Krioukov, Kc Claffy, K. Fall, and A. Brady. On compact routing for the internet. *ACM SIGCOMM*, 37(3), 2007. 46
- [Kel75] F. P. Kelly. Networks of queues with customers of different types. *Journal of Applied Probability*, 12 :542–554, September 1975. 73, 78
- [Kel79] F. P. Kelly. *Reversibility and Stochastic Networks*. Wiley, New York, NY, USA, 1979. 73, 78
- [KFY04] Dmitri V. Krioukov, Kevin R. Fall, and Xiaowei Yang. Compact routing on internet-like graphs. In *IEEE INFOCOM*, 2004. 46
- [KHK02] Arie M.C.A. Koster, C.P.M. van Hoesel, and A.W.J. Kolen. Solving partial constraint satisfaction problems with tree decomposition. *Networks*, 40(3) :170–180, 2002. <http://dx.doi.org/10.1002/net.10046>. 3
- [KM02] Jack H. Koolen and Vincent Moulton. Hyperbolic bridged graphs. *Eur. J. Comb.*, 23 :683–699, August 2002. 16, 19
- [KNS01] Dimitrios Koukopoulos, Sotiris E. Nikolettseas, and Paul G. Spirakis. The range of stability for heterogeneous and FIFO queueing networks. *Electronic Colloquium on Computational Complexity (ECCC)*, (099), 2001. 79

- [KR02] D. Karger and M. Ruhl. Finding nearest neighbors in growth-restricted metrics. In *Proceedings of STOC, 34th Symposium on Theory Of Computing*, pages 741–750. ACM, 2002. 65
- [KS90] P.R. Kumar and T.I. Seidman. Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems. *IEEE Trans. on Automatic Control*, 35 :289–298, 1990. 78
- [KST09] M. Kiwi, M. Soto, and C. Thraves. Adversarial queuing theory with setups. *Theor. Comput. Sci.*, 410 :670–687, March 2009. 5, 6, 74
- [KT05] Jon Kleinberg and Eva Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005. 2
- [Lan01] Jean Marc Lanlignel. *Autour de la décomposition en coupes*. PhD thesis, Université Montpellier II, LIRMM, 2001. 26
- [lHhM99] Wen lian Hsu and Tze heng Ma. Fast and simple algorithms for recognizing chordal comparability graphs and interval graphs. *SIAM Journal on Computing*, 28 :1004–1020, 1999. 22
- [Lov72] László Lovász. A characterization of perfect graphs. *Journal of Combinatorial Theory, Series B*, 13(2) :95 – 98, 1972. 22
- [Loy62] R. M. Loynes. The stability of a queue with non-independent inter-arrival and service times. In *Mathematical Proceedings of the Cambridge Philosophical Society*, 58, pages 472–520, 1962. 79
- [LPSR04] Zvi Lotker, Boaz Patt-Shamir, and Adi Rosén. New stability results for adversarial queuing. *SIAM J. Comput.*, 33 :286–303, February 2004. 79
- [Lu91] Shunong Lu. *A distributed concurrency control protocol considering read-only transactions*. PhD thesis, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 1991. UMI Order No. GAX91-36669. 78
- [MD94] S.P. Meyn and D.D. Down. Stability of generalized jackson networks. *Ann. Appl. Prob.*, 4 :124–148, 1994. 78
- [MdSD⁺09] Pascal Mérindol, Virginie Van den Schrieck, Benoit Donnet, Olivier Bonaventure, and Jean-Jacques Pansiot. Quantifying ases multiconnectivity using multicast information. In *Internet Measurement Conference*, pages 370–376, 2009. 38
- [MOVL09] Clémence Magnien, Frédéric Ouédraogo, Guillaume Valadon, and Matthieu Latapy. Fast dynamics in internet topology : preliminary observations and explanations. *CoRR*, abs/0904.2716, 2009. 41

Références bibliographiques

- [MRC05] Stefanie Muff, Francesco Rao, and Amedeo Caglisch. Local modularity measure for network clusterizations. *Phys. Rev. E*, 72 :056107, Nov 2005. 53
- [MRWK03] Zhuoqing Morley Mao, Jennifer Rexford, Jia Wang, and Randy H. Katz. Towards an accurate as-level traceroute tool. pages 365–378, 2003. 40
- [MS94] Ross M. McConnell and Jeremy P. Spinrad. Linear-time modular decomposition and efficient transitive orientation of comparability graphs. In *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, SODA '94, pages 536–545, Philadelphia, PA, USA, 1994. Society for Industrial and Applied Mathematics. 23
- [New06] M E Newman. Modularity and community structure in networks. *PNAS*, 103(23) :8577–8582, 2006. 53, 55
- [NG04] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(066133), 2004. 2, 53, 54
- [Nog09] Damien Noguès. δ -hyperbolicité et graphes. Master's thesis, MPRI, Université Paris 7, 2009. 18, 20, 32
- [NS09] Onuttom Narayan and Iraj Saniee. The Large Scale Curvature of Networks. *ArXiv e-prints*, jul 2009. 33, 35, 48
- [NZ02] T. S. Eugene Ng and Hui Zhang. Predicting internet network distance with coordinates-based approaches. In *IEEE INFOCOM*, 2002. 34
- [OBDJ07] J.M. Olesen, J. Bascompte, Y.L. Dupont, and P. Jordano. The modularity of pollination networks. *Proceedings of the National Academy of Sciences*, 104(50) :19891, 2007. 53
- [OPW⁺10] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang. The (in)completeness of the observed internet AS-level structure. *IEEE/ACM Transac. on Networking*, 18(1) :109–122, 2010. 36
- [PG98] Jean-Jacques Pansiot and Dominique Grad. On routes and multicast trees in the internet. *Computer Communication Review*, 28(1) :41–50, 1998. 34
- [PLE71] Amir Pnueli, Abraham Lempel, and Shimon Even. Transitive orientation of graphs and identification of permutation graphs. *Can. J. Math.*, 23 :160–175, 1971. 22
- [PMDB09] Jean-Jacques Pansiot, Pascal Mérindol, Benoit Donnet, and Olivier Bonaventure. Internet topology discovery through mrimfo probing. Technical Report TR-2009-01, Université catholique de Louvain, East Lansing, Michigan, October 2009. 38

- [PV06] Charis Papadopoulos and Constantinos Voglis. Drawing graphs using modular decomposition. In Patrick Healy and Nikola Nikolov, editors, *Graph Drawing*, volume 3843 of *Lecture Notes in Computer Science*, pages 343–354. Springer Berlin / Heidelberg, 2006. 22
- [Rao08] Michaël Rao. Solving some NP-complete problems using split decomposition. *Discrete Applied Mathematics*, 156(14) :2768 – 2780, 2008. 26
- [RMK⁺09] Venugopalan Ramasubramanian, Dahlia Malkhi, Fabian Kuhn, Mahesh Balakrishnan, Archit Gupta, and Aditya Akella. On the treeness of internet latency and bandwidth. In *Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, SIGMETRICS '09, pages 61–72, New York, NY, USA, 2009. ACM. 2, 33, 34, 48
- [Rou] University of Oregon RouteViews Project,. [HTTP://WWW.ROUTEVIEWS.ORG/](http://www.routeviews.org/). 37, 39
- [RS86] N. Robertson and P. Seymour. Graph minors ii. algorithmic aspects of treewidth , 7 :309–322, 1986. *J. of Algorithms*, 7 :309–322, 1986. 2, 3
- [RS92] A.N. Rybko and A.L. Stolyar. Ergodity of stochastic processes that describe functioning of open queueing networks. *Problems Inform. Transmission*, 28 :3–26, 1992. 78, 97
- [SC83] Valeriu P. Soltan and Victor Chepoi. Conditions for invariance of set diameters under d-convexification in a graph. *Cybernetics and Systems Analysis*, 19 :750–756, 1983. 10.1007/BF01068561. 19
- [Sch07] S. Schaeffer. Graph clustering. *Computer Science Review*, 1(1) :27 – 64, 2007. 53
- [SMWA04] Neil Spring, Ratul Mahajan, David Wetherall, and Thomas Anderson. Measuring isp topologies with rocketfuel. *IEEE/ACM Trans. Netw.*, 12(1) :2–16, 2004. 34, 35, 38
- [SS05] Yuval Shavitt and Eran Shir. Dimes : let the internet measure itself. *SIGCOMM Comput. Commun. Rev.*, 35 :71–74, October 2005. 40
- [ST04] Yuval Shavitt and Tomer Tankel. On the curvature of the internet and its usage for overlay construction and distance estimation. In *INFOCOM*, 2004. 2
- [ST08] Yuval Shavitt and Tomer Tankel. Hyperbolic embedding of internet graph for distance estimation and overlay construction. *IEEE/ACM Trans. Netw.*, 16(1) :25–36, 2008. 2, 33, 34, 48

Références bibliographiques

- [TCHP08] Marc Tedder, Derek Corneil, Michel Habib, and Christophe Paul. Simpler linear-time modular decomposition via recursive factorizing permutations. In Luca Aceto, Ivan Damgård, Leslie Goldberg, Magnús Halldórsson, Anna Ingólfssdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, volume 5125 of *Lecture Notes in Computer Science*, pages 634–645. Springer Berlin / Heidelberg, 2008. 23
- [Tel97] Proskurowski A. Telle, J.A. Algorithms for vertex partitioning problems on partial $\hat{\Gamma}^0$ -trees. *SIAM Journal on Discrete Mathematics*, 10(4) :529–550, 1997. cited By (since 1996) 82. 3
- [TG96] L. Tassiulas and L. Georgiadis. Any work-conserving policy stabilizes the ring with spatial re-use. *IEEE/ACM Trans. Netw.* 4, 2 :205–208, 1996. 79
- [TGJ⁺02] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topology generators : Degree-based vs. structural. In *ACM SIGCOMM*, page 159, 2002. 35, 47, 48
- [VL05] Fabien Viger and Matthieu Latapy. Efficient and simple generation of random simple connected graphs with prescribed degree sequence. In Lusheng Wang, editor, *Computing and Combinatorics*, volume 3595 of *Lecture Notes in Computer Science*, chapter 45, pages 440–449–449. Springer Berlin / Heidelberg, Berlin/Heidelberg, 2005. 47
- [WAD09] Walter Willinger, David Alderson, and John C. Doyle. Mathematics and the internet : A source of enormous confusion and great potential. *Notices of the AMS*, 56(5) :586–599, 2009. 36
- [WZ07] Zhi Wang and Jianzhi Zhang. In search of the biological significance of modular structures in protein networks. *PLoS Comput Biol*, 3(6) :e107, 06 2007. 53
- [WZ09] Yaokun Wu and Chengpeng Zhang. Chordality and hyperbolicity of a graph. *Arxiv preprint arXiv :0910.3544*, (08) :1–44, 2009. 13
- [WZ11] Yaokun Wu and Chengpeng Zhang. Hyperbolicity and chordality of a graph. *Electr. J. Comb.*, 18(43), 2011. 13, 17, 20, 21, 28

Résumé

Ce travail étudie des propriétés topologiques des graphes et leurs applications aux réseaux de communications, notamment aux graphes représentant la structure d'Internet.

Dans un premier temps, on s'intéresse à l'arborescence des graphes par l'étude de deux paramètres : l'hyperbolicité et la largeur arborescente (treewidth). Pour l'hyperbolicité, on analyse sa relation avec d'autres paramètres de graphes et on montre que certaines décompositions de graphes en permettent un calcul efficace.

On calcule ces deux paramètres dans des instantanés d'Internet pour différents niveaux hiérarchiques et différentes périodes de temps. On y apporte des interprétations structurelles et algorithmiques pour les valeurs obtenues.

On aborde ensuite le problème de partitionnement de graphes (clustering) sous l'angle de la modularité, paramètre qui mesure la qualité d'un partitionnement, largement utilisé dans la littérature. On analyse la modularité du point de vue théorique et son comportement asymptotique pour certaines familles de graphes.

Enfin, on s'intéresse à une approche combinatoire de la théorie des files d'attente où les injections de paquets sont effectuées par un adversaire. On propose une généralisation de ce modèle par l'introduction de différentes classes de requêtes.

Mots clés : Mots clés : Largeur arborescente, hyperbolicité, topologie d'Internet, modularité, files d'attente avec adversaire.

Abstract

This thesis focuses on topological properties of graphs and their application on communication networks, specifically on graphs reflecting Internet structure.

We first look how far from a tree a graph may be by the study of two parameters : hyperbolicity and treewidth. For hyperbolicity, we analyse the relation with others graph parameters, we also show that some graph decompositions allow its efficient computation. We compute both parameters on Internet snapshots at different levels of granularity and time periods. We propose some structural and algorithmic consequences of obtained values. Then, we study the graph clustering problem from the perspective of modularity, which measures a clustering quality and is largely studied in the literature. We analyse modularity from a theoretical point of view and [describe] its asymptotic behaviour for some graph families.

Finally, we deal with adversarial queueing theory, a combinatorial framework derived from classic queueing theory where injection process is under the control of an adversary. We propose a new model generalisation by considering request of distinct types.

Keywords : treewidth, hyperbolicity, Internet topology, modularity, adversarial queueing theory.