Algorithmique des graphes pour les réseaux et la biologie structurale computationnelle

Dorian Mazauric

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

Algorithmique des graphes pour les réseaux et la biologie structurale computationnelle

Dorian Mazauric

Inria Sophia Antipolis - Méditerranée
Équipe-projet Algorithmes et Biologie Structurale (ABS)

Soutenue le 05 novembre 2021 devant le jury composé de:

Cláudia Linhares Sales	Professeur Universidad Federal do Ceará, Brésil (Rapporteure)
Yann Ponty	Directeur de Recherche CNRS (Rapporteur)
Laurent Viennot	Directeur de Recherche Inria (Rapporteur)
Frédéric Cazals	Directeur de Recherche Inria
Xavier G. Defago	Professeur Tokyo Institute of Technology, Japon
Nicolas Nisse	Chargé de Recherche Inria
Joseph G. Peters	Professeur Simon Fraser University, Canada
Jean-Claude Bermond	Directeur de Recherche émérite CNRS (Invité)
Johny Bond	Professeur des Universités - Université Côte d'Azur (Invité)
Igor Litovsky	Professeur des Universités - Université Côte d'Azur (Invité)

Table des matières

Chapitre 1

Introduction générale

Contents

L'objectif de cette introduction est de donner une vue d'ensemble de mes activités de recherche. Elles sont de trois types.

1 Conception, analyse et optimisation d'algorithmes. Les outils utilisés relèvent principalement des mathématiques discrètes avec une modélisation par des graphes. J'ai d'abord travaillé sur des problèmes de réseaux de télécommunication puis j'ai souhaité faire évoluer le centre de gravité des applications de mes travaux en m'intéressant aux réseaux en biologie, et particulièrement ceux en lien avec la biologie structurale computationnelle.

2 Expérimentation, implémentation et intégration des algorithmes dans des librairies existantes.

3 Diffusion de la culture scientifique.

Les activités de type 2 et 3 sont très importantes pour moi et me tiennent à coeur mais je ne les détaillerai pas dans le document. J'en donne juste un aperçu dans cette introduction et mentionne des liens pour plus de détails (Section 1.6 et Section 1.7).

Pour les activités de type 1 je présente brièvement mes contributions dans cette introduction (Section 1.1 à Section 1.5) et détaille un certain nombre d'entre elles dans les chapitres 2, 3, 4 et 5.

Dans la Section 1.1, je rappelle les résultats obtenus dans ma thèse (je ne les détaille pas dans le document). Il faut noter que certains travaux démarrés à cette

époque ont parfois été poursuivis et finalisés bien après. Par exemple les travaux sur l'ordonnancement des liens en présence d'interférence ont été commencés durant mon stage de Master en 2008, présentés en 2011 mais finalisés par un envoi à un journal seulement en 2019. Certains problèmes étaient modélisés par des jeux avec des agents dans les graphes. Dans la continuité de mon doctorat, j'ai été amené à modéliser et à analyser d'autres problèmes de réseaux avec des jeux dans les graphes. Dans ce document, je décris mes contributions pour trois de ces problèmes (Section 1.2 et Chapitre 2). Les travaux de ma thèse concernaient des problèmes pour des réseaux de télécommunication avec essentiellement une modélisation par des graphes. Par la suite je me suis intéressé aux réseaux en biologie, et particulièrement ceux en lien avec la biologie structurale computationnelle. Dans ce domaine, les problèmes se modélisent également avec des graphes. Par exemple, le réseau des protéines formant un assemblage macro-moléculaire peut se représenter par un graphe : un sommet par protéine et une arête représente l'existence d'un contact entre les deux protéines associées. J'ai développé des techniques algorithmiques sur les graphes pour différents problèmes qui concourent tous à la réalisation de l'objectif majeur de mon projet de recherche dans ce domaine : le développement de modèles haute résolution de grands assemblages macro-moléculaires (Section 1.3 et Chapitre 3). J'ai également étudié, analysé et optimisé des réseaux de *partage* (Section 1.4 et Chapitre 4). J'ai notamment travaillé sur la dynamique de formation des groupes de partage dans les réseaux sociaux. Il est aussi possible de voir ce processus dynamique comme des changements de conformations d'une protéine (les utilisateurs représentant les atomes). Je mentionne également d'autres contributions (Section 1.5), certaines étant présentées plus en détails dans le document (Chapitre 5).

1.1 Optimisation discrète dans les réseaux de télécommunication (Thèse de doctorat)

Dans mes travaux de thèse au sein des équipes-projets Maestro[1] et Mascotte[2], je me suis intéressé à différents types de réseaux de télécommunication : réseaux optiques, réseaux cœur, réseaux sans fil, réseaux pair-à-pair. Ces différents types de réseaux ont tous leurs spécificités, mais de nombreuses problématiques leurs sont communes. En effet, pour tous ces réseaux, il est important d'assurer la meilleure qualité de service possible, de garantir la stabilité du système et de minimiser les ressources nécessaires et donc le coût de fonctionnement.

J'ai abordé quatre problèmes importants dans ces réseaux de télécommunication : reconfiguration du routage dans les réseaux optiques, économie d'énergie dans les réseaux cœur, ordonnancement des liens dans les réseaux sans-fil et placement de données dans les réseaux pair-à-pair. Pour les résoudre, j'ai utilisé et développé des outils théoriques des mathématiques discrètes (graphes, configurations, optimisation

[1] Neo (anciennement Maestro) est une équipe-projet Inria Sophia Antipolis - Méditerranée

[2] Coati (anciennement Mascotte) est une équipe-projet commune Inria Sophia Antipolis - Méditerranée et Laboratoire I3S (CNRS, Université Nice Sophia)

combinatoire), d'algorithmique (complexité, algorithmique distribuée) et de probabilités. Certains problèmes ont fait l'objet de recherche et de publications après la fin de ma thèse de doctorat.

Pour le problème de reconfiguration du routage dans les réseaux optiques, il s'agissait de minimiser des paramètres liés aux interruptions de requêtes de connexion lors du reroutage. Les différents problèmes ont été modélisés par des calculs de paramètres de graphes (e.g. jeux avec des agents dans les graphes) [CMN16, CHM08b, CHM08a, CHM12, CMN09, BCM+12, CCM+10, CCM+11, CHM+09, CMN14, CMN16].

Une autre partie de mes travaux de thèse concernait l'analyse et l'optimisation du placement de données dans les réseaux pair-à-pair en termes de disponibilité des données. Mes contributions concernent deux problèmes différents avec le développement de techniques de théorie des Design [BJMMY16] et de chaînes de Markov [CGM+10, CGM+13].

J'ai également travaillé à la détermination de routages efficaces en énergie dans les réseaux cœur. Pour le modèle utilisé, il s'agit de trouver un routage des connexions minimisant le nombre d'équipements du réseau. J'ai obtenu des résultats d'inaproximabilité et des bornes théoriques, et j'ai analysé l'impact des solutions efficaces en énergie sur la longueur des routes et sur la tolérance aux pannes [GMM11, GMMO10, GMM12].

Je me suis aussi intéressé à l'optimisation de l'ordonnancement des liens dans les réseaux sans fil. Nous avons proposé le premier algorithme entièrement local vérifiant les propriétés suivantes : il est valable quel que soit le modèle d'interférence binaire utilisé; il a un *surcoût* constant (indépendant de la taille du réseau et des valeurs des files d'attente associées aux liens du réseau); et il ne requiert pas de connaissance particulière de l'état du réseau [BMN09, BMMN10, BMMN08].

Enfin, j'ai également obtenu des résultats pour des problèmes liés à la coloration impropre pondérée [ABG+12, ABG+11] et au calcul de la longueur de chemin moyen dans les graphes petit monde [GMP10, GMB10, GMP10].

1.2 Jeux dans les graphes (Chapitre 2)

Je me suis intéressé à différents problèmes des réseaux qui se prêtaient parfaitement à une modélisation avec des jeux dans les graphes. Dans la plupart de mes travaux, les jeux se jouent à deux joueurs. Un des joueurs représente *notre algorithme* alors que l'autre est un adversaire qui va émuler les situations les plus difficiles (les pires) du système considéré.

Le jeu du Web surfeur permet de comprendre, analyser et améliorer les mécanismes de préchargement afin de garantir la meilleure qualité de service possible aux utilisateurs (e.g. les Web surfeurs) tout en minimisant les ressources utilisées. Dans ce contexte, le graphe représente un réseau (e.g. celui du Web) pour lequel les sommets représentent des contenus (e.g. des pages Web, des vidéos) et les arêtes représentent des liens entre ces contenus (e.g. entre les pages Web). Le premier

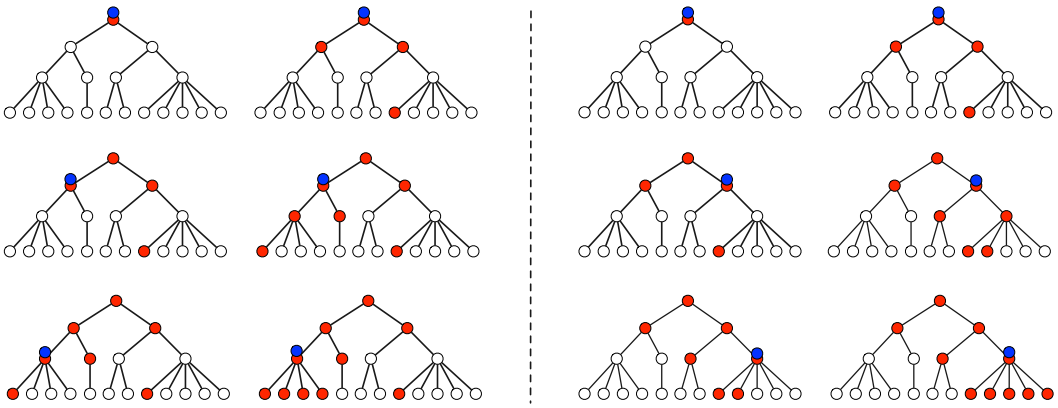

Figure 1.1: Exemple de réseau simple : un arbre composé de 19 sommets. Le rond bleu représente le Web surfeur et les sommets rouges sont les sommets marqués (pages Web préchargées). Au début, le Web surfeur est à la racine. Le premier joueur (navigateur) marque trois sommets rouges. Si le Web surfeur va à gauche (respectivement à droite) dans l'arbre, les figures de gauche (respectivement de droite) décrivent une stratégie qui permet au premier joueur de gagner en marquant 3 sommets à chaque étape. Pour cet exemple, si le premier joueur ne peut marquer que deux pages, alors le Web surfeur gagne en suivant les stratégies indiquées dans les figures.

joueur va émuler un algorithme de préchargement de contenu alors que le deuxième va simuler des comportements d'utilisateurs. Ce jeu se joue à tour de rôle. À chaque étape, le premier joueur marque un certain nombre k de sommets (cela veut dire que les contenus associés de k pages sont préchargés) et le deuxième joueur va déplacer un agent (le Web surfeur) d'un sommet à un autre (qui est un voisin). Le but du premier joueur (navigateur) est que le Web surfeur, qui navigue sur le réseau de page en page, n'arrive jamais sur une page non-préchargée, et donc que le surfeur n'attende jamais. Dans ce cas le premier joueur gagne, sinon le deuxième joueur gagne. Étant donné que le navigateur ne peut pas précharger toutes les pages ou un grand nombre de pages par étape, le problème consiste à trouver le plus petit entier positif tel que le premier joueur peut toujours gagner quelle que soit la stratégie du Web surfeur. Dans ce cas, le Web surfeur n'attend jamais qu'une page se charge et le nombre de pages à précharger à chaque étape (nombre de sommets marqués) est minimisé. Un exemple est décrit dans la Figure 1.2. Mes contributions sont des résultats de complexité et des algorihmes pour calculer les paramètres associés [FGJM+11, FGJM+12b, FGJM+12a, GLM+15] (Section 2.1). Ce travail, réalisé avec Fedor V. Fomin, Frédéric Giroire, Alain Jean-Marie et Nicolas Nisse, est le fruit d'une collaboration entre les équipes-projets Maestro et Mascotte.

Je me suis aussi intéressé au jeu de la localisation d'une cible (invisible et immobile) dans un graphe. Dans ce jeu, introduit par Seager en 2013, une cible est placée secrètement sur un sommet et, à chaque tour, il est possible d'interroger un sommet

et recevoir, comme réponse, la distance exacte entre ce sommet et la cible. L'objectif est de localiser la cible en minimisant le nombre de tours, et ce, quelle que soit sa position. Le premier joueur doit donc déterminer une séquence de questions optimales alors que le deuxième joueur place une cible dans un sommet qui va maximiser le nombre de tours pour le premier joueur. Mes contributions concernent des résultats de complexité et des algorihmes de programmation dynamique pour certaines classes d'instances pour le calcul des paramètres associés, comme la dimension métrique séquentielle [BMMI+18b, BMMI+18a] (Section 2.2). Ce travail, réalisé avec Julien Bensmail, Fionn Mc Inerney, Nicolas Nisse et Stéphane Pérennes, est le fruit d'une collaboration entre les équipes-projets ABS[3] et Coati.

Enfin, j'ai prouvé des résultats de complexité pour un problème de calcul de diamètre dynamique dans les réseaux dynamiques. Ce problème peut être modélisé en termes de jeu dans les graphes. Mes contributions sont des résultats de complexité [GM14] (Section 2.3). Ce travail a été réalisé avec Emmanuel Godard au Laboratoire d'Informatique Fondamentale de Marseille[4].

1.3 Algorithmique pour la biologie structurale (Chapitre 3)

Tous les phénomènes biologiques (cognition, réponse immunitaire, métabolisme...) reposent sur des complexes moléculaires qui interagissent souvent en cascade. Leurs propriétés dépendent de la structure et de la dynamique de leurs sous-unités. Les études expérimentales de ces systèmes font face à des limitations importantes. En effet, obtenir des modèles haute résolution n'est possible que pour des sous-unités (par cristallographie) alors que pour les grands assemblages, seuls des modèles basse résolution sont possibles (par spectrométrie de masse native et/ou par cryo-microscopie électronique). Mes recherches ont pour ambition de développer des méthodes innovantes pour construire des modèles haute résolution d'assemblages macro-moléculaires en combinant des données basse résolution et haute résolution. Naturellement, une sous-unité peut subir des déformations et nous devons donc explorer les conformations (plausibles) des sous-unités (protéines). Mes activités de recherche se décomposent selon les trois problèmes principaux décrits ci-dessous. La Figure 1.2 résume l'articulation de ce programme de recherche.

Premièrement, il s'agit de problèmes d'inférence pour des modèles basse résolution (Section 3.2). Par spectrométrie de masse native, il est possible d'obtenir différents sous-complexes d'un assemblage macro-moléculaire A. Le problème est alors de déterminer un ensemble de contacts plausibles entre les sous-unités de A à partir de l'information obtenue de ces sous-complexes. Le but est de construire des modèles basse résolution de grands assemblages. J'ai contribué à l'analyse et au développement d'algorithmes pour les problèmes d'optimisation combinatoire associés [CHM+18, CHM+17, HMNW20]. Ce travail a été réalisé avec Nathann Cohen, Frédéric Havet,

[3] ABS (Algorithmes et Biologie Structurale) est une équipe-projet Inria Sophia Antipolis - Méditerranée

[4] le laboratoire est devenu Laboratoire d'Informatique et Systèmes après une fusion

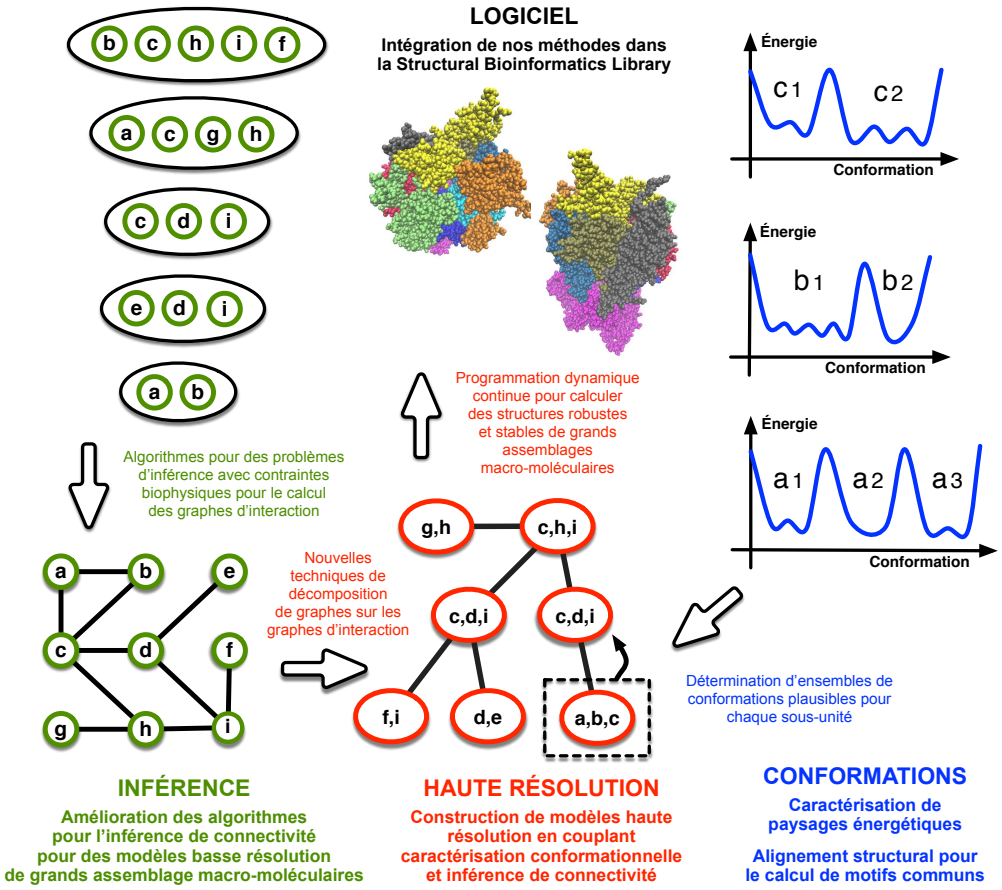

Figure 1.2: Résumé de mes recherches pour l'élaboration de modèles haute résolution de grands assemblages macro-moléculaires.

Viet-Ha Nguyen, Ignasi Sau et Rémi Watrigant.

Deuxièmement, il s'agit du problème de génération de conformations. Une protéine (ou une sous-unité) est composée de n atomes ayant chacun 3 paramètres décrivant sa position. Une position pour les n atomes forme une conformation de la protéine ($3n$ degrés de liberté). Une énergie est associée à chaque conformation et le paysage énergétique d'une protéine est l'ensemble des couples conformation et énergie. Le paysage énergétique encode toutes les propriétés microscopiques et macroscopiques en termes de structure, thermodynamique et dynamique. Étant donné que les minima locaux d'un paysage énergétique correspondent à des conformations de basse énergie, il est nécessaire de générer de large et divers ensembles de minima locaux. Pour cela, il est possible d'utiliser les algorithmes d'exploration de l'état-de-l'art [RDRC16, CMCW16, CM16] ainsi que des méthodes d'analyse de conformations d'ensembles [CDM+15]. Mes recherches visent à analyser et améliorer ces méthodes d'exploration dans le but de générer des ensembles pertinents de conformations. Pour ce faire, j'ai développé deux axes de recherche.

- Tout d'abord, je me suis intéressé à la caractérisation (analyse et comparaison) de paysages énergétiques moléculaires [CDM+15] (Section 3.3). Ces travaux ont été réalisés avec Frédéric Cazals, Tom Dreyfus, Christine Roth (équipe-projet ABS), Charles Robert (IBPC-LBT / CNRS), Joanne M. Carr et David J. Wales (University of Cambridge). Dans ce document, je présente uniquement mes travaux relatifs au problème de comparaison qui permet, entre autres, de comparer différentes méthodes d'échantillonage.

- Ensuite, j'ai développé de nouvelles techniques (e.g. algorithmes de programmation dynamique) pour la recherche de motifs communs de différentes conformations (Section 3.4). Ce travail a été réalisé avec Frédéric Cazals, Maria Guramare (Harvard University) et Romain Tetley. Un des ingrédients majeurs des algorithmes d'exploration de paysages énergétiques est l'ensemble de déplacement (*move set*) qui consiste à proposer une conformation candidate à partir d'une conformation déjà calculée. Ces résultats permettront une amélioration du temps de calcul des méthodes d'exploration de paysages énergétiques, en contraignant encore davantage les déplacements possibles.

Troisièmement, il s'agit de déterminer des modèles haute résolution de grands assemblages macro-moléculaires (Section 3.5). Grâce à l'amélioration des techniques d'exploration de paysages énergétiques (en termes de qualité et en temps de calcul), il sera ainsi possible de générer de larges ensembles de conformations pour chacune des sous-unités (protéines). De plus, à partir de modèles basse résolution obtenus, il sera possible d'obtenir un graphe d'interaction $G = (V, E)$ correspondant à un assemblage macro-moléculaire : V représente l'ensemble des sous-unités et E représente l'ensemble des contacts plausibles entre les sous-unités. Un ensemble de couleurs (conformations) est associé à chaque sommet. Le problème Domino consiste alors à déterminer une conformation pour chaque sous-unité dans le but d'optimiser une fonction de score donnée. Pour ce faire, nous développons des techniques d'optimisation combinatoire. Je co-encadre (avec Frédéric Havet) la thèse de Viet-Ha Nguyen sur ce sujet.

1.4 Réseaux de partage (Chapitre 4)

Je me suis intéressé à différents problèmes des réseaux de partage : analyse de la dynamique du partage d'information dans les réseaux sociaux, comparaison de clusterings et optimisation d'affectactions d'annonces dans un réseau anti-gaspillage.

J'ai étudié la dynamique de formations des groupes dans les réseaux sociaux. Dans le modèle que nous avons considéré, un sous-ensemble d'utilisateurs peuvent changer de groupe si et seulement si leurs utilités respectives augmentent toutes strictement. Un tel sous-ensemble de taille k est appelé une k-déviation. En effet, ces processus sont basés sur l'optimisation de l'utilité individuelle avec la possibilité pour les utilisateurs de former des coalitions de taille maximale fixée. Un problème important est la caractérisation des classes d'instances admettant une partition k-stable (une partition pour laquelle il n'existe pas de k-déviation). Un exemple simple

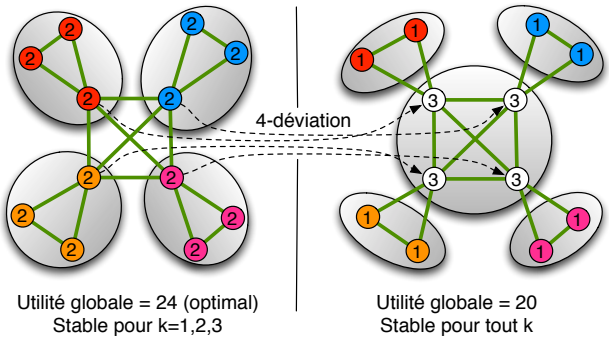

Figure 1.3: Exemple de réseau (social) composé de 12 sommets (utilisateurs). Une arête représente un lien d'amitié alors qu'une non-arête représente un lien d'inimitié. Deux *ennemis* ne peuvent pas être dans un même groupe. Les utilités (nombre d'amis dans le groupe) sont indiquées par les entiers dans les sommets. La partition de gauche est 3-stable mais pas 4-stable. En effet, les quatre sommets au centre peuvent former un nouveau groupe en augmentant strictement leurs utilités. La partition de droite est 4-stable.

est décrit dans la Figure 1.3. Pour ces instances, il est alors intéressant de déterminer le temps de convergence du processus dynamique (nombre de k-déviations avant d'obtenir la k-stabilité) dans le pire des cas notamment. J'ai notamment résolu une conjecture de Jon M. Kleinberg et Katrina Ligett [CDM13, BCDM18] (Section 4.1). Ce travail a été réalisé avec Augustin Chaintreau et Guillaume Ducoffe à Columbia University (département Computer Science).

Je me suis également intéressé à un problème de comparaison de clusterings. Romain Tetley (équique-projet ABS) a étudié le problème de calculer des motifs structurellement conservés (e.g. les motifs commun de deux conformations d'une même protéine). Pour cela, il a développé une méthode complexe mélangeant, entre autres, calcul de distances et de rang d'acides aminés, calcul de la dynamique des composantes connexes formés de ces acides aminés et calcul de persistence. Un problème de comparaison de composantes connexes des deux conformations en est ressorti et a été modélisé en termes de comparaison de partition, avec une application plus générale qui est la comparaison de clusterings. J'ai contribué à l'analyse du problème et au développement d'algorithmes efficaces pour le résoudre [CMTW17, CMTW18] (Section 4.2). Ce travail a été réalisé avec Frédéric Cazals, Romain Tetley et Rémi Watrigant au sein de l'équipe-projet ABS.

Enfin, j'ai développé des algorithmes pour un problème d'affectations d'annonces dans un réseau anti gaspillage (Section 1.6 et Section 4.3). Ce travail, réalisé avec Jean-Baptiste Caillau, Enzo Giusti, Joanna Moulierac et Xuchun Zhang, est le fruit d'une collaboration entre les équipes-projets ABS et Coati.

1.5 Autres contributions (Chapitre 5)

Dans cette section, j'indique brièvement d'autres contributions et en détaillerai deux dans le Chapitre 5.

Je me suis intéressé à la caractérisation de la structure commune de réseaux cérébraux. Voir [WMGDD16, LGD+17] pour tous les détails. Ces travaux ont été réalisés avec Rachid Deriche, Guillermo Gallardo-Diez, Nahuel Lascano et Demian Wassermann en collaboration entre les équipes-projets ABS et Athena[5].

J'ai analysé le phénomène de pannes en cascade dans les réseaux électriques. Un des objectifs était de déterminer les liens sensibles qui généreraient ce type de pannes Voir [SMZ17, SMZ14, MSZ13] pour tous les détails de mes contributions pour ce sujet. Ces travaux ont été réalisés avec Saleh Soltan et Gil Zussman à Columbia University (département Electrical Engineering),

Je me suis aussi intéressé à la représentation des complexes simpliciaux par des arbres. Mes contributions concernent principalement l'analyse de la complexité des problèmes ainsi que des algorithmes pour certaines classes d'instances [BM16] (Section 5.1). Ces travaux ont été réalisés avec Jean-Daniel Boissonnat au sein de l'équipe-projet Geometrica[6].

Enfin, je me suis intéressé à un problème de flot avec des contraintes de délai de type *on/off*. La difficulté réside dans le choix du sous-ensemble de connexions qui supportent un flot non nul et qui doivent respecter des contraintes de délai (pour les autres connexions, ces contraintes peuvent être violées). Mes contributions concernent l'analyse de la complexité et le développement d'algorithmes efficaces pour certaines classes d'instances [BMV14, BMV17] (Section 5.2). Ces travaux ont été réalisés avec Pierre Bonami et Yann Vaxès au Laboratoire d'Informatique Fondamentale de Marseille.

1.6 Implémentation et logiciel

Il est important pour moi d'implémenter les algorithmes que j'ai conçus et de les intégrer dans des librairies existantes.

J'ai choisi de ne pas détailler les résultats d'expérimentation de mes différents travaux. Il est possible de trouver ces parties dans les publications associées. Dans le document global, je ne détaillerai pas les implémentations.

Par exemple, pour le problème d'affectations d'annonces dans un réseau anti gaspillage (Section 4.3), les algorithmes développés et implémentés sont en cours d'intégration (avec InriaTech) dans l'application mobile Pepino de la startup Oui!Greens (`www.ouigreens.com`). Ce travail a été réalisé avec Jean-Baptiste Caillau, Enzo Giusti, Joanna Moulierac et Xuchun Zhang.

J'ai également intégré certaines de mes contributions dans la *Structural Bioinformatics Library* (SBL, `http://sbl.inria.fr`), développée au sein de l'équipe-projet

[5] Athena est une équipe-projet Inria Sophia Antipolis - Méditerranée

[6] DataShape (anciennement Geometrica) est une équipe-projet Inria Sophia Antipolis - Méditerranée

Figure 1.4: Illustration de mes interventions de médiation.

ABS. Cette librairie `C++/Python`, initiée par Frédéric Cazals et Tom Dreyfus, fournit des outils combinatoires, géométriques et topologiques pour résoudre des problèmes en biologie structurale. J'ai contribué aux quatre applications suivantes : *Comparing two clusterings using matchings between clusters of clusters*, *Conformational ensembles comparison*, *Energy landscapes comparison* et *Optimal transportation for graphs*. Ce travail a été effectué avec Frédéric Cazals, Tom Dreyfus, Romain Tetley et Christine Roth.

1.7 Diffusion de la culture scientifique

La diffusion de la culture scientifique est, selon moi, une activité très importante du métier chercheur même si je ne la détaille pas dans le document global.

Je coordonne le projet **Terra Numerica, vers une Cité du Numérique** (`http://terra-numerica.org`) qui a pour but d'accroître le capital de compétences numériques (dans son acceptation la plus large) de tous les citoyens (dont les scolaires), à travers une audience des plus vastes et diversifiées. Les objectifs sont les suivants :

- répondre aux besoins urgents de compréhension et d'appropriation des sciences du numérique par la société,
- sensibiliser et responsabiliser les citoyens aux forts enjeux sociétaux qui en découlent,
- développer la démarche scientifique, susciter les vocations (chez les filles et les garçons), réconcilier les élèves en difficulté,
- associer le plus grand nombre aux évolutions liées aux sciences du numérique.

Terra Numerica a l'ambition de créer un dispositif original, attractif et unique de diffusion, de partage, de rencontres, de convivialité entre les acteurs du numérique : chercheurs, enseignants-chercheurs, enseignants, associatifs, industriels, étudiants, élèves, grand public et citoyens. Il comprendrait un lieu central, la Cité du Numérique (de type Palais de la Découverte des Sciences du numérique) et différents Espaces Partenaires à travers tout le sud-est (mais pas que). L'idée est d'allier deux points forts du territoire : les Sciences & Technologies et le Tourisme. Ressources, activités,

ateliers, supports, objets scientifiques historiques, etc. sont déployés de manière cohérente dans les différents Espaces Partenaires Terra Numerica (établissements scolaires, tiers lieux associatifs, etc.) afin d'aller le plus possible à la rencontre des citoyens. Les ressources et activités Terra Numerica sont décrites sur le site Web. Terra Numerica s'inscrit dans la continuité des projets des acteurs de la diffusion de la culture des sciences du numérique dans les départements des Alpes-Maritimes et du Var. C'est un projet commun entre le CNRS, Inria, Université Côte d'Azur, l'Education Nationale (Académie de Nice) et d'autres partenaires (https://terra-numerica.org/partenaires).

Je suis membre du groupe Médiation et Animation des MAthématiques, des Sciences et Techniques Informatiques et des Communications (MASTIC), Inria Sophia Antipolis - Méditerranée et du projet **GALEJADE** (*Graphes et ALgorithmes : Ensemble de Jeux À Destination des Écoliers, mais pas que*), financé par la Fondation Blaise Pascal, Université Côte d'Azur et Inria. Les partenaires scientifiques et pédagogiques de Galejade sont Inria Sophia Antipolis - Méditerranée et MASTIC, la Direction des Services Départementaux de l'Éducation Nationale (DSDEN) des Alpes-Maritimes et Institut ESOPE 21. Les membres de Galejade sont Pierre Alliez, Sabrina Ballauri, Florence Barbara, Laurent Giauffret, Frédéric Havet, Magali Martin-Mazauric, Nicolas Nisse, Martine Olivi. Tous les détails sont disponibles sur https://galejade.inria.fr.

J'effectue des ateliers et conférences dans des lycées de la région Provence-Alpes-Côte d'Azur (notamment dans le cadre du dispositif régional Culture Science), dans des collèges, dans des écoles primaires des Alpes-Maritimes (notamment dans le cadre du dispositif d'Accompagnement en Sciences et Technologies à l'École Primaire). J'interviens également à l'École Supérieure du Professorat et de l'Éducation (ÉSPÉ) de l'Académie de Nice et j'effectue des formations pour les enseignants dans des écoles des Alpes-Maritimes en collaboration avec la DSDEN des Alpes-Maritimes. De plus, j'effectue des interventions lors de la Fête de la Science et du stage MathC2+ à Inria Sophia Antipolis - Méditerranée. La Figure 1.4 illustre quelques unes de mes activités de diffusion de la culture scientifique en milieu scolaire, pour la fête de la Science...

En collaboration avec Laurent Giauffret de la DSDEN des Alpes-Maritimes, j'ai écrit un livre intitulé *Graphes et Algorithmes - Jeux grandeur nature* [Maz16]. Ce document a pour vocation de présenter, d'expliquer et de jouer avec les graphes et les algorithmes. L'objectif est de présenter des problèmes de graphes et des algorithmes sous la forme la plus simple et la plus ludique possible. Toutes les activités expliquées peuvent se décliner dans trois espaces : l'espace d'une feuille de papier, l'espace d'un plateau de jeu et l'espace grandeur nature (par exemple avec des cerceaux et des lattes en plastique). Tous mes supports de médiation se trouvent dans le document *Graphes et Algorithmes - Diffusion de l'information scientifique* [Maz16]. De plus, j'ai co-écrit un article pour la Fondation la main à la pâte (avec Jean-Claude Bermond). J'ai contribué à l'élaboration d'autres documents (e.g. des posters) avec Frédéric Havet et Nicolas Nisse.

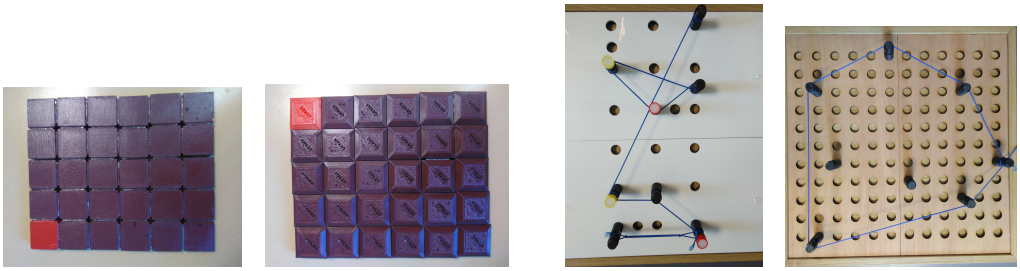

Figure 1.5: (Gauche) Nos deux prototypes de tablette de chocolat. (Droite) Plateau pour le jeu de coloration d'un graphe et le calcul d'enveloppe convexe.

Galejade a également développé des jeux et objets de médiation originaux. Nous avons développé deux prototypes pour le jeu du carreau de chocolat empoisonné (voir les deux photographies de gauche de la Figure 1.5). Un troisième prototype est en cours de réalisation et sera disponible début 2019. Nous réaliserons ensuite une dizaine de tablettes. Nous avons également conçu deux prototypes de plateau en bois pour jouer avec les graphes et les algorithmes : il est possible de jouer avec des graphes prédéfinis ou jouer avec des graphes imaginés par les utilisateurs (élèves). Les deux photographies de droite de la Figure 1.5 illustrent deux jeux possibles. Tous les détails sont disponibles sur `https://galejade.inria.fr`.

Chapitre 2

Jeux dans les graphes

Contents

Dans ce chapitre, j'explique les différents problèmes pour lesquels j'ai été amené à déterminer des modèles sous forme de jeux pour (tenter de) les résoudre.

2.1 Jeu du Web surfeur

Le jeu du Web surfeur est un modèle très original pour traduire certaines questions qui se posent pour les politiques de préchargement (e.g. de pages Web, de vidéos). J'ai choisi de présenter une partie de nos contributions, à savoir un travail réalisé avec Fedor V. Fomin, Frédéric Giroire, Alain Jean-Marie, Nicolas Nisse et Stéphane Pérennes [FGJM+11, FGJM+12b, FGJM+12a]. Une deuxième partie concerne une variante online de ce problème [GLM+15] que je ne détaille pas dans ce document.

2.1.1 Naviguer vite sur le Web sans gaspiller de ressources

En informatique, le *pré-téléchargement* est une technique classique qui exploite le parallélisme entre l'exécution d'une tâche et le transfert des informations nécessaires aux tâches suivantes pour réduire les temps d'attente. Par exemple, dans un processeur, les instructions et les données sont chargées dans la mémoire simultanément

à l'exécution des instructions précédentes. Actuellement, cette technique peut être utilisée dans le contexte du Web où les navigateurs peuvent télécharger les documents accessibles depuis le document en train d'être visioné (page Web, video, etc.). L'accès au document suivant paraît instantané à l'utilisateur et donne l'impression d'une navigation rapide [phd11]. C'est pourquoi le pré-téléchargement a été proposé par Mozilla comme un draft de standard Internet [Inc99]. Cependant, pré-télécharger tous les documents accessibles pourrait résulter en un gaspillage des ressources du réseau (e.g., bande passante, mémoire) puisque tous les documents téléchargés ne seront pas forcément visionnés ou utilisés. Il est donc nécessaire d'établir un compromis entre le gain de temps et la perte des ressources du réseau qui en résulte.

Les modèles développés jusqu'à présent pour l'étude du pré-téléchargement sont essentiellement basés sur la notion de *graphe d'exécution* où les sommets représentent les tâches et les arcs représentent les ordonnancements possibles entre tâches. Par exemple, les sommets représentent les pages Web et les arcs modélisent les liens qu'il faut suivre pour accéder d'une page Web à une autre. L'exécution du programme ou la navigation d'un internaute sur le Web correspondent alors à une marche dans le graphe d'exécution. Il s'agit alors d'optimiser une certaine fonction de coût correspondant à la quantité de ressources consommées par le pré-téléchargement. Cette fonction dépend de la marche suivie dans le graphe d'exécution et de la gêne occasionnée par l'attente d'informations au cours de l'exécution de la tâche ou de la navigation sur le Web. En d'autres termes, il s'agit d'optimiser la bande passante nécessaire au pré-téléchargement, tout en satisfaisant la qualité de service requise. Généralement, il est difficile de résoudre de tels problèmes. Par exemple, dans les modèles Markoviens [JG97] où les arcs du graphe d'exécution sont associés à des transitions de probabilités (modélisant un internaute qui navigue plus ou moins aléatoirement sur le Web), le problème de pré-téléchargement s'inscrit dans le contexte de la programmation dynamique stochastique [GCD02, MJM10]. Sa résolution exacte requiert un effort de calcul exponentiel en le nombre de nœuds du graphe d'exécution (taille de l'espace d'états de ces modèles de Markov).

Dans ce travail, nous considérons le problème du pré-téléchargement *parfait* où l'internaute "surfant" sur le Web ne doit jamais pouvoir accéder à une page Web qui n'a pas été pré-téléchargée auparavant. En d'autres termes, l'internaute est impatient et ne tolère aucune attente. La limite de bande passante se traduit par le fait qu'un nombre limité de pages Web peuvent être pré-téléchargées à chaque étape. Notre problème consiste à déterminer le nombre constant minimum de pages Web qui doivent être pré-téléchargées à chaque étape. Un avantage de notre approche est qu'elle ne nécessite aucune supposition quant au comportement de l'internaute, contrairement à [GCD02, MJM10]. Pour étudier ce problème nous en proposons une modélisation sous forme de jeu de type *Cops and Robber* dans les graphes [FT08, BN11].

Nous formalisons le *jeu de surveillance* d'un graphe dans la Section 2.1.2 et décrivons sa relation avec le problème de pré-téléchargement. En particulier, nous discutons des hypothèses utilisées. Nos résultats sont présentés dans la Section 2.1.3. Nous concluons par quelques perspectives et de nombreuses questions non résolues.

2.1.2 Modélisation du problème par un jeu dans les graphes

Au cours du *jeu de surveillance*, deux joueurs, le *fugitif* et le *surveillant*, s'affrontent sur un graphe (orienté ou non) connexe $G = (V, E)$ à partir d'un sommet $v_0 \in V$ qui est le seul sommet initialement *marqué* dans G et la position initiale du fugitif. Soit $k \in \mathbb{N}$ un entier fixé. Le jeu se déroule tour-à-tour, en commençant par le surveillant. À son tour, le surveillant marque au plus k sommets de G. Un sommet marqué le reste jusqu'à la fin du jeu. Puis, le fugitif peut se déplacer le long d'une arête (ou d'un arc) de G. Le fugitif gagne si, à une étape du jeu, il atteint un sommet non-marqué. Le surveillant gagne dans le cas contraire. Le surveillant a, bien sûr, intérêt à marquer le plus de sommets possible à chaque étape. Notons que le jeu termine au bout d'au plus $\lceil |V|/k \rceil$ étapes lorsque tous les sommets sont marqués.

L'*indice de contrôle*, $ic(G, v_0)$, de $G = (V, E)$ est le plus petit $k \geq 1$ qui permet au surveillant de gagner quels que soient les déplacements du fugitif à partir de $v_0 \in V$. Nous définissons le PROBLÈME DE SURVEILLANCE comme suit.

Nom : PROBLÈME DE SURVEILLANCE
Instance : un graphe connexe $G = (V, E)$, un sommet $v_0 \in V$, un entier $k \geq 1$
Question : $ic(G, v_0) \leq k$?

Dans la suite, nous ferons également référence à la version minimisation du problème sans le mentionner explicitement.

Il est clair que le graphe G joue ici le rôle du graphe d'exécution où les nœuds sont les pages Web et les arcs sont les liens entre pages Web. Le fugitif est l'internaute surfant sur le Web et sa position courante dans G correspond à la page Web en train d'être visualisée. Enfin, marquer un sommet correspond à pré-télécharger la page Web correspondante. Ainsi, l'indice de contrôle représente la vitesse minimum qui permet un pré-téléchargement parfait, c'est-à-dire qui permet de satisfaire l'internaute.

Considérons le graphe de la Figure 2.1 (a) et $k = 2$. Dans la suite, nous décrivons une stratégie gagnante pour le surveillant. Les Figures 2.1 (b), (d), (f), (h), (j) et (l) décrivent les actions du surveillant (c'est-à-dire les deux sommets qu'il marque à chaque étape). Les Figures 2.1 (c), (e), (g), (i) et (k) décrivent les actions du fugitif (c'est-à-dire le déplacement à chaque étape). Il est possible d'observer sur cet exemple que le surveillant gagne quelle que soit la stratégie du fugitif. En effet, le fugitif n'a pas d'intérêt de rester sur place ou de revenir vers son point départ. De plus, par symétrie, le surveillant gagne si le fugitif choisit le chemin du haut. De plus, si $k = 1$, alors il est facile de prouver que le fugitif peut toujours gagner. Donc $ic(G, v_0) = 2$. La Figure 2.2 montrer une stratégie perdante (et donc non optimale) pour le surveillant. En effet, le fugitif arrive à atteindre un sommet non marqué (Figure 2.2 (i)). Une des conclusions de ces deux stratégies est qu'il est parfois nécessaire de marquer des sommets de manière non connexe (c'est-à-dire l'ensemble des sommets bleus n'induisent pas une composante connexe) pour obtenir une stratégie optimale pour le surveillant. La deuxième stratégie a cette propriété de connexité mais n'est pas optimale car le surfeur peut atteindre un sommet non marqué.

Soit Δ le degré maximum de G et $deg(v_0)$ le degré de v_0 : $deg(v_0) \leq ic(G, v_0) \leq \Delta$.

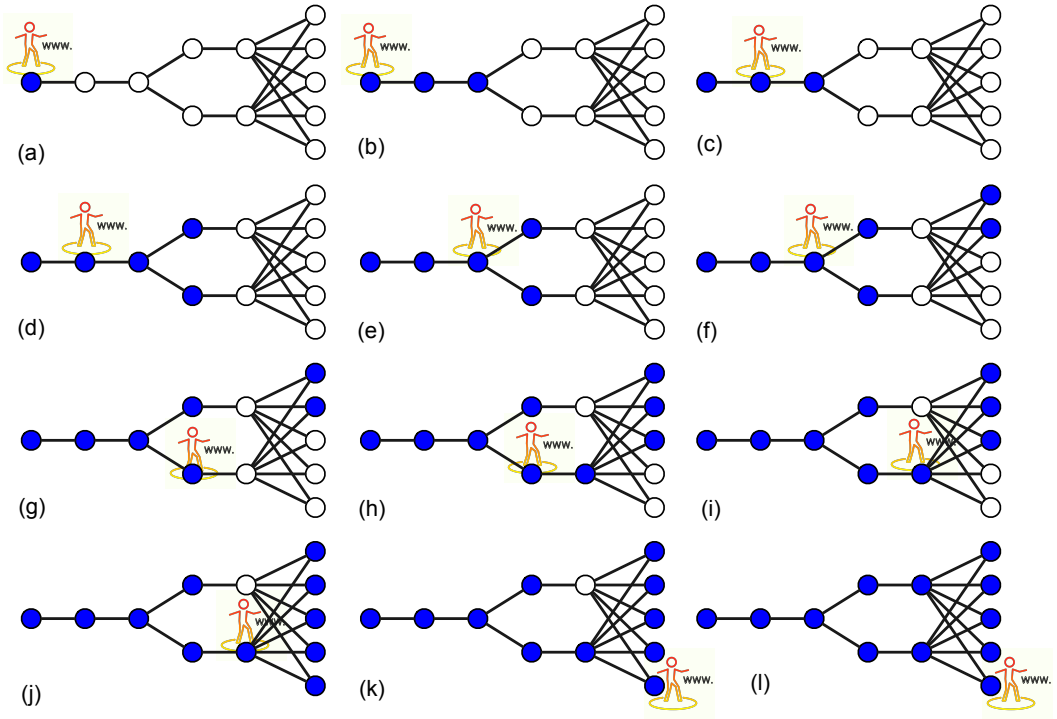

Figure 2.1: Stratégie gagnante pour le surveillant avec $k = 2$.

En effet, si $k < deg(v_0)$, le surveillant ne peut pas marquer tous les voisins de v_0 à la première étape et le fugitif peut atteindre un sommet non-marqué dès son premier déplacement. Au contraire, si $k \geq \Delta$, le surveillant est certain de gagner en marquant tous les voisins (non déjà marqués) de la position courante du fugitif à chaque étape. Pour G non orienté, nous prouvons : $ic(G, v_0) = \Delta$ si et seulement si $deg(v_0) = \Delta$.

Avant de présenter nos résultats (Section 2.1.3), nous discutons des hypothèses de ce modèle. Tout d'abord, notre modèle laisse supposer que toutes les pages Web peuvent être téléchargées à la même vitesse. En fait, notre modèle convient aussi lorsque chaque page Web v à une certaine taille w_v (dont dépend le temps de téléchargement). En effet, dans ce cas, il suffit de considérer le graphe G^p obtenu en remplaçant chaque sommet v de G par une clique K_v de taille w_v et chaque arête $\{u, v\}$ par un biparti complet entre K_v et K_u. Le problème de pré-téléchargement dans G est alors équivalent au problème de surveillance dans G^p.

Une autre hypothèse est la discrétisation du temps et le fait que l'internaute passe un temps identique sur chaque page. Comme nous considérons un pire cas, la durée d'une étape de notre modèle correspond simplement au temps minimum qu'un internaute peut passer sur une page Web. Le cas où le temps minimum que l'internaute passe sur une page Web diffère d'une page Web à une autre est laissé pour de futurs travaux.

L'hypothèse (probablement) la plus forte que nous faisons et celle d'une mémoire

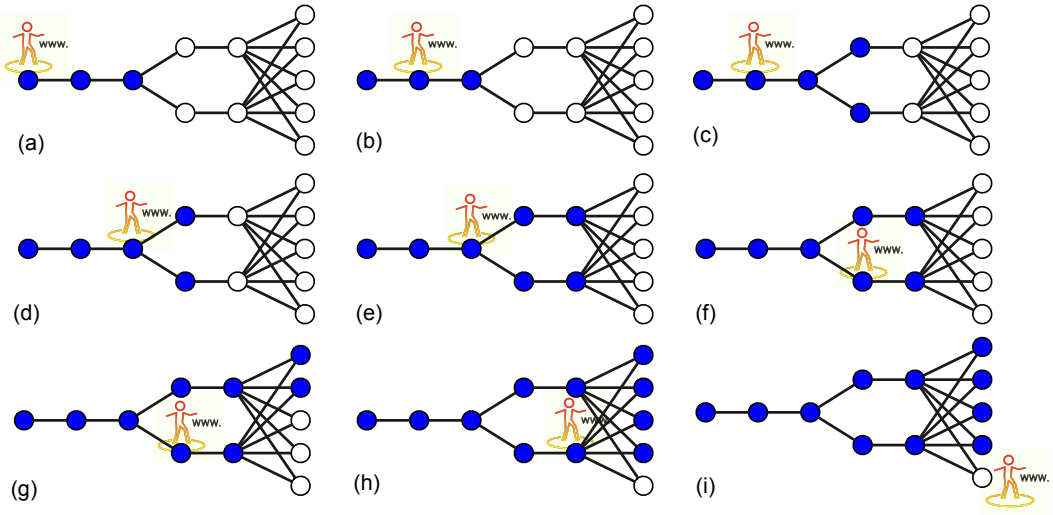

Figure 2.2: Stratégie perdante pour le surveillant avec $k = 2$.

infinie : une page téléchargée (un sommet marqué) le reste jusqu'à la fin. Il s'agit d'un premier modèle où, même avec cette hypothèse simplificatrice, le problème est très dur à résoudre. Comme travaux futurs, la gestion du cache doit être considérée : deux modèles plus réalistes peuvent être étudiés : soit un sommet marqué devient non-marqué après un certain nombre (fixé) d'étapes, soit le surveillant peut "démarquer" des sommets, en respectant que le nombre total de sommets marqués ne dépasse jamais une certaine limite fixée.

2.1.3 Contributions

Dans cette section, nous nous intéressons à la complexité algorithmique du calcul de l'indice de contrôle d'un graphe. Nous prouvons que le problème est difficile même restreint à certaines classes de graphes "simples" comme les graphes cordaux (généralisation des arbres) et les graphes orientés acycliques (DAG). Tout n'est pourtant pas perdu puisque nous proposons des algorithmes polynomiaux pour résoudre le problème dans les arbres et les graphes d'intervalle (généralisation des chemins).

La plupart des preuves des résultats que nous avons obtenus tirent parti du fait que l'on peut se restreindre au cas où le fugitif ne peut suivre que des chemins induits de G. Plus précisément, soit $mic(G, v_0)$ le plus petit $k \geq 1$ qui permet au surveillant de gagner dans le graphe G lorsque le fugitif se déplace à chaque étape et ne peut suivre que des chemins induits débutant en $v_0 \in V(G)$. Évidemment, $mic(G, v_0) \leq ic(G, v_0)$.

Théorème 1 *Pour tout graphe G et $v_0 \in V(G)$, $mic(G, v_0) = ic(G, v_0)$.*

Un *graphe cordal* est un graphe dont tout cycle de longueur au moins 4 contient au moins une corde. Un graphe est dit *séparant* si son ensemble de sommets peut

être partitionné en une clique et un stable. Un *graphe d'intervalle* est le graphe d'intersection d'intervalles de $\mathbb{R}$. Notons que les graphes d'intervalle et les graphes séparants sont deux sous-classes des graphes cordaux.

Théorème 2 *Étant donné un graphe connexe $G = (V, E)$ et un sommet $v_0 \in V$, le* PROBLÈME DE SURVEILLANCE *est :*

1 NP*-difficile dans la classe des graphes cordaux pour $k = 2$ paramètre fixé;*

2 PSPACE*-complet dans la classe des DAGs pour $k = 4$ paramètre fixé ;*

3 NP*-difficile dans la classe des graphes séparants si k fait partie de l'entrée. De plus, dans les graphes séparant, le jeu dure au plus 2 étapes.*

Les assertions 1 et 3 découlent d'une réduction du problème 3-*transverse minimum*[1]. Le fait que le jeu dure au plus 2 étapes dans les graphes séparant vient de ce qu'un plus long chemin induit dans ces graphes a longueur au plus 2. L'assertion 2 découle d'une réduction de 3-QSAT[2].

D'après le théorème précédent, le PROBLÈME DE SURVEILLANCE de graphe n'est pas soluble à paramètre fixé (FPT). De plus, ce problème reste difficile même si le jeu est limité à un nombre fixé de tours.

Théorème 3 *Le* PROBLÈME DE SURVEILLANCE *peut être résolu en temps :*

- *linéaire dans la classe des graphes de degré au plus 3 ;*
- $O(n \log n)$ *dans la classe des arbres de n sommets ;*
- $O(n\Delta^3)$ *dans la classe des graphes d'intervalle de n sommets et de degré maximum Δ ;*
- $O^*(2^n)$ *dans la classe des graphes de n sommets.*

Nous donnons dans la suite les idées de preuve. La première assertion est presque triviale puisque $ic(G, v_0) = 1$ si et seulement si G est un chemin avec v_0 comme extrémité et que $ic(G, v_0) = \Delta$ si et seulement si v_0 est de degré Δ. Deuxièmement, dans le cas d'un arbre T enraciné en $v_0 \in V(T)$, le problème peut être résolu par programmation dynamique. Soit $f_k : V(T) \to \mathbb{N}$ la fonction qui associe 0 aux feuilles et $\max\{0, d - k + \sum_{w \in C} f_k(w)\}$ à tout sommet avec C l'ensemble de ses d fils. Intuitivement, $f_k(v)$ est le nombre minimum de sommets de T_v (différents de v) qui doivent être marqués avant le début du jeu pour gagner avec k dans le sous-arbre T_v enraciné en v, en commençant de v déjà marqué. Nous prouvons que $ic(T, v_0) \leq k$ si et seulement si $f_k(v_0) = 0$. Ensutie, dans les cas des graphes d'intervalle, nous

[1] 3-transverse minimum : étant donnés un ensemble E et $\mathcal{T}$ un ensemble de triplets de E, trouver $H \subseteq E$ de cardinalité minimum tel que $H \cap S \neq \emptyset$ pour tout $S \in \mathcal{T}$.

[2] 3-QSAT : étant donnée une formule F avec $2n$ variables bouléennes $(x_i, y_i)_{i \leq n}$, existe-t-il une assignation de ces variables telle que $\forall x_1, \exists y_1, \cdots, \forall x_n, \exists y_n F$ soit satisfaite ?

montrons que, pour tout $k \leq \Delta$, il existe un ensemble de Δ^2 chemins induits tels que si k suffit à les surveiller, alors $ic(G, v_0) \leq k$. Tester un chemin demandant un temps linéaire, le résultat en découle. Enfin, nous prouvons que décider si $ic(G, v_0) \leq k$ revient à trouver, par programmation dynamique, un chemin dans le graphe dont les sommets sont les configurations (ensemble des sommets marqués et la position du fugitif à une étape) et il y a un arc entre deux configurations si on peut passer de l'une à l'autre en un tour du jeu avec k. Les algorithmes polynomiaux ci-dessus calculent également des stratégies de surveillance optimales.

Théorème 4 *Pour tout graphe G et $v_0 \in V(G)$, $ic(G, v_0) \geq \max_S \lceil \frac{|N[S]|-1}{|S|} \rceil$ avec $v_0 \in S \subseteq V$, S induit un sous-graphe connexe et $N[S]$ le voisinage fermé de S. Il y a de plus égalité dans le cas des arbres.*

Une égalité pour tout graphe dans le théorème précédent impliquerait que le problème de surveillance est co-NP. Sous réserve que P $\neq$ NP, cela contredirait le fait que le problème est PSPACE-complet. Cependant nous n'avons trouvé aucun exemple de graphe G et $v_0 \in V(G)$ tels que $ic(G, v_0) > \max_S \lceil \frac{|N[S]|-1}{|S|} \rceil$.

2.1.4 Perspectives

Plusieurs questions intéressantes restent à résoudre. Le PROBLÈME DE SURVEILLANCE peut-il être résolu en temps polynomial dans les graphes de degré borné ? de largeur arborescente (treewidth) bornée ? Existe-t-il un algorithme pour résoudre ce problème en temps $O^*(c^n)$, $c < 2$, dans les graphes de n sommets ?

Dans le contexte du pré-téléchargement, une variante naturelle du PROBLÈME DE SURVEILLANCE de graphe est celle dans laquelle l'ensemble des sommets marqués est contraint à être connexe. En effet, cela revient à ne pouvoir télécharger que des pages Web liées à des pages Web déjà pré-téléchargées. Nous notons $cic(G, v_0)$ le plus petit $k \geq 1$ qui permet au surveillant de gagner dans le graphe G en satisfaisant la contrainte de connexité. Évidemment, $cic(G, v_0) \geq ic(G, v_0)$. Les résultats énoncés précédemment restent valides pour la variante connexe du problème de pré-téléchargement. En effet, les graphes considérés dans les réductions satisfont l'égalité des paramètres *cic* et *ic*. De plus, pour tout arbre, graphe d'intervalle, ou graphe de degré au plus 3, l'égalité est satisfaite. En ce qui concerne l'algorithme exponentiel exact, il suffit de ne considérer que les configurations connexes (l'ensemble des sommets marqués induit un sous-graphe connexe).

Lemme 1 *Pour tout $k \geq 2$, il y a un graphe G et $v_0 \in V(G)$ tels que $k = ic(G, v_0) < cic(G, v_0) = k + 1$.*

Cette construction généralise le graphe décrit dans les Figures 2.1 et 2.2. Existe-t-il un graphe G et $v_0 \in V(G)$ tels que $ic(G, v_0) + 1 < cic(G, v_0)$? Existe-t-il f, une constante ou une fonction de n tel que $cic(G, v_0) \leq f \cdot ic(G, v_0)$ pour tout graphe G de n sommets et $v_0 \in V$?

Le Web n'étant pas connu *a priori*, nous avons également analysé une variante online de ce problème [GLM+15] que je ne détaille pas dans ce document.

2.2 Dimension métrique séquentielle

Ce travail a été réalisé avec Julien Bensmail, Fionn Mc Inerney, Stéphane Pérennes et Nicolas Nisse [BMMI+18b, BMMI+18a].

2.2.1 Contexte, motivations et problèmes

L'essor récent des téléphones portables dans nos vies a favorisé l'émergence de plusieurs problématiques sociétales, comme celle de la localisation. Supposons que l'on veuille localiser un émetteur au moyen de détecteurs. Si ces derniers déterminent parfaitement la distance (euclidienne) d'où un signal est émis, il suffit de trois détecteurs pour localiser exactement la source (par triangulation). Cependant, des facteurs peuvent rendre la localisation beaucoup plus compliquée : par exemple lorsque la métrique est celle d'un graphe (*i.e.*, les distances sont les longueurs de plus courts chemins dans un graphe) et/ou si le signal de l'émetteur est altéré par des obstacles, ce qui empêche les détecteurs de déterminer la distance exacte à l'émetteur. Nous considérons ce type de problème en étudiant le jeu suivant.

Imaginons un randonneur grièvement blessé et immobile, perdu et isolé dans un environnement modélisé par un graphe dont chaque sommet contient un détecteur. Pour le retrouver, des secouristes peuvent interroger à chaque tour du jeu un nombre limité de détecteurs. Dans le cas d'une communication parfaite, un détecteur retourne sa distance (dans le graphe) au randonneur. Sinon, nous ne pouvons que comparer la puissance des signaux reçus par les détecteurs interrogés (et en déduire lequel est le plus près du randonneur, lequel est le deuxième plus près, etc.). Le temps presse, et, sous toutes ces conditions, le problème est de retrouver le randonneur le plus vite possible (*i.e.*, en minimisant le nombre de tours).

Plus précisément, soient $G = (V, E)$ un graphe et $k \geq 1$ un entier. Nous considérons le jeu tour à tour suivant. Initialement, une *cible* invisible et immobile est placée secrètement sur un sommet c de G. À chaque tour, il est possible d'*interroger* k sommets $v_1, \cdots, v_k$ de G, pour recevoir, en réponse, la distance exacte de c à ceux-ci, c'est-à-dire $(d_G(c, v_1), \cdots, d_G(c, v_k))$ avec $d_G(u, v)$ la longueur d'un plus court chemin de u à v dans G. Le but du jeu est de déterminer c en minimisant le nombre de tours. Nous notons $\lambda_k^{ex}(G)$ ce nombre minimum de tours dans le pire des cas (c'est-à-dire pour les pires placements de la cible) pour G et k. Par exemple, si P est un chemin, il suffit d'interroger une extrémité, *i.e.*, $\lambda_k^{ex}(P) = \lambda_1^{ex}(P) = 1$. Dans le cas d'une étoile S_n à n feuilles, il faut et il suffit d'interroger toutes les feuilles sauf une, *i.e.*, $\lambda_k^{ex}(S_n) = \lceil \frac{n-1}{k} \rceil$.

Nous étudions également la variante de ce jeu dans laquelle l'interrogation de k sommets $v_1, \cdots, v_k$ donne, en réponse, les distances relatives de ceux-ci à c. Plus précisément, la réponse est une partition $(V_1, \cdots, V_p)$ de $\{v_1, \cdots, v_k\}$, où chaque V_j contient les v_i les j^{es} plus proches de c. Autrement dit, pour tous $1 \leq i < j \leq k$, nous apprenons qui de v_i ou v_j est le plus proche de c (ou s'ils sont à égale distance), sans connaître leur distance exacte à c. Soit $\lambda_k^{rel}(G)$ le nombre minimum de tours nécessaires pour localiser la cible pour G et k dans ce cas. Notons que dans cette

variante, il est nécessaire que $k \geq 2$ dès que G a au moins deux sommets.

Pour ces deux problèmes, nous nous intéressons aussi aux paramètres duaux, dénotés $\kappa_\ell^{ex}(G)$ et $\kappa_\ell^{rel}(G)$, qui, pour un $\ell \geq 1$ fixé, représentent le plus petit nombre k tel que la cible peut être localisée en au plus ℓ tours en interrogeant au plus k sommets par tour, lorsque ceux-ci fournissent leur distance exacte et relative, respectivement, à la cible. Notons que $\lambda_k^{ex}(G) \leq \ell$ si et seulement si $\kappa_\ell^{ex}(G) \leq k$, et que $\lambda_k^{rel}(G) \leq \ell$ si et seulement si $\kappa_\ell^{rel}(G) \leq k$. Notons aussi que $\lambda_k^{ex}(G) \leq \lambda_k^{rel}(G)$ et $\kappa_\ell^{ex}(G) \leq \kappa_\ell^{rel}(G)$.

2.2.2 État de l'art

De nombreux jeux liés à la localisation d'une cible ou d'un agent mobile dans un graphe ont été étudiés. Les deux problèmes considérés ici généralisent certains d'entre eux. Notre jeu avec les distances exactes est une généralisation de la notion de *dimension métrique* introduite indépendamment par Slater [Sla75] et Harary et Melter [HM76]. Plus précisément, la dimension métrique $DM(G)$ d'un graphe G est la taille minimum d'un *ensemble résolvant*, *i.e.*, un ensemble de sommets qu'il suffit d'interroger pour localiser la cible immédiatement (en un tour). Ainsi, $DM(G)$ est exactement $\kappa_1^{ex}(G)$. Par ailleurs, $\lambda_k^{ex}(G) \leq \lceil DM(G)/k \rceil$ puisqu'il est possible d'interroger séquentiellement les sommets d'un ensemble résolvant minimum. Cependant, il est possible de montrer que l'écart entre $\lambda_k^{ex}(G)$ et $\lceil DM(G)/k \rceil$ peut être arbitrairement grand (par exemple la famille de graphes dont est issu le graphe G de la Figure 2.3).

Le problème avec les distances relatives est, lui, une généralisation de la notion de *dimension centroïdale* introduite par Foucaud, Klasing et Slater [FKS14]. Plus précisément, la dimension centroïdale de G correspond à $\kappa_1^{rel}(G)$. Calculer la dimension métrique ou centroïdale d'un graphe est NP-complet en général [GJ79, FKS14], ce qui indique que les deux problèmes de calculer $\kappa_1^{ex}(G)$ et $\kappa_1^{rel}(G)$ sont difficiles.

Le problème de déterminer $\lambda_1^{ex}(G)$ a aussi été considéré par Seager [Sea13], qui a donné la valeur exacte de ce paramètre pour des familles restreintes d'arbres. D'autres jeux proches sont étudiés dans la littérature, dans lesquels la cible peut se déplacer le long d'une arête à chaque tour. La minimisation du nombre de tours dans le cas d'une cible mobile avec $k = 1$ et distances exactes a été considérée dans certaines topologies (e.g., arbres) [Sea12, CCD+12, Sea14, BDE+17]. La minimisation du nombre de sommets interrogés par tour pour que la localisation d'une cible mobile soit possible est étudiée dans [BGG+b] (distances exactes) et [BGG+a] (distances relatives).

2.2.3 Contributions

Nous initions l'étude de nos deux problèmes en montrant que calculer la valeur de $\kappa_\ell^{ex}(G)$, $\kappa_\ell^{rel}(G)$, $\lambda_k^{ex}(G)$ ou $\lambda_k^{rel}(G)$ est NP-complet pour tout k ou ℓ fixé. Nous nous intéressons ensuite au problème avec les distances exactes, *a priori* plus simple.

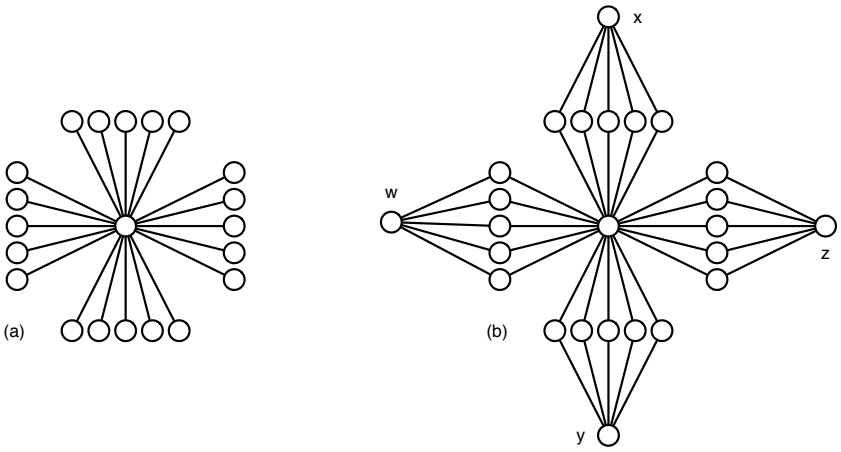

Figure 2.3: Exemple d'un sous-graphe isométrique H (a) d'un graphe G (b).

Dans le cas des arbres, nous résolvons complètement le problème de Seager [Sea13]. Plus précisément, dans le cas d'un arbre T, nous montrons, de manière surprenante, que la difficulté est de choisir les nœuds à interroger au premier tour. Plus précisément, lorsque k fait partie de l'entrée, déterminer $\lambda_k^{ex}(T)$ demeure NP-complet même en se restreignant aux arbres. En revanche, nous donnons un algorithme opérant en temps polynomial qui calcule une stratégie "optimale à partir du deuxième tour d'interrogation", Plus précisément, notre algorithme constitue une $(+1)$-approximation au problème. Une autre conséquence de notre algorithme est que $\lambda_k^{ex}(T)$ peut être déterminé en temps polynomial (en $|V(T)|$) lorsque k est fixé. Nos résultats positifs restent valides même pour les arbres pondérés (lorsque les arêtes peuvent avoir des longueurs).

Difficulté générale des deux problèmes. La complexité de nos deux problèmes peut être attribuée à plusieurs facteurs. Tout d'abord, bien que plusieurs interrogations successives permettent de restreindre l'espace S auquel la cible appartient, il est possible de construire des situations dans lesquelles la prochaine interrogation optimale incluera des sommets hors de S. Ainsi, pour des familles de graphes closes par sous-graphes, une stratégie récursive naïve n'est pas forcément optimale. Un autre facteur rendant compte de la complexité du problème est que savoir jouer dans un graphe G n'implique pas de savoir jouer dans un sous-graphe isométrique H de G. Autrement dit, les deux problèmes ne sont pas clos par sous-graphes isométriques. Ainsi, réduire un graphe à une instance connue dans laquelle les distances entre sommets sont préservées n'aide pas toujours.

Par exemple, la Figure 2.3 représente un graphe G (droite) contenant un sous-graphe isométrique H de G (gauche) vérifiant $\lambda_4^{ex}(G) < \lambda_4^{ex}(H)$ et $\lambda_4^{rel}(G) < \lambda_4^{rel}(H)$. Nous considérons ici le cas $k = 4$ par simplicité, mais la construction se généralise facilement à tout k (l'écart entre les paramètres peut être arbitrairement grand). Nous montrons que $\lambda_4^{ex}(G) < \lambda_4^{ex}(H)$. Notons que H est l'étoile S_{20} à 20 feuilles et

donc $\lambda_4^{ex}(H) = 5$. Considérons maintenant G. Au premier tour, si nous interrogeons les sommets w, x, y et z, dans le pire des cas les positions restantes pour c forment un stable de taille 5 (cas où l'un des quatre sommets indique être à distance 1), impliquant la localisation de c au prochain tour. Ainsi, $\lambda_4^{ex}(G) = 2$. La preuve dans le cas des distances relatives est similaire.

Nous montrons que déterminer $\lambda_k^{ex}(G)$ et $\lambda_k^{rel}(G)$ pour tout k (au moins 1 pour le premier paramètre, au moins 2 pour le second) fixé est NP-complet.

Théorème 5 *Soient $k \geq 1$ et $k' > 1$ fixés. Les problèmes de déterminer $\lambda_k^{ex}(G)$ et $\lambda_{k'}^{rel}(G)$ sont NP-complets dans la classe des graphes G de diamètre au plus 2.*
Soit $\ell \geq 1$ fixé. Les problèmes de déterminer $\kappa_\ell^{ex}(G)$ et $\kappa_\ell^{rel}(G)$ sont NP-complets.

La première partie du théorème est prouvée par une réduction de 3-Dimensional Matching. La seconde est prouvée via deux réductions de Metric Dimension et Centroidal Dimension.

Distances exactes et arbres. Pour un arbre T, la principale difficulté du problème de déterminer $\lambda_k^{ex}(T)$ provient exclusivement du premier tour, *i.e.,* du choix des k premiers nœuds à interroger. Précisément, nous montrons que trouver un bon ensemble de k nœuds à interroger au premier tour est NP-complet, par une réduction de Hitting Set.

Théorème 6 *Étant donné un arbre T et un entier k, le problème de déterminer $\lambda_k^{ex}(T)$ est NP-complet.*

De manière surprenante, nous donnons un algorithme polynomial qui calcule une stratégie "optimale à partir du deuxième tour d'interrogation". Cela vient du fait qu'avec les informations obtenues au premier tour, les positions potentielles restantes pour la cible se situent toutes à même distance d'un certain nœud r. Après une série de réductions, nous pouvons ensuite supposer, pour les tours suivants, que : T est enraciné en un nœud r, que toutes les feuilles de T sont à même distance de r et que la cible se trouve sur une feuille.

Théorème 7 *Soient $k \geq 1$ et T un arbre (possiblement arête-pondéré) à n nœuds enraciné en r, dont les feuilles sont toutes à même distance de r et abritent la cible. Il existe un algorithme qui calcule en temps $O(n \log n)$ (indépendant de k) une stratégie optimale pour localiser la cible.*

Notons $v_1, \cdots, v_d$ les voisins de r, T_{v_i} le sous-arbre de T enraciné en v_i et S l'ensemble des feuilles. Étant donné que la cible occupe un nœud de S (dont la distance à r est connue), une propriété importante est qu'il est possible de savoir si la cible est dans T_{v_i} (ou non) simplement en interrogeant un unique (et arbitraire) nœud de T_{v_i}. Localiser la cible revient donc à identifier le sous-arbre T_{v_i} la contenant, puis à répéter récursivement ce processus dans le sous-arbre où la cible a été détectée. Notons que lorsque nous testons si la cible est dans un sous-arbre T_{v_i}, il est parfois avantageux d'interroger plusieurs nœuds de T_{v_i} car cela permet de jouer *pleinement*

le premier tour du jeu si T_{v_i} en était l'entrée, dans l'éventualité de la présence de la cible dans ce sous-arbre. Nous prouvons que si nous testons les sous-arbres de manière gloutonne, de ceux qui demandent le plus de tours (e.g., une étoile avec de nombreuses feuilles) à ceux qui en demandent le moins (e.g., un chemin), il en résulte une stratégie optimale.

Précisément, l'algorithme fonctionne par programmation dynamique. Pour chaque sous-arbre T_{v_i}, il calcule une stratégie optimale (prenant un nombre de tours minimum ℓ_i) pour détecter la cible sur une feuille de T_{v_i} ainsi que le nombre minimum $\pi_i \leq k$ de nœuds qui doivent être interrogés lors du premier tour d'une telle stratégie optimale. Les sous-arbres sont ordonnés du plus grand au plus petit selon l'ordre lexicographique sur (ℓ_i, π_i). Notre algorithme simule une stratégie qui teste les sous-arbres dans cet ordre. Nous prouvons qu'une telle stratégie est optimale et que (ℓ_r, π_r) peut alors être calculé "facilement".

De l'algorithme précédent, nous déduisons une $(+1)$-approximation du problème de déterminer $\lambda_k^{ex}(T)$ pour un arbre T, et un algorithme polynomial lorsque k est fixé.

Théorème 8 *Soient T un arbre (possiblement arête-pondéré) à n nœuds, et $k \geq 1$ un entier. Il existe un algorithme qui calcule, en temps en $O(n \log n)$, une stratégie pour localiser la cible en au plus $\lambda_k^{ex}(T) + 1$ tours. De plus, en temps $O(n^{k+2} \log n)$, cet algorithme détermine $\lambda_k^{ex}(T)$ et une stratégie optimale.*

Pour localiser la cible en au plus $\lambda_k^{ex}(T) + 1$ tours en temps $O(n \log n)$, il suffit d'interroger un nœud arbitraire r au premier tour. Les nœuds pouvant accueillir la cible sont alors tous à même distance (connue) de r. Il est donc possible d'appliquer l'algorithme du Théorème 7.

Pour calculer $\lambda_k^{ex}(T)$ et une stratégie optimale en temps $O(n^{k+2} \log n)$, il suffit d'essayer les $\binom{n}{k}$ premiers tours possibles puis, pour chaque réponse qu'il est possible d'obtenir (il y en a au plus n), d'appliquer l'algorithme du Théorème 7. L'instance pour laquelle le nombre maximum de tours est minimum donne alors $\lambda_k^{ex}(T)$ et une stratégie optimale.

2.2.4 Perspectives

Notre résultat dans les arbres laisse ouverte la question de savoir si $\lambda_k^{ex}(T)$ peut être calculé en temps FPT (*i.e.*, en temps $f(k) \cdot poly(n)$) dans les arbres T à n nœuds. De manière générale, il serait intéressant d'étudier la localisation d'une cible, en distances exactes, dans d'autres familles de graphes comme les graphes d'intervalles ou les graphes planaires.

Le problème avec les distances relatives semble beaucoup plus compliqué que celui avec les distances exactes, même pour des topologies simples. Vers une meilleure compréhension de celui-ci, une première étape serait de pleinement comprendre le cas des chemins (*i.e.*, déterminer $\kappa_1^{rel}(P)$ pour tout chemin P, à savoir le plus petit nombre de sommets à interroger pour déterminer la cible en un tour), partiellement résolu dans [FKS14], avant de considérer une généralisation aux arbres.

2.3 Diamètre dynamique

Ce travail a été réalisé avec Emmanuel Godard [GM14].

2.3.1 Contexte, motivations, problèmes et état de l'art

Les réseaux statiques ont été largement étudiés dans la littérature. Pour les réseaux dynamiques, les recherches ont été essentiellement axées sur les dynamiques liées aux pannes, c'est-à-dire liées aux anomalies par rapport au comportement attendu. Or, il existe des systèmes pour lesquels les changements perpétuels font intrinsèquement partie de leurs fonctionnements normaux. Ces systèmes avec une instabilité permanente sont de plus en plus répandus (e.g. les réseaux mobiles). Pour de tels systèmes, il est nécessaire de repenser certaines notions fondamentales comme la connexité par exemple. En effet, d'un point de vue instantané, le système peut apparaître comme toujours non-connexe, bien qu'au cours du temps une certaine connexité existe. Nous nous intéressons aux *chemins temporels*, ou *trajets* en utilisant la terminologie de [CFQS12]. Le *diamètre dynamique* correspond à la longueur maximale des trajets "au plus tôt". De manière équivalente, cela représente le nombre de rondes nécessaires pour qu'un nœud puisse influencer causalement n'importe quel autre nœud du réseau (processus de diffusion).

Dans ce travail, nous considérons des réseaux homogènes, une sous-famille spécifique de réseaux dynamiques. Intuitivement un réseau homogène est tel, qu'à tout moment, les évolutions futures possibles sont identiques. En d'autres termes, le comportement du réseau ne dépend pas du moment d'observation (ce qui n'est pas le cas, en général, pour un réseau dynamique). En particulier, le diamètre dynamique ne dépend pas du moment de départ du trajet. Cette famille est donc particulièrement adaptée à l'étude du diamètre dynamique. Par ailleurs, du fait de la simplicité de sa définition, il nous semble que les bornes inférieures de complexité devraient s'appliquer à toute classe pertinente de réseaux dynamiques. Notons que dans le cas déterministe, le calcul du diamètre dynamique est polynomial [BF03].

Nous considérons différentes familles de réseaux homogènes : non-orientés, orientés, statiquement connexes (il existe un sous-graphe couvrant connexe commun à tous les graphes instantanés) et statiquement fortement connexes. Dans la suite, nous décrivons uniquement le modèle dans le cas non-orienté (voir [GM14] pour le modèle orienté).

Nous considérons un modèle à temps discret dans lequel la communication est fiable et s'effectue en rondes, mais avec une topologie de communication pouvant changer d'une ronde à l'autre. Soit $\mathcal{G} = (V, E)$ le graphe représentant le réseau de manière sous-jacente, avec V l'ensemble des nœuds et E l'ensemble des connexions possibles entre les nœuds. La communication avec une topologie donnée est décrite par un sous-graphe couvrant G de $\mathcal{G}$. L'ensemble $\Sigma = \{E' \mid E' \subseteq E\}$ représente toutes les communications instantanées possibles. Pour simplifier les notations, nous identifierons toujours un sous-graphe couvrant à son ensemble d'arêtes. Dans ce contexte, un élément $G \in \Sigma$ est un *graphe instantané*. Une *évolution de*

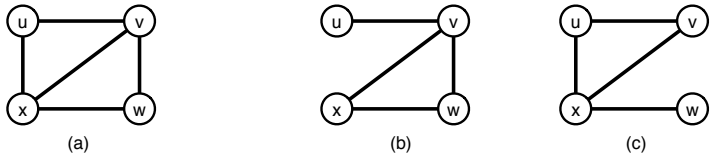

Figure 2.4: **(a)** Exemple de réseau dynamique $\mathbf{G}$ avec deux graphes instantanés possibles : **(b)** G_1 et **(c)** G_2.

communication (ou *évolution*) est une suite $(G_i)_{i\in\mathbb{N}}$ de graphes instantanés. Un *réseau dynamique non-déterministe* est un ensemble d'évolutions (c'est-à-dire un ensemble de suites de graphes instantanés). Étant donné $\mathbf{G} \subseteq \Sigma$, nous définissons $\mathcal{H}(\mathbf{G}) = \{(G_i)_{i\in\mathbb{N}} \mid G_i \in \mathbf{G}\}$ comme l'ensemble de toutes les suites possibles d'éléments de $\mathbf{G}$. Un réseau dynamique est *temps-homogène* (ou *homogène*) s'il existe $\mathbf{G} \subseteq \Sigma$ tel que l'ensemble d'évolution soit $\mathcal{H}(\mathbf{G})$. Étant donné qu'un réseau homogène est décrit par $\mathcal{H}(\mathbf{G})$, nous identifierons le réseau dynamique par $\mathbf{G}$. Étant donné une évolution $(G_i)_{i\in\mathbb{N}}$, la communication entre deux sommets u et v à une ronde donnée $i \in \mathbb{N}$ n'est possible que si l'arête $\{u, v\} \in E(G_i)$.

Soit $(G_i)_{i\in\mathbb{N}}$ une évolution de $\mathbf{G}$. La suite $(u_0, \ldots, u_p)$, $u_i \in V$, est un *trajet* de u_0 à u_p de départ i_0 et arrivée i_0+p si pour tout $0 \leq k < p$, l'arête $\{u_k, u_{k+1}\} \in E(G_{i_0+k})$ ou $u_k = u_{k+1}$. L'entier p est la longueur du trajet. Un trajet de u à v de départ i_0 et d'arrivée $i_0 + p$ est un *trajet au plus tôt* si tout trajet de u à v de départ i_0 est de longueur au moins p.

La distance dynamique au temps i_0 entre $u \in V$ et $v \in V$ est notée $d^{i_0}_{\mathbf{G}}(u, v)$. Cette distance représente la longueur maximale, pour toute évolution de $\mathbf{G}$, des trajets au plus tôt entre u et v de départ i_0. Dans les réseaux homogènes, la distance dynamique ne dépend pas du temps de départ et est donc notée $d_{\mathbf{G}}(u, v)$. De plus, contrairement aux réseaux dynamiques non-orientés généraux, nous avons $d_{\mathbf{G}}(u, v) = d_{\mathbf{G}}(v, u)$.

Le diamètre dynamique d'un réseau dynamique $\mathbf{G}$ est $L(\mathbf{G}) = \max_{u,v\in V} d_{\mathbf{G}}(u, v)$. De manière équivalente, il s'agit du temps maximum pour une diffusion depuis tout nœud de $\mathbf{G}$. Le diamètre d'un réseau homogène est défini si et seulement si tous les graphes instantanés sont connexes. Dans la suite, nous supposerons toujours que les graphes instantanés sont connexes.

Considérons par exemple l'instance décrite dans la Figure 2.4. Étant donné que $|V(\mathbf{G})| = 4$, nous avons nécessairement $L(\mathbf{G}) \leq 3$ car tous les graphes instantanés sont connexes. De plus, nous avons $d_{\mathbf{G}}(u, w) = 3$ en prenant toute évolution du type $(G_1, G_2, G_1, \ldots)$. En d'autres termes, u communique avec v lors de la première ronde, u et v communiquent avec x lors de la deuxième ronde et v et x communiquent avec w lors de la troisième ronde. Nous avons donc $L(\mathbf{G}) = 3$.

2.3.2 Contributions

Pour les réseaux dynamiques non-déterministes, nous montrons que le diamètre dynamique est difficile à calculer, voire difficile à approximer dans certains cas. De plus, même lorsqu'il existe un sous-graphe couvrant qui demeure statiquement connexe au cours de l'évolution du réseau, l'influence des arêtes dynamiques est difficile à évaluer. Plus précisément, nous prouvons le théorème suivant:

Théorème 9 *Le problème de calculer le diamètre dynamique d'un réseau dynamique homogène*

- *n'est pas dans* APX *même si le réseau est non-orienté;*
- *est* NP*-complet même si le réseau est statiquement connexe;*
- *n'est pas dans* APX *même si le réseau est orienté et statiquement fortement connexe.*

2.3.3 Perspectives

Outre la question de l'approximabilité dans le cas statiquement connexe, une autre question intéressante est de considérer le calcul de la longueur maximale des trajets "au plus court" et "au plus rapide" [CFQS12]. De plus, il est important de caractériser des classes d'instances intéressantes pour lesquelles le problème du diamètre dynamique est polynomial. Par exemple, quelle est la complexité du problème lorsque l'ensemble des graphes instantanés est de taille constante ?

Chapitre 3

Algorithmique pour la biologie structurale

Contents

3.1 Introduction

Mes recherches consistent à développer des méthodes algorithmiques pour la détermination de modèles haute-résolution de grands assemblages macro-moléculaires. Pour ce faire, j'ai décomposé le travail en quatre grandes parties décrites ci-dessous (qui correspondent aux quatre sections du chapitre).

- **Inférence pour des modèles basse résolution** (Section 3.2). Par spectrométrie de masse native, il est possible d'obtenir différents sous-complexes d'un assemblage macro-moléculaire A. Le problème est alors de déterminer un ensemble de contacts plausibles entre les sous-unités de A à partir de

l'information obtenue de ces sous-complexes. Le but est de construire des modèles basse résolution de grands assemblages. J'ai contribué à l'analyse et au développement d'algorithmes pour les problèmes d'optimisation combinatoire associés.

- **Caractérisation de paysages énergétiques moléculaires** (Section 3.3). La paysage énergétique d'une molécule est une hypersurface qui représente toutes les conformations de cette molécule. Un problème central est l'exploration de ce paysage énergétique afin de déterminer quelles sont les conformations stables (certains minima locaux) et les transitions entre elles (chemins dans le paysage énergétique). J'ai développé des algorithmes afin d'analyser et de comparer des paysages énergétiques pour améliorer les méthodes d'exploration.

- **Alignement structural pour le calcul de motifs communs** (Section 3.4). J'ai développé de nouvelles techniques (e.g. algorithmes de programmation dynamique) pour la recherche de motifs communs de différentes conformations. Ces résultats permettront une amélioration du temps de calcul des méthodes d'exploration de paysages énergétiques, en contraignant encore davantage les degrés de liberté.

- **Perspectives : Modèles haute résolution de grands assemblages macromoléculaires** (Section 3.5). Les résultats sur la caractérisation de paysages énergétiques moléculaires (Section 3.3) et les nouveaux algorithmes pour la recherche de motifs communs (Section 3.4) permettront d'améliorer les techniques d'exploration de paysages énergétiques (en termes de qualité et en temps de calcul). À partir de modèles basse résolution (Section 3.2), l'idée est ensuite de développer des techniques algorithmiques afin de déterminer les conformations plausibles des différentes protéines composant un assemblage. Je co-encadre la thèse de Viet-Ha Nguyen sur ce sujet. J'ai décidé de ne pas détailler nos premiers résultats dans ce document.

En lien avec la biologie structurale computationnelle, j'ai également travaillé sur des problèmes de représentation compacte de complexes simpliciaux (voir Section 5.1) et sur le problème de comparaison de deux clusterings motivé à l'origine par le caclul de motifs structurellement conservés (voir Section 4.2).

3.2 Inférence pour des modèles basse résolution

Ce travail a été réalisé avec Nathann Cohen (CNRS), Frédéric Havet (Coati), Thi viet ha Nguyen (ABS et Coati), Ignasi Sau Valls (CNRS) et Rémi Watrigant (Université Claude Bernard Lyon 1) [CHM+17, CHM+18].

3.2.1 Motivation et travaux existants

Un assemblage macro-moléculaire A peut être représenté par un graphe $G = (V, E)$: chaque sommet $v \in V$ correspond à une sous-unité (*e.g.* une protéine) et il y a une

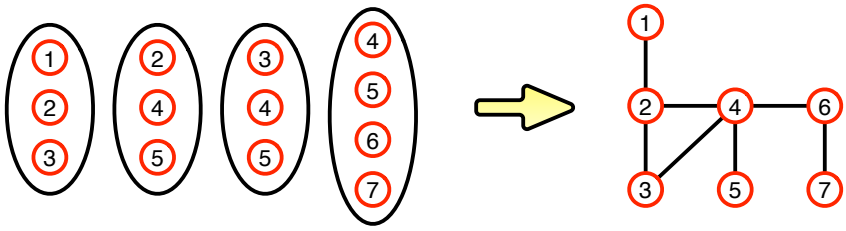

Figure 3.1: (Gauche) Quatre sous-complexes $C_1 = \{v_1, v_2, v_3\}$, $C_2 = \{v_2, v_4, v_5\}$, $C_3 = \{v_3, v_4, v_5\}$, $C_4 = \{v_4, v_5, v_6, v_7\}$. (Droite) Solution optimale E composée de $|E| = 7$ arêtes pour le PROBLÈME INFÉRENCE DE CONNECTIVITÉ MINIMUM.

arête entre deux sommets $u \in V$ et $v \in V$ si les deux sous-unités correspondantes à u et v sont en contact dans A. L'objectif des modèles basse résolution est de déterminer l'ensemble E. L'ensemble des sommets est supposé connu.

Par spectrométrie de masse native, il est possible d'obtenir de l'information concernant la structure d'un assemblage macro-moléculaire A. Plus précisément, nous avons la composition (en termes de sous-unités) de différents sous-complexes de A. Notons $C_1 \subseteq V$, ..., $C_t \subseteq V$ les t sous-complexes obtenus par spectrométrie de masse native. Naturellement, chaque sous-complexe forme une composante connexe de l'assemblage A. En termes de graphe, cela signifie que chaque sous-complexe forme un sous-graphe connexe de G. Autrement dit, le sous-graphe $G[C_i]$ induit par les sommets de C_i est un sous-graphe connexe de G, pour chaque $i \in \{1, \ldots, t\}$.

Étant donné un ensemble de sommets V et t sous-ensembles $C_1 \subseteq V$, ..., $C_t \subseteq V$, le PROBLÈME INFÉRENCE DE CONNECTIVITÉ MINIMUM consiste à trouver le plus petit ensemble d'arêtes E tel que, pour chaque $i \in \{1, \ldots, t\}$, $G[C_i]$ est connexe. Ce problème a été proposé pour déterminer les contacts plausibles entre les sous-unités étant donné que chacune d'entre elles est en contact avec un nombre limité d'autres sous-unités. Nous formalisons ci-dessous la version décision du problème.

Nom : PROBLÈME INFÉRENCE DE CONNECTIVITÉ
Instance : un entier $k \geq 1$, un ensemble de sommets V, t sous-ensembles $C_1 \subseteq V$, ..., $C_t \subseteq V$
Question : existe-t-il un ensemble d'arêtes E tel que $|E| \leq k$ et $G[C_i]$ est connexe pour tout $i \in \{1, \ldots, t\}$?

Remarque. Soit E une solution du problème. Le graphe $G = (V, E)$ n'est pas nécessairement connexe même si, d'un point de vue de l'application en biologie structurale, la contrainte de connexité du graphe semble naturelle.

Considérons l'exemple de la Figure 3.1 qui décrit un assemblage macro-moléculaire A composé de 7 protéines. L'ensemble des sommets représentant les protéines est $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$. La figure de gauche représente quatre sous-complexes $C_1 = \{v_1, v_2, v_3\}$, $C_2 = \{v_2, v_4, v_5\}$, $C_3 = \{v_3, v_4, v_5\}$, $C_4 = \{v_4, v_5, v_6, v_7\}$. La figure de droite montre une solution optimale E composée de $|E| = 7$ arêtes pour le PROBLÈME INFÉRENCE DE CONNECTIVITÉ MINIMUM. Observons que $G[C_i]$ est

connexe pour tout $i \in \{1, \ldots, t\}$. Par exemple, le sous-graphe induit par le sous-ensemble de sommets $\{v_4, v_5, v_6, v_7\}$ est un chemin de quatre sommets. Pour le PROBLÈME INFÉRENCE DE CONNECTIVITÉ, il n'y a pas de solution pour $k \leq 6$.

Le PROBLÈME INFÉRENCE DE CONNECTIVITÉ MINIMUM est également connu sous les noms de SUBSET INTERCONNECTION DESIGN, MINIMUM TOPIC-CONNECTED OVERLAY ou INTERCONNECTION GRAPH PROBLEM. Il a été étudié par différentes communautés dans le contexte de conception de systèmes de vide [DK95, DM88], des réseaux superposés évolutifs [CMTV07, HHI+12, OR11], des réseaux d'interconnexion reconfigurables [FHWE08, FW08], et avec des variantes dans le contexte d'inférence de réseaux sociaux [AAR10], de détermination de vainqueurs dans des systèmes d'enchères combinatoires [CDS04] et dans la représentation d'hypergraphes [BCPS10, KMN14, JP87, KS03].

Le PROBLÈME INFÉRENCE DE CONNECTIVITÉ MINIMUM a été prouvé NP-complet et difficile à approximer [AAC+13, ACCC15].

3.2.2 Problème de recouvrement d'un hypergraphe par un graphe

Nous avons étudié un problème plus général pour lequel chaque sous-complexe doit contenir un sous-graphe couvrant figurant dans une liste fixé de sous-graphes. Cela permet de prendre en compte des informations basse-résolution connues (e.g. ensemble de contacts probables dans les sous-complexes).

Formellement, pour une famille de graphes fixée (possiblement infinie) $\mathcal{F}$, un graphe G *recouvre* $\mathcal{F}$ dans un hypergraphe H si $V(H)$ est égal à $V(G)$ et le sous-graphe de G induit par chaque hyperarête de H contient un membre de $\mathcal{F}$ comme sous-graphe couvrant. Il est facile d'observer que le graphe complet à $|V(H)|$ sommets recouvre $\mathcal{F}$ dans un hypergraphe H dès que le problème admet une solution. Ainsi, nous avons étudié le PROBLÈME $\mathcal{F}$-RECOUVREMENT qui consiste à déterminer s'il existe un tel graphe G avec au plus k arêtes, pour un entier $k \in \mathbb{N}$ donné en entrée.

Nom : PROBLÈME $\mathcal{F}$-RECOUVREMENT
Instance : un hypergraphe H et un entier k
Question : existe-t-il un graphe G qui *recouvre* $\mathcal{F}$ dans H tel que $|E(G)| \leq k$?

Le PROBLÈME $\mathcal{F}$-RECOUVREMENT MINIMUM correspond à la version minimisation et consiste à calculer le nombre minimum d'arêtes over$_{\mathcal{F}}(H)$ pour recouvrir $\mathcal{F}$ dans H. Ce problème généralise le PROBLÈME INFÉRENCE DE CONNECTIVITÉ MINIMUM. En effet, si la famille $\mathcal{F}$ contient tous les graphes connexes, alors le PROBLÈME $\mathcal{F}$-RECOUVREMENT MINIMUM correspond au PROBLÈME INFÉRENCE DE CONNECTIVITÉ MINIMUM introduit précédemment.

Nous avons aussi étudié des variantes du problème avec une contrainte sur le degré des sommets. En effet, une sous-unité a un nombre limité de contacts avec les autres sous-unités (moins de 10). Ainsi un sommet a nécessairement un degré petit. Cette contrainte fait qu'il n'y a plus nécessairement une solution admissible. Les problèmes que nous avons étudiés sont décrits ci-dessous.

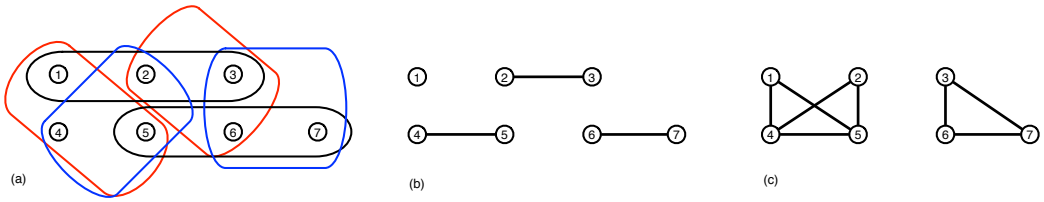

Figure 3.2: Exemple pour le PROBLÈME $(\Delta \leq k)$-$\mathcal{F}$-RECOUVREMENT et pour le PROBLÈME MAX $(\Delta \leq k)$-$\mathcal{F}$-RECOUVREMENT. **(a)** Une instance H. **(b)** Un graphe G avec $\Delta(G) \leq 1$ qui est une solution pour le PROBLÈME $(\Delta \leq 1)$-O_3-RECOUVREMENT avec $k = 3$ (avec O_3 le graphe composé de trois sommets et une arête). **(c)** Une solution pour le PROBLÈME MAX $(\Delta \leq 3)$-C_3-RECOUVREMENT avec $s = 3$ (avec C_3 le cycle composé de trois sommets).

Nom : PROBLÈME $(\Delta \leq k)$-$\mathcal{F}$-RECOUVREMENT
Instance : un hypergraphe H et un entier positif k.
Question : existe-t-il un graphe G qui $\mathcal{F}$-recouvre H tel que $\Delta(G) \leq k$?

Nous notons $\text{over}_{\mathcal{F}}(H, G)$ le nombre d'hyperarêtes de H qui sont $\mathcal{F}$-recouvertes par G. Une généralisation naturelle est de trouver $\text{over}_{\mathcal{F}}(H, k)$, le nombre maximum d'hyperarêtes $\mathcal{F}$-recouvertes par un graphe de degré maximum au plus k.

Nom : PROBLÈME MAX $(\Delta \leq k)$-$\mathcal{F}$-RECOUVREMENT
Instance : un hypergraphe H et un entier positif s.
Question : existe-t-il un graphe G tel que $\Delta(G) \leq k$ et $\text{over}_{\mathcal{F}}(H, G) \geq s$?

La Figure 3.2 illustre ces deux problèmes avec une instance simple : hypergraphe composé de 7 sommets et 6 hyperarêtes (les couleurs des hyperarêtes servent uniquement à optimiser la lecture).

3.2.3 Contributions

Nous avons tout d'abord prouvé un résultat de dichotomie concernant la complexité (polynomial versus NP-complet) du PROBLÈME $\mathcal{F}$-RECOUVREMENT MINIMUM. Plus précisément, nous montrons dans le Théorème 10 que les cas faciles (lorsque les graphes sans arête de bonnes tailles sont dans $\mathcal{F}$ ou si $\mathcal{F}$ contient seulement des cliques) sont les seules familles qui admettent algorithme polynomial: toutes les autres sont NP-complets. Étant donné un famille de graphes $\mathcal{F}$ et un entier positif p, $\mathcal{F}_p = \{F \in \mathcal{F} : |V(F)| = p\}$. Nous notons K_p le graphe complet composé de p sommets et $\overline{K_p}$ le graphe sans arête composé de p sommets.

Théorème 10 *Soit $\mathcal{F}$ une famille de graphes. Si, pour tout $p > 0$, soit $\mathcal{F}_p = \emptyset$, ou $\mathcal{F}_p = \{K_p\}$ ou $\overline{K_p} \in \mathcal{F}_p$, alors le* PROBLÈME $\mathcal{F}$-RECOUVREMENT MINIMUM *admet un algorithme polynomial. Sinon, le problème est* NP*-complet.*

Nous avons également analysé la complexité paramétrée du problème en prouvant des conditions suffisantes sur $\mathcal{F}$ afin d'obtenir des résultats de W[1]-difficulté, W[2]-

difficulté ou des problèmes FPT lorsque le paramètre est la taille de la solution (nombre d'arêtes). Cela donne une dichotomie FPT/W[1]-difficulté pour un problème relâchée pour lequel chaque hyperarête de H doit contenir un membre de $\mathcal{F}$ comme sous-graphe mais plus nécessairement couvrant.

Avec des contraintes sur les degrés, nous avons obtenu un résultat de dichotomie pour la complexité du PROBLÈME MAX $(\Delta \leq k)$-F-RECOUVREMENT et du PROBLÈME $(\Delta \leq k)$-F-RECOUVREMENT lorsque $F = O_p$ avec O_p le graphe composé de p sommets et d'une seule arête (Théorème 11).

Théorème 11 *Soient k et p deux entiers positifs. Si $p = 2$ ou si $p = 3$ et $k = 1$, alors le* PROBLÈME MAX $(\Delta \leq k)$-O_p-RECOUVREMENT *et le* PROBLÈME $(\Delta \leq k)$-O_p-RECOUVREMENT *sont dans* P*. Sinon, ils sont* NP*-complets.*

Pour le PROBLÈME MAX $(\Delta \leq k)$-F-RECOUVREMENT, nous avons caractérisé toutes les classes d'instances admettant un algorithme polynomial et celles pour lesquelles le problème est NP-complet (Théorème 12).

Théorème 12 *Le* PROBLÈME MAX $(\Delta \leq k)$-F-RECOUVREMENT *est dans* P *si $\Delta(F) > k$, ou F est un graphe vide, ou $F = K_2$ ou $k = 1$ et $F = O_3$. Sinon, il est* NP*-complet.*

Le Théorème 13 décrit quatre classes d'instances qui admettent un algorithme polynomial pour le PROBLÈME $(\Delta \leq k)$-$\mathcal{F}$-RECOUVREMENT MINIMUM.

Théorème 13 *Il existe un algorithme en temps polynomial pour les problèmes*

- $(\Delta \leq k)$-K-RECOUVREMENT *pour tout graphe complet K et tout entier $k \geq 1$,*
- $(\Delta \leq k)$-F-RECOUVREMENT *pour tout graphe k-régulier connexe F,*
- $(\Delta \leq 3)$-C_4-RECOUVREMENT*,*
- $(\Delta \leq 2)$-$\mathcal{P}$-RECOUVREMENT*.*

Pour le dernier problème, il existe un algorithme en temps linéaire.

3.2.4 Perspectives

Dans les travaux futurs, l'accent sera mis sur le développement d'algorithmes (d'approximation, exacts) qui seront intégrés dans la SBL. Le but est de pouvoir les utiliser en pratique avec des assemblages de grande taille. Nous nous focaliserons également sur les problèmes d'énumération associés car plusieurs solutions peuvent être nécessaires pour l'étape finale de mon programme de recherche sur la caractérisation haute-résolution d'assemblages macro-moléculaires (voir Section 3.5).

3.3 Caractérisation de paysages énergétiques moléculaires

Ces travaux ont été réalisés avec Frédéric Cazals, Tom Dreyfus, Christine Roth (équipe-projet ABS), Charles Robert (IBPC-LBT / CNRS), Joanne M. Carr et David J. Wales (University of Cambridge) [CMCW16, CDM+15, CM16]. Comme mentionné dans l'introduction générale, j'ai choisi de présenter mes contributions sur la comparaison de paysages énergétiques (avec Frédéric Cazals).

3.3.1 Contexte et motivations

Une protéine est composée de n atomes ayant chacun 3 paramètres décrivant sa position. Une position pour les n atomes forme une conformation de la protéine ($3n$ degrés de liberté). Une énergie est associée à chaque conformation et le paysage énergétique d'une protéine est l'ensemble des couples conformation et énergie. Un problème central est de comprendre les transitions entre les états biologiquement stables (qui correspondent à des minima locaux du paysage énergétique). En illustration, le problème de repliement d'une protéine (*folding*) consiste à savoir comment une protéine fait pour passer d'un état déplié à un état replié, et le problème d'amarrage moléculaire (*docking*) consiste à comprendre comment deux ou plusieurs molécules s'orientent afin de former un complexe stable. Il s'agit alors de comprendre les transitions dans le paysage énergétique. En particulier, cela revient à déterminer les chemins reliant les minima dans le graphe associé au paysage énergétique. Dans ce graphe, appelé *graphe de transitions compressé*, un sommet représente un minimum local et son bassin d'attraction (ensemble des conformations associées) et il y a une arête entre deux sommets s'il existe un col entre les deux minima associés à ces deux sommets. Le poids d'un sommet représente le volume du bassin associé au minimum correspondant à ce sommet.

Je me suis intéressé à l'analyse et à la comparaison d'échantillons de paysages énergétiques de protéines. Dans la suite, je présente uniquement mes travaux relatifs au problème de comparaison. Ce problème de comparaison est important car il permet, entre autres, de comparer différentes méthodes d'échantillonage.

3.3.2 Problème de flot avec contraintes de connectivité

Nous avons modélisé le problème de comparaison de deux échantillons d'un paysage énergétique d'une protéine par un problème de flot avec la prise en compte de contraintes de connectivité. Les deux échantillons sont représentés par deux graphes (sommet-valués) de transitions compressés G (graphe source) et G' (graphe demande). La valuation de chaque sommet v de G représente le volume maximum de flot que peut envoyer v aux sommets de G'. La valuation de chaque sommet v' de G' représente le volume de flot demandé par v'. Pour notre problème, cela peut par exemple représenter le volume des bassins associés aux sommets. De plus, le réel $c_{v,v'}$ représente le coût linéaire d'envoyer une unité de flot de $v \in V(G)$ à $v' \in V(G')$. Ce coût peut représenter la distance entre les deux conformations associées aux minima représentés par v et v'. Nous avons, entre autres, utilisé la *least root mean*

square deviation (lRMSD). Le PROBLÈME DE FLOT DE COÛT MINIMUM (*earth mover distance*, EMD) consiste à déterminer un flot (des sommets de G vers les sommets de G') de coût minimum satisfaisant les contraintes d'offres et de demandes.

Nous avons introduit de nouveaux problèmes de transport de masse en y intégrant des contraintes de connectivité. Plus précisément, pour chaque sous-ensemble de sommets de G induisant un sous-graphe connexe de G, nous imposons que l'ensemble des sommets de G' recevant un flot non nul de cet ensemble induise un sous-graphe connexe de G'. De plus, nous introduisons une contrainte sur le nombre de paires (v, v') pouvant supporter un flot non nul (typiquement linéaire en le nombre total de sommets). Ces contraintes sont notamment motivées d'un point de vue de l'application en biophysique. En effet, les contraintes de connectivité apportent une garantie topologique à la comparaison et la contrainte sur le nombre (linéaire) de paires traduit l'idée qu'un bassin représente l'union d'un certain nombre de bassins de plus petit volume ou correspond à une partie d'un plus gros bassin. Contrairement au problème classique EMD qui se résout par programmation linéaire, ces nouveaux problèmes sont très difficiles à résoudre en raison des contraintes décrites précédemment.

La Figure 3.3 décrit un exemple schématique de comparaisons de paysages énergétiques moléculaires. Le flot décrit permet d'établir une correspondance entre les bassins (et les minima locaux) des deux paysages.

3.3.3 Contributions

Nous avons introduit deux nouveaux problèmes de transport optimal : le PROBLÈME DE FLOT DE COÛT MINIMUM AVEC CONTRAINTES DE CONNECTIVITÉ (EMD-CC) et le PROBLÈME DE FLOT MAXIMUM AVEC CONTRAINTES DE COÛT ET DE CONNECTIVITÉ (EMD-CCC). En premier lieu, nous avons montré que, contraitement à EMD, EMD-CC n'induit pas une métrique. En effet, l'inégalité triangulaire n'est, en général, pas satisfaite. Nous avons prouvé que le problème de décision associé à EMD-CC est NP-complet même pour des classes d'instances très simples. Nous en avons déduit que le problème de décision associé à EMD-CCC est également NP-complet et que EMD-CC n'est pas dans APX. En d'autres termes, il n'existe pas d'algorithme polynomial pour EMD-CC garantissant un facteur d'approximation constant, à moins que P = NP. En revanche nous avons prouvé un schéma d'approximation polynomial pour EMD-CC pour certaines classes d'instances. Nous avons ensuite développé un algorithme heuristique glouton retournant une solution admissible de complexité en temps $O(n^3m^2)$ avec n et m les nombres de sommets de G et de G', respectivement.

Nous avons implémenté nos solutions en C++ générique dans la *Structural Bioinformatics Library* (SBL). D'un point de vue expérimental, nous avons tout d'abord comparé des payasages énergétiques synthétiques (paysages de Voronoï). Un diagramme de Voronoï est modélisé par un graphe G sommet-valué. Un sommet de G représente une cellule de Voronoï et il y a une arête entre deux sommets si les deux cellules correspondantes à ces sommets partagent une arête dans la triangulation de

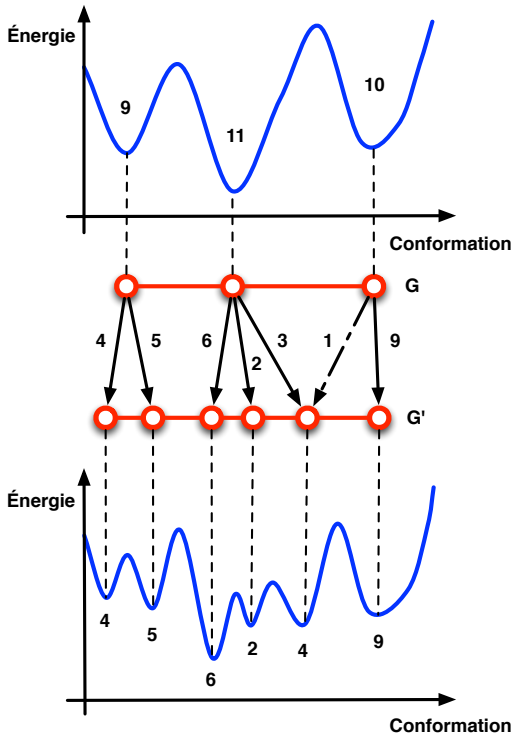

Figure 3.3: Deux paysages énergétiques (bleu) et les deux graphes G et G' correspondant (rouge) Les nombres sur les deux paysages énergétiques sont ici les volumes des bassins et sont les valeurs attribuées aux sommets des graphes. Le coût unitaire entre les sommets de G et les sommets de G' ne sont pas représentés. Les arcs de G vers G' sur la figure représentent un exemple de flot. Les contraintes de connectivité sont satisfaites. Dans cet exemple schématique, il est possible d'établir les correspondance entre les bassins des deux paysages (e.g. en enlevant un flot de volume 1 de la solution).

Delaunay associée à ce diagramme. Le poids d'un sommet représente la surface de la cellule de Voronoï correspondante à ce sommet. Nous avons comparé les plans de transport calculés par notre méthode gloutonne et ceux calculés par le programme linéaire associé à EMD (sans contraintes de connectivité) dans le but de quantifier les similarités entre deux diagrammes de Voronoï. Malgré sa complexité, notre algorithme permet de calculer des solutions pour des graphes composés d'au plus 1 200 sommets. Les solutions calculées pour EMD montrent que les contraintes de connectivité sont satisfaites par plus de 89% des sommets et par plus de 75% des arêtes de G. Les solutions calculées pour EMD-CCC montrent que, dans le pire des cas, les demandes de flot sont satisfaites à 62%. Les coûts des solutions calculées pour EMD-CCC sont en général inférieures à ceux des solutions pour EMD car pour ce dernier les demandes de flot doivent toutes être satisfaites. Le nombre d'arêtes de

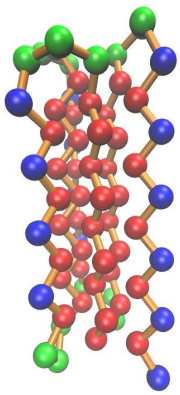

Figure 3.4: Exemple de conformation du modèle simplifié de protétines BLN69 composé de trois pseudo acides aminés : hydrophobe en rouge, hydrophyle en bleu et neutre en vert.

nos solutions sont linéaires en le nombre total de sommets de G et de G'.

Dans un deuxième temps, nous avons considéré un modèle simplifié de protéines BLN composé de trois pseudo acides aminés : hydrophobe (B), hydrophyle (L) et neutre (N). Ce modèle est composé d'une chaîne linéaire de k billes et de $k-1$ liaisons covalentes liant chaque paire de billes consécutives. Nous avons considéré $k = 69$ et chaque conformation est un point en dimension $d = 207$. La Figure 3.4 représente un exemple de conformation de BLN69 : hydrophobe en rouge, hydrophyle en bleu et neutre en vert. La base de données calculée par Oakley et al. recense 458 082 minima locaux et 378 913 cols. Dans la suite, je décris uniquement la validation concernant la comparaison de paysages énergétiques (voir [CDM+15, CMCW16] pour les autres résultats). Nous avons comparé des paysages énergétiques échantillonnés autour des dix plus petits minima locaux (correspondant aux états stables de la protéine). Les solutions calculées pour EMD montrent que les contraintes de connectivité sont satisfaites par 88% des sommets et par 79% des arêtes en moyenne (41% et 24% dans le pire des cas, respectivement). Pour EMD-CCC, notre algorithme calcule des solutions avec plus de 99% de demandes satisfaites (et respectant les contraintes de connectivité) pour chacune des 45 comparaisons effectuées. De plus, le coût du plan de transport pour EMD-CCC est quasi identique au coût calculé pour EMD.

3.3.4 Autres résultats et perspectives

Nous avons également considéré des variantes pour lesquelles la contrainte de connectivité était moins stricte. Plus précisément, il s'agissait de borner les distances entre les différentes composantes connexes recevant du flot d'un sommet ou d'un ensemble de sommets. En illustration, ces variantes permettent d'accélérer les temps de calcul avant de trouver une solution admissible avec un volume de flot minimum (au prix de ne pas avoir le strict respect des contraintes de connectivité). Une perspective de

recherche est d'analyser quantitativement les compromis possibles pour différentes classes d'instances.

Enfin, une question importante est d'utiliser ces algorithmes afin de comparer et d'améliorer les méthodes d'exploration de paysages énergétiques. Dans la section suivante, je décris une autre direction de recherche concourant á ce même objectif.

3.4 Alignement structural pour le calcul de motifs communs

Je me suis intéressé à un problème d'alignement structural dans le but de déterminer des motifs communs entre conformations d'une protéine (ou de protéines homologues). L'objectif est d'utiliser de nouveaux algorithmes d'alignement pour améliorer les méthodes d'exploration de conformations.

Ce travail a été réalisé avec Frédéric Cazals, Maria Guramare (Harvard University) et Romain Tetley. La Figure 3.5 schématise ces problèmes d'alignements structuraux (bleu) et l'apport pour les modèles haute-résolution (Section 3.5). Le rectangle bleu de gauche décrit nos contributions (détaillées dans la suite) et le rectangle bleu de droite représente les perspectives et travaux futurs.

3.4.1 Contexte, motivations et travaux existants

Comme expliqué précédemment, les alignements structuraux seront le premier ingrédient pour caractériser les motifs communs et améliorer les algorithmes d'exploration. Les méthodes FATCAT [YG05], Kpax [RGMV12] ou Apurva [AMDY11] optimisent une fonction globale. Apurva est un aligneur structural basé sur des cartes de contact, favorisant des alignements longs et flexibles. Kpax est un aligneur strucural basé sur une représentation géométrique du backbone de la protéine, favorisant une mesure géométrique. Il a été montré que les résultats obtenus ne permettaient pas de mettre en évidence les différentes échelles des motifs structuraux communs entre deux conformations ou deux protéines homologues [CT19]. Notre approche a pour but de prendre en compte ces différentes échelles.

Nous expliquons ci-dessous la modélisation formelle du problème avant d'expliquer les algorithmes de la littérature. Soient N et N' deux structures de deux protéines contenant n et n' acides aminés. Soient $c_{i,j}$ le coût de coupler l'acide aminé $i \in [\![1, n]\!]$ de N avec l'acide aminé $j \in [\![1, n']\!]$ de N'. Soit w_i (w_j, respectivement) le coût de ne pas coupler l'acide aminé $i \in [\![1, n]\!]$ de N ($j \in [\![1, n]\!]$ de N', respectivement). Le problème d'Alignement Structural consiste à déterminer un couplage optimal entre

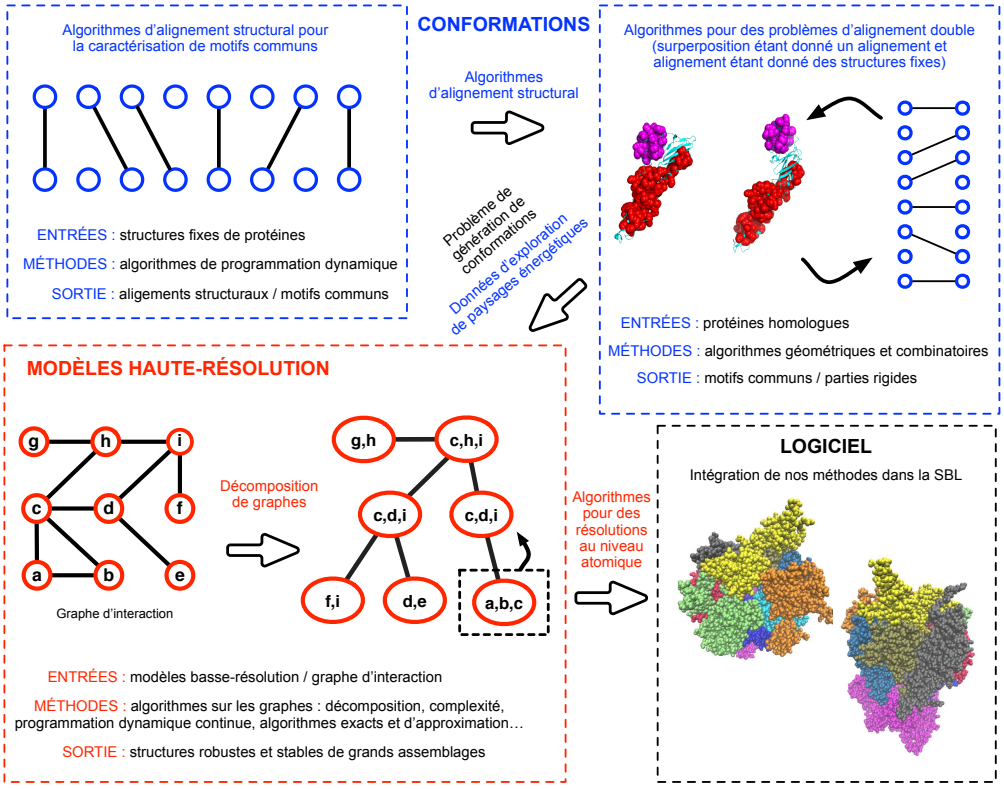

Figure 3.5: Résumé des contributions pour le problème de calcul de motifs communs de conformations (bleu) et liens avec le problème de modèles haute-résolution d'assemblages macro-moléculaires (rouge).

acides aminés de N et acides aminés de N'. Formellement,

$$\begin{cases} \text{Minimiser } \sum_{i\in[\![1,n]\!]}(1-m_i)w_i + \sum_{j\in[\![1,n']\!]}(1-m_j)w_j + \sum_{i\in[\![1,n]\!]}\sum_{j\in[\![1,n']\!]} m_{i,j}\ w_{i,j} + \\ \text{sous les contraintes} \\ \sum_{j\in[\![1,n']\!]} m_{i,j} = m_i \quad \forall i \in [\![1,n]\!], \\ \sum_{i\in[\![1,n]\!]} m_{i,j} = m_j \quad \forall j \in [\![1,n']\!], \\ \sum_{i_1\in[\![1,i-1]\!]}\sum_{j_1\in[\![j+1,n']\!]} m_{i_1,j_1} + \sum_{i_1\in[\![i+1,n]\!]}\sum_{j_1\in[\![1,j-1]\!]} m_{i_1,j_1} \leq (1-m_{i,j})nn' \\ \qquad\qquad \forall i \in [\![1,n]\!], \forall j \in [\![1,n']\!], \\ m_i, m_j, m_{i,j} \in \{0,1\} \leq 1 \quad \forall i \in [\![1,n]\!], \forall j \in [\![1,n']\!]. \end{cases} \tag{3.1}$$

La variable $m_i = 1$ si l'acide aminé $i \in [\![1,n]\!]$ de N est couplé avec un acide aminé de N'; si $m_i = 0$, alors l'acide aminé n'est pas couplé. (Similairement pour N'.) La variable $m_{i,j} = 1$ si l'acide aminé $i \in [\![1,n]\!]$ de N est couplé avec l'acide aminé $j \in [\![1,n']\!]$ de N'; si $m_{i,j} = 0$, alors les acide aminés i de N et j de N' ne sont pas couplés. Les contraintes de l'avant dernière ligne permettent d'empêcher un double couplage (i_1, j_1) et (i_2, j_2) avec, par exemple, $i_1 < i_2$ et $j_2 < j_1$.

Dans la littérature, ce problème a été modélisé par un problème de plus court

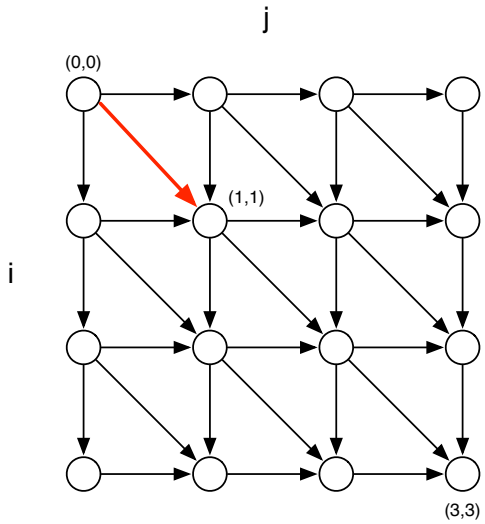

Figure 3.6: Modèle initial F pour la problème d'Alignement Structural avec $n = n' = 3$. Un arc diagonal représente un couplage entre deux acides aminés et un arc vertical (horizontal, respectivement) représente un non-couplage pour un acide aminé de la première protéine N (deuxième protéine, respectivement). Considérons une solution pour laquelle le seul couplage est celui de l'acide aminé 1 de N avec l'acide aminé 1 de N'. Il y a $6 = \binom{4}{2}$ chemins différents de $s = (0, 0)$ à $t = (3, 3)$ qui correspondent à ce couplage. Cela motive l'introduction d'un autre modèle.

chemin dans un graphe orienté F en forme de grille de dimension $(n + 1) \times (n' + 1)$. La Figure 3.6 montre un exemple avec $n = n' = 3$. Les arcs verticaux (horizontaux, respectivement) modélisent un non-couplage pour les acides aminés i de N (j de N', respectivement). Les arcs diagonaux représentent les couplages. Dans la Figure 3.6, l'arc rouge modélise le couplage entre les deux premiers acides aminés de N et de N'. Les coûts sur les arcs sont les coûts de couplages et de non-couplages mentionnés précédemment. Le problème d'Alignement Structural revient alors à trouver un plus court chemin entre le sommet $s = (0, 0)$ en haut à gauche de la grille et le sommet $t = (n + 1, n' + 1)$ en bas à droite de la grille.

La limite principale est la multiplicité de différents chemins orientés de s à t qui correspondent à un même couplage d'acides aminés. En effet, le nombre de chemins de $u_{i,j}$ à $u_{k,l}$ qui utilisent que des arcs horizontaux et verticaux est donné par le coefficient binomial $\binom{k-i+l-j}{k-i}$. Dans l'exemple de la Figure 3.6, si seuls les deux premiers acides aminés de N et de N' sont couplés, alors il y a 6 chemins différents possibles.

Dans nos travaux, nous avons utilisé un nouveau graphe ayant la propriété que deux chemins orientés différents correspondent nécessairement à deux couplages différents. Cela est notamment motivé par le développement d'algorithmes d'énumération efficace.

3.4.2 Problèmes de plus courts chemins contraints

Pour pallier la principale limite du modèle utilisé dans la littérature, nous définissons un graphe orienté $G = (V, A, c, w)$ qui est arc-coloré (fonction c) et arc-pondéré (fonction w). Chaque sommet (sauf la source et le puits) représente soit un couplage entre deux acides aminés ou, un ou plusieurs non-couplages d'acides aminés. L'ensemble des arcs peut être partitioné en deux sous-ensembles disjoints : l'ensemble des arcs rouges (avec l'étiquette 1) qui représente un couplage entre deux acides aminés et l'ensemble des arcs bleus (avec l'étiquette 0) qui représentent des non-couplages. Les poids des arcs sont les coûts des couplages et des non couplages. Le nombre de sommets et le nombre d'arcs est quadratique en le nombre total d'acides aminés. Une propriété importante est que deux chemins différents de la source s au puits t représentent deux alignements différents.

Dans la suite, nous expliquons deux nouveaux problèmes que nous avons introduits. Soit $\kappa \geq 0$ un entier. Un κ-chemin simple $P_{s,t}$ de G entre s et t est un chemin simple contenant exactement κ arcs colorés 1 (rouge). En d'autres termes, $\sum_{e \in A(P_{s,t})} c(e) = k$.

Nom : Problème chemin rouge
Instance : un graphe orienté arc-coloré et arc-pondéré $G = (V, A, c, w)$, un sommet source $s \in V$, un sommet puits $t \in V$, un entier $\kappa \geq 0$ et un réel $W > 0$
Question : existe-t-il un κ-chemin $P_{s,t}$ de G entre s et t tel que $\sum_{a \in A(P_{s,t})} w(a) \leq W$?

Soit $\lambda \geq 1$ un entier. Un chemin $P_{s,t}$ de G entre s et t contient une composante rouge de taille au moins λ si (et seulement si) il y a un sous-chemin P' de $P_{s,t}$ tel que $c(a) = 1$ (rouge) pour tout $a \in A(P')$ et $|A(P')| \geq \lambda$.

Nom : Problème composante rouge
Instance : un graphe orienté arc-coloré et arc-pondéré $G = (V, A, c, w)$, un sommet source $s \in V$, un sommet puits $t \in V$, un entier $\pi \geq 0$, un entier $\lambda \geq 1$ et un réel $W > 0$
Question : existe-t-il un chemin simple $P_{s,t}$ de G entre s et t tel que $P_{s,t}$ contient au moins π composantes rouges de taille au moins λ et tel que $\sum_{a \in A(P_{s,t})} w(a) \leq W$?

3.4.3 Contributions

Nous prouvons que les versions minimisations des problèmes sont très difficiles à approximer.

Théorème 14 *Les versions minimisations des* problèmes chemin rouge *et* composante rouge *ne sont pas dans APX.*

Nous prouvons cependant des algorithmes de programmation dynamique polynomiaux dans le cas des graphes sans circuit.

Théorème 15 *Il existe un algorithme en temps* $O(\kappa|A(G)|)$ *pour le* PROBLÈME CHEMIN ROUGE *et il existe un algorithme en temps* $O(\lambda^2 p|A(G)|)$ *pour le* PROBLÈME COMPOSANTE ROUGE.

3.5 Perspectives : Modèles haute résolution de grands assemblages macro-moléculaires

Comme décrit précédemment, mon programme de recherche est basé sur le couplage entre les modèles basse résolution obtenus par des algorithmes pour l'inférence de connectivité (les arêtes du graphe) et sur les possibles conformations de chaque protéine (sommet) obtenus avec des algorithmes d'exploration de paysages énergétiques. Voir Figure 1.2 et Figure 3.5. Il s'agit alors de développer des techniques algorithmiques pour résoudre le problème suivant. Étant donné un graphe (les contacts entre les protéines) et des ensembles de conformations pour chaque sommet (protéine), trouver une conformation par sommet qui minimise une fonction de score globale. Je co-encadre actuellement la thèse de Viet-Ha Nguyen sur ce sujet.

Chapitre 4

Réseaux de partage

Contents

4.1 Dynamique des groupes de partage

Ce travail a été réalisé avec Augustin Chaintreau, Guillaume Ducoffe et Jean-Claude Bermond [CDM13, BCDM18].

4.1.1 Contexte et motivation

"Qui reçoit de moi une idée augmente son instruction sans diminuer la mienne." De cet idéal énoncé par Thomas Jefferson on peut retenir que partager de l'information produit parfois un bénéfice mutuel. L'importance d'être *bien* informé motive une palette d'activité sociale (groupes, communautés, réseautage), récemment transformés par des outils en lignes, avec un net risque de surexposition. Facebook, Twitter, Weibo simplifient le partage et suppriment les frontières géographiques ou de classes. Mais ils transforment aussi le contexte de cette information, traditionnellement contrôlée par la diffusion entre cercles et groupes sociaux qui tiennent à l'écart ceux que l'on ne veut pas tenir informés de tout, voire du tout.

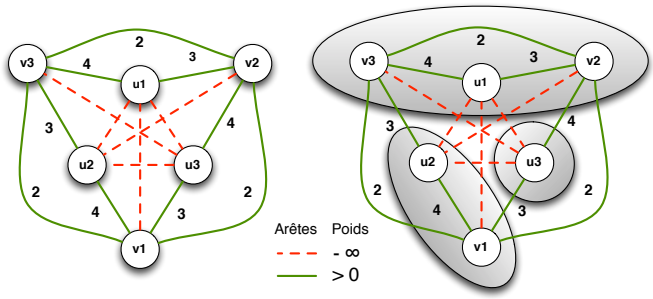

Figure 4.1: L'instance de gauche représente un graphe arête-valué avec $\mathcal{W} = \{-\infty, 2, 3, 4\}$. La partition décrite à droite est 1-stable mais il n'existe pas de partition 2-stable pour cette instance.

Comment se constituent – ou pourrait mieux se constituer – des groupes de partage d'information dans un graphe où les arêtes représentent soit un bénéfice mutuel, soit une incompatibilité ? Éviter les paires incompatibles en formant des partitions où l'information se propage peut sembler efficace. Assigner ces groupes par un algorithme de coloration de graphe, par contre, semble *simpliste* et peut se révéler *instable* car les noeuds, suivant leurs intérêts, peuvent dévier vers une configuration qui les favorisent. En effet, un tel algorithme n'optimise pas, en général, l'utilité individuelle. Nous étudions la *dynamique*, encore inconnue, de ce jeu distribué.

4.1.2 Modélisation du réseau social, des groupes

Le réseau est modélisé par un graphe arête-valué $G = (V, E, w)$ avec V représentant l'ensemble des utilisateurs. Le poids $w_{u,v}$ (positif ou négatif) entre deux sommets u et v représente l'utilité engendrée pour u et v si ces deux derniers partagent de l'information. Nous supposons que le graphe est complet en ajoutant des arêtes de poids 0 si nécessaire. Nous notons $\mathcal{W}$ l'ensemble des poids pris par les arêtes. La Figure 4.1 représente un exemple de graphe valué avec $\mathcal{W} = \{-\infty, 2, 3, 4\}$. Un poids $-\infty$ entre deux sommets représente le cas où les deux utilisateurs sont *ennemis* et ne veulent en aucun cas partager de l'information. Nous supposons enfin que les groupes de partage forment une *partition des sommets*. Autrement dit, un sommet appartient à un unique groupe. Étant donnée une partition C, l'utilité $f_u(C)$ d'un sommet u est la somme des poids des arêtes adjacentes à u dans son groupe. Formellement $f_u(C) = \sum_{v \in C(u)} w_{u,v}$ où $C(u)$ représente l'ensemble des sommets dans le même groupe que u. L'utilité du sommet v_3 pour la partition décrite dans la Figure 4.1 est 6. Notons que deux sommets ennemis (poids $-\infty$) ne sont jamais dans le même groupe mais que deux sommets liés par un poids négatif (autre que $-\infty$) peuvent être dans un même groupe en raison de compensations.

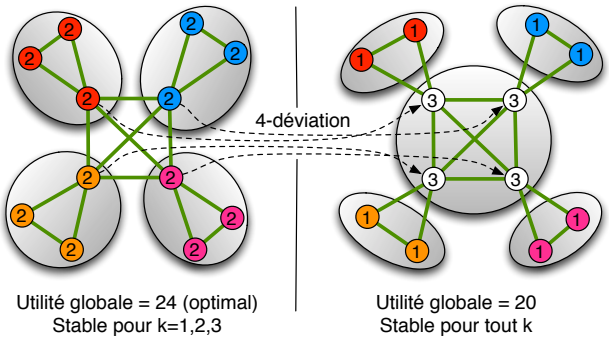

Figure 4.2: Exemple de 4-déviation avec $\mathcal{W} = \{-\infty, 1\}$. Dans la figure, une non-arête représente un poids $-\infty$ et une arête verte représente un poids de 1. La partition de gauche est 3-stable mais pas 4-stable, et l'utilité globale est maximale. La partition de droite est 4-stable mais l'utilité globale a diminué.

4.1.3 Dynamique du système

La dynamique du système est la suivante. Initialement, chaque utilisateur forme un groupe seul. Soit $k \geq 1$ la taille maximale constante d'une coalition. Étant donnés des groupes de partage, un sous-ensemble d'au plus k utilisateurs peut rejoindre un groupe existant ou créer un nouveau groupe si, et seulement si, leurs utilités respectives augmentent strictement. Ce changement est appelé une *k-déviation*. La partition est *k-stable* si, et seulement si, il n'y a pas de k-déviation possible.

La partition décrite dans la Figure 4.1 est 1-stable. En effet, aucun sommet ne peut créer un nouveau groupe seul et augmenter strictement son utilité. De plus, aucun sommet ne peut rejoindre (seul) un groupe existant car soit un ennemi y est présent soit il aurait une utilité strictement positive mais inférieure à son utilité courante. En revanche, cette partition n'est pas 2-stable car les deux sommets v_1 et v_2 peuvent former une coalition et rejoindre le groupe formé par u_3. L'utilité de v_1 passe de 4 à 5 et l'utilité de v_2 passe de 5 à 6. Après une telle 2-déviation, la partition obtenue est isomorphe à la précédente. Donc cette partition n'est pas 2-stable et, plus généralement, il est possible d'observer que cette instance n'admet pas de partition 2-stable.

Un problème important est de caractériser les classes d'instances admettant une partition k-stable. Pour ces instances, il est alors intéressant de déterminer si une partition k-stable est atteignable par le processus dynamique opérant dans le système. Pour ces instances, nous nous sommes intéressés au temps de convergence du processus dynamique (nombre de k-déviations avant d'obtenir la k-stabilité) dans le pire des cas notamment.

4.1.4 État de l'art

Kleinberg et Ligett ont montré dans [KL10] que si $\mathcal{W} = \{-\infty, 1\}$, alors il existe toujours une partition k-stable pour tout $k \geq 1$. La Figure 4.2 décrit une instance simple pour laquelle une 4-déviation existe entre une partition k-stable pour $k \leq 3$ qui maximise la sommet des utilités individuelles et une partition k-stable pour tout k mais dont la somme des utilités n'est plus optimale. Cette classe d'instances peut correspondre au cas où il n'y a que des *amis* et des *ennemis*, et où, pour des raisons de vie privée, un individu ne veut surtout pas partager d'information avec un ennemi. Pour ces instances particulières, ils ont montré que le temps de convergence est polynomial pour $k \in \{1, 2, 3\}$. Ils ont en revanche laissé ouvert le problème de la polynomialité du temps de convergence pour $k = 4$.

4.1.5 Contributions

Nous avons montré que, pour $k = 4$ et $\mathcal{W} = \{-\infty, 1\}$, le temps de convergence pouvait être $\Omega(n^{c \log(n)})$ avec c une constante et n le nombre d'utilisateurs. Pour $k \in \{1, 2\}$, nous avons prouvé une formule close pour le nombre exact de k-déviations dans le pire des cas (Tableau 4.1). De plus, nous avons montré que pour $\mathcal{W} = \{-\infty, 0, 1\}$, il existe pour tout réseau, une partition 1 et 2-stable (avec des temps de convergence polynomiaux) mais que certains réseaux n'admettent pas de partition 3-stable. Cette classe d'instances modélise le cas où les utilisateurs peuvent avoir des relations *neutres*. Plus généralement, nous montrons des résultats d'existence pour des poids généraux (Tableau 4.2) et nous prouvons que le problème de décider s'il existe une partition k-stable est NP-complet en général. L'intégralité de nos résultats et de nos preuves se trouve dans [DMC12].

Nous débutons par l'analyse du temps de convergence pour l'ensemble de poids $\mathcal{W} = \{-\infty, 1\}$. Pour cet ensemble de poids, nous représentons toute partition C par un *vecteur de partition* $\Lambda = (\lambda_n, \ldots, \lambda_1)$ de taille n où λ_i représente le nombre de groupes de taille exactement i pour tout $1 \leq i \leq n$. Nous avons montré que, pour tout $k \geq 1$ et pour toute partition, n'importe quelle k-déviation augmente le vecteur de partition selon l'ordre lexicographique. Donc le processus local converge toujours vers une partition k-stable. Autrement dit, il n'y a pas de cycle de k-déviations pouvant empêcher le processus d'atteindre une partition k-stable.

Un paramètre important est alors le nombre maximal de k-déviations avant d'atteindre la k-stabilité. Étant donnés $k \geq 1$ et $n \geq 1$, nous notons $L(k, n)$ ce temps de convergence, dans le pire des cas, pour un graphe avec au plus n sommets. Notons que $L(k', n) \geq L(k, n)$ pour tout $k' \geq k \geq 1$ et pour tout $n \geq 1$.

Kleinberg et Ligett ont montré dans [KL10] que $L(1, n) = O(n^2)$, $L(2, n) = O(n^2)$ et $L(3, n) = O(n^3)$. Pour $k \in \{1, 2\}$, leur preuve s'appuie sur le fait que la somme des utilités augmente strictement après n'importe quelle k-déviation. Le résultat s'obtient ensuite en remarquant que cette somme est bornée supérieurement par $O(n^2)$ car l'utilité individuelle est bornée supérieurement par $n - 1$. Pour $k = 3$, ils ont montré que la somme des carrés des utilités augmente toujours strictement après

k	Littérature	Nos résultats
1	$O(n^2)$	$\sim \frac{2}{3}n^{3/2}$
2	$O(n^2)$	$\sim \frac{2}{3}n^{3/2}$
3	$O(n^3)$	$\Omega(n^2)$
≥ 4	$O(2^n)$	$\Omega(n^{c\log(n)})$, $O(e^{\sqrt{n}})$

Tableau 4.1: Temps de convergence maximal $L(k,n)$ pour $\mathcal{W} = \{-\infty, 1\}$.

$\mathcal{W}$	$k(\mathcal{W})$
$\{-\infty, a, b\}$, $0 < a < b$	1
$\{-\infty, -\mathbb{N}, 0, 1\}$	2
$\{-\infty, b\}$, $b > 0$	∞
$\mathcal{W} \subseteq \mathbb{N}$; $\mathcal{W} \subseteq -\mathbb{N}$	∞
$-\mathbb{N} \cup \{N\}$ [1]	∞

Tableau 4.2: Existence de partition k-stable représenté par $k(\mathcal{W})$.

n'importe quelle k-déviation (l'utilité globale peut ne pas augmenter) et donc que $L(3,n) = O(n^3)$. Ils ont en revanche laissé ouvert le problème de la polynomialité pour $k = 4$. Dans ce cas, toute fonction (additive) potentielle bornée par un polynôme peut décroître strictement pour certaines 4-déviations. Nous avons résolu le problème ouvert de [KL10] en montrant que $L(k,n)$ n'est pas polynomial pour tout $k \geq 4$.

Théorème 16 *$L(4,n) = \Omega(n^{c\log(n)})$ avec c une constante.*

En utilisant les techniques développées dans la preuve du Théorème 16, nous avons également prouvé une borne inférieure pour $k = 3$: $L(3,n) = \Omega(n^2)$. Nous avons également prouvé que $L(k,n) = O(e^{\sqrt{n}})$ améliorant la meilleure borne supérieure exponentielle connue. La preuve s'appuie sur le fait que le nombre de vecteurs de partitions différents est $O(e^{\sqrt{n}})$. Rappelons qu'un vecteur de partition augmente strictement selon l'ordre lexicographique après n'importe quelle k-déviation.

Pour améliorer les bornes supérieures, nous avons montré que nous pouvons nous ramener au cas $\mathcal{W} = \{1\}$. Autrement dit, pour tout $k \geq 1$ et pour tout $n \geq 1$, $L(k,n)$ est atteint pour $\mathcal{W} = \{1\}$. Nous avons alors prouvé que $L(1,n) = L(2,n)$ en montrant que cette valeur était égale à la longueur de la plus longue chaine dans le treillis des partitions. Utilisant les résultats de [GK86, GK93], nous déduisons la formule close :

Théorème 17 *$L(1,n) = L(2,n) = 2\binom{m+1}{3} + mr$, où r et m sont les uniques solutions de $n = \frac{m(m+1)}{2} + r$, $0 \leq r \leq m$. Cela implique que $L(1,n) = L(2,n) \sim \frac{2}{3}n\sqrt{n}$ quand n est grand.*

Le Tableau 4.1 résume les résultats des travaux existants et nos contributions.

Nous terminons par expliquer nos contributions sur l'analyse d'existence de partition k-stable pour des poids généraux. Étant donné un ensemble de poids $\mathcal{W}$, $k(\mathcal{W})$ est défini de la manière suivante : pour tout $k \leq k(\mathcal{W})$, il existe une partition k-stable pour tout graphe et il existe un graphe qui n'est pas $(k(\mathcal{W})+1)$-stable. S'il existe une partition k-stable pour tout $k \geq 1$, nous définissons $k(\mathcal{W}) = \infty$ (*e.g.* $k(\{-\infty, 1\}) = \infty$).

Nous avons prouvé que $k(\mathcal{W}) \geq 1$ pour tout $\mathcal{W}$, c'est-à-dire qu'il existe toujours une partition 1-stable. Nous avons ensuite caractérisé les ensembles $\mathcal{W}$ tels que $k(\mathcal{W}) = \infty$. Précisément $k(\mathcal{W}) = \infty \Leftrightarrow \mathcal{W} = \{-\infty, b\}$ avec $b > 0$, $\mathcal{W} \subseteq \mathbb{N}$

ou $\mathcal{W} \subseteq -\mathbb{N} \cup \{N\}$ [1]. Nous avons également montré que si nous ajoutons des relations *neutres* entre individus à la classe d'instances ennemis et amis, alors la 3-stabilité n'est plus garantie. Autrement dit, pour $\mathcal{W} = \{-\infty, 0, 1\}$, il existe un graphe G qui n'admet pas de partition 3-stable et il existe toujours une partition 2-stable (avec temps de convergence polynomial). De plus, nous avons prouvé que $k(\{-\infty, 0, 1\}) = k(\{-\infty, -\mathbb{N}, 0, 1\})$. Enfin pour $\mathcal{W} = \{-\infty, a, b\}$, $0 < a < b$, nous avons montré que $k(\mathcal{W}) = 1$. Le Tableau 4.2 résume nos résultats.

Pour conclure, nous avons démontré que le problème de décider s'il existe une partition k-stable est NP-complet. Nous avons utilisé le problème de l'ensemble indépendant de cardinalité maximale dans notre réduction.

Théorème 18 *Pour tout $\mathcal{W}$ contenant $-\infty$ et pour tout $k > k(\mathcal{W})$, étant donné un graphe G avec les poids $\mathcal{W}$, le problème de décider s'il existe une partition k-stable pour G est* NP*-complet.*

4.1.6 Perspectives

En plus de poursuivre notre étude pour des poids généraux, nous proposons d'étudier la formation des groupes de partage dans les réseaux sociaux lorsque les sommets peuvent appartenir à plusieurs groupes différents. Certains réseaux n'admettant pas de configuration k-stable dans le cas de la partition, peuvent maintenant avoir une configuration k-stable. Mais nos premiers résultats montrent également que cela peut rendre le réseau instable. En effet, la 3-stabilité n'est plus garantie pour l'ensemble de poids $\mathcal{W} = \{-M, 1\}$ lorsque les sommets appartiennent à deux groupes. Une autre extension intéressante est de prendre en compte des *utilités transitives*. Un utilisateur peut ne pas vouloir être dans un groupe comprenant simultanément une personne de son cercle amical et une personne de son cercle professionnel (alors que les utilités respectives sont intrinsèquement positives). Nous envisageons une modélisation à base d'hypergraphes.

4.2 Comparaison de clusterings

Dans sa thèse soutenue en novembre 2018, Romain Tetley (équique-projet ABS) a étudié le problème de calculer des motifs structurellement conservés. En illustration, étant donné deux conformations d'une même protéine, il s'agit de déterminer quels sont les motifs communs à ces deux conformations. Pour cela, Romain a développé une méthode complexe mélangeant, entre autres, calcul de distances et de rang d'acides aminés, calcul de la dynamique des composantes connexes formés de ces acides aminés et calcul de persistence. Romain a alors soulevé un problème de comparaison de composantes connexes des deux conformations. Pour plus de détails sur le problème et sur la méthode, voir [Tet18]. Dans la suite, nous étudierons

[1] Le poids $N > 0$ est plus grand que n fois la valeur absolue du poids négatif le plus petit (différent de $-\infty$). Nous pouvons considérer qu'il vaut $+\infty$ quand on le compare aux autres poids (de valeur finie).

ce problème en termes de comparaison de deux clusterings et expliquerons nos contributions. Ce travail a été réalisé avec Frédéric Cazals, Romain Tetley et Rémi Watrigant [CMTW17, CMTW18].

4.2.1 Contexte, motivations et état de l'art

Le clustering, qui consiste à regrouper des points de données en ensembles disjoints d'éléments similaires, est une tâche fondamentale en analyse de données. De nombreuses classes de méthodes ont été développées (approches hiérarchiques [DH73], K-means [AV07], approches basées sur la densité [Che95, CM02]...). Toutefois, bien que des statistiques existent pour comparer globalement deux clusterings [Mei07], la recherche de correspondances entre les deux ensembles de clusters fait défaut. Nous avons pallier ce manque en présentant un modèle théorique permettant d'établir des groupements de clusters (ou meta-clusters). Notre approche est basée sur un problème d'optimisation combinatoire sur les graphes : le D-family-matching. Nous avons montré que ce problème est NP-difficile mais avons proposé des algorithmes exacts pour certaines classes d'instances et un algorithme générique dans le cas général. Enfin, nous avons illustré l'utilité d'un tel modèle en l'appliquant au grainage de K-means.

4.2.2 Formalisation du problème et exemples

Soit $t \geq 1$ un entier. Considérons un ensemble d'éléments $Z = \{z_1, \ldots, z_t\}$ et deux clusterings F et F' de Z. Formellement, $F = \{F_1, \ldots, F_r\}$, $r \geq 1$, avec $F_i \subseteq Z$, $F_i \neq \emptyset$ et $F_i \cap F_j = \emptyset$ pour tout $i, j \in \{1, \ldots, r\}$, $i \neq j$. De manière analogue, $F' = \{F'_1, \ldots, F'_{r'}\}$, $r' \geq 1$, avec $F'_i \subseteq Z$, $F'_i \neq \emptyset$ et $F'_i \cap F'_j = \emptyset$ pour tout $i, j \in \{1, \ldots, r'\}$, $i \neq j$. La Figure 4.3 (a) décrit deux clusterings F et F' avec $t = 40$, $r = 2$ et $r' = 5$. Le graphe d'intersections arête-pondéré $G = (V, E, w)$ associé à Z, F et F', est tel que $V = \{F_1, \ldots, F_r\} \cup \{F'_1, \ldots, F'_{r'}\}$, $E = \{\{F_i, F'_j\} \mid F_i \cap F'_j \neq \emptyset, 1 \leq i \leq r, 1 \leq j \leq r'\}$ et le poids d'une arête $e = \{F_i, F'_j\} \in E$ est $w_e = |F_i \cap F'_j|$. Autrement dit, les sommets représentent les clusters et le poids d'une arête est le nombre d'éléments communs aux deux clusters correspondant aux deux sommets (si cette intersection est vide, alors il n'y a pas d'arête). Ainsi, tout graphe d'intersections est un graphe biparti. De plus, on peut montrer que tout graphe biparti est le graphe d'intersections de deux clusterings d'un ensemble Z. La Figure 4.3 (b,c) représente le graphe d'intersections de l'exemple décrit précédemment. Nous définissons maintenant la notion de D-family-matching.

Définition 1 (D-family-matching) *Soient $D \geq 1$ un entier positif et $G = (V, E, w)$ un graphe d'intersections de deux clusterings. Un D-family-matching pour G est une famille $\mathcal{S} = \{S_1, \ldots, S_k\}$ telle que pour tout $i, j \in \{1, \ldots, k\}$, $i \neq j$, alors $S_i \subseteq V$, $S_i \neq \emptyset$, $S_i \cap S_j = \emptyset$, et le graphe $G[S_i]$ induit par l'ensemble de sommets S_i a diamètre au plus D. Le score $\Phi(\mathcal{S})$ d'un D-family-matching $\mathcal{S}$ est $\Phi(\mathcal{S}) = \sum_{i=1}^{k} \sum_{e \in E(G[S_i])} w_e$.*

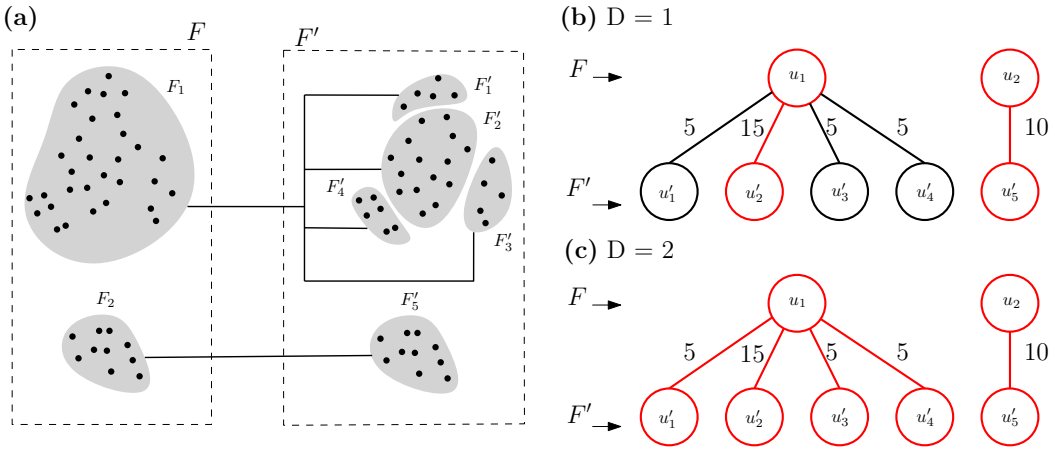

Figure 4.3: Comparaison de deux clusterings du même ensemble de données 2D composé de 40 points. **(a)** Clustering F composé de 2 clusters de 30 et 10 points. Clustering F' composé de 5 clusters de 5, 15, 5, 5 et 10 points. Le graphe d'intersections associé aux deux clusterings est représenté dans (b) et (c): un sommet par cluster, une arête entre deux sommets si les clusters correspondants partagent au moins un point, le poids d'une arête est le nombre de points en commun. Notre méthode regroupe les clusters en meta-clusters et est paramétré par le diamètre D des sous-graphes connectant les clusters à l'intérieur des meta-clusters (en rouge). Les méthodes existantes sont basées sur les couplages maximaux d'un graphe et correspondent au cas $D = 1$. **(b)** Avec $D = 1$, un couplage est obtenu: F_1 avec F'_2 et F_2 avec F'_5. **(c)** Avec $D = 2$, $\{F_1\}$ est couplé avec le meta-cluster $\{F'_1, F'_2, F'_3, F'_4\}$ et $\{F_2\}$ est couplé avec $\{F'_5\}$.

Nom : Problème D-family-matching
Instance : Un graphe d'intersections G, un réel positif ζ
Question : existe-t-il un D-family-matching $\mathcal{S}$ pour G tel que $\Phi(\mathcal{S}) \geq \zeta$?

La version maximisation consiste à déterminer $\Phi_D^G = \max_{\mathcal{S} \in \mathcal{S}_D(G)} \Phi(\mathcal{S})$ avec $\mathcal{S}_D(G)$ l'ensemble de tous les D-family-matching pour G. Intuitivement, le problème revient à déterminer un D-family-matching qui minimise les inconsistences. La Figure 4.3 décrit une instance simple du problème D-family-matching et des solutions optimales pour différentes valeurs de D.

Dans la suite, nous nous focalisons sur la version maximisation du problème D-family-matching sans le mentionner explicitement.

4.2.3 Contributions

Pour $D = 1$, le problème est équivalent à celui du couplage de poids maximum dans les graphes bipartis. En effet, comme G est biparti, tout sous-graphe de diamètre au plus 1 est nécessairement une arête ou un sommet. Comme le problème du couplage maximum admet un algorithme de complexité $O(n^2 \log n + nm)$ [FT87], nous en

déduisons la même complexité pour le problème 1-family-matching. Comme nous allons le voir, la complexité du problème est différente si $D > 1$.

Nous prouvons dans le Théorème 19 que le problème est difficile à approximer dès que le diamètre est au moins 2.

Théorème 19 *Soit $D \geq 2$. Le* PROBLÈME D-FAMILY-MATCHING *est* APX*-difficile pour :*

- *les graphes bipartis de degré maximum* 3 *(avec des poids unitaires si $D = 2$);*
- *les graphes bipartis de degré maximum* 4 *même si le poids maximum est constant.*

Concernant l'approximabilité du problème, une approche naturelle serait de choisir de manière gloutonne une collection de sous-graphes de diamètre D. Cette approche, qui donne une 2-approximation dans le cas $D = 1$ pour des graphes non bipartis de degré maximum constant, échoue malheureusement dans notre cas pour $D = 2$. Une autre stratégie serait alors de partir d'une solution (exacte ou approchée) du problème $(D-1)$-family-matching afin d'obtenir un D-family-matching. Malheureusement, il est possible de construire des instances pour lesquelles une solution optimale avec des meta-clusters de diamètre $D-1$ est arbitrairement plus mauvaise qu'une solution optimale avec des meta-clusters de diamètre D. La question de l'existence d'un algorithme d'approximation (avec un rapport constant) reste ouverte.

Nous prouvons dans le Théorème 20 des algorithmes de programmation dynamique polynomiaux lorsque le graphe d'intersections est une union d'arbres, de chemins ou de cycles. Le nombre de sommets est noté n.

Théorème 20 *Soit $D \geq 1$. Il existe un algorithme résolvant optimalement le* PROBLÈME D-FAMILY-MATCHING *en temps*

- *$O(D^2\Delta^2 n)$ si G est une union disjointe d'arbres de degré maximum Δ,*
- *$O(Dn)$ si G est une union disjointe de chemins,*
- *$O(D^2 n)$ si G est une union disjointe de cycles.*

Nous avons aussi développé un algorithme générique pour notre problème, qui consiste à considérer des arbres couvrants de G (de manière séquentielle) et de calculer des D-family-matching pour G en se basant sur ces arbres. Les trois ingrédients principaux sont les suivants:

- *Un générateur d'arbres couvrants $\mathcal{R}(G,t)$.* Cette fonction calcule l'arbre couvrant enraciné T^t de G qui est utilisé à l'étape $t \geq 1$ par l'Algorithme $\mathcal{A}$.
- *Un algorithme $\mathcal{A}(G, T^t, D)$.* Cet algorithme calcule un D-family-matching $\mathcal{S}^t$ pour G basé sur l'arbre couvrant $\mathcal{R}(G,t) = T^t$ de G. L'algorithme $\mathcal{A}(G, T^t, D)$ peut être celui décrit dans le Théorème 20.

- *Une propriété d'arrêt* $\Pi(\mathcal{M})$. Cette dernière dépend de l'ensemble de solutions $\mathcal{M}$ calculées précédemment. Tant que celle-ci n'est pas satisfaite, nous générons un autre arbre couvrant enraciné T^t de G (en utilisant $\mathcal{R}$) et calculons un D-family-matching $\mathcal{S}^t$ pour G basé sur T^t (en utilisant $\mathcal{A}$).

Nous avons également prouvé un algorithme de programmation dynamique sur un arbre couvrant mais qui prend également en compte les autres arêtes du graphe (qui ne sont pas dans l'arbre). Nous avons montré qu'il existe au moins un arbre couvrant tel que cet algorithme retourne une solution optimale au problème. Le prix à payer est de ne plus avoir nécessairement une complexité polynomiale. Voir [CMTW17] pour plus de détails. Une question intéressante est de savoir pour quelles classes d'instances, il existe un résultat analogue (exact ou approximation) pour l'algorithme polynomial décrit dans le Théorème 20.

La partie expérimentations n'est pas expliquée dans ce document (Voir [CMTW17]).

4.3 Réseau anti-gaspillage

Ce travail a été réalisé avec Joanna Moulierac, Jean-Baptiste Caillau, Enzo Giusti, et Xuchun Zhang.

4.3.1 Contexte et problématique

Sur les 10 millions de tonnes de produits alimentaires gaspillés chaque année en France, 50% sont des fruits et légumes. Une des raisons de ce gaspillage réside dans les normes de calibrage et d'aspect imposées par les grossistes qui peuvent pousser un maraîcher à jeter près de 20% de sa production. En parallèle, même si le besoin de se mobiliser contre le gaspillage fait consensus dans l'opinion publique, le passage à l'acte est compliqué par le manque de temps ou de savoir-faire. Pourtant, quelle que soit sa position dans la filière alimentaire, chaque acteur a de la valeur de par l'utilisation qu'il fait des produits. Le problème est que l'offre et la demande sont déconnectées au mauvais moment.

D'où la mission de Oui!Greens : trouver pour chaque produit alimentaire moche, hors calibre ou abimé mais consommable le ou les acteurs de la filière capables de le revaloriser avant qu'il ne soit gaspillé. Nous proposons donc aux maraîchers et distributeurs qui disposent de surplus de produits consommables, une application mobile qui identifie les acteurs locaux les plus à même d'éviter que ces produits ne soient perdus et les met en relation.

Pour cela, Oui!Greens facilite l'action pour tous les acteurs de la chaîne, vendeurs comme acheteurs. Plutôt que de se contenter de signaler les disponibilités des produits, Oui!Greens mobilise la spécificité des besoins de chaque maillon de la chaine pour revaloriser tout type de produit alimentaire brut, quel que soit son état. Il s'agit de trouver pour chaque produit, le ou les bons acteurs au bon endroit et au bon moment. Trop d'annonces : l'utilisateur est noyé, ne se sent pas concerné et n'agit pas. Trop peu : il est frustré / découragé et le produit ne trouve pas

preneur à temps. Dans les deux cas, vendeur et acheteur en reviennent à percevoir l'action anti-gaspillage comme une contrainte. L'affectation des annonces est donc primordiale. Dans la suite, nous proposons une première modélisation de ce problème en termes de problème combinatoire dans les graphes.

4.3.2 Modélisation et contributions

Dans notre modélisation, nous avons défini un problème d'optimisation combinatoire dans les graphes. Soit $G = (A \cup P, E)$ un graphe biparti. L'ensemble A représente les acteurs, l'ensemble P représente les produits et l'ensemble E représente les liens entre un acteur et un produit. Les deux ensembles A et P sont indépendants (graphe biparti). Si un acteur $a \in A$ n'a pas d'intérêt pour un produit $p \in P$, alors il n'y a pas d'arête entre a et p. Sans perte de généralité, nous supposons que $|N_G(x)| \geq 1$, $\forall x \in A \cup P$. Nous notons $\delta_a \in \mathbb{N}$ pour tout $a \in A$ le nombre maximum de publicités que a souhaite recevoir (e.g. par jour). Dans sa version non pondérée, le problème que nous avons étudié est le suivant.

Nom : Problème Oui!Greens
Instance : un graphe biparti $G = (A \cup P, E)$, des réels positifs δ_a pour tout $a \in A$ et des réels positifs α_p pour tout $p \in P$ et un réel positif ζ
Question : existe-t-il un sous-graphe $H = (A \cup P, E^*)$, $E^* \subseteq E$, tel que $|N_H(a)| \leq \delta_a$ et $\min_{p \in P} \alpha_p \frac{|N_H(p)|}{|N_G(p)|} \geq \zeta$?

Nous avons également considéré le problème de maximisation associé qui consiste à déterminer un sous-graphe $H = (A \cup P, E^*)$, $E^* \subseteq E$, tel que $|N_H(a)| \leq \delta_a$ et tel que $\min_{p \in P} \alpha_p \frac{|N_H(p)|}{|N_G(p)|}$ est maximum. Intuitivement, nous cherchons à maximiser la plus faible utilité parmi tous les produits tout en respectant la contrainte du nombre de publicités maximum pour que chaque acteur. Les coefficients α_p modélisent plusieurs propriétés des produits (e.g. rareté, quantité, durée de vie). L'idée est que plus un produit $p \in P$ est rare, plus la quantité est élevée et/ou plus la durée de vie est faible, alors plus α_p est petit.

Théorème 21 *La version décision du* problème Oui!Greens *est* NP*-complet.*

Nous avons développé différents algorithmes (exacts et heuristiques) pour ce problème. Un des algorithmes est en cours d'intégration dans l'application mobile Pepino (de la startup Oui!Greens) car celui-ci a surclassé les autres en termes de qualité de solutions tout en ayant une complexité en temps adaptée à une exécution quotidienne (les algorithmes exacts sont trop longs). En accord avec la startup, je ne donne pas plus de détails dans ce document.

Chapitre 5

Autres contributions

Contents

Dans ce chapitre, je développe quelques autres contributions (de manière non exhaustive).

5.1 Représentation compacte de complexes simpliciaux

Dans cette section, je décris le travail que j'ai réalisé avec Jean-Daniel Boissonnat [BM16]. Il s'agissait de représenter les complexes simpliciaux avec des arbres de manière compacte mais tout en garantissant l'existence d'algorithmes efficaces pour la recherche d'un simplexe donné ou le changement de la structure (ajout et/ou suppression de simplexes).

5.1.1 Contexte, motivations et état de l'art

Les complexes simpliciaux sont très utilisés en topologie combinatoire ou computationnelle. Un grand nombre d'applications utilise ces objets mathématiques (e.g. en analyse de données topologiques). Un des problèmes majeurs est que la taille des complexes est souvent extrêmement grande et augmente significativement avec la dimensions des structures. En conséquence, l'utilisation des complexes simpliciaux est limitée en pratique. Une des questions centrales est donc de représenter ces objets par des structures compactes. Une des possibilités naturelles et efficaces, est de représenter un complexe simplicial par un arbre enraciné sommet-étiqueté. Intuitivement, chaque simplex maximal (au sens de l'inclusion) est représenté par un chemin entre la racine et une feuille de l'arbre.

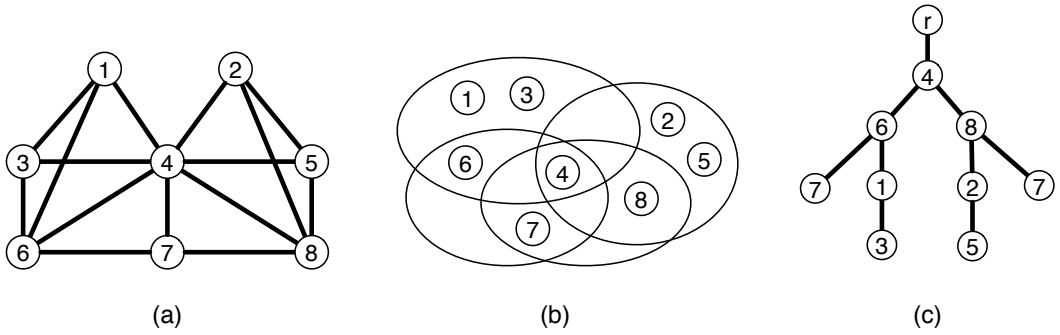

Figure 5.1: **(a)** Complexe simplicial $\mathcal{K}$ composé de huit sommets $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$. Un sommet avec $i \in [\![1, 8]\!]$ représente v_i. L'ensemble des simplexes maximaux est composé de deux tétraèdres induits par $\{v_1, v_3, v_4, v_6\}$ et $\{v_2, v_4, v_5, v_8\}$, et de deux triangles induits par $\{v_4, v_6, v_7\}$ et $\{v_4, v_7, v_8\}$. (b) Hypergraphe $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, avec $\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$ et $\mathcal{E} = \{\{v_1, v_3, v_4, v_6\}, \{v_2, v_4, v_5, v_8\}, \{v_4, v_6, v_7\}, \{v_4, v_7, v_8\}\}$, qui représente $\mathcal{K}$. Un sommet avec $i \in [\![1, 8]\!]$ représente v_i. **(c)** Arbre T représentant $\mathcal{H}$. Un sommet u avec $i \in [\![1, 8]\!]$ est tel que $L_1(u) = v_i$. $L_1(r)$ étant choisi de manière arbitraire, nous avons choisi d'indiquer "r" dans le sommet racine de l'arbre.

Nous considérons le problème de représenter tous les simplexes maximaux d'un complexe simplicial donné $\mathcal{K}$ par un arbre sommet-étiqueté enraciné. Pour cela, $\mathcal{K}$ est modélisé par un hypergraphe $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ avec $\mathcal{V}$ l'ensemble des sommets de $\mathcal{K}$ et l'ensemble d'hyperarêtes $\mathcal{E}$ est l'ensemble des simplexes maximaux de $\mathcal{K}$. Notons que $e \not\subseteq e'$ pour tous $e, e' \in \mathcal{E}$, $e \neq e'$. Dans la suite, nous utilisons la notion d'hypergraphe pour décrire le complexe.

Définissons quelques notations. Soit $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ un hypergraphe. Soit $v \in \mathcal{V}$ un sommet de $\mathcal{H}$. L'ensemble $N_{\mathcal{H}}(v)$ représente l'ensemble des voisins de v dans $\mathcal{H}$, c'est-à-dire $N_{\mathcal{H}}(v) = \{v' \in V \mid v' \neq v, \exists e \in \mathcal{E}, v' \in e, v \in e\}$. Le voisinage fermé de v est noté $N_{\mathcal{H}}[v] = N_{\mathcal{H}}(v) \cup \{v\}$. Soit $|e|$ la taille de l'hyperarête $e \in \mathcal{E}$, c'est-à-dire le nombre de sommets qui sont contenus dans e. Soit $d_{\mathcal{H}} = \max_{e \in \mathcal{E}} |e|$ la dimension de $\mathcal{H}$. Soient $\mathcal{E}_v = \{e \setminus \{v\} \mid v \in e, e \in \mathcal{E}\}$ et $\bar{\mathcal{E}}_v = \{e \mid v \notin e, e \in \mathcal{E}\}$. Soit $n = |\mathcal{V}|$. Soit $\Sigma(\mathcal{V})$ l'ensemble des ordres sur $\mathcal{V}$ et soit $\sigma = (\sigma_1, \ldots, \sigma_n) \in \Sigma(\mathcal{V})$ un ordre quelconque sur $\mathcal{V}$.

Soit $T = (V, E)$ un arbre enraciné en $r \in V$ et soit $u \in V$ un sommet de T. L'arbre $T[u]$ est un sous-arbre de T enraciné en u, c'est-à-dire l'arbre induit par l'ensemble des sommets $\{u' \in V \mid u \in V(P_{u',r})\}$, avec $P_{u',r}$ le chemin simple dans T entre u' et r.

Comme mentionné précédemment, l'objectif est de représenter les simplexes maximaux par un arbre sommet-étiqueté enraciné. L'idée est de factoriser la représentation de sommets qui apparaissent dans plusieurs simplexes maximaux (hyperarêtes), dans le but de minimiser la taille des structures. Nous pouvons alors définir la notion de *représentation en arbre* d'un complexe, formalisée dans la Définition 2.

Définition 2 (Représentation en arbre d'un complexe) *Soit $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ un hypergraphe. Une représentation en arbre de $\mathcal{H}$ est un arbre sommet-étiqueté $T = (V, E, L_1)$ enraciné en $r \in V$, $L_1 : V \to \mathcal{V}$, tel que :*

1. *pour toute hyperarête $e \in \mathcal{E}$, il existe un chemin simple P_e dans T entre la racine r et une feuille tel que $e = V(P_e) \setminus \{r\}$.*
2. *le nombre de feuilles de T est $|\mathcal{E}|$.*

Notons que $L_1(r)$ est choisi de manière arbitraire. La propriété (1) signifie que pour chaque hyperarête, il existe un chemin simple entre la racine et une feuille qui représente cette hyperarête. La propriété (2) signifie que tout chemin simple entre la racine et une feuille représente une hyperarête. Il y a alors une bijection entre l'ensemble des hyperarêtes et l'ensemble des chemins simples de l'arbre.

La Figure 5.1(a) représente un complexe simplicial $\mathcal{K}$, la Figure 5.1(b) est l'hypergraphe $\mathcal{H}$ modélisant $\mathcal{K}$ et la Figure 5.1(c) décrit une représentation en arbre T de $\mathcal{H}$. Le sommet v_4 (v_6, v_8, respectivement) apparaît dans quatre (deux, respectivement) hyperarêtes mais il y a unique sommet étiqueté v_4 (v_6, v_8, respectivement) dans T.

La représentation en arbre n'est pas satifaisante en raison de la complexité de tout algorithme pour chercher, ajouter ou supprimer un complexe maximal. En effet, des contraintes supplémentaires doivent être intégrées afin d'avoir des algorithmes efficaces. Avant de décrire les nouvelles structures que nous proposons, nous dressons un succinct état de l'art. Le diagramme de Hasse d'un complexe simplicial $\mathcal{K}$ est le graphe qui associe un sommet à chaque simplexe $\tau_1 \in \mathcal{K}$ et une arête entre deux sommets si les simplexes associés τ_1 and τ_2 satisfont $\tau_1 \subset \tau_2$ et $\dim(\tau_1) = \dim(\tau_2) - 1$. Le diagramme de Hasse ne permet pas de représenter efficacement (en termes de taille) un complexe simplicial. La notion de *simplex tree* a été introduit dans [BM14] pour représenter de manière compacte les complexes simpliciaux. Plus récemment, dans [BCST15], le problème de compresser les *simplex tree* a été étudié. La contrainte est que le *compact simplex tree* doit préserver les fonctionnalités de la structure originale (e.g. admettant des algorithmes efficaces pour la recherche d'un simplexe donné). Notre travail se focalise sur une représentation en arbre globale (équivalente à une des structures introduites dans [BCST15]) et sur une représentation en arbre locale. Ces deux représentations satisfont les contraintes décrites précédemment.

5.1.2 Nouvelles représentations en arbre

Comme mentionné précédemment, la représentation en arbre ne permet d'avoir des algorithmes *efficaces* pour, entre autres, la recherche d'un simplexe maximal donné. Pour illustrer ce propos, considérons la représentation en arbre $T = (V, E, L_1)$ enraciné en $r \in V$ décrite dans la Figure 5.2(a). Nous ne pouvons pas vérifier rapidement que l'hyperarête $e = \{a_1, a_2, a_3, a_4, b_1, c_1\}$ appartient à l'hypergraphe $\mathcal{H}$ representé par T. En effet, la racine r de T a deux voisins $u, u' \in N_T(r)$ tels que $L_1(u) = a_1 \in e$ et $L_1(u') = b_1 \in e$. Ainsi, nous ne savons pas si l'hyperarête e est

représentée par un chemin $(r, u, \ldots)$ ou par un chemin $(r, u', \ldots)$. Nous proposons donc deux nouvelles représentations en arbre plus contraintes : la première a une propriété *locale* alors que la deuxième en a une *globale*.

Définition 3 (représentation en arbre locale) *Soit $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ un hypergraphe. Une représentation en arbre locale de $\mathcal{H}$ est un arbre sommet-étiqueté $T = (V, E, L_1, L_2)$ enraciné en $r \in V$, $L_1 : V \to \mathcal{V}$, $L_2 : V \to [\![0, |\mathcal{V}|]\!]$, tel que :*

1. *si $|\mathcal{E}| = 0$, alors $T = (\{r\}, \emptyset)$;*
2. *si $|\mathcal{E}| \geq 1$, alors il existe un sommet $u \in N_T(r)$, avec $L_1(u) = v \in \mathcal{V}$, tel que :*
   (a) *pour tout $u' \in N_T(r) \setminus \{u\}$, alors $L_2(u) < L_2(u')$;*
   (b) *l'arbre $T[u]$ enraciné en $r' = u$ est une représentation en arbre locale de $(\mathcal{V} \setminus \{v\}, \mathcal{E}_v)$;*
   (c) *l'arbre $T \setminus T[u]$ enraciné en r est une représentation en arbre locale de $(\mathcal{V} \setminus \{v\}, \bar{\mathcal{E}}_v)$.*

La propriété (2.b) traduit le fait que toutes les hyperarêtes contenant v sont représentées dans $T[u]$ et la propriété (2.c) traduit le fait que toutes les autres hyper-arêtes sont représentées dans $T \setminus T[u]$. La propriété (2.a) assure l'existence d'un algorithme efficace pour la recherche d'une hyperarête (voir le lemme 2). En effet, L_2 induit un ordre (local) sur l'ensemble des étiquettes L_1 des sommets voisins de r. Plus précisément, le sommet u qui a la plus petite valeur de L_2 est tel que le sous-arbre $T[u]$ va représenter toutes les hyper-arêtes contenant $L_1(u)$, le sommet u^2 qui a la deuxième plus petite valeur de L_2 est tel que le sous-arbre $T[u^2]$ va représenter toutes les hyper-arêtes ne contenant pas $L_1(u)$ mais contenant $L_1(u^2)$, et ainsi de suite.

Nous formalisons dans la définition 4 la notion de représentation en arbre globale en utilisant un ordre sur les sommets de l'hypergraphe.

Définition 4 (représentation en arbre globale) *Soit $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ un hypergraphe. Un arbre $T = (V, E, L_1, L_2)$ enraciné en $r \in V$, $L_1 : V \to \mathcal{V}$, $L_2 : V \to [\![0, |\mathcal{V}|]\!]$, est une représentation en arbre globale de $\mathcal{H}$ si et seulement si*

1. *T est une représentation en arbre locale de $\mathcal{H}$;*
2. *pour tout $u, u' \in V$, alors $L_1(u) = L_1(u')$ si et seulement si $L_2(u) = L_2(u')$;*
3. *pour tout chemin simple $P = (r, u_1, \ldots, u_t)$ de T, alors $L_2(u_i) < L_2(u_{i+1})$ pour tout $i \in [\![1, t-1]\!]$.*

La propriété (2) stipule que si deux sommets différents de l'arbre représentent un même sommet de l'hypergraphe, alors ces deux sommets ont la même étiquette pour L_2. La propritété (3) certifie que tout chemin dans l'arbre est strictement croissant pour pour la fonction L_2. La Figure 5.2(c) décrit une représentation en arbre globale pour un hypergraphe.

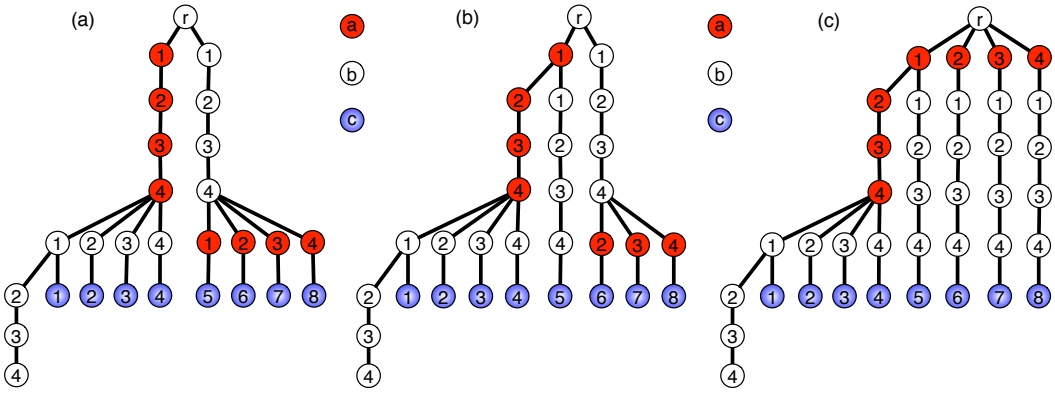

Figure 5.2: Soit $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ avec $\mathcal{V} = \{a_1, \ldots, a_4, b_1, \ldots, b_4, c_1, \ldots, c_8\}$ et $\mathcal{E} = \{\{a_1, a_2, a_3, a_4, b_1, c_1\}, \{a_1, a_2, a_3, a_4, b_2, c_2\}, \{a_1, a_2, a_3, a_4, b_3, c_3\}, \{a_1, a_2, a_3, a_4, b_4, c_4\}, \{b_1, b_2, b_3, b_4, a_1, c_5\}, \{b_1, b_2, b_3, b_4, a_2, c_6\}, \{b_1, b_2, b_3, b_4, a_3, c_7\}, \{b_1, b_2, b_3, b_4, a_4, c_8\}, \{a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4\}\}$. Un sommet rouge u avec $i \in [\![1, 4]\!]$ est tel que $L_1(u) = a_i$ et $L_2(u) = i$. Un sommet blanc u avec $i \in [\![1, 4]\!]$ est tel que $L_1(u) = b_i$ et $L_2(u) = i + 4$. Un sommet bleu u avec $i \in [\![1, 8]\!]$ est tel que $L_1(u) = c_i$ et $L_2(u) = i+8$. $L_1(r)$ étant choisi de manière arbitraire, nous avons choisi d'indiquer "r" dans le sommet racine de l'arbre. **(a)** Représentation en arbre de $\mathcal{H}$. **(b)** Représentation locale en arbre de $\mathcal{H}$. Soit u_1 tel que $L_1(u_1) = a_1$. L'arbre $T[u_1]$ enraciné en u_1 est une représentation locale de $(\mathcal{V} \setminus \{a_1\}, \mathcal{E}_{a_1})$. **(c)** Représentation globale en arbre de $\mathcal{H}$. Un ordre correspondant est $\sigma = (a_1, \ldots, a_4, b_1, \ldots, b_4, c_1, \ldots, c_8)$.

Nous définissons à présent les deux problèmes associés à ces nouvelles représentations en arbre.

Nom : Problème représentation en arbre locale
Instance : un hypergraphe $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, un entier positif ζ
Question : existe-t-il une représentation en arbre locale T_{locale} de $\mathcal{H}$ telle que $|V(T_{locale})| \leq \zeta$?

Nom : Problème représentation en arbre globale
Instance : un hypergraphe $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, un entier positif ζ
Question : existe-t-il une représentation en arbre globale $T_{globale}$ de $\mathcal{H}$ telle que $|V(T_{globale})| \leq \zeta$?

Les problèmes d'optimisation combinatoire associés consistent à déterminer une représentation en arbre (avec les différentes contraintes) qui minimise le nombre de sommets. Nous appelons T^*, T^*_{locale} et $T^*_{globale}$ des représentations en arbre optimales pour les différentes représentations étudiées. Pour illustration, nous avons $|V(T^*)| = 28$ (Figure 5.2(a)), $|V(T^*_{local})| = f(\mathcal{V}, \mathcal{E}) + 1 = 31$ (Figure 5.2(b)) et $|V(T^*_{global})| = 39$ (Figure 5.2(c)).

5.1.3 Contributions

Dans un premier temps, nous comparons les deux structures proposées avant d'analyser la complexité de calculer celles-ci. Nous prouvons dans le Lemme 2 des bornes de complexité pour le problème de recherche, suppresssion et ajout d'un simplexe maximal pour les deux structures : locale et globale.

Lemme 2 *Soient $\mathcal{H}$ un hypergraphe et T (T', respectivement) une représentation en arbre locale (globale, respectivement) de $\mathcal{H}$. Il existe un algorithme en temps $O(d_{\mathcal{H}}^2 \log_2(\Delta_T))$ ($O(d_{\mathcal{H}} \log_2(\Delta_T))$, respectivement) pour le problème de trouver, supprimer, ajouter un simplexe maximal donné.*

La complexité en temps semble meilleure pour la représentation en arbre globale. Cependant, nous prouvons que la taille d'une telle représentation est toujours plus grande la taille d'une représentation en arbre locale.

Propriété 1 *Soit $\mathcal{H}$ un hypergraphe : $|V(T^*_{global})| \geq |V(T^*_{local})| \geq |V(T^*)|$.*

Nous prouvons dans le Lemma 3 qu'il existe une classe d'hypergraphes infinie $\mathcal{C}$ telle que pour chaque hypergraphe $\mathcal{H} = (\mathcal{V}, \mathcal{E}) \in \mathcal{C}$, alors $|V(T^*_{local})| = O(|\mathcal{V}|)$ et $|V(T^*_{global})| = \Omega(|\mathcal{V}|^2)$.

Lemme 3 *Pour tout $n \geq 1$, il existe un hypergraphe $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, avec $n = |\mathcal{V}|/4$, tel que $|V(T^*_{local})| \leq 8n$ et $|V(T^*_{global})| \geq n^2$.*

Pour résumer, ces résultats ne nous permettent pas de déterminer quelle est la meilleure représentation dans l'absolu car la complexité (des algorithmes de recherche) est meilleure pour la représentation en arbre globale (d'un facteur $O(\Delta_T) = O(|V(T)|)$) mais cela peut être compensé par le fait que la taille d'un représentation en arbre peut être bien plus petite (possiblement du même facteur).

Nous étudions la complexité de calculer des représentations en arbre (locale et globale) optimales. Dans le cas des graphes (chaque hyperarête est de taille deux), nous prouvons que les problèmes sont tous NP-complets. Notons que dans ce cas, les tailles optimales pour chacune des représentations en arbre sont les mêmes.

Théorème 22 *Les versions décisions des* PROBLÈMES REPRÉSENTATIONS EN ARBRES *sont* NP*-complets même si le graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ est planaire, $|N_{\mathcal{G}}(v)| \leq 3$ et $|N_{\mathcal{G}}(v) \cap N_{\mathcal{G}}(v')| \leq 1$ for all $v, v' \in \mathcal{V}$, $v \neq v'$.*

Nous étudions ensuite le cas des hypergraphes de degré borné. Le problème est polynomial lorsque le degré maximum est au plus 2.

Théorème 23 *Les* PROBLÈMES REPRÉSENTATION EN ARBRE GLOBALE *et* LOCALE *sont dans P pour les hypergraphes de degré maximum au plus deux.*

Le résultat le plus difficile à montrer est le fait que les problèmes sont APX-complets lorsque le degré est au plus 3.

Théorème 24 *Les* PROBLÈMES REPRÉSENTATION EN ARBRE GLOBALE *et* LOCALE *sont* APX*-complets même si le degré maximum de l'hypergraphe est trois.*

Enfin, nous avons prouvé une approximation générale pour nos problèmes.

Théorème 25 *Les* PROBLÈMES REPRÉSENTATION EN ARBRE GLOBALE *et* LOCALE *admettent un algorithme polynomial garantissant une* $\frac{k}{2}$*-approximation pour la classe des hypergraphes de degré maximum* k *avec* $k \geq 3$ *une constante.*

5.2 Flot avec contrainte de délai de type on/off

Dans cette section, je décris mes contributions concernant un problème de flot à plusieurs commodités avec des contraintes de délai. Ce travail a été réalisé en collaboration avec Yann Vaxès et Pierre Bonami [BM16, BMV14].

5.2.1 Contexte, motivations et état de l'art

Le problème de flot à plusieurs commodités a été largement étudié dans la littérature. Étant donné un réseau, un ensemble de capacités sur les arêtes et un ensemble de demandes (commodités), le problème consiste à déterminer un flot satisfaisant toutes les demandes et respectant les contraintes de capacité et de conservation de flots. La version entière est un problème NP-complet [EIS76] même pour deux commodités et des capacités unitaires (rendant le problème fortement NP-complet dans ce cas). Cependant, si un flot fractionnaire est autorisé, alors le problème devient polynomial car il peut se formuler comme un programme linéaire [AMO93].

Les opérateurs des réseaux de télécommunication doivent satisfaire certaines exigences concernant la qualité de service pour leurs clients. Un des paramètres les plus importants est le délai de bout en bout d'une unité de flot entre un nœud source et un nœud destination. Cette exigence n'est pas prise en compte dans le problème de flot à plusieurs commodités. Le délai sur un lien dépend du volume de flot supporté par ce lien; classiquement cela est modélisé par une fonction convexe. Le délai de bout en bout pour une demande et un chemin associé est la somme des délais de tous les liens du chemin. Certains articles se sont focalisés sur la minimisation de la moyenne du délai de bout en bout. Ce problème consiste à minimiser une fonction convexe avec des contraintes linéaires [OMV97] et peut être résolu en utilisant la programmation semi définie [TAG03]. D'autres articles se sont focalisés sur la recherche d'un flot à plusieurs commodités satisfaisant les demandes et minimisant le plus grand délai de bout en bout [CSM07]. Comme les auteurs de [BAO06], nous pensons qu'un problème plus réaliste consiste à ajouter des contraintes strictes sur les délais de bout en bout pour toutes les connexions. En effet, il y a différentes classes de service dans les réseaux de communication et, pour chacune d'entre elles, il est crucial de respecter un certain niveau de qualité de service, c'est-à-dire respectant un seuil pour le délai de bout en bout.

Dans ce travail, nous nous focalisons sur un problème de flot à plusieurs commodités dans lequel chaque arête a une fonction de latence proportionnelle qui décrit

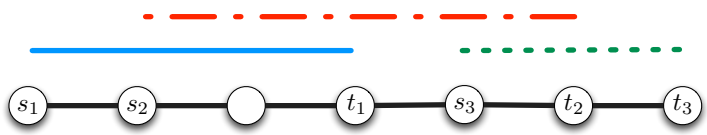

Figure 5.3: Le graphe $G = (V, E)$ est un chemin composé de sept nœuds et $\alpha_e = 1$ pour tout $e \in E$. Il y a trois paires source-destination : (s_1, t_1), (s_2, t_2) et (s_3, t_3). Les trois chemins sont représentés au dessus de G.

le délai commun pour les flots utilisant cette arête comme une fonction du flot total sur cette arête. Nous nous intéressons au problème de maximiser la somme du flot sous des contraintes de délai proportionel. Nous autorisons des flots fractionnaires pour toutes les connexions. Comme mentionné précédemment, nous avons une contrainte sur le délai de bout en bout pour chaque demande. Mais si un chemin associé à une connexion supporte un flot nul, alors la contrainte n'est pas considérée. Ces contraintes de type "on-off" rendent le problème beaucoup plus difficile que la résolution d'un programme linéaire [HBCO12].

5.2.2 Modélisation du problème et exemple

Soit $G = (V, E)$ un graphe connexe (représentant un réseau) avec un coefficient α_e pour toute arête $e \in E$. Soient $\{(s_1, t_1), \ldots, (s_m, t_m)\}$ un ensemble de m paires source-destination (connexions). Soit $\mathcal{P} = \{P_1, \ldots, P_m\}$ un ensemble de m chemins dans G. Le chemin P_i correspond à la paire (s_i, t_i) pour tout $i = 1, \ldots, m$. Sans perte de généralité, nous supposons qu'il y a un unique chemin pour chaque paire source-destination (si tel n'était pas le cas, nous considérerions certaines paires source-destination de manière multiple). Nous notons x_i le flot supporté par le chemin P_i pour tout $i = 1, \ldots, m$. Nous supposons que le délai τ_e pour traverser l'arête $e \in E$ est proportionnel au flot total $\sum_{i:e\in E(P_i)} x_i$ supporté par l'arête e, i.e. $\tau_e = \alpha_e \sum_{i:e\in E(P_i)} x_i$. Soit $\lambda > 0$. Pour tout $i = 1, \ldots, m$, nous exigeons pour le chemin P_i que, si $x_i > 0$, alors le délai de bout en bout $\sum_{e\in E(P_i)} \tau_e$ est au plus λ. Par une mise à l'échelle des coefficients α_e (division par λ), nous pouvons supposer que $\lambda = 1$. Un flot à plusieurs commodités $x = (x_1, \ldots, x_m)$ satisfait la contrainte de latence si pour tout $j = 1, \ldots, m$ tel que $x_j > 0$, la contrainte suivante pour le chemin P_j est respectée : $\sum_{e\in E(P_j)} \tau_e \leq 1$. En notant $\beta_{i,j} := \sum_{e\in E(P_i)\cap E(P_j)} \alpha_e$, cette contrainte revient à : $\sum_{i=1}^{m} \beta_{i,j} x_i \leq 1$. Nous pouvons définir le problème de flot à plusieurs commodités avec contrainte de délai comme suit.

Nom : Problème flot avec contraintes de délai
Instance : Un graphe $G = (V, E)$, un coefficient α_e pour tout $e \in E$, m connexions et un réel positif ζ
Question : existe-t-il un flot $x = (x_1, \ldots, x_m)$ tel que $\sum_{i=1}^{m} x_i \geq \zeta$, $\sum_{i=1}^{m} \beta_{i,j} x_i \leq 1$ pour tout $j \in \{1, \ldots, m\}$ tel que $x_j > 0$ et $x_i \geq 0$ pour tout $i \in \{1, \ldots, m\}$?

Dans la suite, nous nous focaliserons sur la version maximisation (sans le mentionner explicitement) qui consiste à trouver, parmi les solutions satisfaisant les contraintes, une solution de valeur $\sum_{i=1}^{m} x_i$ maximale. En d'autres termes, le problème revient à résoudre :

$$\begin{cases} \text{Max} & \sum_{i=1}^{m} x_i & & \\ & \sum_{i=1}^{m} \beta_{i,j} x_i \leq 1 & j = 1, \ldots, m & x_j > 0 \\ & x_i \geq 0 & i = 1, \ldots, m & \end{cases}$$

La difficulté du PROBLÈME FLOT AVEC CONTRAINTES DE DÉLAI réside dans le choix des chemins supportant un flot non nul. En effet, si l'ensemble des chemins $\mathcal{P}^* \subseteq \mathcal{P}$ supportant un flot non nul est donné, alors le problème devient polynomial car il se ramène à résoudre un programme linéaire.

Le graphe d'intersection des chemins H a pour ensemble de sommets $V(H) = \{h_1, \ldots, h_m\}$ correspondant à l'ensemble des chemins $\mathcal{P} = \{P_1, \ldots, P_m\}$. Pour $i, j \in \{1, \ldots, m\}$, $i \neq j$, il y a une arête $\{h_i, h_j\} \in E(H)$ entre deux sommets $h_i \in V(H)$ et $h_j \in V(H)$ si, et seulement si, il existe une arête $e \in E$ telle que $e \in E(P_i) \cap E(P_j)$, c'est-à-dire lorsque P_i et P_j partagent au moins une arête.

Considérons l'exemple suivant. Soit le chemin $G = (V, E)$ de la Figure 5.3 et les trois paires source-destination (s_1, t_1), (s_2, t_2) et (s_3, t_3). Le chemin P_i est l'unique chemin simple entre s_i et t_i dans G pour tout $i \in \{1, 2, 3\}$. Notons que H est un chemin composé de trois sommets. Posons $\alpha_e = 1$ pour tout $e \in E$. Le problème revient à maximiser $x_1 + x_2 + x_3$ sous les contraintes $x_1(3x_1 + 2x_2) \leq x_1$, $x_2(2x_1 + 4x_2 + x_3) \leq x_2$, $x_3(x_2 + 2x_3) \leq x_3$ et $x_1, x_2, x_3 \geq 0$. La solution $x^* = (1/3, 0, 1/2)$ est optimale. La connexion (s_2, t_2) supporte un flot nul, c'est-à-dire $x_2 = 0$, expliquant pourquoi $2x_1 + 4x_2 + x_3 = 7/6 > 1$.

5.2.3 Contributions

Nous analysons tout d'abord la qualité de la solution obtenue par le programme linéaire pour une variante du PROBLÈME FLOT AVEC CONTRAINTES DE DÉLAI pour laquelle toutes les contraintes de délai doivent être satisfaites même pour les connexions qui supportent un flot nul. Nous en déduisons un algorithme d'approximation polynomial. En particulier, cela donne une L-approximation avec L la taille d'un plus long chemin de $\mathcal{P}$ et une 2-approximation lorsque G est un chemin. Dans le cas des chemins, nous donnons une autre 2-approximation qui s'exécute en temps linéaire. Voir [BMV17] pour plus de détails.

Nous prouvons ensuite un schéma d'approximation polynomial lorsque le graphe d'intersection des chemins H a une largeur arborescente bornée. Pour cela, nous définissons tout d'abord une variante du problème : le PROBLÈME FLOT DISCRET AVEC CONTRAINTES DE DÉLAI. Le seul changement est que, étant donné un ensemble fini X de valeurs positives contenant zéro, le flot x_i doit appartenir à X pour tout $i = 1, \ldots, m$. La solution $x = (1/3, 0, 1/3)$ est une solution optimale pour cette variante pour l'instance de la Figure 5.3 avec $X = \{0, 1/3, 2/3, 1\}$. Nous prouvons dans le Lemme 4 un algorithme exact de programmation dynamique pour le PROBLÈME FLOT DISCRET AVEC CONTRAINTES DE DÉLAI.

Lemme 4 *Soit X un ensemble fini de valeurs positives contenant zéro. Il existe un algorithme exact de complexité $O(m|X|^{tw(H)+1})$ pour le* PROBLÈME FLOT DISCRET AVEC CONTRAINTES DE DÉLAI *avec $tw(H)$ la largeur arborescente de H.*

Soient $x_{max} = \max_{i=1,\dots,m} 1/\sum_{e\in E(P_i)} \alpha_e$ et $x_{min} = \min_{i=1,\dots,m} 1/\sum_{e\in E(P_i)} \alpha_e$. Nous prouvons dans le Théorème 26 un schéma d'approximation polynomial pour le PROBLÈME FLOT AVEC CONTRAINTES DE DÉLAI lorsque H a une largeur arborescente bornée et lorsque x_{max}/x_{min} est borné.

Théorème 26 *Soient $b, t \geq 1$ des constantes. Pour tout $\varepsilon > 0$, il existe un algorithme polynomial avec facteur d'approximation $1+\varepsilon$ pour le* PROBLÈME FLOT AVEC CONTRAINTES DE DÉLAI *lorsque $tw(H) \leq t$ et $x_{max}/x_{min} \leq b$.*

Nous déduisons du Théorème 26 qu'il existe un schéma d'approximation polynomial lorsque G est un arbre et lorsque x_{max}/x_{min}, Δ_G et χ_G sont bornés par des constantes, avec Δ_G le degré maximum de G et χ_G le nombre maximum de chemins qui partagent une même arête. Cela montre aussi l'existence d'un schéma d'approximation polynomial lorsque G est un chemin et lorsque x_{max}/x_{min} et χ_G sont bornés par des constantes. Cependant, la complexité du problème reste ouverte lorsque G est un chemin si χ_G n'est pas borné par une constante. Plus précisément, nous ne savons pas si le problème est NP-difficile ou s'il existe un algorithme polynomial avec un rapport d'approximation strictement plus petit que 2. Enfin, les résultats précédents sont valides même si les seuils de délai de bout en bout sont quelconques (si $x_i > 0$, alors $\sum_{e\in E(P_i)} \tau_e \leq \lambda_i \ \forall i = 1, \dots, m$).

5.2.4 Perspectives

Pour conclure, nous mentionnons quelques pistes de recherche et problèmes ouverts. Quelle est la complexité du PROBLÈME FLOT AVEC CONTRAINTES DE DÉLAI quand G est un chemin ? Un premier problème ouvert intéressant est de déterminer la complexité du problème lorsque G est un chemin tel que toutes les connexions partagent une arête $e \in E$ (H est un graphe complet). Existe-t-il un algorithme polynomial garantissant un facteur d'approximation strictement plus petit que 2 quand le graphe est un chemin avec χ_G non borné ? Existe-t-il un algorithme exact polynomial lorsque $tw(H)$ est bornée ? Existe-t-il une classe d'instances avec $tw(H)$ non bornée qui admettent un schéma d'approximation polynomial ? Est-ce que le PROBLÈME FLOT AVEC CONTRAINTES DE DÉLAI est dans APX ?

Un axe intéressant est le développement de règles de réduction polynomiales à partir, par exemple, de solutions obtenues pour le problème où toutes les contraintes de délai sont satisfaites (même pour une connexion avec un flot nul). De telles solutions sont obtenues en temps polynomial par la résolution d'un programme linéaire. Considérons un chemin $P_i \in \mathcal{P}$. Notons $LP(\mathcal{P})$ la valeur optimale et notons $LP(\mathcal{P} \setminus \{P_i\})$ la valeur optimale de l'instance sans le chemin P_i. Pour quelles classes d'instances pouvons-nous supprimer P_i si $LP(\mathcal{P} \setminus \{P_i\}) > LP(\mathcal{P})$? Pour quelles classes d'instances $LP(\mathcal{P} \setminus \{P_i\}) < LP(\mathcal{P})$ implique l'existence d'une solution optimale telle que $x_i > 0$?

Chapitre 6

Conclusion générale et perspectives

Contents

Pour conclure, je suis très investi dans la Recherche pour la Société (médiation scientifique en particulier). Je coordonne le projet d'envergure Terra Numerica (`https://terra-numerica.org`) qui vise la création d'une Cité du Numérique dans le sud-est. Concernant mes activités de Recherche (pour faire avancer le front des connaissances), je mentionne dans la suite les axes de recherche que je souhaite continuer à développer, notamment ceux liant optimisation combinatoire et biologie structurale computationnelle.

6.1 Algorithmique pour les réseaux et la biologie structurale

Dans ce document, j'ai décrit les techniques algorithmiques que j'ai développées pour résoudre des problèmes d'optimisation combinatoire dans différents domaines comme les réseaux de communication et la biologie structurale notamment. Il est important de souligner la portée et la généricité de ces techniques algorithmiques. Pour illustration, j'ai contribué aux recherches de différentes équipes-projets Inria depuis ma thèse : Mascotte/Coati, Maestro/Neo, ABS, Geometrica/DataShape, Athena, Biovision.

Dans ce document, je n'ai pas mentionné tous les travaux en cours. Dans l'un deux, nous nous intéressons à un probl'eme dans les réseaux d'interaction protéine-protéine (*PPIN*). Le but est de mieux caractériser les protéines impliquées dans la mort cellulaire. Ce travail en cours est réalisé avec Frédéric Cazals, Alain Jean-Marie, Jérémie Roux et Guilherme Santa Cruz. Je peux résumer mes recherches en deux grandes parties : conception d'algorithmes dans les réseaux de communication et algorithmique pour la biologie structurale. Le problème mentionné précédemment peut les lier car il s'agit de concevoir des algorithmes dans les réseaux de communication entre protéines.

6.2 Perspectives : Biologie structurale computationnelle appliquée à l'optimisation combinatoire

Les protéines sont capables de résoudre, très rapidement, des problèmes très compliqués ! Pourquoi ne pourraient-elles pas résoudre des problèmes d'optimisation combinatoire compliqués ?

Un paysage énergétique potentiel (PEL) d'un système moléculaire est une fonction qui représente l'énergie potentielle pour chaque conformation du système. Les variables sont les $3n$ coordonnées des n atomes composant le système moléculaire. L'exploration de tels paysages est un problème central en biologie structurale computationnelle car identifier les minima locaux de plus basses énergies permettra de connaître les conformations de grand intérêt. Ces dernières peuvent en effet être très utiles pour les problèmes de repliement et d'amarrage moléculaires. Des méthodes état-de-l'art d'exploration de PEL ont été développées au sein d'ABS [RDRC16]. Voir l'application d'exploration de paysages énergétiques dans la Structural Bioinformatics Library (SBL, `sbl.inria.fr`) [CD17].

Les graphes (et les hypergraphes) peuvent modéliser un grand nombre de structures (de nature diverse) : routes entre des villes, liens dans un réseau de télécommunication, contacts entre sous-unités d'une protéine, liens d'amitié entre utilisateurs dans les réseaux sociaux... Des problèmes importants dans ces réseaux peuvent se modéliser par des problèmes d'optimisation combinatoire dans les graphes correspondants. Malheureusement, beaucoup de ces problèmes sont connus pour être NP-complets. Intuitivement, il n'existe pas d'algorithme polynomial pour résoudre ces problèmes, à moins que P = NP. Des algorithmes d'approximation, des algorithmes pour certaines classes d'instances... ont été développés. Voir par exemple [Vaz01] pour des algorithmes d'approximation.

Étant donné un problème combinatoire Π, l'idée est de modéliser (traduire) les instances de Π en termes de paysages énergétiques (appelés paysages énergétiques combinatoires) pour lesquels les solutions (quasi-)optimales pour Π sont des minima locaux (de basses énergies). Ensuite, les algorithmes d'exploration permettront d'identifier ces minima importants et donc de trouver des solutions (optimales, quasi-optimales) pour le problème Π.

L'objectif de ce projet est double. Tout d'abord, il s'agit de traduire un problème d'optimisation combinatoire en un problème d'exploration d'un paysage énergétique combinatoire auxiliaire tel que les minima locaux de plus basses énergies sont des solutions (quasi-)optimales. Nous avons entamé ces recherches par des problèmes liés aux décomposition de graphes (e.g. largeur arborescente, largeur de chemin), le problème de l'ensemble indépendant maximum et le problème de couplage avec distances internes. Ensuite, il s'agit d'intégrer dans la SBL nos méthodes et algorithmes de représentation en paysages énergétiques combinatoires et de les tester avec les algorithmes d'exploration état-de-l'art présents dans la SBL.

Bibliographie

[AAC+13] D. Agarwal, J.-C. Silva Araujo, C. Caillouet, F. Cazals, D. Coudert, and S. Pérennes. *Connectivity Inference in Mass Spectrometry Based Structure Determination*, pages 289–300. 2013.

[AAR10] D. Angluin, J. Aspnes, and L. Reyzin. Inferring social networks from outbreaks. In *Proceedings of the International Conference on Algorithmic Learning Theory (ALT '10)*, pages 104–118, 2010.

[ABG+11] J. Araujo, J.-C. Bermond, F. Giroire, F. Havet, D. Mazauric, and R. Modrzejewski. Weighted Improper Colouring. In *In Proceedings of the 22th International Workshop on Combinatorial Algorithms (IWOCA 2011)*, volume 7056 of *Lecture Notes in Computer Science (LNCS)*, Victoria, Canada, July 2011. Springer.

[ABG+12] J. Araujo, J.-C. Bermond, F. Giroire, F. Havet, D. Mazauric, and R. Modrzejewski. Weighted improper colouring. *Journal of Discrete Algorithms*, 16:53–66, 2012.

[ACCC15] D. Agarwal, C. Caillouet, D. Coudert, and F. Cazals. Unveiling Contacts within Macro-molecular assemblies by solving Minimum Weight Connectivity Inference Problems. *Molecular and Cellular Proteomics*, 14:2274–2284, April 2015.

[AMDY11] R. Andonov, N. Malod-Dognin, and N. Yanev. Maximum Contact Map Overlap Revisited. *J. of Computational Biology*, 18(1):1–15, January 2011.

[AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[AV07] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *ACM-SODA*, page 1035. Society for Industrial and Applied Mathematics, 2007.

[BAO06] W. Ben-Ameur and A. Ouorou. Mathematical models of the delay constrained routing problem. *Algorithmic Operations Research*, 1(2), 2006.

[BCDM18] J.-C. Bermond, A. Chaintreau, G. Ducoffe, and D. Mazauric. How long does it take for all users in a social network to choose their communities? In *9th International Conference on Fun with Algorithms (FUN 2018)*, La Maddalena, Italy, 2018.

[BCM+12] S. Belhareth, D. Coudert, D. Mazauric, N. Nisse, and I. Tahiri. Reconfiguration with physical constraints in WDM networks. In *In Proceedings of ICC Workshop on New Trends in Optical Networks Survivability (ICC 2012). IEEE.*, pages 6346–6350, Ottawa, Canada, June 2012.

[BCPS10] U. Brandes, S. Cornelsen, B. Pampel, and A. Sallaberry. Blocks of hypergraphs - applied to hypergraphs and outerplanarity. In *Proceedings of the 21st International Workshop on Combinatorial Algorithms (IWOCA '10)*, pages 201–211, 2010.

[BCST15] J.-D. Boissonnat, Karthik C. S., and S. Tavenas. Building Efficient and Compact Data Structures for Simplicial Complexes. In *International Symposium on Computational Geometry 2015*, Eindhoven, Netherlands, June 2015.

[BDE+17] A. Brandt, J. Diemunsch, C. Erbes, J. Legrand, and C. Moffatt. A robber locating strategy for trees. *Discrete Applied Math.*, 232:99–106, 2017.

[BF03] S. Bhadra and A. Ferreira. Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. In *ADHOC-NOW'03*, pages 259–270, 2003.

[BGG+a] B. Bosek, P. Gordinowicz, J. Grytczuk, N. Nisse, J. Sokol, and M. Sleszynska-Nowak. Centroidal localization game. arXiv:1711.08836 (2017).

[BGG+b] B. Bosek, P. Gordinowicz, J. Grytczuk, N. Nisse, J. Sokol, and M. Sleszynska-Nowak. Localization game on geometric and planar graphs. CoRR abs/1709.05904 (2017), à paraître dans *Discrete Applied Maths.*

[BJMMY16] J.-C. Bermond, A. Jean-Marie, D. Mazauric, and J. Yu. Well Balanced Designs for Data Placement. *Journal of Combinatorial Designs*, 24(2):55–76, February 2016.

[BM14] J.-D. Boissonnat and C. Maria. The simplex tree: An efficient data structure for general simplicial complexes. *Algorithmica*, 70(3):406–427, 2014.

[BM16] J.-D. Boissonnat and D. Mazauric. On the complexity of the representation of simplicial complexes by trees. *Theoretical Computer Science*, 617:17, February 2016.

[BMMI+18a] J. Bensmail, D. Mazauric, F. Mc Inerney, N. Nisse, and S. Pérennes. Localiser une cible dans un graphe. In *ALGOTEL 2018 - 20èmes*

Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Roscoff, France, May 2018.

[BMMI+18b] J. Bensmail, D. Mazauric, F. Mc Inerney, N. Nisse, and S. Pérennes. Sequential Metric Dimension. In *16th Workshop on Approximation and Online Algorithms (WAOA 2018)*, Helsinki, Finland, August 2018.

[BMMN08] J.-C. Bermond, D. Mazauric, V. Misra, and P. Nain. Distributed Call Scheduling in Wireless Network. Submitted to WINE, 2008.

[BMMN10] J.-C. Bermond, D. Mazauric, V. Misra, and P. Nain. A Distributed Scheduling Algorithm for Wireless Networks with Constant Overhead and Arbitrary Binary Interference. In *In Proceedings of ACM SIGMETRICS 2010*, New York, USA, June 2010. Columbia University.

[BMN09] J.-C. Bermond, D. Mazauric, and P. Nain. Algorithmes distribués d'ordonnancement dans les réseaux sans-fil. In Alexandre Caminada, editor, *10es Journées Doctorales en Informatique et Réseaux (JDIR 2009)*, Belfort, France, February 2009.

[BMV14] P. Bonami, D. Mazauric, and Y. Vaxès. Flot maximum avec contrainte de délai proportionnel. In *In Proceedings of the 16es Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel 2014)*, pages 1–4, Le Bois-Plage-en-Ré, France, June 2014.

[BMV17] P. Bonami, D. Mazauric, and Y. Vaxès. Maximum flow under proportional delay constraint. *Theoretical Computer Science*, 689:58–66, 2017.

[BN11] A. Bonato and R. Nowakovski. *The game of Cops and Robber on Graphs*. American Math. Soc., 2011.

[CCD+12] J. Carraher, I. Choi, M. Delcourt, L. H. Erickson, and D. B. West. Locating a robber on a graph via distance queries. *Theor. Computer Science*, 463:54–61, 2012.

[CCM+10] N. Cohen, D. Coudert, D. Mazauric, N. Nepomuceno, and N. Nisse. Tradeoffs in process strategy games with application in the WDM reconfiguration problem. In Paola Boldi and Luisa Gargano, editors, *Fifth International conference on Fun with Algorithms (FUN 2010)*, volume 6099 of *Lecture Notes in Computer Science (LNCS)*, pages 121–132, Ischia Island, Italy, June 2010. Springer.

[CCM+11] N. Cohen, D. Coudert, D. Mazauric, N. Nepomuceno, and N. Nisse. Tradeoffs in process strategy games with application in the WDM reconfiguration problem. *Journal of Theoretical Computer Science (TCS)*, 412(35):4675–4687, 2011.

[CD17] F. Cazals and T. Dreyfus. The Structural Bioinformatics Library: modeling in biomolecular science and beyond. *Bioinformatics*, 7(33):1–8, 2017.

[CDM13] A. Chaintreau, G. Ducoffe, and D. Mazauric. De la difficulté de garder ses amis (quand on a des ennemis) ! In Nicolas Nisse, Franck Rousseau, and Yann Busnel, editors, *In Proceedings of the 15es Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel 2013)*, pages 1–4, Pornic, France, May 2013. Prix du meilleur article étudiant.

[CDM+15] F. Cazals, T. Dreyfus, D. Mazauric, A. Roth, and C.H. Robert. Conformational ensembles and sampled energy landscapes: Analysis and comparison. *J. Comp. Chem.*, 36(16):1213–1231, 2015.

[CDS04] V. Conitzer, J. Derryberry, and Tuomas Sandholm. Combinatorial auctions with structured item graphs. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI '04)*, pages 212–218, 2004.

[CFQS12] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. *Int. Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012.

[CGM+10] S. Caron, F. Giroire, D. Mazauric, J. Monteiro, and S. Pérennes. P2P Storage Systems: Data Life Time for Different Placement Policies. In Maria Gradinariu Potop-Butucaru and Hervé Rivano, editors, *In Proceedings of the 12es Rencontres Francophones sur les Aspects Algorithmiques de Télécommunications (AlgoTel 2010)*, pages 17–20, Belle Dune - Côte d'Opale, France, June 2010.

[CGM+13] S. Caron, F. Giroire, D. Mazauric, J. Monteiro, and S. Pérennes. P2P Storage Systems: Study of Different Placement Policies. *Peer-to-Peer Networking and Applications*, 7(4), March 2013.

[Che95] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE PAMI*, 17(8):790–799, 1995.

[CHM08a] D. Coudert, F. Huc, and D. Mazauric. A distributed algorithm for computing and updating the process number of a forest. In Gabi Taubenfeld, editor, *In Proceedings of the 22nd International Symposium on Distributed Computing (DISC 2008)*, volume 5218, pages 500–501, Arcachon, France, September 2008. Springer.

[CHM08b] D. Coudert, F. Huc, and D. Mazauric. Computing and updating the process number in trees. In Theodore P. Baker, Alain Bui, and Sébastien Tixeuil, editors, *In Proceedings of the 12th International*

Conference On Principles Of DIstributed Systems (OPODIS 2008), volume 5401 of *Lecture Notes in Computer Science (LNCS)*, Luxor, Egypt, December 2008. Springer.

[CHM+09] D. Coudert, F. Huc, D. Mazauric, N. Nisse, and J.-S. Sereni. Reconfiguration of the Routing in WDM Networks with Two Classes of Services. In *In Proceedings of the 13th IEEE Conference on Optical Network Design and Modeling (ONDM 2009)*, Braunschweig, Germany, February 2009.

[CHM12] D. Coudert, F. Huc, and D. Mazauric. A Distributed Algorithm for Computing the Node Search Number in Trees. *Algorithmica*, 63(1):158–190, 2012.

[CHM+17] N. Cohen, F. Havet, D. Mazauric, I. Sau, and R. Watrigant. Complexity Dichotomies for the Minimum F-Overlay Problem. In *IWOCA: International Workshop on Combinatorial Algorithms*, page 12, Newcastle, Australia, July 2017.

[CHM+18] N. Cohen, F. Havet, D. Mazauric, I. Sau, and R. Watrigant. Complexity dichotomies for the Minimum F-Overlay problem. *Journal of Discrete Algorithms*, 52-53:133–142, September 2018.

[CM02] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[CM16] F. Cazals and D. Mazauric. Optimal transportation problems with connectivity constraints. Research Report RR-8991, Inria Sophia Antipolis ; Université Côte d'Azur, December 2016.

[CMCW16] J. Carr, D. Mazauric, F. Cazals, and D. J. Wales. Energy landscapes and persistent minima. *The Journal of Chemical Physics*, 144(5):4, 2016.

[CMN09] D. Coudert, D. Mazauric, and N. Nisse. On Rerouting Connection Requests in Networks with Shared Bandwidth. In A. Koster and V. Lozin, editors, *In Proceedings of the DIMAP Workshop on Algorithmic Graph Theory (AGT 2009)*, volume 32 of *Electronic Notes in Discrete Mathematics*, Warwick, United Kingdom, March 2009.

[CMN14] D. Coudert, D. Mazauric, and N. Nisse. Experimental Evaluation of a Branch and Bound Algorithm for computing Pathwidth. In *In Proceedings of the 13th International Symposium on Experimental Algorithms (SEA 2014)*, volume 8504 of *Lecture Notes in Computer Science (LNCS)*, pages 46–58, Copenhagen, Denmark, June 2014. Springer.

[CMN16] D. Coudert, D. Mazauric, and N. Nisse. Experimental Evaluation of a Branch and Bound Algorithm for Computing Pathwidth and Directed Pathwidth. *ACM Journal of Experimental Algorithmics*, 21(1):23, 2016.

[CMTV07] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg. Constructing scalable overlays for pub-sub with many topics. In *Proceedings of the 26th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC '07)*, pages 109–118, 2007.

[CMTW17] F. Cazals, D. Mazauric, R. Tetley, and R. Watrigant. Comparing two clusterings using matchings between clusters of clusters. 2017. Under revision.

[CMTW18] F. Cazals, D. Mazauric, R. Tetley, and R. Watrigant. Comparaison de deux clusterings par couplage entre clusters de clusters. In *ALGOTEL 2018 - 20èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, Roscoff, France, May 2018.

[CSM07] J. R. Correa, A. S. Schulz, and N. E. Stier Moses. Fast, fair, and efficient flows in networks. *Operations Research*, 55(2):215–225, 2007.

[CT19] F. Cazals and R. Tetley. Multiscale analysis of structurally conserved motifs. 2019. Submitted.

[DH73] R.O. Duda and P.E. Hart. *Pattern classification and scene analysis*. Wiley, 1973.

[DK95] D.-Z. Du and D. F. Kelley. On complexity of subset interconnection designs. *Journal of Global Optimization*, 6(2):193–205, 1995.

[DM88] D.-Z. Du and Z. Miller. Matroids and subset interconnection design. *SIAM Journal on Discrete Mathematics*, 1(4):416–424, 1988.

[DMC12] G. Ducoffe, D. Mazauric, and A. Chaintreau. Convergence of coloring games with collusions. Technical report, Columbia University, August 2012. available at `www.cs.columbia.edu/∼augustin/pub/DMC.TR13.pdf`.

[EIS76] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. Comput.*, 5(4):691–703, 1976.

[FGJM+11] F. Fomin, F. Giroire, A. Jean-Marie, D. Mazauric, and N. Nisse. To Satisfy Impatient Web surfers is Hard. Research Report RR-7740, LIRMM ; INRIA, September 2011.

[FGJM+12a] F. V. Fomin, F. Giroire, A. Jean-Marie, D. Mazauric, and N. Nisse. Satisfaire un internaute impatient est difficile. In Fabien Mathieu and Nicolas Hanusse, editors, *In Proceedings of the 14es Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel 2012)*, pages 79–82, La Grande Motte, France, 2012. Prix du meilleur article.

[FGJM+12b] F. V. Fomin, F. Giroire, A. Jean-Marie, D. Mazauric, and N. Nisse. To Satisfy Impatient Web surfers is Hard. In E. Kranakis, D. Krizanc, and F. Luccio, editors, *In Proceedings of the Sixth International Conference on Fun with Algorithms (FUN 2012)*, volume 7288 of *Lecture Notes in Computer Science (LNCS)*, pages 166–176, San Servolo Island, Venice, Italy, June 2012. Springer.

[FHWE08] H. Fan, C. Hundt, Y.-L. Wu, and J. Ernst. Algorithms and implementation for interconnection graph problem. In *Proceedings of the 2nd Annual International Conference on Combinatorial Optimization and Applications (COCOA '08)*, pages 201–210, 2008.

[FKS14] F. Foucaud, R. Klasing, and P. J. Slater. Centroidal bases in graphs. *Networks*, 64:96–108, 2014.

[FT87] M. Fredman and R. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34(3):596–615, July 1987.

[FT08] F.V. Fomin and D. M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theor. Comput. Sci.*, 399(3):236–245, 2008.

[FW08] H. Fan and Y.-L. Wu. Interconnection graph problem. In *Proceedings of the 2008 International Conference on Foundations of Computer Science (FCS '08)*, pages 51–55, 2008.

[GCD02] R. Grigoras, V. Charvillat, and M. Douze. Optimizing hypervideo navigation using a Markov decision process approach. In *ACM Multimedia*, pages 39–48, 2002.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[GK86] C. Greene and D. J. Kleitman. Longest chains in the lattice of integer partitions ordered by majorization. *Eur. J. Comb.*, 7(1):1–10, January 1986.

[GK93] E. Goles and M. A. Kiwi. Games on line graphs and sand piles. *Theoretical Computer Science*, 115(2):321 – 349, 1993.

[GLM+15] F. Giroire, I. Lamprou, D. Mazauric, N. Nisse, S. Pérennes, and R. Soares. Connected surveillance game. *Theoretical Computer Science*,

584:131 – 143, 2015. Special Issue on Structural Information and Communication Complexity.

[GM14] E. Godard and D. Mazauric. Computing the dynamic diameter of non-deterministic dynamic networks is hard. In *ALGOSENSORS 14*, Lecture Notes in Computer Science, 2014.

[GMB10] P. Giabbanelli, D. Mazauric, and J.-C. Bermond. On the average path length of deterministic and stochastics recursive networks. In *In Proceedings of Second Workshop on Complex Networks (CompleNet 2010). Communications in Computer and Information Science (CCIS)*, volume 116, Rio de Janeiro, Brazil, October 2010. Springer-Verlag.

[GMM11] F. Giroire, D. Mazauric, and J. Moulierac. Routage efficace en énergie. In Bertrand Ducourthial and Pascal Felber, editors, *13es Rencontres Francophones sur les Aspects Algorithmiques de Télécommunications (AlgoTel 2011)*, Cap Estérel, France, 2011.

[GMM12] F. Giroire, D. Mazauric, and J. Moulierac. Energy Efficient Routing by Switching-Off Network Interfaces. In Naima Kaabouch and Wen-Chen Hu, editors, *Energy-Aware Systems and Networking for Sustainable Initiatives*, pages 207–236. IGI Global, June 2012.

[GMMO10] F. Giroire, D. Mazauric, J. Moulierac, and B. Onfroy. Minimizing Routing Energy Consumption: from Theoretical to Practical Results. In *In Proceedings of IEEE/ACM International Conference on Green Computing and Communications (GreenCom 2010)*, pages 252–259, Hangzhou, China, December 2010.

[GMP10] P. Giabbanelli, D. Mazauric, and S. Pérennes. Computing the average path length and a label-based routing in a small-world graph. In Maria Gradinariu Potop-Butucaru and Hervé Rivano, editors, *In Proceedings of the 12es Rencontres Francophones sur les Aspects Algorithmiques de Télécommunications (AlgoTel 2010)*, pages 47–50, Belle Dune - Côte d'Opale, France, June 2010.

[HBCO12] H. Hijazi, P. Bonami, G. Cornuéjols, and A. Ouorou. Mixed-integer nonlinear programs featuring "on/off" constraints. *Comp. Opt. and Appl.*, 52(2):537–558, 2012.

[HHI+12] J. Hosoda, J. Hromkovic, T. Izumi, H. Ono, M. Steinová, and K. Wada. On the approximability and hardness of minimum topic connected overlay and its special instances. *Theoretical Computer Science*, 429:144 – 154, 2012.

[HM76] F. Harary and R. A. Melter. On the metric dimension of a graph. *Ars Combin.*, 2:191–195, 1976.

[HMNW20] Frédéric Havet, Dorian Mazauric, Viet-Ha Nguyen, and Rémi Watrigant. Overlaying a hypergraph with a graph with bounded maximum degree. In *CALDAM 2020 - 6th Annual International Conference on Algorithms and Discrete Applied Mathematics*, Hyderabad, India, February 2020.

[Inc99] Zona Research Inc. The economic impacts of unacceptable website download speeds. White paper, Redwood City, CA, April 1999. www.webperf.net/info/wp_downloadspeed.pdf.

[JG97] D. Joseph and D. Grunwald. Prefetching using Markov predictors. In *ISCA*, pages 252–263, 1997.

[JP87] D. S. Johnson and H. O. Pollak. Hypergraph planarity and the complexity of drawing venn diagrams. *Journal of Graph Theory*, 11(3):309–325, 1987.

[KL10] J. M. Kleinberg and K. Ligett. Information-Sharing and Privacy in Social Networks. *paper in progress (available at arxiv.org/abs/1003.0469)*, 2010.

[KMN14] B. Klemz, T. Mchedlidze, and M. Nöllenburg. Minimum tree supports for hypergraphs and low-concurrency euler diagrams. In *Proceedings of the 14th Scandinavian Symposium and Workshops (SWAT '14)*, pages 265–276, 2014.

[KS03] E. Korach and M. Stern. The clustering matroid and the optimal clustering tree. *Mathematical Programming*, 98(1):385–414, 2003.

[LGD+17] N. Lascano, G. Gallardo, R. Deriche, D. Mazauric, and D. Wassermann. Extracting the Groupwise Core Structural Connectivity Network: Bridging Statistical and Graph-Theoretical Approaches. In *Information Processing in Medical Imaging*, Boone, United States, 2017.

[Maz16] D. Mazauric. Graphes et Algorithmes - Diffusion de l'information scientifique, 2016. Slides de médiation scientifique pour comprendre les graphes et les algorithmes de manière ludique.

[Mei07] M. Meilă. Comparing clusterings–an information based distance. *Journal of multivariate analysis*, 98(5):873–895, 2007.

[MJM10] O. Morad and A. Jean-Marie. Optimisation en temps-réel du téléchargement de vidéos. In *Proc. of 11th Congress of the French Operations Research Soc.*, 2010.

[MSZ13] D. Mazauric, S. Soltan, and G. Zussman. Computational analysis of cascading failures in power networks. In *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling*

of Computer Systems, SIGMETRICS '13, pages 337–338, New York, NY, USA, 2013. ACM.

[OMV97] A. Ouorou, P. Mahey, and J.-Ph. Vial. A survey of algorithms for convex multicommodity flow problems, 1997.

[OR11] M. Onus and A. W. Richa. Minimum maximum-degree publish-subscribe overlay network design. *IEEE/ACM Trans. Netw.*, 19(5):1331–1343, October 2011.

[phd11] http://www.phdcomics.com/comics/archive.php?comicid=1456, 2011.

[RDRC16] A. Roth, T. Dreyfus, C.H. Robert, and F. Cazals. Hybridizing rapidly growing random trees and basin hopping yields an improved exploration of energy landscapes. *J. Comp. Chem.*, 37(8):739–752, 2016.

[RGMV12] D. Ritchie, A. Ghoorah, L. Mavridis, and V. Venkatraman. Fast protein structure alignment using Gaussian overlap scoring of backbone peptide fragment similarity. *Bioinformatics*, 28(24):3274–3281, 2012.

[Sea12] S. Seager. Locating a robber on a graph. *Discrete Math.*, 312:3265–3269, 2012.

[Sea13] S. Seager. A sequential locating game on graphs. *Ars Combin.*, 110:45–54, 2013.

[Sea14] S. Seager. Locating a backtracking robber on a tree. *Theor. Computer Science*, 539:28–37, 2014.

[Sla75] P. J. Slater. Leaves of trees. In *Proc. 6th Southeastern Conf. Combin., Graph Theory, Computing in Congressus Numer.*, volume 14, pages 549–559, 1975.

[SMZ14] S. Soltan, D. Mazauric, and G. Zussman. Cascading failures in power grids: Analysis and algorithms. In *Proceedings of the 5th International Conference on Future Energy Systems*, e-Energy '14, pages 195–206, New York, NY, USA, 2014. ACM.

[SMZ17] S. Soltan, D. Mazauric, and G. Zussman. Analysis of failures in power grids. *IEEE Transactions on Control of Network Systems*, 4(2):288–300, June 2017.

[TAG03] C. Touati, E. Altman, and J. Galtier. Semi-definite programming approach for bandwidth allocation and routing in networks. *Game Theory and Applications*, 9:169–179, 2003.

[Tet18] R. Tetley. *Mixed sequence-structure based analysis of proteins, with applications to functional annotations.* Theses, Université Côte d'Azur, November 2018.

[Vaz01] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, Heidelberg, 2001.

[WMGDD16] D. Wassermann, D. Mazauric, G. Gallardo-Diez, and R. Deriche. Extracting the Core Structural Connectivity Network: Guaranteeing Network Connectedness Through a Graph-Theoretical Approach. In *MICCAI 2016*, Athens, Greece, September 2016.

[YG05] Y. Ye and A. Godzik. Multiple flexible structure alignment using partial order graphs. *Bioinformatics*, 21(10):2362–2369, 2005.