État des lieux - Graphes dans la fédération NormaSTIC

Luc Brun (GREYC) - Géraldine Del Mondo (LITIS)

9 décembre 2022

FÉDÉRATION NORMANDE
DE RECHERCHE EN SCIENCES
ET TECHNOLOGIES DE L'INFORMATION
ET DE LA COMMUNICATION

Chapitre 1 Introduction

1.1 Objectifs et méthodologie

L'objectif premier de l'Action Transverse Graphes (ATG) est de **favoriser les échanges entre chercheurs de la fédération NormaSTIC autour de l'objet *Graphe***. Qu'est-ce qu'un *Graphe* ? Les modélisations qui utilisent les graphes sont très nombreuses dans les travaux recensés au sein de la fédération, mais force est de constater que tous les chercheurs ne mettent pas forcément les mêmes définitions dernière ce terme, et les traitements qui leurs sont appliqués sont eux aussi potentiellement très divers. L'Action Transverse Graphes (ATG) a donc notamment pour but de recenser et de faire connaître les différentes variations qui existent autour des graphes afin que chacun puisse comprendre les travaux développés sur ce thème dans la fédération.

L'enjeu est double, il s'agit de contribuer à l'émergence de nouvelles collaborations au sein de la fédération mais aussi de pouvoir **communiquer vers l'extérieur et promouvoir les compétences autour des graphes dans NormaSTIC**, afin de susciter des collaborations avec des chercheurs externes à la fédération.

Pour mener à bien ces objectifs, nous avons réalisé un premier **état des lieux des différentes variations de graphes et leurs usages dans la fédération afin de souligner complémentarités et différences dans la définition et l'usage des graphes**. Sur une durée d'un an environ nous avons rencontré les chercheurs identifiés comme "représentants" d'une variation pour qu'ils présentent informellement leur travaux au cours d'un entretien. Un objectif de ces échanges est aussi l'identification des collaborations existantes entre membres de la fédération sur ce thème et les liens avec les GdRs. A la suite de chaque entretien les chercheurs ont été invités à rédiger une fiche nous permettant par la suite de faire la synthèse des différents travaux.

La re-transcription de ce travail est prévue en 3 étapes :

1. Le présent document est la première étape, il a pour but de résumer la situation existante, d'essayer de prendre du recul sur l'ensemble des travaux afin d'identifier problématiques scientifiques et applications réalisées autour des graphes, et de les mettre en perspective.
2. La deuxième étape sera la présentation du travail de synthèse et des perspectives de recherche identifiées, dans le cadre d'une journée au mois d'octobre prochain. L'objectif est de suciter les échanges autour du travail réalisé.
3. Enfin, le site web [(1)] sera mis à jour afin de donner une cartographie des travaux menés sur les graphes visible à l'exterieur de la fédération.

L'ATG se proposera ensuite de poursuivre cette dynamique autour de journées communes et d'actions en faveur des échanges et de la diffusion des travaux autour des graphes. Par ailleurs, dans la mesure où il était difficile de viser l'exhaustivité, cet état des lieux pourra bien sur être étoffé par la suite.

(1). `http://www.normastic.fr/department/action-transverse-graphes/`

1.2 Remerciements

Nous remercions tous les chercheurs qui ont bien voulu répondre à nos sollicitations et nous accorder de leur temps pour réaliser un entretien et rédiger une fiche de synthèse : Sébastien Adam, Nicolas Bedon, Bertrand Cuissard, Paul Dorbec, Benoît Gaüzère, Pierre Heroux, Thierry Lecroq, Olivier Lézoray, Yoann Pigné, François Rioult, Brigitte Vallée.

Et nous remercions bien sur aussi tous ceux qui ont été sollicités via ces "représentants";-)

1.3 Plan du document

Ce rapport se présente en trois parties : l'existant (section 2), la synthèse (section 3) et les perspectives et opportunités (section 4). La première partie a pour but de présenter factuellement les fiches de synthèse qui nous ont été adressées suite aux entretiens. L'objectif est d'essayer de les présenter sous un format commun. La deuxième partie commence à dessiner une vision plus macroscopique et identifie, parmi ces retours, des approches et des courants caractéristiques. La dernière partie se propose de mettre en perspective les courants méthodologiques émergents de cette étude et tente d'ouvrir sur des perspectives de collaborations. Une conclusion boucle ce document et les fiches de synthèse "brutes" sont disponibles en annexes.

Chapitre 2 L'existant

2.1 Introduction

A partir des entretiens et des fiches de synthèse produites par les personnes interrogées (en italiques dans la liste des personnes impliquées) nous reprenons dans cette partie les éléments factuels qui nous ont été communiqués. Ces éléments ont été éventuellement re-travaillés pour permettre une meilleure re-transcription.

2.2 Graphes moléculaires

Mots clés Data mining, Chemoinformatique

Problématiques scientifiques Graph matching, Fouille de données sur graphe

Domaines applicatifs (exemples) Chémo-informatique, Drug design

Personnes de la fédération impliquées

LITIS	GREYC
Sébastien Adam (APP)	Sébastien Bougleux (IMAGE)
Géraldine Del Mondo (APP)	Luc Brun (IMAGE)
Benoît Gaüzère (APP)	Bruno Crémilleux (CODAG)
Pierre Heroux (APP)	*Bertrand Cuissard (CODAG)*
Laurent Heutte (APP)	Abdelkader Ouali (CODAG)
Thierry Lecroq (TIBS)	François Rioult (CODAG)
Thierry Paquet (APP)	Antoine Widlöcher (CODAG)
Lina Soualmia (TIBS)	Albrecht Zimmemrmann (CODAG)

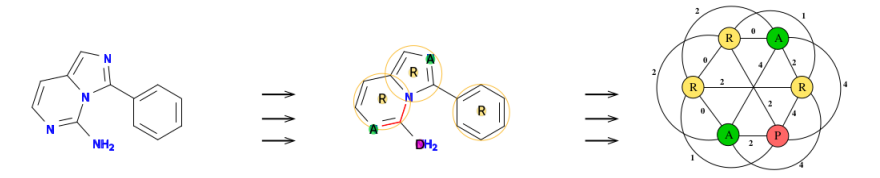
Quel(s) graphe(s) ? Trois types de graphes (moléculaires) identifiés :

- Graphe forme squelettique (un nœud – un atome / une arête – une liaison atomique)
- Graphe des fonctions pharmacophoriques (un nœud - un ensemble d'atomes / une arête – distance entre deux ensembles)
- Diagramme de Hasse pour l'organisation des graphes pharmacophoriques (un nœud – un pharmacophore / un arc – une relation d'inclusion)

Graphes potentiellement étiquettés. **Nombre de sommets traités** (en général) : 10-30, quelques dizaines (voir récemment quelques centaines) de milliers de graphes par jeu de données.

Contexte général Les méthodes informatiques traitées ici sont dédiées à la **manipulation de graphes moléculaires**. Un graphe moléculaire est un graphe qui modélise une molécule. La figure 2.1 montre deux modélisations d'une molécule exploitées actuellement : la forme squelettique usuelle et une forme de "graphes des fonctions pharmacophoriques" beaucoup moins courante. En prenant un compte un jeu de molécules, une grande partie des travaux consiste à **générer des pharmacophores**. Ensuite, l'**analyse**

des pharmacophores contient des traitements automatiques qui orientent des explorations manuelles du chimiste afin de découvrir des liens dans les résultats et de construire sa théorie chimique.



Formule squelettique | Occurrences de fonctions pharmacophoriques | Graphe des fonctions pharmacophoriques

FIGURE 2.1. Deux modélisations d'une molécule sous la forme d'un graphe

2.3 Graphes - arbres et automates pour la bio-informatique

Mots clés Trie, arbres des suffixes, Automate, Graphe de de Bruijn, Transformée de Burrows- Wheeler, Indexation, Recherche de mot

Problématiques scientifiques Algorithmique du texte, Recherche de motifs

Domaines applicatifs (exemples) Bio-informatique

Personnes de la fédération impliquées

LITIS

Richard Groult (TIBS)
Yannick Guesnet (TIBS)
Thierry Lecroq (TIBS)
Arnaud Lefebvre (TIBS)
Laurent Mouchard (TIBS)
Élise Prieur-Gaston (TIBS)

Quel(s) graphe(s) ? Arbres (voir Figure 2.2 à gauche), Automates (voir Figure 2.2 à droite)
Nombre de sommets traités (en général) : jusqu'à quelques millions de sommets

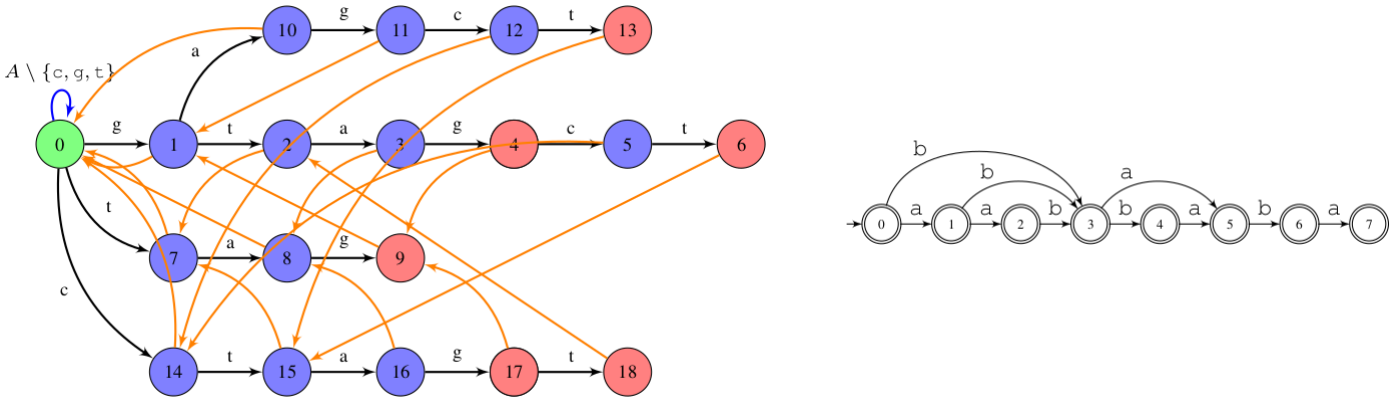


FIGURE 2.2. **A gauche** : Machine de Aho-Corasick pour l'ensemble fini de mots X = {gtagct, tag, gagct, ctagt} ; **A droite** : L'oracle des facteurs de *aabbaba*

Contexte général L'**algorithmique du texte** vise à développer des **structures de données et des algorithmes efficaces pour manipuler des suites de symboles** (appelées mots, textes ou séquences). L'aspect sémantique sous-jacent à ces suites de symboles est rarement pris en compte. En algorithmique du texte, plusieurs types de graphes sont utilisés. Pour la **recherche de motifs** par exemple, on

utilise fréquemment des arbres [36]. Des automates sont aussi utilisés (oracle des facteurs) pour réduire l'espace mémoire nécessaire au stockage des séquences [55]. En effet, l'une des problématiques qui a émergé avec l'accumulation des données (notamment avec le séquençage à haut débit) consiste à **trouver des alternatives économiques à ces graphes qui deviennent trop volumineux**. Les graphes de Bruijn ont également été utilisés pour la correction de données de séquençage de troisième génération produisant des fragments longs et bruités [67, 61, 66].

2.4 Graphes en informatique mathématique

Mots clés : Théorie des graphes, algorithmique des graphes, combinatoire, famille de graphes (cordaux, de Cayley, de Bruce Reed,...,), paramètre de graphes, complexité algorithmique, trie, arbres des suffixes, automates, hypergraphes, modèle RAM, DAG, modélisation.

Problématiques scientifiques : Étude de paramètres et d'algorithmes de graphes, y compris sur des sous-classes de graphes. Génération et énumération de graphes contraints. Expressivité des modèles.

Domaines applicatifs (exemples). Même si les travaux sont essentiellement théoriques, ils sont susceptibles d'application dans les domaines suivants : transmission, stockage et protection de l'information, représentation des documents, compilation, spécification et vérification, calcul formel, logique formelle.

Personnes de la fédération impliquées.

LITIS	GREYC
L'équipe C&A (*Nicolas Bedon*)	*Paul Dorbec (AmacC)*, Julien Clément (AmacC), Julien Courtiel (AmacC), Matthieu Dien (AmacC), Etienne Grandjean (AmacC), Loick Lhote (AmacC), *Brigitte Vallée (AmacC)*

Quel(s) graphe(s) ? Les types de graphes sont très variés. On distingue néanmoins les graphes *nus*, correspondant aux définitions classiques de graphes, et les graphes *décorés*, qui sont des graphes nus auxquels on ajoute des informations. Les graphes décorés sont utilisés comme structures de données ou comme modèles. La Figure 2.3 donne des exemples.
Nombre de nœuds traités (en général) : très variable. Quand l'étude reste théorique il n'y a pas de limite sur la taille de l'objet.

Contexte général. On étudie des **propriétés de familles particulières** de graphes, par exemple les graphes de degré au moins 3 pour la problématique de la couverture minimale. On transpose des notions (et propriétés) bien connues sur des graphes à des modèles plus généraux comme les hypergraphes. On utilise aussi les graphes pour modéliser des algorithmes ou des systèmes afin d' en étudier les propriétés. Via des études de complexité algorithmique, des problèmes (par exemple en cryptographie) sont réduits à des problèmes sur les graphes. Enfin, les graphes sont aussi utilisés comme des **structures de données** dont on étudie les aspects structurels et algorithmiques.

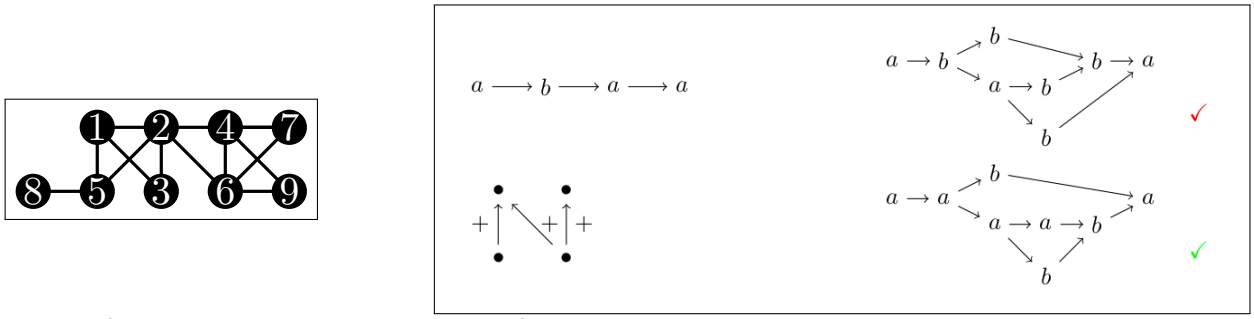


FIGURE 2.3. À gauche, un 1-2-arbre étiqueté. À droite, des ordres étiquetés représentés par des graphes.

2.5 Graphes dynamiques dans le sport

Mots clés Graphes dynamiques, Trajectoires, Sport, E-sport

Problématiques scientifiques Détection dans des graphes dynamiques de motifs récurrents, Fouille de données

Domaines applicatifs (exemples) Handball, Sport collectif, E-sport collectif

Personnes de la fédération impliquées

LITIS	GREYC
Rodolphe Charrier (RIIC)	*François Rioult (CODAG)*
Géraldine Del Mondo (APP)	Albrecht Zimmermann (CODAG)
Éric Sanlaville (RIIC)	
Alexandre Pauchet (MIND)	

Quel(s) graphe(s) ? Graphes simple orientés ou non, étiquettés, éventuellement spatialisés. Les sommets représentent les entités du jeu (e.g. joueurs, ballon, zones) et les arêtes les relations entre ces entités (e.g. distance). **Nombre de sommets traités** (en général) : par graphe une vingtaine, sur des séquences de plusieurs centaines de graphes.

Contexte général L'objectif général est d'être capable d'analyser les tactiques de jeux en sport collectif. Dans ce but plusieurs approches à base de graphe ont été envisagées, dont l'utilisation de graphes dynamiques sémantisés (Figure 2.4 à gauche) ou des diagrammes de Voronoï (Figure 2.4 à droite). Plusieurs méthodologies sont à l'étude, notamment **rechercher des patterns de jeux existants** (i.e. enclenchements) dans des suites de graphes ou des démarches plus exploratoires d'**émergence de patterns récurrents**.

2.6 Graphes dynamiques - réseaux d'interactions

Mots clés Graphes dynamiques, Intelligence Collective, Auto-organisation, Systèmes complexes, réseaux mobiles, Algorithmique distribuée

Problématiques scientifiques Analyse des réseaux complexes et dynamiques, Algorithmes d'optimisation, Algorithmique des systèmes distribués

Domaines applicatifs (exemples) Organisation du territoire et modélisation spatialisée, Logistique et planification en environnement dynamiques et incertains, Réseaux de capteurs, Drones en essaim, Sys-

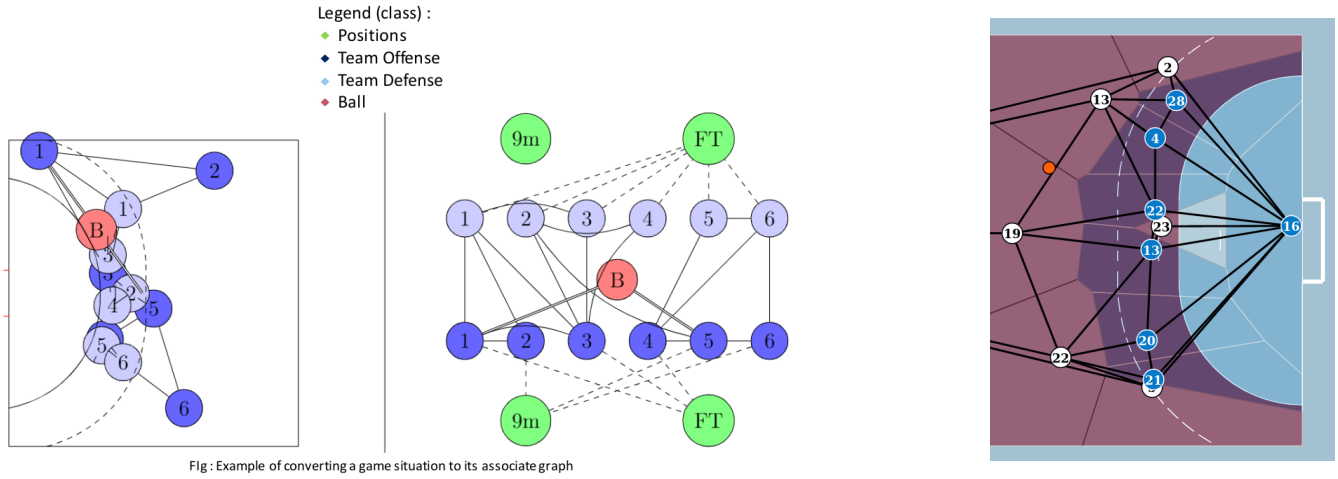


FIGURE 2.4. Deux modélisations de situations de jeu en Handball

tèmes d'information distribués, Analyse des stratégies en sports collectifs

Personnes de la fédération impliquées

LITIS	GREYC
L'équipe RIIC (*Yoann Pigné*)	Morgan Barbier (AMACC)
Géraldine Del Mondo (APP)	Paul Dorbec (AMACC)
Pierrick Tranouez (APP)	Julien Clément (AMACC)
	Julien Courtiel (AMACC)
	François Rioult (CODAG)
	Albrecht Zimmerman (CODAG)

Quel(s) graphe(s) ? Graphes simples orientés ou non, étiquettés ou non. Les sommets et les arêtes peuvent représenter des entités et relations du monde réel (e.g. robots et distances entre ces robots dans le cadre de l'alignement autonome de robots sans communication ni information de placement) ou être abstraits pour des travaux plus fondamentaux (e.g. travaux sur la persistance des composantes connexes) **Nombre de sommets traités** (en général) : quelques milliers à quelques dizaines de milliers de sommets et des séquences de quelques milliers d'étapes.

Contexte général Les verrous scientifiques tournent autour de la notion d'**analyse et de contrôle des réseaux d'interaction**. Cette notion peut être en partie représentée par des graphes, auxquels s'ajoute une dimension temporelle qui permet de **capturer la dynamique des réseaux**. Ces graphes incluant l'évolution de réseaux portent de multiples noms dans la littérature : *time-varying graphs*[32], *evolving graphs*[19], *dynamic networks*[30], *temporal networks*[52], en fonction de leur domaine d'application d'origine. On trouve principalement trois méthodologies générales d'utilisation des graphes.

La première, l'**analyse des réseaux complexes et dynamiques** cherche à comprendre la génèse de ces réseaux et de leurs structures (morphogénèse), et à la simuler à partir de modèles utilisant les graphes dynamiques. Il s'agit également de caractériser l'**émergence de propriétés** globales sur ces réseaux.

La deuxième traite des **algorithmes d'optimisation**. De nombreux problèmes sur les systèmes complexes réels s'expriment comme des problèmes d'optimisation, par exemple de type planification ou tournées de véhicules. L'idée est de les résoudre à l'aide des outils issus de la **recherche opérationnelle** : programmation mathématique, metaheuristiques et algorithmes classiques sur les graphes, revisités

pour satisfaire les exigences dues à la dynamicité et parfois à la nature distribuée des systèmes.

La troisième concerne l'**algorithmique des systèmes distribués**. Il s'agit de proposer des algorithmes permettant de piloter ces systèmes en garantissant certaines propriétés liées à la connectivité, la sécurité, l'économie d'énergie, le maintien de structures géométriques. L'application phare de cet axe concerne le vol d'une formation autonome de drones.

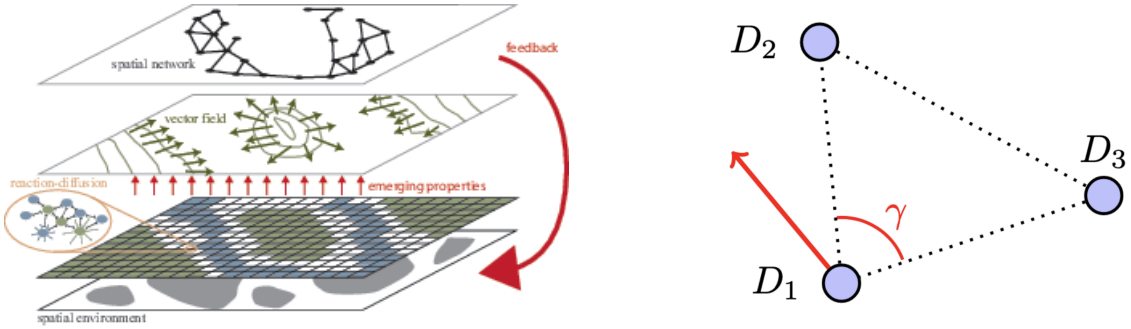


FIGURE 2.5. Illustrations de 2 axes de recherche : **à gauche** analyse de réseaux complexes (Morphogénèse de réseaux urbains), **à droite** l'algorithmique distribuées (positionnement de robots)

2.7 Apprentissage sur graphes

Mots clés Graph Matching, Distance d'édition et noyaux sur graphes, GNN

Problématiques scientifiques Définition de métriques sur graphes, pré image de graphes, matching/isomorphisme de graphes, réseaux de neurones sur graphes.

Domaines applicatifs (exemples) Chemo-Informatique, Analyse d'images de documents

Personnes de la fédération impliquées

LITIS	GREYC
Sébastien Adam (APP)	Luc Brun (IMAGE)
Maxime Berar (APP)	Sébastien Bougleux (IMAGE)
Benoît Gaüzère (APP)	
Pierre Héroux (APP)	
Paul Honeine (APP)	

Quel(s) graphe(s) ? Graphes moléculaires, de protéines, graphes de documents ou d'images.
Nombre de sommets traités (en général) : de quelques dizaines à quelques centaines de sommets.

Contexte général Dans cette approche, chaque graphe décrit un objet affecté d'une propriété (une classe ou une valeur scalaire). L'activité de recherche consiste à concevoir des **méthodes d'apprentissage machine** adaptées à ces objets.

Une première approche consiste à définir une métrique dans l'espace des graphes. Ceci peut se faire via la conception de méthodes efficaces de calcul de la **distance d'édition** sur graphes(Figure 2.6 dessus). L'avantage de cette approche est que l'interprétabilité de la distance d'édition est facilement compréhensible par des experts. Son principal inconvénient est d'être peu compatible avec les méthodes usuelles d'apprentissage machine.

Inversement, le calcul de similarité entre graphes peut être calculé à l'aide de **méthodes à noyaux** (Figure 2.6 en bas à droite). Les résultats sont plus difficilement interprétables par un expert, mais les

noyaux offrent une plus grande flexibilité dans la définition de la mesure de similarité et sont facilement interfaçable avec de nombreuses méthodes d'apprentissage. Notons que l'interprétabilité des résultats peut être améliorée par des travaux portant sur la pré image, qui consiste par exemple à expliciter le graphe moyen calculé avec la métrique déduite des noyaux.

Enfin les méthodes d'apprentissage machine récentes et dédiées aux graphes se focalisent sur la transcription des méthodes d'**apprentissage de représentations** au domaine des graphes. Cela nécessite de transcrire à la fois les opérations de convolution et de pooling (sous échantillonage de graphe).

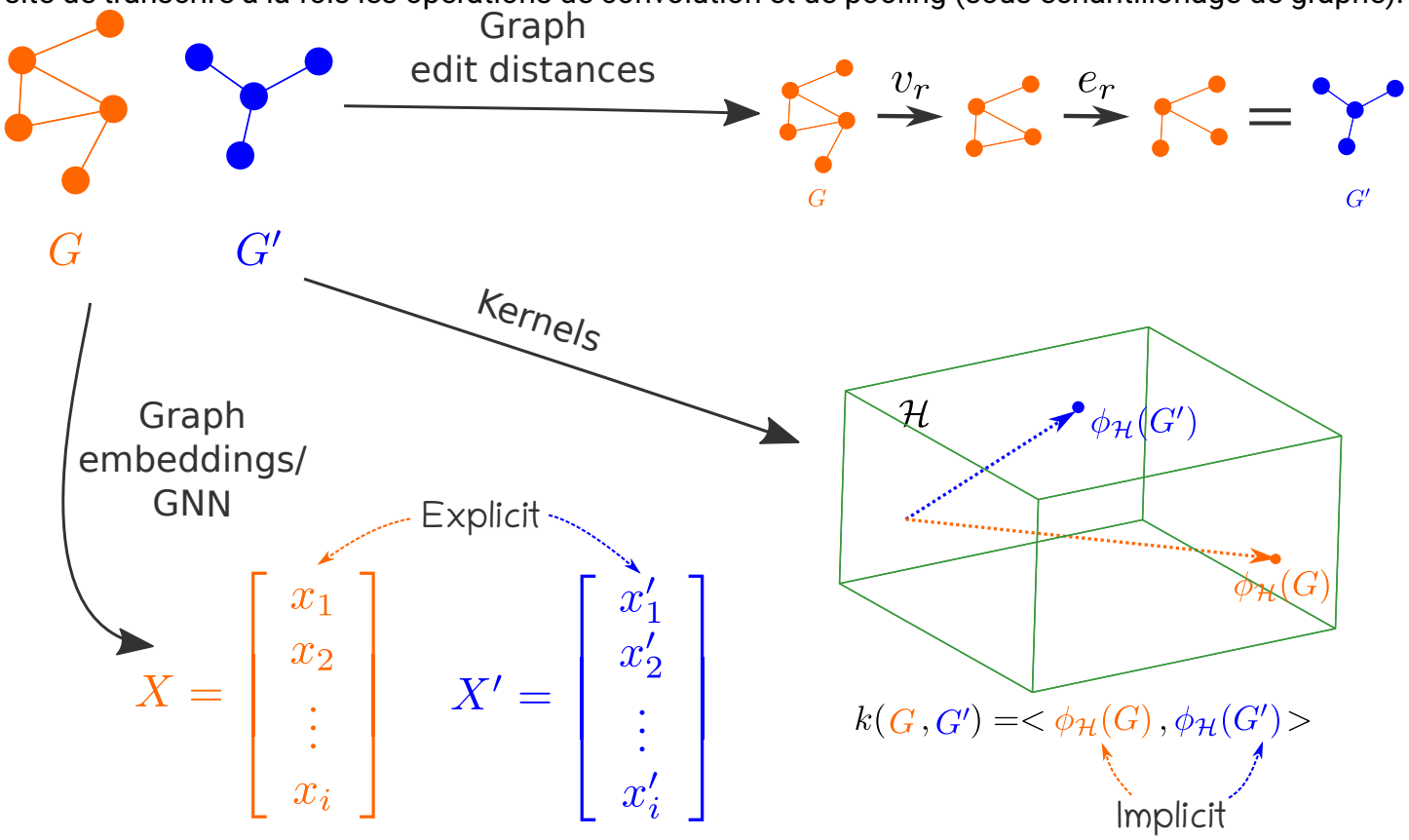


FIGURE 2.6. Différentes approches pour connecter Graphes et apprentissage machine

Notons que les propriétés à prédire, peuvent également être localisées sur les sommets des graphes (plutôt que sur le graphe complet). On parle dans ce cas de problèmes transductifs. On a alors en entrée un graphe, généralement complet, pour lequel on connaît la structure et les caractéristiques de tous les sommets. Il s'agit alors d'inférer la classe de sommets inconnus à partir d'un nombre restreint de sommets étiquetés.

2.8 Traitement sur signal sur graphes

Mots clés Traitement de signaux et morphologie mathématique multivariée sur graphes, rehaussement de détails, débruitage, inpainting, GNN

Problématiques scientifiques Étendre des familles de méthodes définies dans $\mathbb{R}^n$ (traitement du signal, morphologie mathématique) à des espaces de topologie plus complexe définis par des graphes, Apprentissage profond et réseaux de neurones à convolution sur graphes.

Domaines applicatifs (exemples) Imagerie médicale, Affective computing, Réalité virtuelle, Sécurité dans les transports

Personnes de la fédération impliquées

GREYC

Sébastien Bougleux (IMAGE)
Christophe Charrier (SAFE)
Luc Brun (IMAGE)
Olivier Lézoray (IMAGE)
Abder Elmoataz (IMAGE)

Quel(s) graphe(s) ? Les graphes peuvent coder des images, des maillages 3D ou l'ensemble des objets d'une base de donnée. Les sommets sont les objets à traiter et les arêtes les distances de similarité entre ces objets.
Nombre de sommets traités (en général) : plusieurs milliers à millions de sommets.

Contexte général La plupart des méthodes de **traitement d'images** exploitent explicitement le fait de travailler sur une grille discrète où la distance entre deux élément se déduit de leur distance Euclidienne. Ce point ne permet pas d'adresser de nombreuses applications où l'on dispose de similarités (ou distances) entre les objets sans que ceux-ci soient disposés sur une grille. L'idée est donc d'étendre les méthodes usuellement utilisées en traitement d'images en définissant un graphe dont les sommets sont les objets à traiter et les arêtes codent la similarité entre ces objets. Cette approche permet de réinvestir de vieux domaines tels que le **traitement du signal ou la morphologie mathématique pour les adapter aux graphes**. Ceci permet par exemple d'effectuer des rehaussement de contours sur des maillages 3D (Figure 2.7).

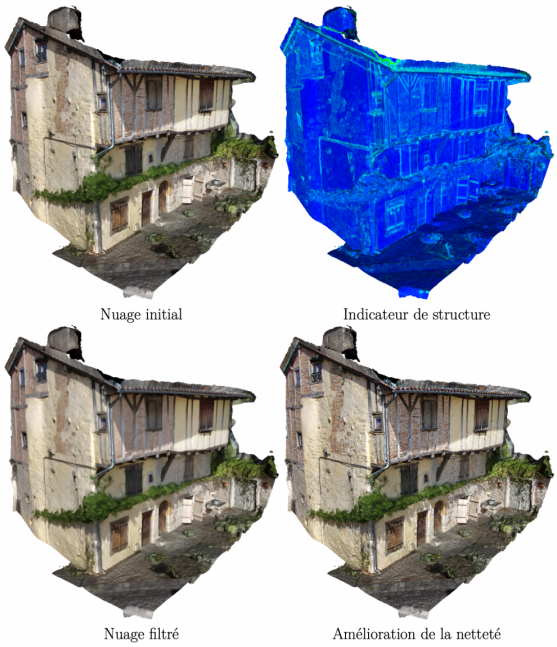


FIGURE 2.7. Simplification adaptative de maillages 3D colorés pour leur rehaussement

2.9 Graphes (spatio-temporels) sémantisés

Mots clés Modélisation de l'information spatiale et temporelle, Théorie des graphes, Raisonnement qualitatif, Représentations multi-échelles, Aide à la décision, Sciences de l'Information Géographique

Problématiques scientifiques Modélisation de l'information spatiale et temporelle par des modèles à base de graphes ; Traitements automatisés sur des graphes (statiques ou dynamiques), e.g. isomorphismes de sous-graphes, similarité entre graphes

Domaines applicatifs (exemples) Gestion des risques, Dynamiques spatiales historiques, Analyse du trafic routier, Mobilité, Logistique portuaire (gouvernance), Sport

Personnes de la fédération impliquées

LITIS	GREYC
Sébastien Adam (APP)	Paul Dorbec (AAMAC)
Géraldine Del Mondo (APP)	Bruno Crémilleux (CODAG)
Benoît Gaüzère (APP)	Bertrand Cuissard (CODAG)
Eric Sanlaville (RIIC)	François Rioult (CODAG)
Pierrick Tranouez (APP)	

Quel(s) graphe(s) ? D'une manière générale, les sommets représentent les entités du monde et les arêtes (ou arcs) les interactions entre ces entités. Il n'y a pas a priori de restriction sur les types de graphes utilisés (e.g. orientés ou non, graphes simples ou non, hyper-graphes (e.g. Figure 2.8 à droite), statique ou dynamique (e.g. Figure 2.8 au centre) etc. **Nombre de sommets traités** (en général) : entre 10 et 1000.

Contexte général Ce travail de recherche est principalement axé sur la **modélisation de l'information spatiale et temporelle** en utilisant des modèles à base de graphes. A partir d'une situation ou d'un phénomène réels il s'agit de trouver la modélisation la plus appropriée permettant l'application de traitement

automatique de processus sur cette structure de données. L'idée est de choisir la **structure** de graphe (e.g. hypergraphe, bigraphe, multigraphe) et de la "customiser" (i.e quelles informations doivent être représentées, cela passe notamment par des choix dans la valuation des arcs et des sommets) pour parvenir à répondre au problème posé. L'information quantitative étant traditionnellement intégrée (et traitée) dans la plupart des modèles, on essaye au maximum d'intégrer une dimension qualitative dans les choix de représentation **sémantique** de l'information. L'hypothèse est que cette dimension peut enrichir les modèles voire permettre d'optimiser certains traitements. Il n'est pas nécessaire de disposer de grandes masses de données, au contraire la plupart des travaux conduits ont pour pour hypothèse un ensemble de données réduit.

Par ailleurs, ce travail est décliné aux **représentations à plusieurs niveaux d'abstraction** (i.e. multi-échelles, e.g. Figure 2.8 à gauche), représenter ces niveaux permet un accès à l'information plus ciblé (peut améliorer l'efficacité de certains processus) et la combinaison d'information de plusieurs niveaux peut générer de la connaissance qui n'était pas directement présente au départ.

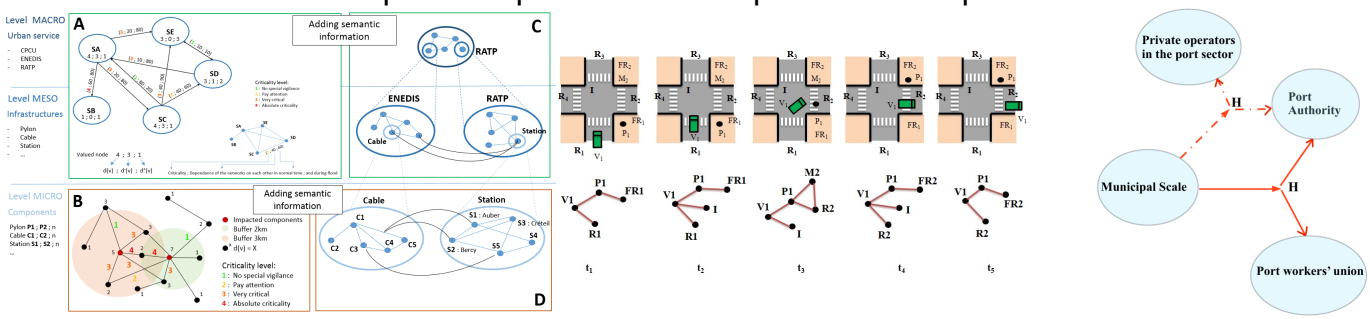


FIGURE 2.8. Modélisation de l'information spatiale (et éventuellement temporelle) dans 3 domaines d'application : risque, mobilité et logistique

Chapitre 3 Synthèse

Cette partie présente dans un premier temps une synthèse des approches que nous avons pu identifier. L'idée est d'une part de distinguer ce qui est de l'ordre de l'approche méthodologique de ce qui fait référence plus directement à des applications, et d'autre part de regrouper autant que possibles les approches. Nous avons identifié **3 courants méthodologiques** et mis en avant **5 applications phares**. L'objectif de ce chapitre est de décrire ces éléments et de commencer à **mettre en perspectives les approches** de la partie précédente pour "justifier" les courants méthodologiques mais surtout pour préparer la dernière partie de ce document qui consistera à proposer des opportunités et perspectives de collaborations entre ces groupes.

3.1 Les différentes approches sur graphes au sein de la fédération

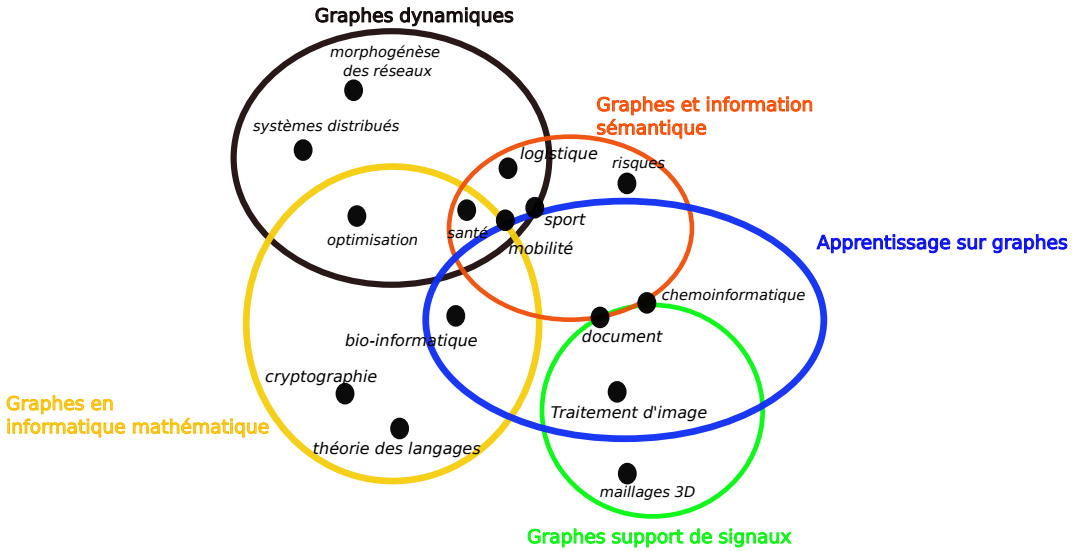


FIGURE 3.1. Les principales approches sur graphes au sein de NormaSTIC avec des exemples d'applications.

3.1.1 Analyse d'ensembles de Graphes

Présentation

L'analyse d'ensembles de graphes doit **traiter une multitude de graphes**, chaque graphe étant affecté d'une propriété (une classe ou une valeur réelle). L'activité de recherche consiste à définir des méthodes permettant de **prédire la valeur d'un graphe inconnu** à partir d'une base d'apprentissage de couples (graphe, propriété). Ce genre de problématique est classiquement résolu soit par des méthodes de **fouille de données**, soit par des méthodes d'**apprentissage machine**. Chaque approche recouvre bien sur plusieurs problématiques scientifiques. Les tailles des graphes incriminés sont généralement petites (une dizaine de sommets) à moyennes (quelques centaines).

Principales problématiques scientifiques

Fouille de données sur graphes Dans cette problématique, on doit manipuler un ensemble de graphes (codant généralement des molécules) divisé entre des graphes vérifiant une certaine propriété et des graphes ne la vérifiant pas. L'objet de la fouille de données va être de trouver des sous graphes fortement

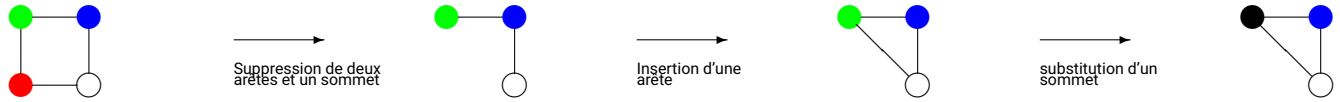


FIGURE 3.2. Un exemple de chemin d'édition

présents dans les graphes vérifiant la propriété et faiblement présent dans ceux ne la vérifiant pas. Dans le cadre de la recherche de médicaments, ces sous graphes sont appelés des pharmacophores. En fonction de la taille des ensembles envisagés, cette recherche peut produire des milliers ou des centaines de milliers de pharmacophores potentiels. Cet ensemble est ensuite filtré (par exemple grâce à la notion de sous-graphe) afin d'extraire un ensemble réduit de pharmacophores qui pourront être étudiés plus avant par un biologiste.

Métriques sur graphes L'essence de l'apprentissage machine est d'affecter une propriété à un objet inconnu à partir des propriétés d'objets supposés proches de celui-ci. Ceci passe donc nécessairement par la définition d'une métrique sur l'ensemble des objets (qui sont ici des graphes).

La **distance d'édition** est une métrique historique dans l'espace des graphes. Celle-ci est liée à la notion de **chemin d'édition** (Figure 3.2). Un chemin d'édition entre deux graphes G_1 et G_2 permet de transformer G_1 en G_2 à l'aide de suppression/insertion de sommets ou d'arêtes ainsi que de changement de label (valeur) de sommets ou d'arêtes. Chaque opération élémentaire d'un chemin d'édition peut être affectée d'un coût, ce qui permet de définir le coût d'un chemin d'édition comme la somme de ses coûts élémentaires. La notion de distance d'édition se définit alors simplement comme le plus court chemin (ou la transformation la moins coûteuses) entre deux graphes. Le GREYC et le LITIS ont contribués de nombreuses fois à cette problématique soit séparément soit en collaboration.

Par exemple, de 2017 à 2019 le GREYC et le LITIS ont participé au projet régional AGAC [(1)] regroupant le GREYC, le LITIS, le COBRA, le LCMT et le CERMN. Dans le cadre de ce projet un algorithme de calcul de graphe médian a été développé. Ce projet a également permis de jeter les bases d'un projet ANR portant sur les réseaux de neurones sur graphes(voir infra).

Les **méthodes à noyaux** sur graphes permettent, comme la distance d'édition, de définir une métrique sur l'espace des graphes. Toutefois, à la différence de la distance d'édition, la métrique obtenue peut être combinée avec l'ensemble des méthodes à noyaux (la métrique de la distance d'édition n'est pas définie positive) Ceci permet de combiner ces noyaux avec l'ensemble des méthodes à noyaux définies en apprentissage machine (SVM, KPCA,...). On peut ainsi classifier des graphes, calculer l'axe principal de variations d'un ensemble de graphes ainsi que de nombreuses autres opérations jusqu'à présent restreintes aux données vectorielles. Le GREYC et le LITIS ont mené des travaux dans ce domaine. Les deux principaux domaines d'applications visés ont été la **reconnaissance de formes** (au sens shape) [82, 42] et la chémoinformatique [44, 46].

Bien que très efficaces, les méthodes à noyaux sur graphes fournissent généralement des résultats difficilement interprétables par un expert. L'équipe APP du LITIS travaille sur ce problème en cherchant à calculer la pré-image d'un ensemble de graphes. Il s'agit dans ce cas d'expliciter un graphe dont on ne connaît (via les noyaux) que les similarités avec les graphes d'un jeu de données(Figure 2.6). La technique utilisée et proche d'un calcul de graphe médian et combine des méthodes à noyaux et la distance d'édition. Dans le cadre d'une classification non supervisée, ces travaux permettent par exemple, d'expliciter les graphes codant le centre de chacune des classes obtenues.

(1). Page Web du projet

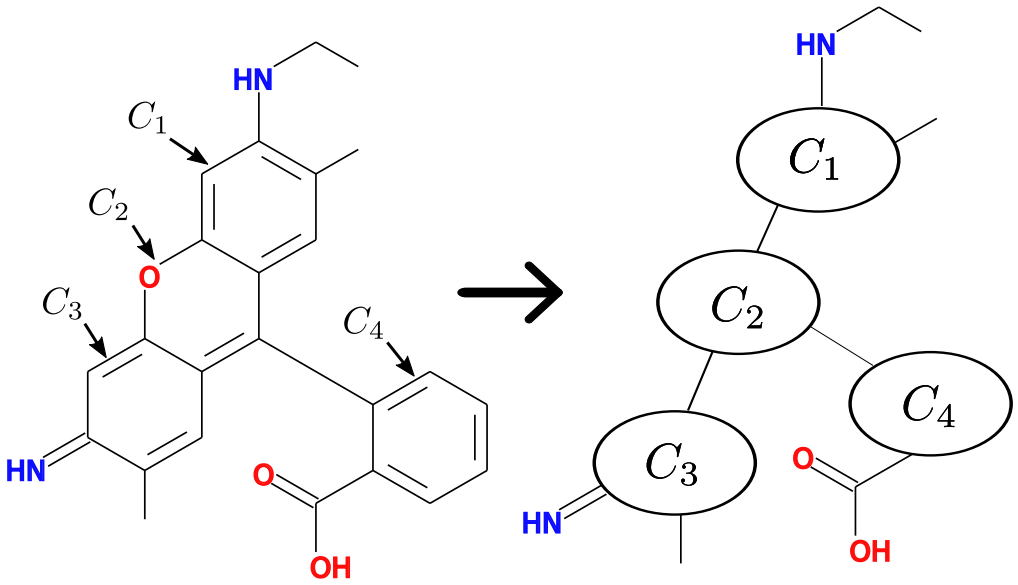


FIGURE 3.3. Une représentation compacte des graphes moléculaires pour le calcul des noyaux sur graphes.

Réseaux de neurones sur graphes Dans le cadre du projet AGAC (voir distance d'édition), le LITIS s'est intéressé aux méthodes de **réseaux de neurones** sur graphes et plus précisément aux méthodes de convolution sur graphes. De cette étude à débouché une théorie permettant d'unifier les deux grandes familles de convolution sur graphes (spectrales et spatiales) et de pointer leurs limites [11, 76]. Suite au projet AGAC, le GREYC et le LITIS ont monté avec le laboratoire LIFAT de Tours un projet ANR (CoDeGNN) (2) qui a été acceptée et est actuellement en cours. Les principaux domaines d'applications de ces travaux sont la bio et chémo informatique ainsi que l'étude des réseaux sociaux.

3.1.2 Traitement du signal sur graphes

Présentation

A l'inverse de la Section 3.1.1 où chaque donnée est représentée par un graphe et où on s'efforce de définir une métrique entre ces objets, on considère ici que chaque objet est codé par un sommet. Les poids des arêtes reliant ces sommets codent la similarité entre les objets (et donc la métrique définie sur l'espace codé par le graphe). Les arêtes du graphe peuvent être dirigées permettant ainsi de rendre la métrique non symétrique. Cette approche connue sous le nom de Traitement du signal sur graphe permet de **traiter des ensembles d'objets de grandes dimensions** et peut se décomposer en plusieurs sous domaines.

Principales problématiques scientifiques

Régularisation adaptative de signaux Dans cette problématique, la **valeur des sommets est considérée comme un signal** sur le graphe qu'il convient de régulariser en fonction de la métrique définie par les poids des arêtes. Le gradient de la fonction le long d'une arête, peut être interprété comme une dérivée partielle de la fonction et se calcule à partir des valeurs de la fonction sur deux sommets adjacents, du poids de l'arête reliant ces sommets et de l'ensemble des poids des arêtes incidentes aux deux sommets. Cette notion de gradient permet d'introduire la notion de régularisation qui se formalise

(2). Page Web du projet

(a). original (b). Contour initial (en vert) (c). Recolorisation

FIGURE 3.4. Contour actifs sur graphes (ici un maillage 3D) pour une tache d'édition : recolorisation

comme la minimisation d'une fonctionnelle avec un terme de régularisation (dépendant du gradient) et une attache aux données.

Morphologie mathématique sur graphes L'introduction du gradient sur graphe permet d'introduire d'autres opérateurs tels que la divergence et le p-Laplacien. L'ensemble de ces opérateurs permet de considérer de nombreux problèmes définis dans le cas continue comme la solution d'une équation aux dérivées partielles. On parle dans le cas des **graphes d'équations aux différences partielles**. Les opérateurs de **morphologie mathématique**, à commencer par l'érosion et la dilatation, peuvent être définis comme la solution d'une équation différentielle non linéaire [26]. Ce pont entre morphologie mathématique et calcul différentiel permet de définir des opérateurs morphologique sur graphes via la notion de gradient.

Contours actifs sur graphes La définition d'opérateurs différentiels sur graphes permet d'adapter l'équation Eikonale qui est la base des méthodes type Level set de **segmentation d'images**. Son extension aux graphes permet donc de segmenter des données beaucoup moins structurées (Figure 3.4) ou de segmenter des images en introduisant des métriques non locales via des arêtes reliant des pixels distants.

3.1.3 Graphes dynamiques

Présentation

Ce thème a pour objet principal d'étude l'**analyse et le contrôle des réseaux d'interaction**. Cette notion peut être en partie représentée par des graphes, auxquels s'ajoute une dimension temporelle qui permet de capturer la **dynamique des réseaux**. Ces graphes incluant l'évolution de réseaux portent de multiples noms dans la littérature : *time-varying graphs*[32], *evolving graphs*[19], *dynamic networks*[30], *temporal networks*[52], en fonction de leur domaine d'application d'origine. Si dans ce thème les sommets du graphes représentent souvent des objets comme dans la Section 3.1.2, les arêtes en revanche représentent des interactions au sens large et non spécifiquement la similarité entre chaque objet. Les types de graphes peuvent être divers (orientés/non orientés, éventuellement valués). La taille des don-

nées ici représente quelques milliers à quelques dizaines de milliers de sommets et des séquences de quelques milliers d'étapes.

Principales problématiques scientifiques

Analyse des réseaux complexes et dynamiques L'objectif est de **comprendre la génèse des réseaux d'interactions et de leurs structures** (morphogénèse), et à la simuler à partir de modèles utilisant les graphes dynamiques. Il s'agit également de caractériser l'**émergence de propriétés** globales sur ces réseaux. A titre d'exemple les travaux [84, 83] abordent les aspects computationnels de la morphogénèse de réseaux complexes, en proposant un modèle général, capable de simuler leur formation. Les réseaux sont générés sous l'influence de contraintes qui s'expriment par l'intermédiaire d'un champ vectoriel qui est déterminé à l'aide d'un système de réaction-diffusion. Un modèle de Gray-Scott produisant une grande variété de motifs dynamiques est utilisé. Le champ vectoriel obtenu contrôle la géométrie et le taux de croissance du réseau construit qui rétroagit sur le processus de réaction-diffusion.

Algorithmes d'optimisation Cet axe a pour but de répondre à des problèmes portant sur des **systèmes complexes réels** et pouvant s'exprimer comme des problèmes d'optimisation. Les méthodes utilisées relèvent de la **recherche opérationnelle**, programmation mathématique, metaheuristiques et algorithmes classiques sur les graphes, revisités pour satisfaire les exigences dues à la dynamicité et parfois à la nature distribuée des systèmes. A titre d'exemple, des travaux [87, 86] s'intéressent à la notion de connectivité dans les graphes dynamiques. Un graphe non orienté est défini sur un intervalle de temps fini et discret. Les arcs peuvent apparaître ou disparaître pendant l'intervalle de temps. Le premier objectif de ce travail est d'étendre la notion de composante connexe aux graphes dynamiques d'une manière nouvelle (Figure 3.5). Les composantes connexes persistantes sont définis par leur taille, correspondant au nombre de sommets, et leur longueur, correspondant au nombre de pas de temps consécutifs sur lesquels ils sont présents. Le deuxième objectif de ce travail est de développer un algorithme calculant les plus grandes (en termes de taille et de longueur) composantes connexes persistantes dans un graphe dynamique. L'algorithme PICCNIC (PersIstent Connected CompoNent In-Cremental Algorithm) est un algorithme polynomial en temps de complexité minimale.

Algorithmique des systèmes distribués L'objectif est ici de proposer des algorithmes permettant de **piloter ces systèmes** en garantissant certaines propriétés liées à la connectivité, la sécurité, l'économie d'énergie, le maintien de structures géométriques. Il s'agit notamment de permettre le vol d'une formation autonome de drones uniquement guidés par leurs perceptions respectives et leurs interactions respectives, **sans nécessiter l'intervention d'un guidage centralisé**. Au delà du challenge technologique, le projet fait appel aux notions d'algorithmiques distribuées et à la modélisation par graphes dynamiques. Dans [47] par exemple, les auteurs s'intéressent à l'alignement autonome de robots sans communication ni information de placement. Seule la vision des robots est utilisée pour évaluer les positions angulaires des voisins.

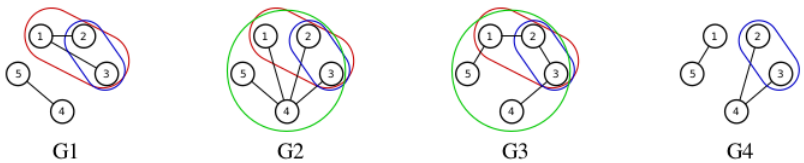


FIGURE 3.5. Une problématique autour d'algorithmes d'optimisation appliquée aux graphes dynamiques (composantes connexes persistantes)

3.1.4 Graphes (spatio-temporels) sémantisés

Présentation

Ici le graphe a vocation a représenter la réalité le plus fidèlement possible sans tomber dans l'écueil d'une modélisation à une échelle trop fine. Le graphe n'est pas seulement un couple de sommets et d'arêtes, il dispose aussi d'**information sémantique** lui permettant de capturer les caractéristiques du monde réel, spécifiques au domaine applicatif pour lequel il est défini. En cela ce thème se distingue de celui présenté en Section 3.1.3, a fortiori sur l'aspect **qualitatif** de l'information. Tout type de graphe peut-être choisi et adapté (simple ou non, orienté ou non, hyper-graphe, multi-niveaux, statique, dynamique etc). Tout comme dans la section 3.1.3, l'**interdisciplinarité** est au coeur de ce domaine, mais ici un focus fort (qui correspond également à une des problématiques de recherche de ce thème) est la définition du modèle de graphe sémantique qui sera en mesure de répondre au problème posé. Ce modèle devant potentiellement aussi prendre en compte d'autres contraintes plus bas niveau comme l'espace mémoire utilisé (ce qui a un impact sur sa définition, Figure 3.6 à gauche). Les autres problématiques de recherche portent sur les traitements opérés sur ces graphes qui peuvent être assez divers, mais souvent en lien avec la **similarité entre graphes** ou le **matching de sous-graphes** ainsi que la **recherche de patterns** (potentiellement dynamique). Il est souvent nécessaire d'adapter les définitions d'algorithmes ou de métriques classiques définis sur des graphes statiques simples à d'autres types de graphes (e.g. hyper-graphes, multi-graphe, bi-graph). Les méthodes de résolution sous-jacentes sont essentiellement algorithmiques et sont communes aux méthodes décrites dans la section 3.1.3, partie algorithmes d'optimisation. Toutefois, l'un des verrous de ce thème est la prise en compte de la dimension sémantique dans ces algorithmes. La taille des données peut varier mais ne concerne pas de grande masse de données. Par ailleurs, la recherche de solutions pour un nombre restreint de données est aussi l'un des challenges de cette thématique.

Principales problématiques scientifiques

Définition du modèle de graphe L'aspect modélisation est en soit pour ce thème une problématique scientifique. La **définition du graphe dans sa structure** est une première étape qui consiste à rechercher le type de graphe le plus approprié pour représenter la réalité du problème posé. Des graphes multi-échelles [74] aux hypergraphes en passant par des graphes simples, l'idée est de percevoir les caractéristiques permettant par la suite de répondre au mieux à la problématique initialement posée.

A cette structure s'ajoute la **composante sémantique** qui permet d'enrichir l'information contenue dans le modèle au travers de **valuation quantitative et/ou qualitative** (un focus privilégié est fait sur ce deuxième aspect) des arêtes et des sommets. Les traitements pourront éventuellement s'abstraire de la sémantique pour des approches purement structurelles, ou au contraire en tenir compte. Souvent il s'agit de deux étapes distinctes.

De récents travaux ont suggéré la possibilité de lier ces graphes sémantiques à des graphes de connaissances dans des modèles plus complexes afin d'aller plus loin dans l'enrichissement du contenu et permettre l'inférence de solutions [39].

Recherche de patterns (statique ou dynamique) Ce traitement opéré sur les graphes sémantiques a pour objectif de rechercher des patterns spécifiques (e.g. zones de vulnérabilités dans un réseau viaire,

situations particulières dans le trafic routier[69, 71] (Figure 3.6 au centre), changements spatiaux dans des parcellaires historiques, enclenchements dans un match de handball). Ce qui est appelé "pattern" est un sous-graphe qui peut être défini sur un graphe statique ou dynamique. Par graphe dynamique on entend plutôt ici séquence de graphes statiques, même si des réflexions ont pu être menées sur des graphes dynamiques agrégés. Le matching de graphe est aussi un enjeu sous-jacent dans le sens où il est parfois nécessaire de faire le lien entre des données à des temps différents avant de pouvoir appliquer des algorithmes de recherche de patterns dynamiques (Figure 3.6 à droite).

Définition et calcul de métriques sur graphe L'analyse des graphes sémantiques conduit à l'utilisation et/ou à la définition de métriques sur graphes permettant de répondre aux problématiques pour lesquelles ces graphes ont été définis. Par exemple, la **comparaison de graphes** est souvent au coeur de beaucoup d'applications et nécessite la définition de métriques spécifiques sur ces graphes. S'il est possible d'appliquer des métriques classiques, il faut souvent les adapter aux types de graphes utilisés qui ne correspondent pas aux hypothèses initiales de ces métriques classiques. Par ailleurs, l'objectif est aussi d'en définir de nouvelles qui permettent d'être plus précis dans cette comparaison. Cette thématique pourrait être rapprochée de ce qui se fait dans la Section 3.1.1 mais sur des ensembles de graphes beaucoup plus restreint et avec des méthodologies différentes. L'adaptation des métriques aux aspects dynamiques comme dans la Section 3.1.3 est aussi une problématique potentielle de ce thème.

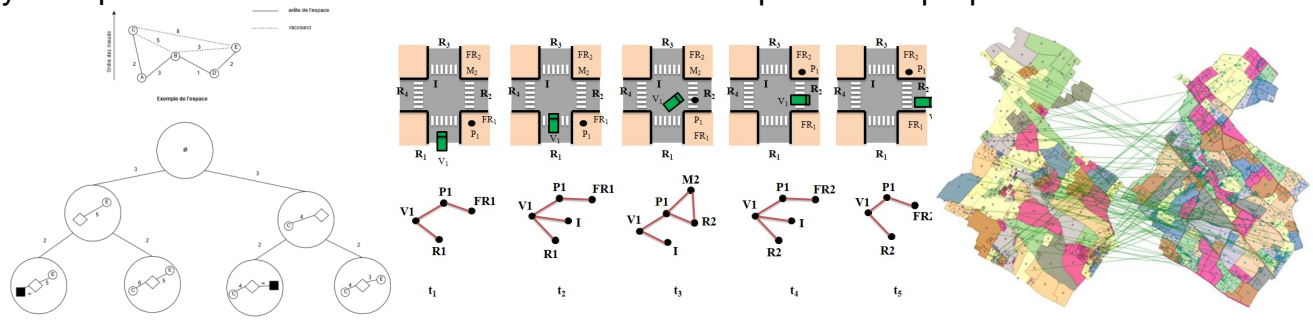


FIGURE 3.6. Modélisation de l'information spatiale (et éventuellement temporelle) dans 3 domaines d'application : risques, mobilité et dynamiques spatiales historiques

3.1.5 Graphes en informatique mathématique

Présentation.

En informatique mathématique, on étudie les classes de complexité algorithmique des problèmes associés, et on en déduit le cas échéant de l'algorithmique efficace. Les graphes étudiés peuvent être des graphes définis usuellement (graphe *nus*) ou des graphes nus auxquels on ajoute des informations (graphes *décorés*). On peut utiliser les graphes comme structures de données ou supports de modélisation. Les contraintes de modélisation ou l'algorithmique incitent souvent à restreindre la famille d'objets étudiée. On peut aussi chercher à transposer des propriétés connues sur les graphes classiques à des extensions de graphes (graphes dynamiques, hypergraphes [37]).

Principales problématiques scientifiques.

Étude des graphes nus. Un graphe est habituellement défini comme un couple (E, V), avec E un ensemble de *sommets* et V un ensemble d'*arêtes*. Cette définition peut être déclinée par exemple en

changeant le type des ensembles utilisés ou en imposant des propriétés sur V. Ce sont les graphes *nus*. Voici quelques exemples de graphes nus contraints :

- un graphe cordal est obtenu par ajouts successifs de sommets, de telle sorte qu'à chaque ajout, le voisinage du sommet dans le graphe forme une clique. Ils apparaissent dans de nombreux domaines, en algorithmique paramétrée (via la notion de treewidth), en statistique (réseaux bayésiens) ou en bio-informatique (phylogénie parfaite) ;
- les graphes de degré borné (degré = nombre de sommets voisins), où l'on étudie [41], par exemple, la conjecture de Bruce-Reed [75] qui relie, dans le cas connexe, le degré d'un graphe et la taille minimale d'un ensemble dominant (un ensemble D est dominant si tout sommet du complémentaire D^c est voisin d'un sommet de D).

On se pose d'abord des problèmes d'optimisation combinatoire (coloration, domination, ensembles indépendants) et on étudie les algorithmes correspondant : même si beaucoup de ces problèmes restent NP-difficiles (même sur ces sous-classes), divers algorithmes donnent des solutions satisfaisantes dans ces cas usuels, et on veut analyser leur complexité sur ces sous-classes. On réfléchit aussi à la liaison entre les propriétés qui définissent la sous-classe et l'estimation des principaux paramètres du graphe (dans le pire des cas ou en moyenne) ; c'est par exemple le cas de la conjecture de Bruce-Reed. On se pose aussi des questions liées au comptage et à la génération aléatoire de graphes d'une sous-classe donnée, comme dans le cas des graphes cordaux, par exemple.

Graphes nus supports de problématiques ou reliés à des algorithmes. Il s'agit ici d'utiliser les graphes nus comme support d'une problématique : la résolution d'un problème est ramenée à un problème de graphes, ou la caractérisation d'un système (algébrique par exemple) à une caractérisation sur les graphes. Voici quelques exemples :

- les monoïdes de traces [40] sont des objets algébriques particulièrement utilisés dans l'étude théorique des programmes concurrents ou dans lesquels certaines instructions peuvent commuter. La relation de commutation (ou de non commutation) dans l'étude des monoïdes de traces peut naturellement être représentée par un graphe ; des sous-familles permettent de caractériser et étudier des classes particulières de monoïdes de traces ayant de bonnes propriétés.
- le problème « Permutation-Équivalence de codes » [62] est essentiel dans la classification des codes et intervient en particulier en cryptanalyse. En supposant qu'on dispose d'un algorithme décidant si deux graphes pondérés non dirigés sont isomorphes, la permutation-équivalence des codes sur un corps arbitraire est décidable [13] de manière déterministe (sous hypothèse).
- l'ensemble des versions successives d'un logiciel, avec les liens de parenté entre les versions, définissent un graphe orienté. L'algorithme git-bisect [34] (utilisé dans le logiciel de développement collaboratif git) cherche un point de régression parmi les versions (une première version qui contient un bug) avec des méthodes proches de la recherche dichotomique. Il s'agit d'analyser finement la complexité de cet algorithme.

Graphes décorés comme structures de données. Quand on lui ajoute des attributs, le graphe devient une structure de données efficace pour organiser et stocker l'information ; elle se révèle donc essentielle dans beaucoup de cadres algorithmiques. Nous l'utilisons pour modéliser des processus, en algorithmique du texte et en bio-informatique. Il s'agit ici de définir des structures de données compactes et permettant d'opérer efficacement [48, 36] : ce sont des automates (oracle des facteurs [2], mais aussi

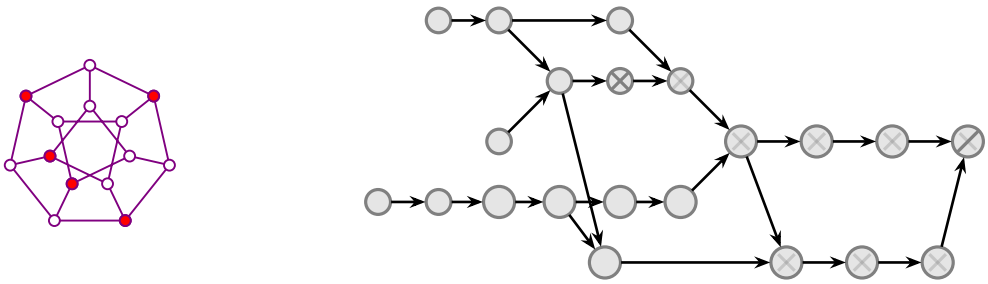


FIGURE 3.7. **À gauche** : Graphe de Peterson généralisé G14 et un ensemble dominant minimum (en rouge) ; **À droite** : Un graphe des versions d'un logiciel.

d'autres automates de types variés [15]), des arbres (tries, suffix-tries), des DAG, parfois des graphes de de Bruijn [68]. L'algorithmique reliée permet de rechercher des motifs, corriger les erreurs dans le séquençage, de passer d'une structure à une autre, etc... Les applications sont nombreuses : l'exemple de la bio-informatique est développé à la Section 3.2.2.

Extensions possibles. On cherche aussi à étendre les problématiques bien maîtrisées sur les graphes à certaines de leurs généralisations (graphes dynamiques, hypergraphes). Pour les graphes dynamiques, c'est un lien fort avec la Section 3.1.3 et potentiellement la Section 3.1.4. Pour les hypergraphes, le calcul de toutes les traverses minimales d'un hypergraphe est un problème qui apparaît dans de nombreux domaines, en particulier en fouille de données. On recherche un algorithme efficace qui résolve la question.

3.2 Quelques domaines d'application phare

3.2.1 Analyse de jeux de molécules

Les molécules se prêtent particulièrement bien à une représentation par graphes. En effet, un molécule peut, en première approche, être décrite par un ensemble d'atomes (carbone, oxygène,...) reliés par des liaisons (simple, double, triples, aromatiques,...). La prédiction des propriétés chimiques, physiques ou biologiques des molécules est un champs disciplinaire à part entière nommé la *chémoinformatique*. Notons que la prédiction de propriétés assez spécifiques des molécules peut requérir des modélisations par graphes plus haut niveau que les simples liaisons atomiques. C'est par exemple souvent le cas pour les propriétés biologiques.

Une des clés pour analyser les propriétés curatives (ou plus simplement biologiques) des molécules consiste à générer des pharmacophores. On appelle ici *pharmacophore* un sous-graphe dont les occurrences dans le jeu de données satisfont certaines conditions exigées par l'analyse. La figure 3.8 donne un exemple d'occurrences de pharmacophore dans une molécule.

Ces conditions peuvent être des conditions exigeant un nombre minmum d'occurrences, une différence de fréquences selon les sous-groupes du jeu de données ....

Il résulte de la première étape un grand nombre de pharmacophores, plusieurs centaines de milliers. L'analyse de données s'appuie sur ces pharmacophores et sur les molécules qui les contiennent. Cette analyse est mixte : elle contient des traitements automatiques qui orientent des explorations manuelles du chimiste afin de découvrir des liens dans les résultats et de construire sa théorie chimique.

Ces explorations nécessitent de trouver une structure pour organiser les pharmacophores et les

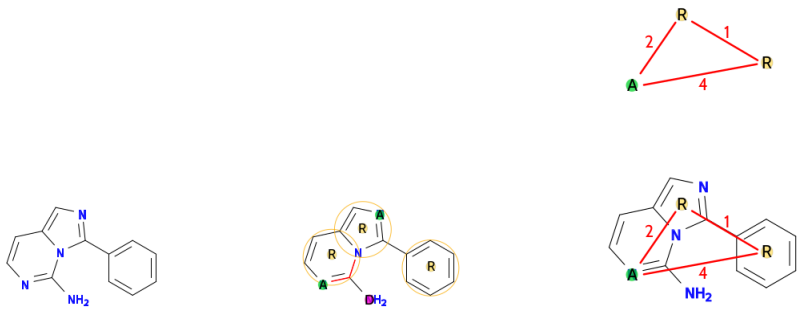


FIGURE 3.8. L'occurrence d'un pharmacophore dans une molécule

molécules. Dans ce but, il s'agit souvent de s'appuyer sur la relation d'inclusion, produisant ainsi des diagrammes de Hasse.

Ces diagrammes sont de très grands graphes orientés donc chaque sommet représente un élément de l'analyse (un pharmacophore) et chaque arête un relation d'inclusion. La figure 3.9 montre un exemple de diagramme de Hasse.

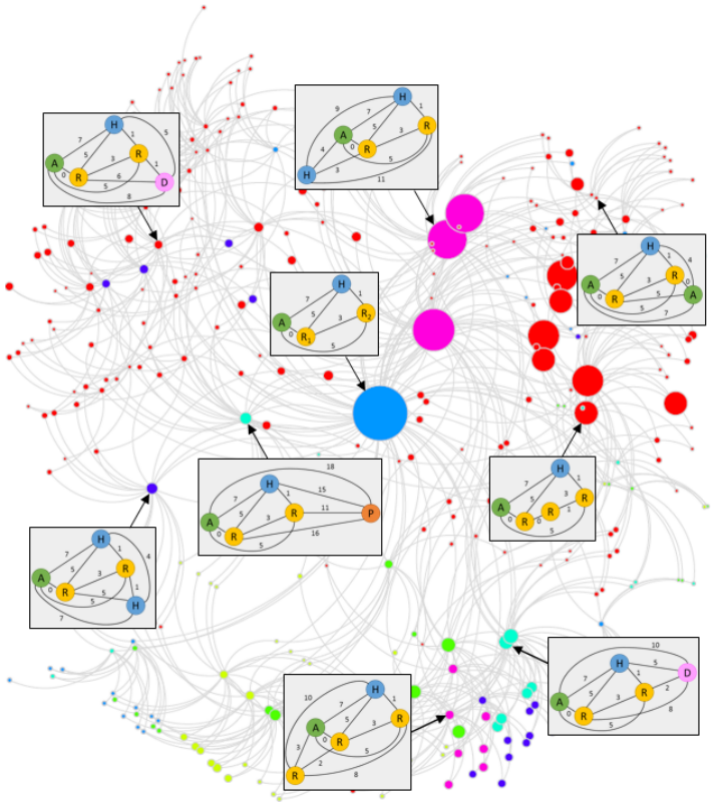


FIGURE 3.9. Diagramme de Hasse

Une autre approche, plus orientée apprentissage machine consiste à prédire les propriétés biologiques mais également physiques ou chimiques des molécules soit au travers de réseaux de neurones sur graphes soit à partir de mesures de similarité ou de distance entre les molécules. Dans ces derniers cas, on utilise soit des méthodes à noyaux soit la distance d'édition sur graphes.

3.2.2 Utilisation des graphes en bio informatique

Les travaux développés en bio-informatique visent à rechercher, indexer et extraire des informations pertinentes notamment dans des données biologiques (génomes et expression des génomes). Dans ce but il s'agit de définir des structures de données qui minimisent l'espace mémoire et permettent des traitements efficaces (e.g. indexation, recherche de motifs).

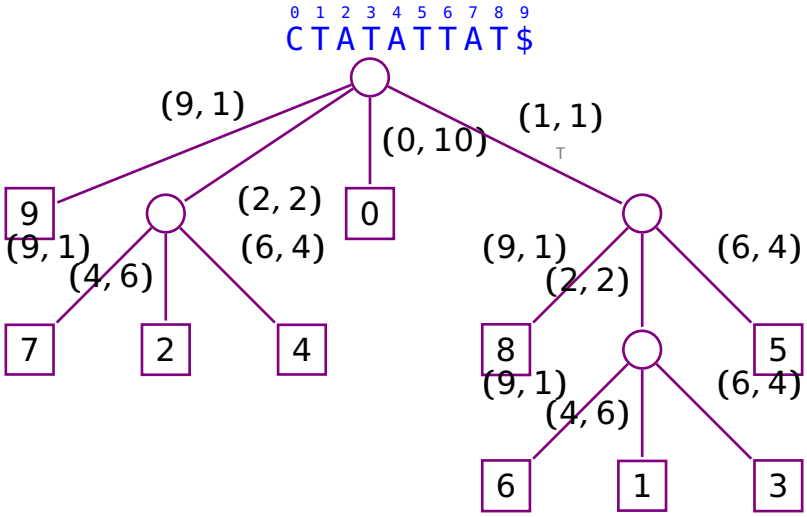


FIGURE 3.10. L'arbre des suffixes de CTATATTAT$. Les nœuds internes sont représentés par des cercles alors que les feuilles sont représentées par des carrés et contiennent la position de début des suffixes auxquels elles sont associées. Tous les nœuds internes ont au moins deux descendants. Les arcs sont étiquetés par des couples (i, ℓ) correspondant au facteur commençant en position i de longueur ℓ. Tout chemin de la racine à une feuille correspond à un suffixe du texte. Par exemple le chemin de la racine à la feuille 4 correspond au suffixe qui commence en position 4, soit ATTAT$ qu'on peut lire sur l'arbre avec l'étiquette (2,2) = AT et l'étiquette (6, 4) = TAT$.

Pour cela différents types de graphes sont utilisés, comme des arbres (e.g. arbre compact des suffixes, Figure 3.10). Cette structure s'avère encore trop gourmande en espace pour de longues séquences. Il existe plusieurs méthodes pour encore réduire l'espace nécessaire. L'une d'elle, l'oracle des facteurs [3] est un automate fini déterministe qui reconnait un peu plus que l'ensemble des facteurs d'une séquence. Par ailleurs, les génomes ne peuvent pas être séquencés en un seul morceau. Ils sont donc fragmentés, les fragments sont ensuite amplifiés et enfin séquencés. À l'aide des multiples copies des fragments générés il faut alors reconstruire le génome d'origine. Pour cela deux types de graphes sont principalement utilisés : les graphes de chevauchement et les graphes de de Bruijn [33]. Notons toutefois qu'à partir d'une certaine taille de données, les graphes ne sont plus la structure de donnée la plus pertinente. Des structures de données alternatives, souvent basées sur la transformée de Burrows-Wheeler permettent de remplacer avantageusement ces derniers.

3.2.3 Analyse d'évènements sportifs

L'analyse d'un jeu collectif (football, handball,...) se base principalement sur les positions relatives des joueurs, d'un éventuel ballon et leurs positions vis à vis du terrain. Tout ceci peut être modélisé par un graphe qui sera nécessairement dynamique du fait de l'évolution des positions des joueurs au cours du temps. Il s'agit dans ce cadre de détecter des motifs récurrents (au cours du temps) caractéristiques de réussites tactiques. Les sommets de tels graphes peuvent être simplement les différents éléments sus-cités ou un diagramme de Voronoï utilisant les positions des joueurs et du ballon comme centres (Figure 3.11). Les arêtes du graphe peuvent être valuées par une distance, une expérimentation sur la possibilité d'inclure des relations spatiales qualitatives a aussi été envisagée.

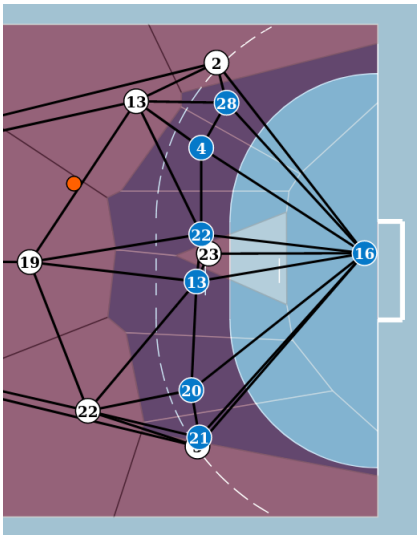


FIGURE 3.11. Une situation de handball avec diagramme de Voronoï et triangulation de Delaunay

3.2.4 Logistique et mobilité

Les réseaux viaires se prêtent particulièrement à la modélisation par graphe. Différentes versions existent comme de prendre les intersections pour les sommets du graphe et les tronçons routier entre ces intersections pour les arêtes du graphe, la réciproque est aussi utilisée. Ces graphes peuvent tenir compte de l'orientation du réseau, de la présence de plusieurs voies (multi-graphe) etc. La valuation des arêtes peut par exemple représenter une distance, un temps de parcours, une vitesse...

Ces réseaux peuvent être étudiés d'un point de vue statique ou dynamique (Figure 3.12). Dans ce second cas, on tient compte par exemple du déplacement des véhicules sur ce réseau. Les problématique de logistique, de gestion portuaire, détection de patterns de mobilité font partie des thèmes qui utilisent ce type de modélisation et peuvent être couplée à des simulations multi-agents.

Une tout autre type de modélisation par graphe peut également être utilisé, il s'agit de représenter les acteurs au sein de la gestion logistique d'un système comme les sommets d'un graphe statique, où les arêtes caractérisent les relations entre ces acteurs. L'objectif est ici de caractériser la gouvernance au sein d'une place portuaire et de pouvoir comparer plusieurs places portuaires par la comparaison des graphes sous-jacents.

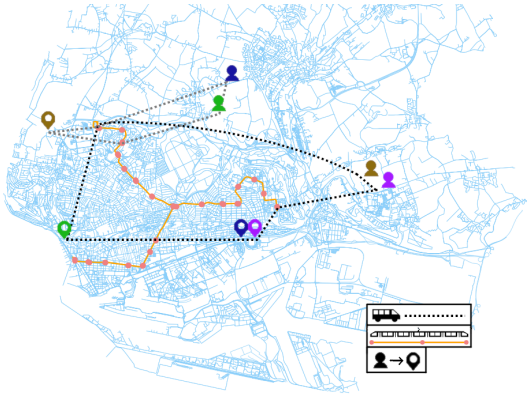
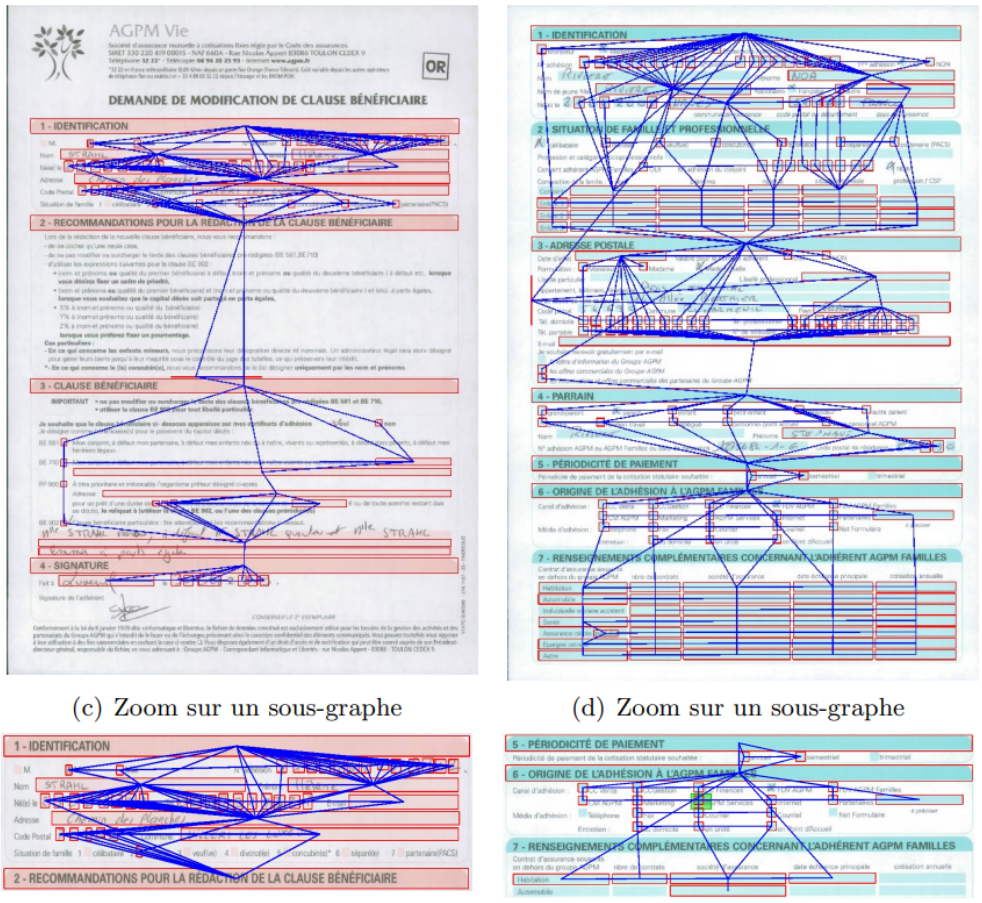


FIGURE 3.12. Illustration des graphes dynamique en mobilité - Transport de personnes [72]

3.2.5 Analyse de documents

Un document fait par des êtres humains est généralement fortement structuré, Cette structuration spatiale correspond à une structure logique ou sémantique sous-jacente. Le codage des relations phy-

(a) Graphe d'un document complet (b) Graphe d'un document complet



(c) Zoom sur un sous-graphe (d) Zoom sur un sous-graphe

FIGURE 3.13. Structuration d'un document administratif

siques par un graphe permet donc indirectement de capturer de l'information sémantique (Figure 3.13).

Dans le cas de lettres ou de documents administratifs, les champs tels que l'adresse, l'objet, le destinataire, la présence de tableaux… peuvent être des éléments discriminants. La détection de ces éléments et leur mise en relation par un graphe permet une première indexation des documents. Notons, que malgré la numérisation, les administrations continuent à recevoir des centaines de milliers de courrier chaque jour.

De même, dans le cadre de l'analyse de plans, les éléments de ces plans (portes, tables, murs,…) peuvent être décomposés en plusieurs parties qui auront sensiblement les mêmes relations d'adjacence dans différents documents. Le codage de ces éléments par des graphes et la recherche de sous graphes dans les plans est une technique couramment utilisée pour détecter toutes les occurrences d'un ou plusieurs objets (e.x. une table contre un mur). Ce type de recherche se nomme du pattern spotting.

Enfin, dans le cadre de l'analyse de documents anciens, les images peintes ou dessinées présentent également une forte structure qui peut être utilisée, via un codage par graphe pour opérer du pattern spotting afin de détecter leur occurences dans des banques de documents.

Chapitre 4 Perspectives et opportunités

L'objectif de cette dernière partie est de poursuivre l'analyse commencée dans le chapitre précédent afin de mettre en perspective les différentes approches. Trois principaux courants émergent de cette réflexion : **Graphes & Informatique Mathématique**, **Graphes & masse de données** et **Graphes dynamiques & sémantiques** (Figure 4.1). L'idée est de les présenter au travers des complémentarités des approches qui les composent, et également de montrer les liens qui existent entre eux. Nous soulignerons aussi certaines différences car nous pensons qu'elles peuvent permettent de mieux cerner les différentes approches.

En complément à cette confrontation méthodologique, nous présenterons brièvement deux indicateurs : les **GDR** et les **lieux de publication** sous-jacents aux travaux présentés. Ces deux derniers éléments permettent de souligner que de nombreux membres de la fédération évoluent dans des milieux proches voire commun et qu'il est souvent possible d'envisager des publications dans des revues ou conférences qui satisfassent les différents partis.

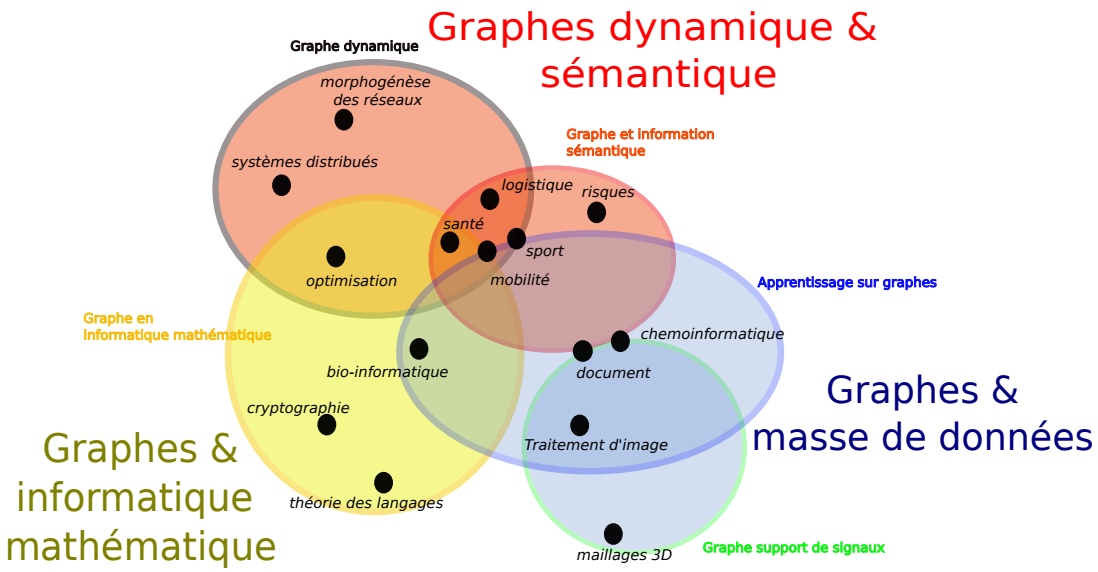


FIGURE 4.1. Trois principaux courants autour des graphes identifiés au sein de la fédération NormaSTIC.

4.1 Les graphes dans NormaSTIC

L'interdisciplinarité est au coeur des enjeux de nombreux travaux, mais particulièrement en ce qui concerne les approches relatives aux GRAPHES DYNAMIQUES et SÉMANTISÉS. Certains des verrous scientifiques (e.g. adaptation d'algorithmes/concepts sur des graphes statiques aux graphes dynamiques) et les stratégies de résolution sont proches et complémentaires (e.g. algorithmique sur graphe, optimisation). Si les GRAPHES SÉMANTISÉS n'excluent pas de travailler sur des structures statiques, la dimension temporelle est très souvent nécessaire pour modéliser les phénomènes sur lesquels porteront les analyses, et en cela les GRAPHES DYNAMIQUES sont d'un réel apport pour ces approches. Réciproquement, actuellement les GRAPHES DYNAMIQUES utilisent très peu l'information sémantique pour valuer les

graphes étudiés, a fortiori la dimension qualitative de cette information est sous-exploitée. Par ailleurs, de nombreuses applications (e.g. logistique, sport) sont d'ors et déjà sujets à des travaux communs ce qui démontre des liens forts entre ces deux visions des graphes.

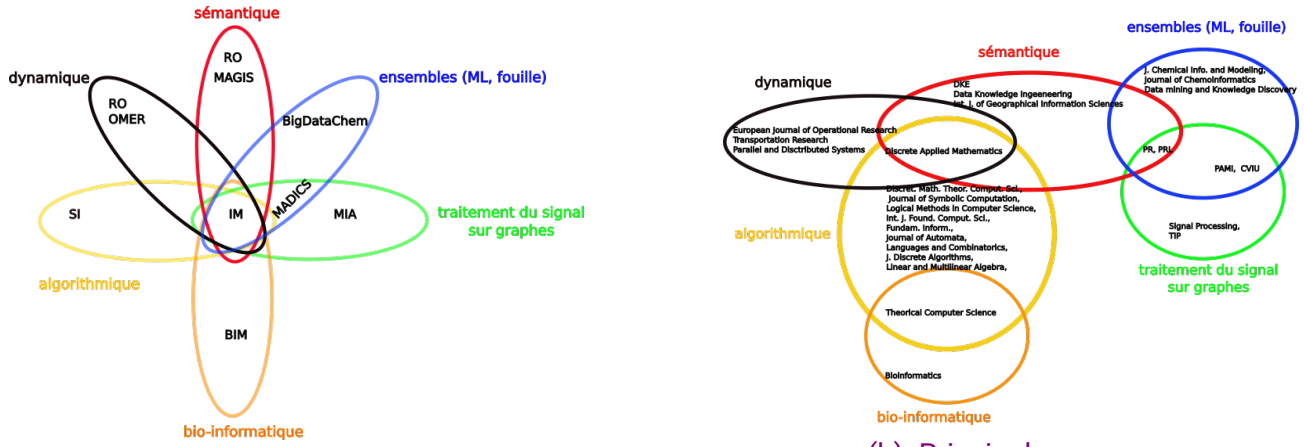
Le courant des **graphes dynamiques & sémantiques** a régulièrement besoin de s'intéresser aux propriétés structurelles des graphes et à l'optimisation des algorithmes appliqués à ces graphes. Ces deux enjeux peuvent être adressés au courant **graphes & informatique mathématique** qui traite de problématiques complémentaires voire communes. Les **graphes dynamiques & sémantiques** ont aussi des besoins d'adaptation de métriques classiques sur des graphe qui peuvent être travaillés selon des points de vue très généraux (sans application) comme par exemple des calcul de densité sur des hypergraphes orientés. D'autres défis, comme l'adaptation de la notion de chemin dans les GRAPHES DYNAMIQUES ou d'autres problématiques de recherche opérationnelle peuvent accueillir l'expertise du courant **graphes & informatique mathématique**. Réciproquement, le courant des **graphes dynamiques & sémantiques** peut apporter d'autres sources d'applications que celles déjà traitées par les **graphes & informatique mathématique** (e.g. bioinformatique) et une autre vision méthodologique de l'approche des graphes. On note déjà des travaux communs entre ces deux courants (e.g. DyNet, stage NormaSTIC) mais le potentiel semble là pour aller plus loin dans cette démarche.

Des problématiques d'isomorphismes de graphes et de comparaison de graphes en général sont très présentes au sein du courant **graphes & masses de données**. Elles le sont aussi dans les approches précédemment décrites de manière plus ou moins transposables. L'approche GRAPHES ET APPRENTISSAGE et le courant **graphes dynamiques & sémantiques** peuvent aisément se comprendre autour de la comparaison de graphes (e.g. travaux en mobilité [70] en cours de soumission). Notamment dans le cadre de la recherche de patterns dans un graphe (potentiellement dynamique), la recherche d'occurences etc. Dans ce cadre, il serait intéressant de pouvoir tester les méthodes utilisées dans l'approche GRAPHES ET APPRENTISSAGE sous réserve du seuil critique de données nécessaires pour que ces méthodes fonctionnent. Par ailleurs, pour faire le lien avec le courant **graphes & informatique mathématique**, notons que de nombreuses méthodes de comparaison de graphes sont dérivées des algorithmes A^*. D'autres formulations se basent soit sur de la programmation par contraintes, soit reformulent le problème comme un problème de minimisation en nombre entier sous contrainte. Par ailleurs, toutes les contraintes liées au temps de calcul, complexité, et au stockage de l'information, qui sont des problématiques adressées par le courant **graphes & informatique mathématique** sont au coeur des préoccupations des deux autres courants.

L'extension du TRAITEMENT DE SIGNAL AUX GRAPHES fournit un support théorique aux convolutions ainsi qu'aux représentations multi-échelles de signaux sur graphes. Ces deux thématiques scientifiques sont au coeur des problématiques des réseaux de neurones sur graphes abordés dans GRAPHES ET APPRENTISSAGE.Il existe donc une voie de convergence entre ces deux approches combinant un monde de graphes (GRAPHES & APPRENTISSAGE) et un monde codé par un graphe (TRAITEMENT DU SIGNAL SUR GRAPHES). Les deux approches peuvent également converger autour de pré traitement à apporter aux graphes avant analyse. Il serait par exemple intéressant d'étudier des méthodes de segmentation de document à partir de graines en utilisant des équations aux différences partielles sur graphes.

4.2 Indicateurs complémentaires : GDR et lieux de publication

Nous pouvons constater sur la Figure 4.2a que chaque approche est organisée autour d'un ou plusieurs GDR. Toutefois, toutes les approches se sentent concernées par les problématiques du GdR



(a). Principaux GDR (b). Principales revues

FIGURE 4.2. Principaux GDR(a) et revues(b) des différentes approches

IM. Cela confirme le rôle central de l'informatique mathématiques dans l'analyse et le traitement de graphes. Notons également que les approches GRAPHES & APPRENTISSAGE et TRAITEMENT DU SIGNAL SUR GRAPHES se retrouvent toutes deux au sein des problématiques du GdR MADICS (Masse de données). Cela confirme les possibles convergences soulignées dans la section précédente.

Les principales revues dans lesquelles publient les approches que nous avons identifiées (Figure 4.2b) présentent un aspect moins en étoile que celui des GdR et comportent plusieurs points d'articulation :

Discrete Applied Mathematics : occupe une place centrale puisqu'il établi un pont entre **Graphes dynamiques & sémantiques** et **Graphes et informatique mathématique**.

Pattern Recognition (PR) et Pattern Recognition Letters(PRL) : se retrouvent dans trois approches (GRAPHES SÉMANTIQUES, ML SUR GRAPHES, TRAITEMENT DU SIGNAL SUR GRAPHES). Ces deux journaux font donc le lien entre **Graphes et masses de données** et **graphes dynamiques et sémantiques**.

Computer Vision and Image Understanding (CVIU) et Pattern Analysis and Machine Intelligence (PAMI) : Ces journaux représentent une intersection complémentaire entre ML SUR GRAPHES et TRAITEMENT DU SIGNAL SUR GRAPHES plus axée sur des problématiques analyse d'image.

4.3 Une petite conclusion

L'objectif de ce document était de présenter une première cartographie des différentes variations de graphes au sein de la fédération NormaSTIC et de mettre en avant les potentialités de collaboration entre ses membres. A sa lecture, il faut avoir conscience que sa rédaction a été réalisée à partir de données inévitablement non exhaustives. Par ailleurs, les rédacteurs sont forcément biaisés par leurs propres domaines de recherche.. Ceci étant, nous avons constaté que les graphes sous-jacents aux figures 3.1, 4.1 et 4.2 étaient tous connexes... nous espérons donc que l'analyse et les suggestions qui sont ressorties de cette étude permettront une meilleure vision de "la problématique graphe" au sein de la fédération et que ce travail donnera prochainement des idées de travaux communs à venir présenter dans une future journée de l'Action Transverse Graphes.

Annexes

Nous avons regroupé dans ces annexes l'ensemble des fiches de synthèse récoltées lors des entretiens dans leur état brut initial.

Chapitre A Entretien Bertrand Cuissard (GREYC - CODAG) 30 mars 2021

Analyse de données à partir de graphes moléculaires

Mots clés : Data mining, Graphes, Chemoinformatique

Problématiques scientifiques :

- *Correspondance de graphes (Graph matching)* : isomorphisme de graphes, de sous-graphes, plus grand sous-graphe commun, similarité entre graphes, distance d'édition, ...
- *fouille de données représentées par des graphes* : extraction des sous-graphes fréquents, des sous-graphes émergents

Domaines applicatifs (exemples) : chémoinformatique, drug design,

Personnes de la fédération impliquées (principaux et permanents [(1)]) :

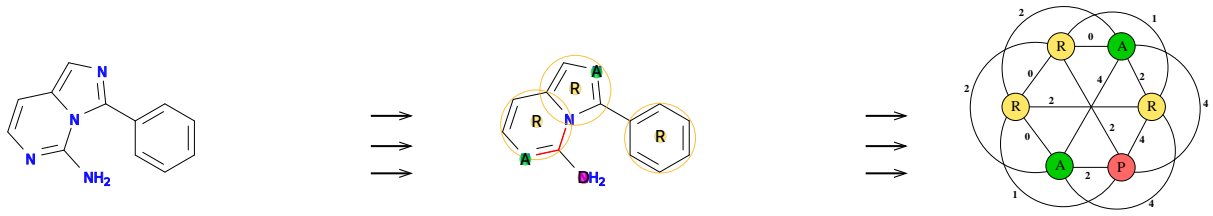
- GREYC, CoDaG : Albrecht Zimmemrmann, Abdelkader Ouali, Antoine Widlöcher, François Rioult, Bruno Crémilleux, ...
- GREYC, autres : Sébastien Bougleux, Luc Brun
- LITIS, APP : Sébastien Adam, Pierre Heroux, Benoît Gaüzère, Thierry Paquet, Laurent Heutte, Géraldine Del Mondo ; LITIS, TIBS : Thierry Lecroq, Lina Soualmia

GdR(s) concerné(s) [(2)] :GdR BigDataChim(Chémoinformatique), Gdr MADICS

Quelques revues et conférences que vous visez [(3)] : Journal of Chemical Information and Modelling ; Journal of Chemoinformatics ; Data Mining and Knowledge Discovery ; Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases

Contexte et méthodologie générale

L'équipe CoDaG est spécialisée dans le développement de méthodes informatiques pour les sciences des données. Dans ce cadre, elle s'intéresse aux graphes en général.



Formule squelettique — Occurrences de fonctions pharmacophoriques — Graphe des fonctions pharmacophoriques

FIGURE A.1. Deux modélisations d'une molécule sous la forme d'un graphe

Le graphe moléculaire Son activité s'appuie sur une participation à de nombreux projets interdisciplinaires : elle entretient une collaboration importante avec des spécialistes du traitement de l'infor-

(1). Vous pouvez citer des personnes avec lesquelles vous avez travaillé sans forcément que ça ait donné lieu à un projet ou une publication

(2). Il n'est pas nécessaire d'être actif pour citer un GdR, a fortiori parmi ISIS/IM/MADICS, précisez celui ou ceux dont vous vous rapprochez le plus

(3). Il n'est pas nécessaire d'y avoir déjà publié ;-))

mation chimique. En considérant cette partie de l'activité, les méthodes informatiques sur lesquelles nous travaillons sont dédiées à la manipulation de graphes moléculaires. Un graphe moléculaire est un graphe qui modélise une molécule. La figure A.1 montre deux modélisations d'une molécule que nous exploitons actuellement : la forme squelettique usuelle et une forme de "graphes des fonctions pharmacophoriques" beaucoup moins courante.

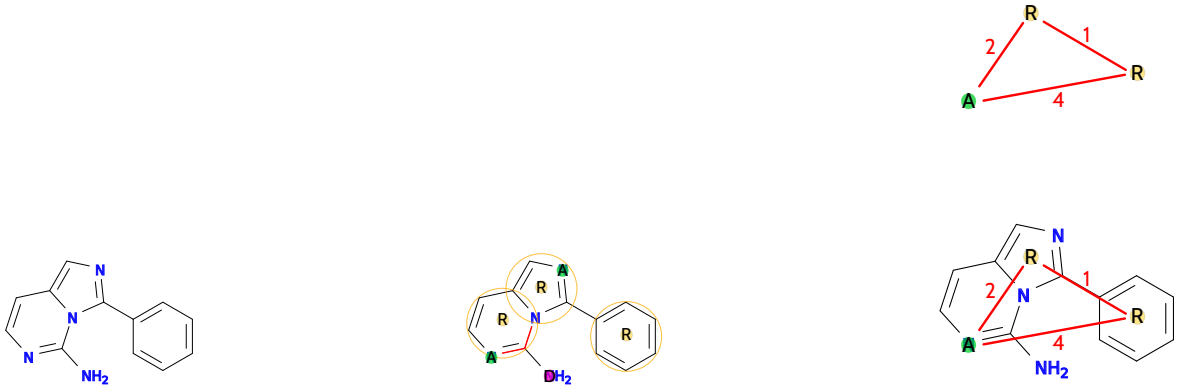


FIGURE A.2. L'occurrence d'un pharmacophore dans une molécume

Le pharmacophore : une sous-structure "remarquable" En prenant un compte un jeu de molécules, une grande partie de nos travaux consiste à générer des pharmacophores. On appelle ici *pharmacophore* un sous-graphe dont les occurrences dans le jeu de données satisfont certaines conditions exigées par l'analyse. Ces conditions peuvent être des conditions exigeant un nombre minimum d'occurrences, une différence de fréquences selon les sous-groupes du jeu de données . . .. La figure A.2 donne un exemple d'occurrences de pharmacophore dans une molécule.

Les graphes manipulés Les graphes manipulés modélisent les interactions atomiques en chimie organique : ce sont des graphes de l'ordre d'un trentaine de sommets. L'analyse de données s'appuie sur des définition de correspondance possible, ces définitions se traduisent par un étiquetage des sommets et arêtes des graphes. Les jeux de données peuvent contenir jusqu'à quelques dizaines de milliers de graphes, voire une centaine de milliers récemment.

L'analyse des pharmacophores Il résulte de la première étape un grand nombre de pharmacophores, plusieurs centaines de milliers. L'analyse de données s'appuie sur ces pharmacophores et sur les molécules qui les contiennent. Cette analyse est mixte : elle contient des traitements automatiques qui orientent des explorations manuelles du chimiste afin de découvrir des liens dans les résultats et de construire sa théorie chimique. Ces explorations nécessitent de trouver une structure pour organiser les pharmacophores et les molécules. Dans ce but, nous nous appuyons souvent sur la relation d'inclusion, produisant ainsi des diagrammes de Hasse. Ces diagrammes sont de très grands graphes orientés donc chaque sommet représente un élément de l'analyse (un pharmacophore) et chaque arête un relation d'inclusion. La figure A.3 montre un exemple de diagramme de Hasse produit avec nos outils d'analyse.

- Métivier JP, Cuissart B, Bureau R, Lepailleur A. The Pharmacophore Network : A Computational Method for Exploring Structure-Activity Relationships from a Large Chemical Data Set. J Med Chem. 2018 Apr 26 ;61(8) :3551-3564. doi : 10.1021/acs.jmedchem.7b01890. Epub 2018 Apr 18. PMID : 29648816.

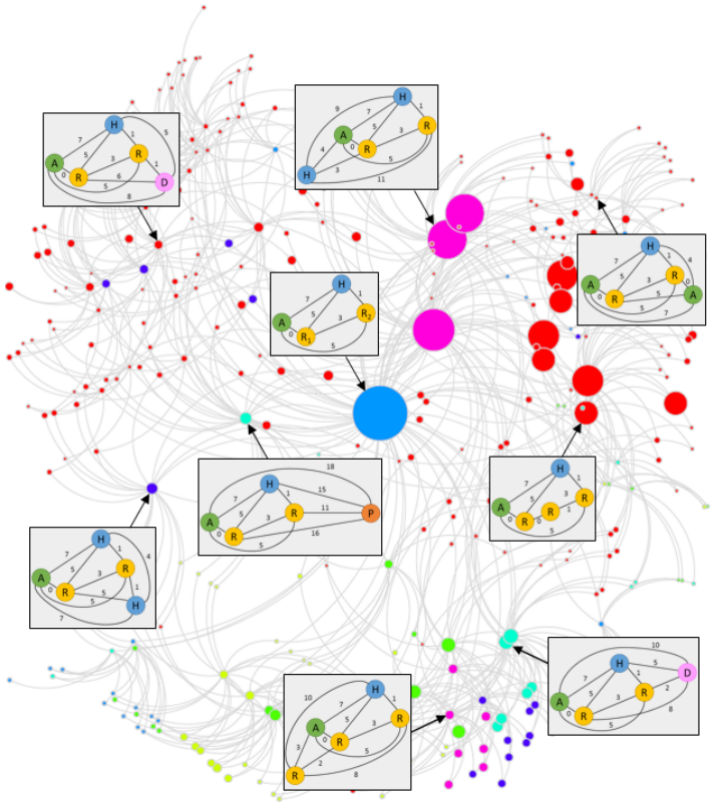


FIGURE A.3. Diagramme de Hasse

Chapitre B Entretien Olivier LÉZORAY (GREYC - IMAGE) 2021

Mots clés : Traitement de signaux sur graphe, Morphologie mathématique multivariée, Photographie computationnelle, Apprentissage profond et réseaux de neurones à convolution sur graphes.

Problématiques scientifiques :

- Traitement de signaux sur graphes : Je m'intéresse à la formulation de modèles et algorithmes de traitement et d'analyse de données de grandes dimensions qui résident sur les noeuds ou les arêtes d'un graphe, et que l'on nomme *signaux sur graphes*. Les graphes représentent des données non nécessairement Euclidiennes et peuvent être dirigés. Les problématiques visées concernent : l'expression de p-Laplaciens sur graphes dirigés et leur exploitation pour la régularisation adaptative de signaux, la morphologie mathématique par EDP sur graphes, la propagation de fronts via l'équation Eikonale sur graphes, les contours actifs sur graphes, la décomposition hiérarchique de signaux sur graphes, la représentation multi-résolution de signaux sur graphes, l'estimation de la saillance pour les maillages 3D, l'estimation de la qualité (avec ou sans référence) des maillages 3D, le filtrage spatiallement variable avec préservation de structure de signaux sur graphes.
- Morphologie Mathématique Multivariée : La construction d'opérateurs morphologiques algébriques repose sur des treillis complets, ce qui impose de disposer d'une relation d'ordre entre les éléments à traiter. Cela pose problème pour le traitement de données multivariées. J'ai proposé plusieurs formulations permettant l'extension de la morphologie mathématique pour des données multivariées qui exploitent des représentations du signal associées à une transformée de rang : une première approche supervisée qui repose sur l'apprentissage d'un treillis complet par apprentissage de variété dans le domaine spectral associé au graphe, une seconde approche par construction d'un chemin Hamiltonien sur le graphe du signal.
- Photographie computationnelle : Je m'intéresse plus particulièrement au développement de formalismes et algorithmes permettant de manipuler le contenu de données 2D ou 3D sur graphes pour les améliorer ou les manipuler visuellement : extraction interactive d'objets dans des données 2D/3D, rehaussement de détails, débruitage, inpainting, etc.
- Apprentissage profond et réseaux de neurones à convolution sur graphes : récemment je me suis intéressé à la généralisation des RNC à des graphes (GCNN) pour des tâches de classification.

Domaines applicatifs (exemples) : Applications en imagerie médicale, affective computing, réalité virtuelle, sécurité dans les transports.

- imagerie médicale : traitement et analyse de grandes images microscopiques histologiques ou cytologiques, d'images dermoscopiques.
- affective computing : reconnaissance d'émotion dans des enregistrements vidéos multimodaux.
- réalité virtuelle : reconnaissance des gestes de la main.
- sécurité dans les transports : segmentation sémantique, ré-identification de passagers, reconnaissance d'actions dans des vidéos.

Personnes de la fédération impliquées (principaux et permanents [(1)]) : Sébastien Bougleux, Luc Brun,

(1). Vous pouvez citer des personnes avec lesquelles vous avez travaillé sans forcément que ça ait donné lieu à un projet

Christophe Charrier.

Labos/Industries impliqués en dehors de la fédération : Université Gustave Eiffel (IFSTTAR-LEOST Lille).

GdR(s) concerné(s) [(2)] : ISIS, IM, MIA, MADICS

Quelques revues et conférences que vous visez [(3)] :

- Revues : IEEE Transactions on (Image Processing, Pattern Analysis and Machine Intelligence), Signal Processing : Image Communication, Pattern Recognition Letters, Pattern Recognition, Computer Vision and Image Understanding.
- Conférences : ICIP, ICASSP, ICIP, EUSIPCO, CVPR, ECCCV, SIGGRAPH, CAIP, IJCNN, ICANN, DGMM

Ressources logicielles que vous auriez développées : Asgard (Librairie C++ templatée pour le traitement de signaux sur graphes)

Projets en cours ou passés [(4)] :

- Partenariat Hubert Curien Toubkbal Franco-Marocain, 2022-2024, "Exploitation de maillages 3D Colorés par réseaux de convolution". Partenaires : UNICAEN (coordonnateur), Université Mustapha Stambouli de Mascara en Algérie.
- Partenariat Hubert Curien Tassili Franco-Algérien 2019-2021, "dermato.ai". Partenaires : UNICAEN (coordonnateur), Université Mustapha Stambouli de Mascara en Algérie.
- HomeKeeper 2018-2020 (Créer un assistant domestique intelligent et vocal pour distribuer des biens et services, et en opérer la réalisation de bout en bout), Programme Investissements d'Avenir. Partenaires : Digital Airways (coordonnateur), 44, Casusbelli, UNA, Manche Numérique, Normastic (Laboratoires GREYC et LITIS). `https://home-keeper.io`
- ANR GRAPHSIP 2015-2019 (GRAPH SIgnal Processing), Programme ANR Fondements du numérique 2014. Partenaires : UNICAEN (coordonnateur), ENS Lyon, UPEM, Gipsa-lab. `https://graphsip.greyc.fr`
- T2PSL 2020-2022 (Traitement Perceptuel de données 3D issues du Patrimoine Saint-Lois) de l'agglomération Saint-Loise. Partenaire : UNICAEN. `https://t2psl.greyc.fr`
- IGIL 2018-2020 (Intuitive Gesture Interface Lab) de la Région Normandie/FEDER. Partenaires : ENSICAEN (coordonnateur), INSA Rouen Normandie, société Itekube. `https://www.ensicaen.fr/projet-igil-intuitive-gesture-interface-lab/`

Exemples :

ou une publication

(2). Il n'est pas nécessaire d'être actif pour citer un GdR, a fortiori parmi ISIS/IM/MADICS, précisez celui ou ceux dont vous vous rapprochez le plus

(3). Il n'est pas nécessaire d'y avoir déjà publié ;-))

(4). Nom, année et URLs si possible

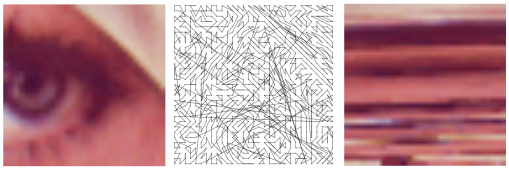


FIGURE B.1. Construction d'un treillis complet par un chemin Hamiltonien dans un graphe représentant une image.

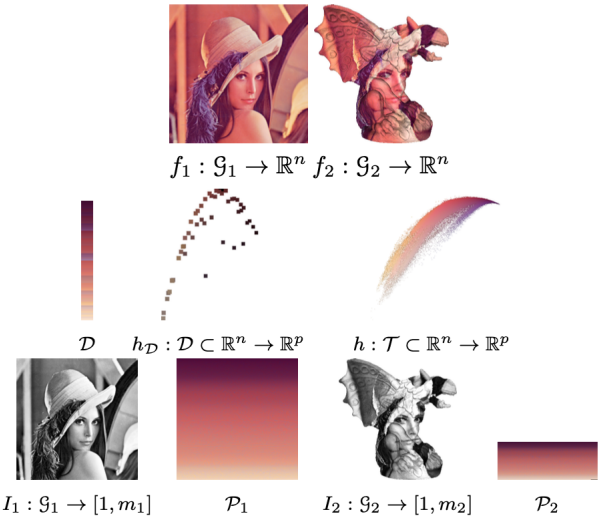


FIGURE B.2. Construction d'un treillis complet par apprentissage de variété.

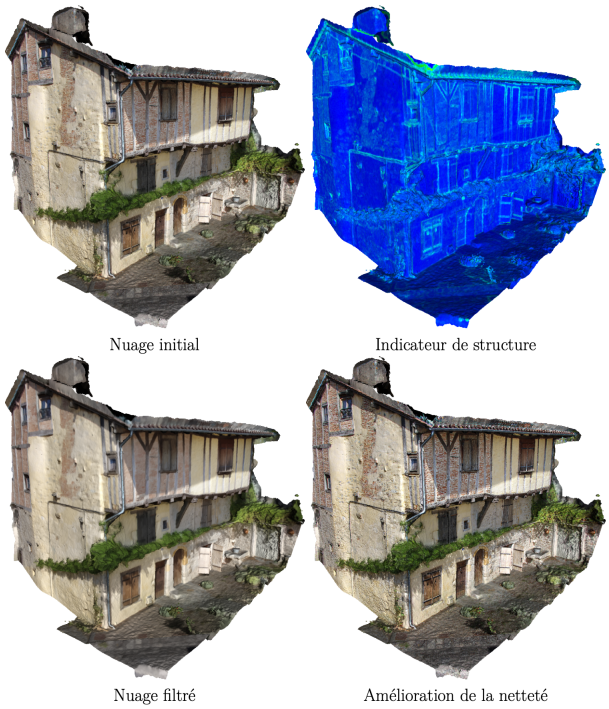


FIGURE B.3. Simplification adaptative de maillages 3D colorés pour leur rehaussement.

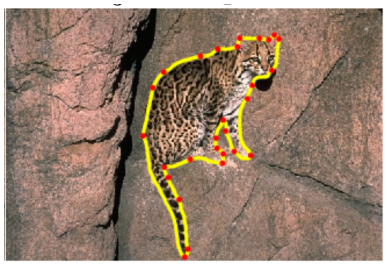


FIGURE B.4. Extraction de contours d'objets dans des images à partir de points.

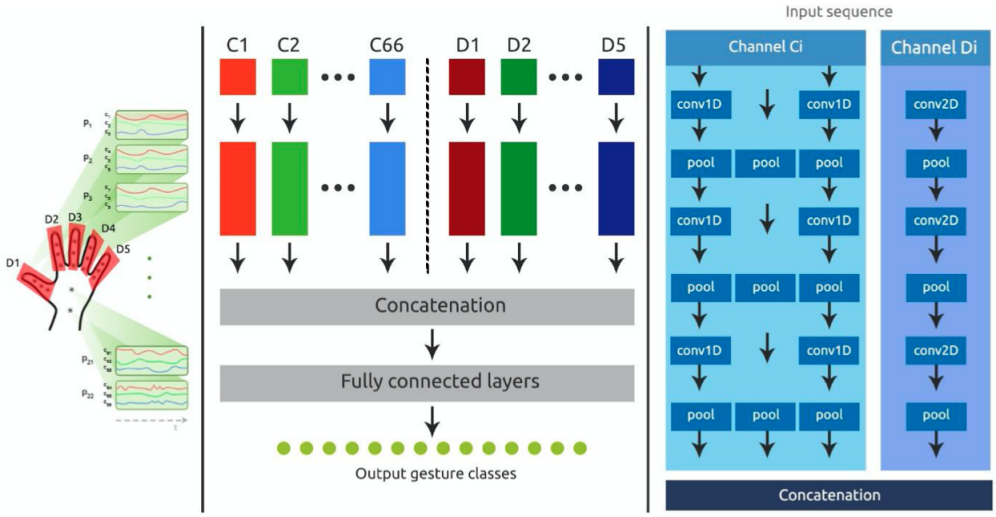


FIGURE B.5. Reconnaissance des gestes de la main.

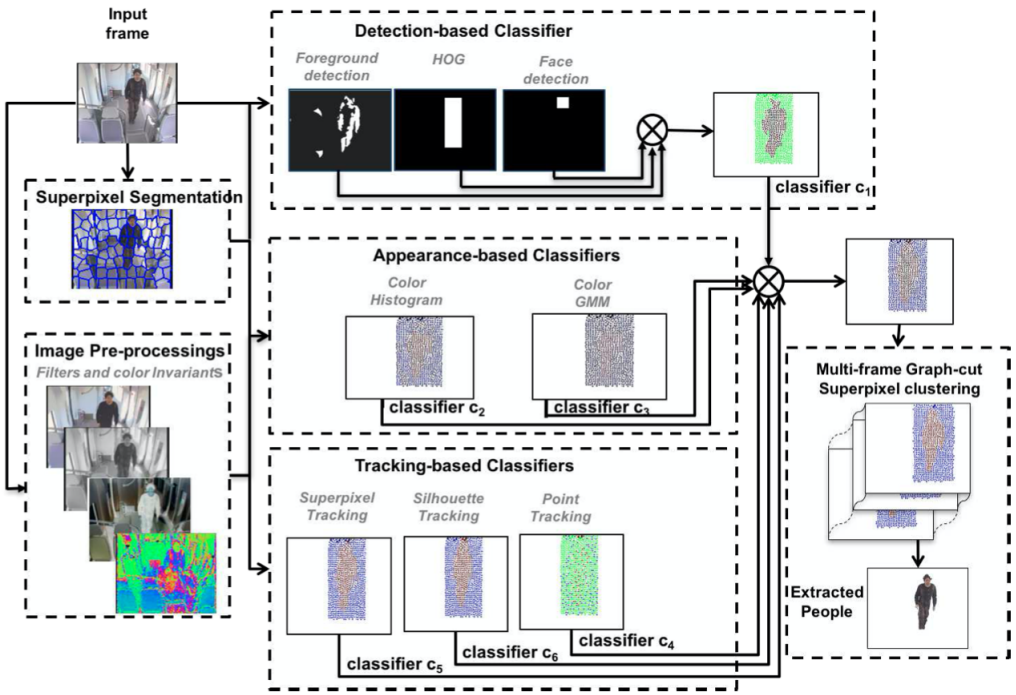
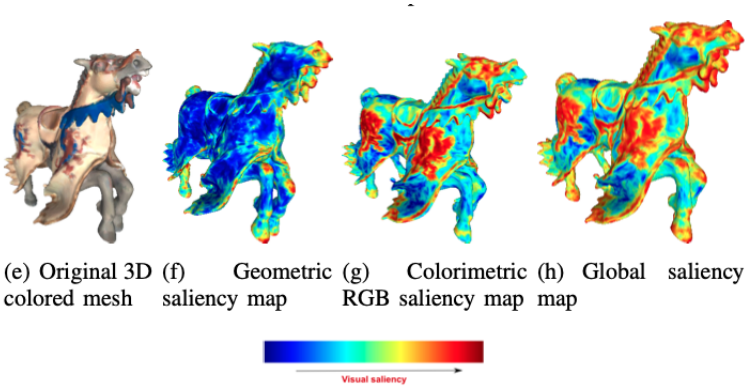


FIGURE B.6. Segmentation et suivi de personnes dans des vidéos.



(e) Original 3D colored mesh (f) Geometric saliency map (g) Colorimetric RGB saliency map (h) Global saliency map

(i) Colormap

FIGURE B.7. Estimation de la saillance de maillages 3D colorés.

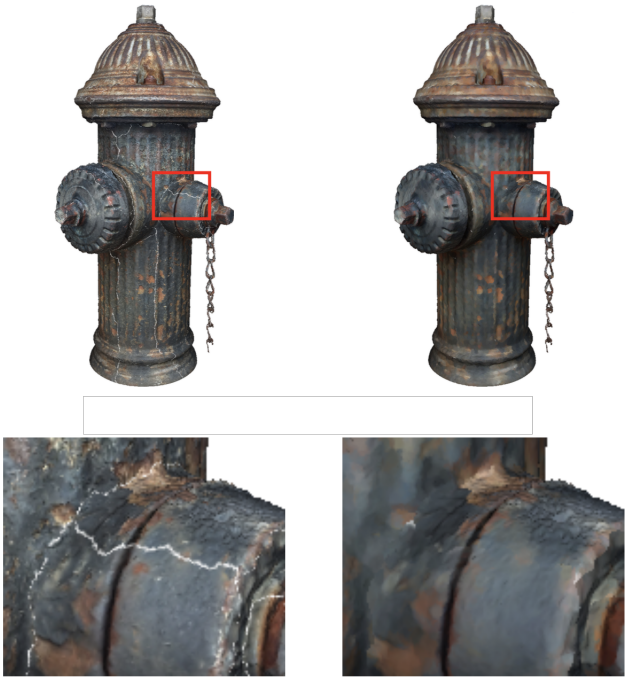


FIGURE B.8. Restauration de maillages 3D colorés.

FIGURE B.9. Contours actifs sur graphes (ici un maillage 3D coloré) pour une tâche d'édition : recolorisation.

Chapitre C Entretien Paul DORBEC (GREYC - AMACC) 20 avril 2021

Théorie et algorithmique des graphes

Mots clés : Théorie des graphes, algorithmique des graphes, combinatoire, famille de graphes, paramètre de graphes, complexité algorithmique.

Problématiques scientifiques : Étude de paramètres et d'algorithmes de graphes, y compris sur des familles de graphes particuliers. Génération et énumération de graphes contraints.

Domaines applicatifs (exemples) :

Personnes de la fédération impliquées (principaux et permanents [(1)]) : Paul Dorbec (AmacC, Greyc), Julien Clément (AmacC, Greyc), Julien Courtiel (AmacC, Greyc), Matthieu Dien (AmacC, Greyc), ...

GdR(s) concerné(s) [(2)] : IM

Quelques revues et conférences que vous visez [(3)] : STACS, Journal of Graph Theory, Journal of Combinatorial Theory, ...

Contexte et méthodologie générale

Nous avons une approche fondamentale des problématiques de graphes, et les étudions principalement comme un objet mathématique. Un graphe est un modèle (des sommets, des arêtes), mais nous n'utilisons a priori pas de propriété du graphe qui serait induite par une application ou une motivation concrète.

Trois exemples sont proposés ici de problématiques étudiées, afin d'illustrer les questions que nous nous posons.

Propriété de graphes et théorèmes/algorithmes généraux

De nombreux problèmes d'optimisation combinatoires sont désormais classiques en théorie des graphes. La plupart sont montrés NP-difficiles (coloration, domination, ensembles indépendants), mais divers algorithmes donnent des solutions satisfaisantes dans les cas courants (que l'on imagine proche des situations concrètes, sans en avoir de preuve, bien que cela puisse être argumenté...).

Nous nous intéressons à des questions qui se formuleraient ainsi : « si mon graphe a telle propriété, que puis-je en déduire sur tel paramètre ? ». La propriété sur le graphe définit une sous famille de graphes sur laquelle nous travaillons. La preuve de la propriété permet généralement de déduire un algorithme.

Deux exemples parlant sur cette problématique :

Exemple 1 : conjecture de Bruce Reed Dans un graphe, un ensemble dominant est un ensemble de sommets tels que tout sommet qui n'est pas dans l'ensemble est voisin d'un sommet de l'ensemble. En 1996, Bruce Reed [75] a prouvé que dans un graphe à n sommets dont tous les sommets ont au moins trois voisins, la taille minimum d'un ensemble dominant était au plus $\frac{3n}{8}$. Il a conjecturé que si le

(1). Vous pouvez citer des personnes avec lesquelles vous avez travaillé sans forcément que ça ait donné lieu à un projet ou une publication

(2). Il n'est pas nécessaire d'être actif pour citer un GdR, a fortiori parmi ISIS/IM/MADICS, précisez celui ou ceux dont vous vous rapprochez le plus

(3). Il n'est pas nécessaire d'y avoir déjà publié ;-))

graphe est connexe et que tous les sommets sont de degré égal à trois, alors $\lceil \frac{n}{3} \rceil$ sommets suffisent à dominer le graphe. Ce résultat n'est malheureusement pas vrai dans le cas général, mais tout résultat dans ce sens est devenu un challenge, sur lequel Paul Dorbec travaille depuis plusieurs années.

Pour prouver un tel résultat, la stratégie de preuve permet généralement d'en déduire un algorithme, simplement en interprétant les choix faits. Une approche adoptée pour attaquer ce problème consiste à identifier de petits sous-graphes que l'on peut dominer avec un nombre de sommets suffisamment petit, et les retirer du graphes. Si on trouve un ensemble de sous-graphes qui permet de décomposer tout le graphe, nous avons une preuve, et un algorithme par déconstruction du graphe. Cette approche a déjà été adoptée avec succès dans [41].

FIGURE C.1. Graphe de Peterson généralisé G_{14} et un ensemble dominant minimum (en rouge).

Exemple 2 : analyse d'un algorithme sur les graphes Un problème sur lequel nous travaillons actuellement dans le cadre de la thèse de Romain Lecoq est la question de l'algorithme gitbisect. L'outil de version git permet de gérer les versions successives d'un logiciel (*commits*). L'ensemble des versions, les différentes branches et les liens de parenté entre les versions, définissent un graphe orienté (par le temps et le lien de parenté). L'algorithme gitbisect [34] est utilisé pour chercher un point de régression dans le dépôt, ou pour simplifier un commit qui introduit un bug dans le dépôt. L'hypothèse de travail est que le bug est présent dans la dernière version du dépôt, et que les versions qui présentent le bug ont toutes un ancêtre commun, la version issue de l'introduction du bug (cf Figure C.2 Pour identifier le commit défectueux, on applique des tests au différentes versions, jusqu'à trouver la version défectueuse dont aucun parent n'est défectueux.

Le problème consistant à trouver le plus petit nombre de tests nécessaires pour identifier le bug où qu'il soit est prouvé NP-difficile [31] . L'algorithme gitbisect s'inspire de la recherche dichotomique pour proposer une méthode de recherche assez intuitive, et qui semble assez efficace. Nous avons montré que dans le cas général, cette méthode pouvait être très lente (utiliser un nombre de tests exponentiel en le nombre de requêtes optimal). Nous avons aussi montré à l'inverse que si dans le dépôt, les opérations de fusion de branches (*merge*) se limitaient à la fusion de deux branches (i.e. le graphe orienté a degré entrant au plus 2), alors gitbisect était une bonne approximation de l'optimal (au plus $\equiv 1.71$ fois le nombre de requêtes optimal, borne atteinte sur certains graphes). Nous avons aussi identifié un autre

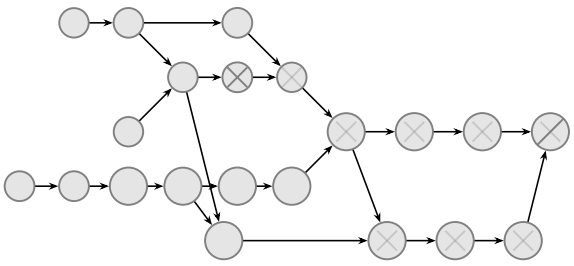


FIGURE C.2. Exemple d'un graphe représentant un dépôt. La croix noire représente le premier commit d'introduction du bug, tous les commits qui en découlent portent le bug.

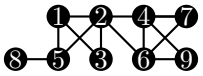


FIGURE C.3. Exemple de 1-2-arbre, étiqueté par un ordre simplicial

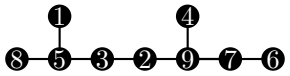


FIGURE C.4. L'arbre de Cayley correspondant

algorithme qui améliore un peu ce facteur d'approximation (pour atteindre environ 1.4 fois l'optimal au pire).

Approche combinatoire, comptage, énumération

Dans une direction assez différente, nous nous intéressons à des questions de dénombrement, de lien entre des objets différents, et de génération aléatoire, sur les objets combinatoires en général, dont les graphes.

Pour les graphes, la question se pose ainsi : étant donné une famille de graphes, combien de graphes à n sommets existe-t-il dans la famille, en fonction de n. Un travail récent avait par exemple pour objectif de compter le nombre de graphes cordaux : un graphe cordal est un graphe que l'on peut obtenir à partir d'un sommet en ajoutant successivement des sommets, de telle sorte qu'à chaque ajout d'un sommet, son voisinage dans le graphe forme une clique. Un ordre d'ajout des sommets selon ce schéma est appelé ordre simplicial.

Nous avons commencé par nous intéresser aux 1-2-arbres, les graphes cordaux obtenus à partir d'une clique de taille au plus 3, en ajoutant successivement des sommets dont le voisinage consiste en un sommet ou une arête (une clique de taille au plus deux) (cf Figure C.3). Par construction, tout graphe de largeur arborescente (*treewidth*) au plus 2 est sous-graphe d'un 1-2-arbre. En comptant ces objets, nous avons identifié une similarité avec les arbres de Cayley (comptés selon une numérotation des sommets de 1 à n, cf Figure C.4, et pu mettre en évidence une bijection explicite entre les deux familles.

Un intérêt de ce genre de résultat est de transposer les propriétés des arbres de Cayley, très étudiés, sur les 1-2-arbres. Un autre intérêt est dans la génération. Une fois qu'une famille de graphe est bien caractérisée, il est bien possible de générer de façon aléatoire uniforme un objet de cette famille, en particulier.

Graphes dynamiques : adaptation des problèmes classiques

Enfin, dans un projet commun avec Le Havre et l'équipe de Eric Sanlaville, nous avons identifié une problématique que nous souhaitons étudier finement sur les graphes dynamiques. Un graphe dynamique est défini sur un ensemble de sommets statique, et un ensemble d'arêtes, chacune munie d'un étiquetage correspondant aux intervalles de temps (dates) où l'arête est présente (la disparition d'un sommet peut simplement être modélisée par son isolement sur la fin de la période).

Certains problèmes sont très pertinents dans les graphes dynamiques. Par exemple, la notion de trajet dans un graphe dynamique consiste à trouver un ensemble de paires (arête, date) qui constitue un chemin d'un sommet à un autre. Cela représente par exemple le choix d'un itinéraire avec escale, et on peut voir ce problème comme une généralisation de la notion de chemin. Sur cette base, de nouvelles définitions de la connexité ont été proposées.

Sur d'autres problèmes, il semble difficile de définir des définitions qui enrichissent les probléma-

tiques classiques. Par exemple, Pour adapter la définition d'un ensemble indépendant aux graphes dynamiques, on peut avoir envie que l'ensemble soit indépendant à tout instant. Le problème revient alors à chercher un ensemble indépendant dans le graphe comportant toutes les arêtes qui existent à un moment ou à un autre dans le graphe, revenant au problème d'ensemble indépendant dans les graphes statiques. On observera que le problème de l'ensemble dominant n'a pas ce même problème, puisqu'un ensemble peut-être dominant sur chacune des dates sans l'être sur le graphes correspondant à l'intersection des différentes dates.

Nous souhaitons entre autre au sein de ce projet catégoriser les problèmes et les différentes adaptations possibles, pour identifier les questions intéressantes.

Chapitre D Entretien Yoann PIGNE (LITIS - RIIC) janvier 2021

Graphes dynamiques

Mots clés : Graphes dynamiques ; Intelligence Collective ; Auto-organisation ; Systèmes complexes ; Réseaux mobiles ; Algorithmique distribuée.

Problématiques scientifiques :

- Analyse des réseaux complexes et dynamiques
- Algorithmes d'optimisation
- Algorithmique des systèmes distribués

Domaines applicatifs (exemples) :

- Organisation du territoire et modélisation spatialisée
- Logistique et planification en environnement dynamiques et incertains
- Réseaux de capteurs
- Drones en essaim
- Systèmes d'information distribués
- Analyse des stratégies en sports collectifs

Personnes de la fédération impliquées (principaux et permanents [(1)]) : La notion de graphes dynamique implique toute l'équipe RI2C. En dehors de l'équipe et au sein de la fédération, les interactions autour de cette notion sont les suivantes :

- Paul DORBEC (PU, Université Normandie Caen) dans le cadre du projet (RIN Tremplin) "DynNet : Dynamic Networks". Autres chercheurs du Greyc impliqués : Julien Courtiel (CR CNRS), Julien Clément (MC, UCN), Morgan Barbier (MC, UCN) (équipe HAMAC).
- Géraldine Del Mondo (MC, INSA Normandie Rouen), Pierrick Tranouez (IGR URN), dans le cadre du projet RIN Recherche "PORTERR – Ports et Territoires".
- François Rioult (MC HdR, UCN), Albrecht Zimmerman (MC, UCN), équipe CODAG, Géraldine Del Mondo. Collaboration sur les stratégies en sports collectifs (Handball, Basketball) modélisées par des graphes dynamiques, via 3 stages co-encadrés et 2 thèses débutées en 2021 (une au GREYC RIN nd Alexis Mortelier, une au CETAPS/LITIS RIN NSH Quentin Bourgeais).

Labos/Industries/Associations impliqués en dehors de la fédération :

- UMR Sebio, Ineris, ULHN, U. Reims
- Le Havre Seine Métropole
- HAROPA PORT Le Havre
- CIDN (Centre d'Innovation Drones Normandie)

(1). Vous pouvez citer des personnes avec lesquelles vous avez travaillé sans forcément que ça ait donné lieu à un projet ou une publication

- SQUADRONE
- ORMES (office des risques majeurs de l'estuaire de la Seine)
- LaBRI
- GREAH EA 3220
- CETAPS (URN)
- NIBIO, Institut de recherche Norvégien
- Computer Science Dept, POZNAN univ of technology (Poland).

GdR(s) concerné(s) [(2)] :

- GDR IM (informatique et mathématique)
  - ❀ GT graphes
- GDR RO (recherche opérationnelle)
  - ❀ GT GT2L (transport-logistique)
- GDR OMER (océan et mers)

Quelques revues et conférences que vous visez [(3)] :

Conférences et Congrès : ROADEF, Complex Networks

Revues : European Journal of Operational Research, Discrete Applied Mathematics, Transportation Research, Parallel and Distributed Systems

Ressources logicielles que vous auriez développées :

- GraphStream (`https://graphstream-project.org`)

Projets en cours ou passés [(4)] :

- "DynNet", RIN Tremplin Région Normandie, 2020-2022
- "PorTerr", RIN Tremplin Région Normandie, 2019-2021
- SolarFarm, Prestation pour NIBIO Norvège, 2018-2021, `https://www.nibio.no/en/projects/solarfarm`

Contexte et méthodologie générale

S'il est vrai que les activités et les domaines d'applications de l'équipe RI2C sont variés, les **verrous scientifiques** que l'équipe se fixe ont, quant à eux, un contour clairement identifié autour de la notion d'analyse et de contrôle des **réseaux d'interaction**. Cette notion peut être en partie représentée par des graphes, auxquels s'ajoute une dimension temporelle qui permet de capturer la dynamique des réseaux. Ces graphes incluant l'évolution de réseaux portent de multiples noms dans la littérature : *time-varying graphs*[32], *evolving graphs*[19] *dynamic networks*[30], *temporal networks*[52], en fonction de leur domaine d'application d'origine.

On trouve principalement trois méthodologies générales d'utilisation des graphes dans l'équipe.

(2). Il n'est pas nécessaire d'être actif pour citer un GdR, a fortiori parmi ISIS/IM/MADICS, précisez celui ou ceux dont vous vous rapprochez le plus

(3). Il n'est pas nécessaire d'y avoir déjà publié ;-))

(4). Nom, année et URLs si possible

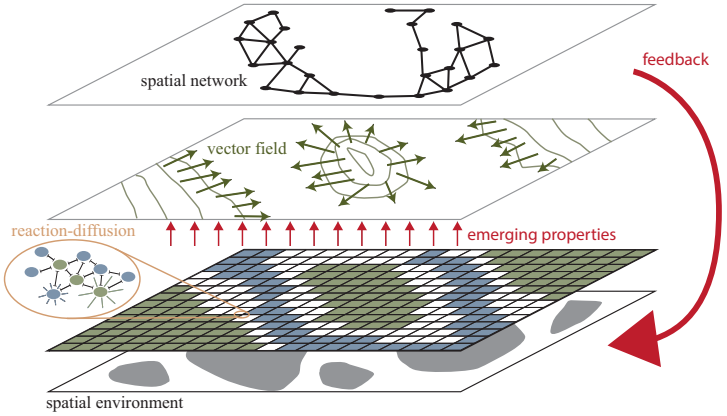


FIGURE D.1. Concept général de morphogènes qui gouvernent la formation du réseau qui rétroagit sur leur comportement.

Celles-ci correspondent aux trois axes de recherche de l'équipe et sont détaillées ici avec chacune un exemple.

En parallèle de ces axes nous avons continué le développement de GraphStream, l'outil logiciel de l'équipe sur les graphes dynamiques.

Analyse des réseaux complexes et dynamiques

Nous cherchons à comprendre la genèse de ces réseaux et de leurs structures (morphogénèse), et à la simuler à partir de modèles utilisant les graphes dynamiques. Nous caractérisons également l'émergence de propriétés globales sur ces réseaux.

Exemple 1 : Morphogénèse de réseaux urbains. Dans son travail de thèse, [84, 83] Michele Tirico aborde les aspects computationnels de la morphogénèse de réseaux complexes, en proposant un modèle général, capable de simuler leur formation. Les réseaux sont générés sous l'influence de contraintes qui s'expriment par l'intermédiaire d'un champ vectoriel qui est déterminé à l'aide d'un système de réaction-diffusion. Un modèle de Gray-Scott produisant une grande variété de motifs dynamiques est utilisé. Le champ vectoriel obtenu contrôle la géométrie et le taux de croissance du réseau construit qui rétroagit sur le processus de réaction-diffusion. La Figure D.1 illustre le processus de morphogénèse de réseaux.

Une étude a été réalisée sur l'influence des motifs et des processus de rétroaction sur la structure des réseaux obtenus en s'appuyant sur des mesures de réseaux complexes et de multi-fractalités (Fig. D.2, à gauche)).

Une démarche de validation et d'évaluation du comportement du modèle a été effectuée et appliquée en comparant les réseaux obtenus à ceux structurant des villes françaises (Fig. D.2, à droite).

Algorithmes d'optimisation

De nombreux problèmes sur les systèmes complexes réels s'expriment comme des problèmes d'optimisation, par exemple de type planification ou tournées de véhicules. Nous les résolvons à l'aide des outils issus de la recherche opérationnelle : programmation mathématique, metaheuristiques et algorithmes classiques sur les graphes, revisités pour satisfaire les exigences dues à la dynamicité et parfois à la nature distribuée des systèmes.

Exemple 2 : Composantes connexes persistantes. Ce travail fut réalisé durant la thèse de Mathilde Vernet [87, 86]. Ce travail, en cours de publication, s'intéresse à la notion de connectivité dans les

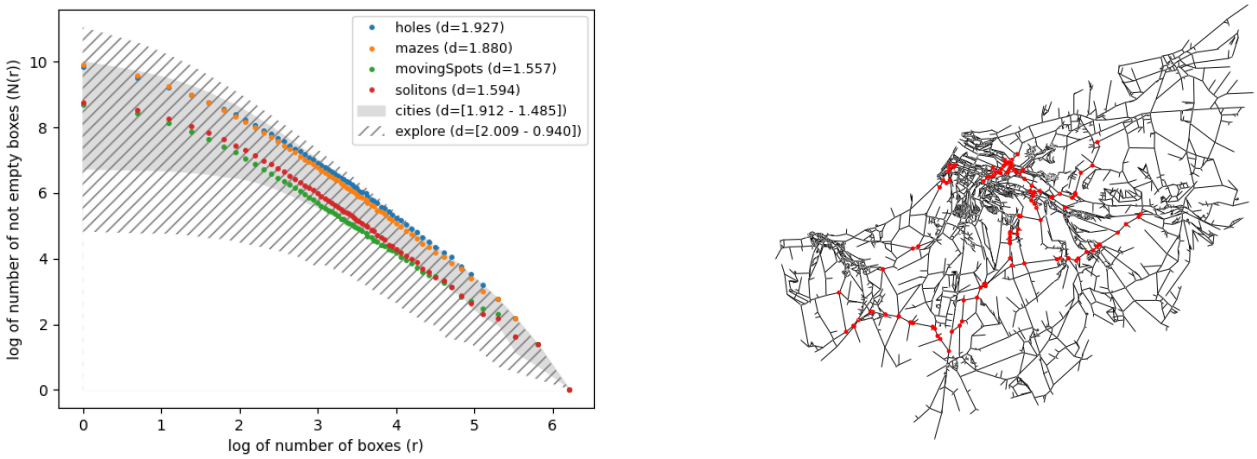


FIGURE D.2. Mesure de fractalités des réseaux générés et comparaison avec des villes réelles de référence (à gauche) ; Comparaison du modèle avec des villes réelles. Le cas de Fécamp (76). (à droite)

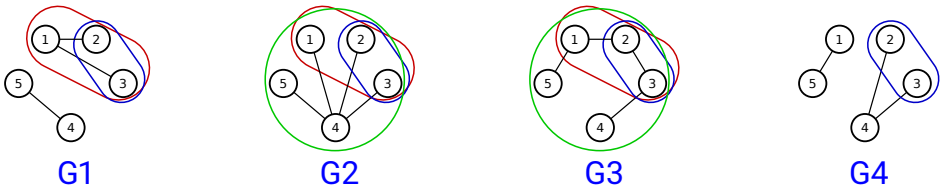


FIGURE D.3. Composantes connexes persistantes (entourées avec des cercles de couleur) dans un graphe dynamique (4 étapes de temps).

graphes dynamiques. Un graphe non orienté est définit sur un intervalle de temps fini et discret. Les arcs peuvent apparaître ou disparaître pendant l'intervalle de temps. Le premier objectif de ce travail est d'étendre la notion de composante connexe aux graphes dynamiques d'une manière nouvelle. Les composants connexes persistants sont définis par leur taille, correspondant au nombre de sommets, et leur longueur, correspondant au nombre de pas de temps consécutifs sur lesquels ils sont présents. La Fig D.3 illustre la notion de composante connexe persistante dans un graphe dynamique.

Le deuxième objectif de ce travail est de développer un algorithme calculant les plus grandes, en termes de taille et de longueur, des composantes connexes persistantes dans un graphe dynamique. L'algorithme PICCNIC (PersIstent Connected CompoNent InCremental Algorithm) est un algorithme polynomial en temps de complexité minimale. Un autre avantage de cet algorithme est qu'il fonctionne en ligne : connaître l'évolution du graphe dynamique n'est pas nécessaire pour l'exécuter. L'algorithme PICCNIC est implémenté à l'aide de la bibliothèque GraphStream [(5)] et expérimenté afin d'étudier attentivement le résultat de l'algorithme selon différents types de graphes d'entrée, ainsi que des réseaux de données réels [(6)], pour vérifier la complexité théorique et confirmer sa faisabilité pour les graphes de grande taille (Fig. D.4).

Algorithmique des systèmes distribués

Nous proposons des algorithmes permettant de piloter ces systèmes en garantissant certaines propriétés liées à la connectivité, la sécurité, l'économie d'énergie, le maintien de structures géométriques.

Exemple 3 : Vol Autonome de formations de drones. La volonté générale de cet axe de recherches est le permettre le vol d'une formation autonome de drones uniquement guidés par leurs perceptions

(5). https://graphstream-project.org/
(6). http://snap.stanford.edu/data/#temporal

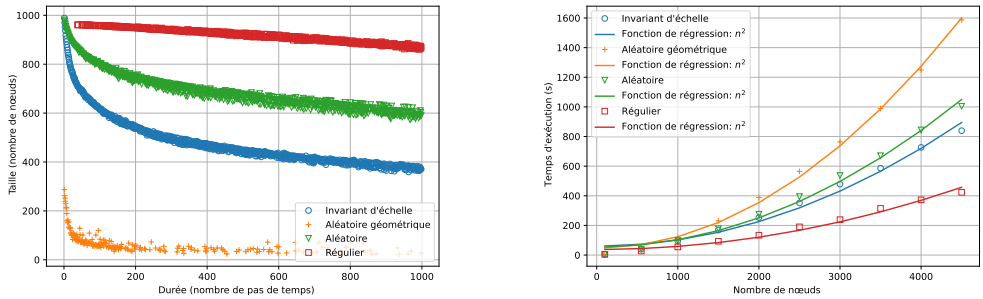


FIGURE D.4. Résultats sous forme de fronts de Pareto (à gauche) et validation de la complexité ($O(n^2\dot{T})$) (à droite)

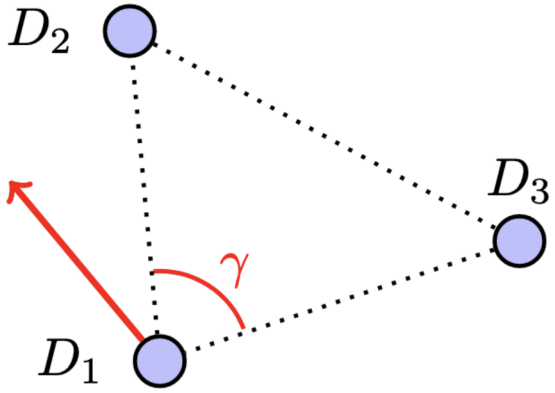


FIGURE D.5. Position arbitraire de trois robots. Chaque robot mesure sa distance angulaire avec ses voisins.

respectives et leurs interactions respectives, sans nécessiter l'intervention d'un guidage centralisé. Au delà du challenge technologique, le projet fait appel aux notions d'algorithmiques distribuées et à la modélisation par graphes dynamiques. Une vidéo de vulgarisation présente les grandes lignes du projet et ses perspectives [(7)]

Dans [47] par exemple, les auteurs s'intéressent à l'alignement autonome de robots sans communication ni information de placement. Seule la vision des robots est utilisée pour évaluer les positions angulaires des voisins (Fig. D.5).

(7). https://webtv.normandie-univ.fr/permalink/v1261be518d742lj61vh/iframe/

Chapitre E Entretien Thierry Lecroq (LITIS - TIBS) 19 mai 2021

Mots clés : trie, arbres des suffixes, automate, graphe de de Bruijn, transformée de Burrows-Wheeler, indexation, recherche de mot

Problématiques scientifiques : L'algorithmique du texte vise à développer des structures de données et des algorithmes efficaces pour manipuler des suites de symboles (appelées mots, textes ou séquences). L'aspect sémantique sous-jacent à ces suites de symboles est rarement pris en compte. En algorithmique du texte, plusieurs types de graphes sont utilisés.

Pour la recherche de motifs, on utilise fréquemment des arbres (graphes sans cycles). Par exemple, l'algorithme de recherche des occurrences d'un ensemble fini de mots d'Aho-Corasick [1] utilise un *trie* (mot provenant du terme anglais *information re**trie**val*) qui permet de représenter l'ensemble fini de mots en factorisant les préfixes communs (voir Figure E.1).

Lorsqu'il s'agit d'indexer tous les facteurs d'une séquence, ce type de *trie* s'avère trop gourmand en espace. Il est possible de réduire l'espace nécessaire en considérant l'arbre compact des suffixes [88] obtenu à partir du *trie* des suffixes en ne gardant que les nœuds internes de degré sortant au moins égal à 2 et les feuilles, en ayant pris soin d'ajouter un terminateur à la fin de la séquence de manière à identifier les feuilles et les suffixes de la séquence (voir Figure. E.2).

Cet structure s'avère encore trop gourmande en espace pour de longues séquences. Il existe plusieurs méthodes pour encore réduire l'espace nécessaire. L'une d'elle s'appelle l'oracle des facteurs [3], c'est un automate fini déterministe qui reconnaît un peu plus que l'ensemble des facteurs d'une séquence (voir Figure E.3). Des membres de l'équipe ont développé une méthode rapide pour calculer de longues répétitions dans la séquence d'origine sur cette structure [54].

Les génomes ne peuvent pas être séquencés en un seul morceau. Ils sont donc fragmentés, les fragments sont ensuite amplifiés et enfin séquencés. À l'aide des multiples copies des fragments générés il faut alors reconstruire le génome d'origine. Pour cela deux types de graphes sont principalement utilisés : les graphes de chevauchement et les graphes de de Bruijn. Pour les graphes de chevauchement, les sommets sont les fragments et il existe un arc entre deux sommets u et v si un suffixe de u est égal à un préfixe de v. L'arc est généralement valué par la longueur du chevauchement. Pour les graphe de de Bruijn (d'ordre k), les sommets sont les k-mers des fragments et il existe un arc entre deux sommets u et v si le suffixe de u de longueur $k-1$ est égal au préfixe de v. de longueur $k-1$. L'arc peut être étiqueté par la dernière lettre de v (voir Figure E.4).

Nous avons établi un lien entre les arbres compacts des suffixes et les graphes de de Bruijn [33].

L'une des problématiques qui a émergé avec l'accumulation des données (notamment avec le séquençage à haut débit) consiste à trouver des alternatives économique à ces graphes qui sont trop volumineux. La transformée de Burrows-Wheeler en est une. En effet, couplée à un tableau des suffixes échantillonné, il est possible de retrouver sensiblement les mêmes informations que dans l'arbre des suffixes compact avec un espace moindre que la séquence d'origine. Cette structure s'appelle un FM-index (voir Figure E.5).

Domaines applicatifs (exemples) :

Les *tries* et les automates sont largement utilisés pour la recherche de motifs [36].

L'oracle des facteurs a été utilisés pour détecter des répétitions dans les séquences génomiques [56], pour la compression de texte sans perte [55] et pour la génération de musique [8].

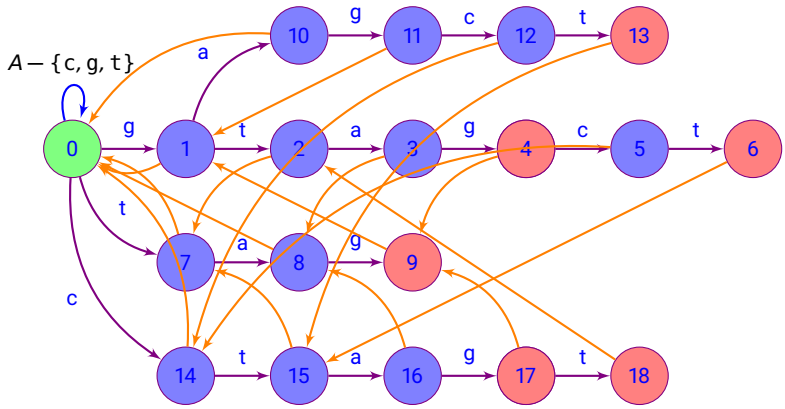


FIGURE E.1. Machine de Aho-Corasick pour l'ensemble fini de mots $X = \{\texttt{gtagct}, \texttt{tag}, \texttt{gagct}, \texttt{ctagt}\}$. Les sommets représentent les préfixes des mots de X, les arcs noirs permettent de passer d'un préfixe u à un préfixe ua. Pour rechercher efficacement les mots de X dans un texte on rajoute un lien de suppléance (en orange) pour tous les sommets différent de la racine. Ce lien associe un sommet u à son plus long suffixe qui est un préfixe d'un mot de X. On rajoute également une boucle sur la racine (en bleu) pour toutes les lettres de l'alphabet qui ne sont pas préfixe d'un mot de X. Les sommets roses correspondent à des mots qui ont au moins un mot de X parmi leurs suffixes.

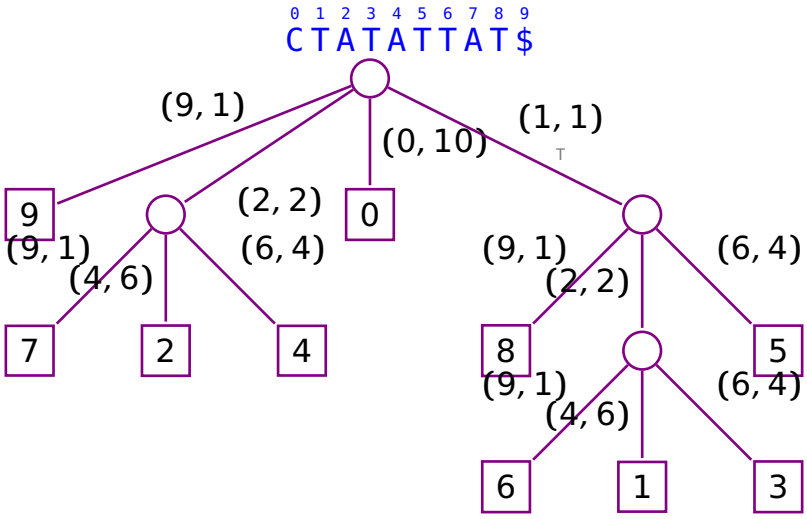


FIGURE E.2. L'arbre des suffixes de CTATATTAT$. Les nœuds internes sont représentés par des cercles alors que les feuilles sont représentées par des carrés et contiennent la position de début des suffixes auxquels elles sont associées. Tous les nœuds internes ont au moins deux descendants. Les arcs sont étiquetés par des couples (i, ℓ) correspondant au facteur commençant en position i de longueur ℓ. Tout chemin de la racine à une feuille correspond à un suffixe du texte. Par exemple le chemin de la racine à la feuille 4 correspond au suffixe qui commence en position 4, soit ATTAT$ qu'on peut lire sur l'arbre avec l'étiquette (2,2) = AT et l'étiquette (6, 4) = TAT$.

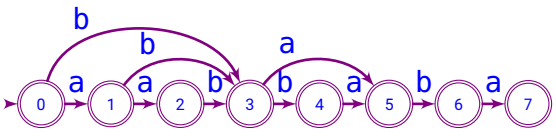


FIGURE E.3. L'oracle des facteurs de aabbaba accepte tous les facteurs de aabbaba mais aussi abab qui ne l'est pas. Cet oracle peut être représenté par ses transitions externes (sans les étiquettes) $(0, 3)$, $(1, 3)$ et $(3, 5)$ en plus du mot aabbaba.

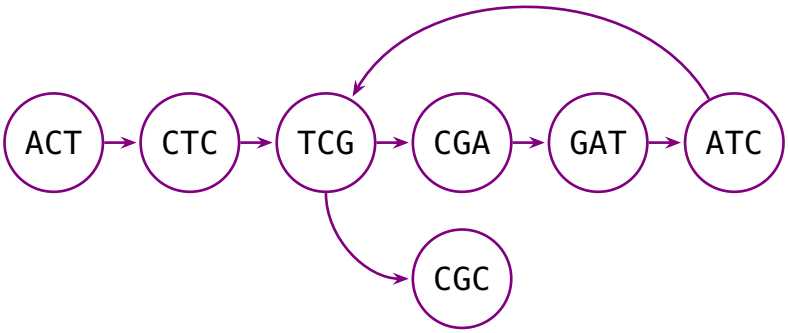


FIGURE E.4. Graphe de de Bruijn d'ordre 3 de $S = \{\texttt{ACTCGA}, \texttt{TCGAT}, \texttt{CGATC}, \texttt{TCGC}\}$. Chaque sommet du graphe correspond à un élément de S. Il existe une arête de TCG vers CGA car le suffixe de TCG dont la longueur est 2 est égal au préfixe de même longueur de CGA, c'est-à-dire à CG.

i	SA	F										L	B^a	B^c	B^t
0	10	$	a	c	a	a	a	c	a	t	a	t	0	0	1
1	2	a	a	a	c	a	t	a	t	$	a	c	0	1	0
2	3	a	a	c	a	t	a	t	$	a	c	a	1	0	0
3	0	a	c	a	a	a	c	a	t	a	t	$	0	0	0
4	4	a	c	a	t	a	t	$	a	c	a	a	1	0	0
5	8	a	t	$	a	c	a	a	a	c	a	t	0	0	1
6	6	a	t	a	t	$	a	c	a	a	a	c	0	1	0
7	1	c	a	a	a	c	a	t	a	t	$	a	1	0	0
8	5	c	a	t	a	t	$	a	c	a	a	a	1	0	0
9	9	t	$	a	c	a	a	a	c	a	t	a	1	0	0
10	7	t	a	t	$	a	c	a	a	a	c	a	1	0	0
												BWT			

FIGURE E.5. La transformée de Burrows-Wheeler (BWT) de `acaaacatat$` correspond à la dernière colonne (L) de la matrice formée par les rotations cycliques de `acaaacatat$` triée dans l'ordre lexicographique. Cette dernière colonne est inversible (elle permet de retrouver la séquence d'origine) et peut être représentée par des vecteurs de bits (à droite). La colonne SA (*Suffix Array*) correspond aux positions de début des rotations cycliques. Elle peut être échantillonnée. Les vecteurs de bits et le SA échantillonné permettent de calculer les mêmes informations que l'arbre compact des suffixes de la même séquence.

Les graphes de de Bruijn ont également été utilisés pour la correction de données de séquençage de troisième génération produisant des fragments longs et bruités [67, 61, 66].

Personnes de la fédération impliquées (principaux et permanents [(1)]) :

Les membres suivants de TIBS participent à des travaux comme ceux décrits précédemment : Richard Groult, Yannick Guesnet, Thierry Lecroq Arnaud Lefebvre, Laurent Mouchard et Élise Prieur-Gaston.

GdR(s) concerné(s) [(2)] :

Les membres présentent régulièrement leurs travaux aux journées du groupe de travail SeqBIM communs aux GDR IM et BIM.

Quelques revues et conférences que vous visez [(3)] :

Les principales revues visées sont *Theoretical Computer Science* pour les aspects d'algorithmique et *Bioinformatics* pour les aspects applicatifs en bioinformatique.

Les conférences du domaine sont CPM (*Combinatorial Pattern Matching*), SPIRE (*String Processing and Information Retrieval*) et PSC (*Prague Stringology Conference*).

(1). Vous pouvez citer des personnes avec lesquelles vous avez travaillé sans forcément que ça ait donné lieu à un projet ou une publication

(2). Il n'est pas nécessaire d'être actif pour citer un GdR, a fortiori parmi ISIS/IM/MADICS, précisez celui ou ceux dont vous vous rapprochez le plus

(3). Il n'est pas nécessaire d'y avoir déjà publié ;-))

Chapitre F Entretien François RIOULT (GREYC - CODAG) 2021

Mots clés : Graphes dynamiques, trajectoires, sport, e-sport

Problématiques scientifiques :

Détection dans des graphes dynamiques de motifs récurrents caractéristiques d'une réussite tactique. Nos graphes dynamiques ont pour sommets les joueurs de handball, le ballon et éventuellement des zones du terrain. Les arêtes représentent les rapports de proximité immédiate entre les joueurs ou la possession du ballon.

Domaines applicatifs (exemples) : Handball, sport collectif, e-sport collectif.

Personnes de la fédération impliquées (principaux et permanents [(1)]) :

Thésards : Alexis Mortelier (Dir. F. Rioult), Quentin Bourgeais (Dir. Éric Sanlaville et Rodolphe Charrier il me semble + Ludovic Seifert, Cetaps Rouen).

Permanents : Éric Sanlaville, Géraldine Del Mondo et Rodolphe Charrier (LITIS), Albrecht Zimmermann (GREYC)

Labos/Industries impliqués en dehors de la fédération :

John Komar, DynamicsLab@NIE, National Institute of Education, Nanyang Technological University de Singapour.

GdR(s) concerné(s) [(2)] :

MADICS, Sport et Activités Physiques

Quelques revues et conférences que vous visez [(3)] :

Ressources logicielles que vous auriez développées :

Projets en cours ou passés [(4)] :

Une situation de handball avec diagramme de Voronoï et triangulation de Delaunay

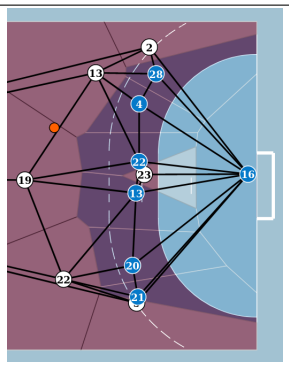


FIGURE F.1

Exemple :

(1). Vous pouvez citer des personnes avec lesquelles vous avez travaillé sans forcément que ça ait donné lieu à un projet ou une publication

(2). Il n'est pas nécessaire d'être actif pour citer un GdR, a fortiori parmi ISIS/IM/MADICS, précisez celui ou ceux dont vous vous rapprochez le plus

(3). Il n'est pas nécessaire d'y avoir déjà publié ;-))

(4). Nom, année et URLs si possible

Chapitre G Entretien Sébastien ADAM, Benoît GAUZERE, Pierre HEROUX (LITIS - APP) Juin 2021

Complément de Luc Brun, Sébastien Bougleux (GREYC - IMAGE)

graphes pour le ML

Mots clés : Matching, Graph Matching, distance d'édition, graph neural network, recherche d'isomorphisme des sous graphes

Problématiques scientifiques :

Domaines applicatifs (exemples) : Chemo-Informatique, Analyse d'images de documents

Personnes de la fédération impliquées (principaux et permanents [(1)]) : Paul Honeine, Pierre Héroux, Benoît Gaüzère, Sébastien Adam, Luc Brun, Sebastien Bougleux.

Labos/Industries impliqués en dehors de la fédération : LIFAT, LMI, LAMSADE, COBRA, CERMN, LCMT,Univ Salerno (Italie),... / ITESOFT, TOSHIBA...

GdR(s) concerné(s) [(2)] : MADICS

Quelques revues et conférences que vous visez [(3)] : ICML, ICLR, NEURIPS, GbPR, GREC, ICDAR, IJDAR, PR, PRL, PAMI, ICPR, S+SSPR

Ressources logicielles que vous auriez développées :

ILPISO : http ://litis-ilpiso.univ-rouen.fr/ILPIso/gem++.html ,

graphkit-learn : https ://github.com/jajupmochi/graphkit-learn,

LSAPE toolbox : https ://bougleux.users.greyc.fr/lsape/ ;

GEDLIB [20] : https ://github.com/dbblumenthal/gedlib,

Binding Python de GEDLIB : https ://forge.greyc.fr/projects/gedlibpy/repository

(résultat du projet RIN AGAC, et d'une collaboration LITIS/GREYC).

Ajouter la lib de LinLin ?

Projets en cours ou passés [(4)] :

- RIN AGAC 2017-2019,
  `http://www.normastic.fr/analyse-de-graphes-appliquee-a-la-chemoinformatique/`
- ANR APi : 2019-2023 `https://anr.fr/Projet-ANR-18-CE23-0014`
- ANR CodeGNN : 2021-2025, `http://www.normastic.fr/projet-anr-codegnn/`

Contexte et méthodologie générale

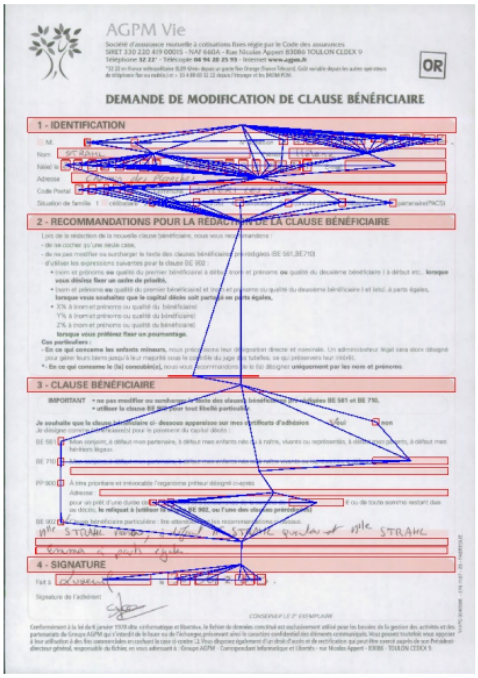
(1). Vous pouvez citer des personnes avec lesquelles vous avez travaillé sans forcément que ça ait donné lieu à un projet ou une publication

(2). Il n'est pas nécessaire d'être actif pour citer un GdR, a fortiori parmi ISIS/IM/MADICS, précisez celui ou ceux dont vous vous rapprochez le plus

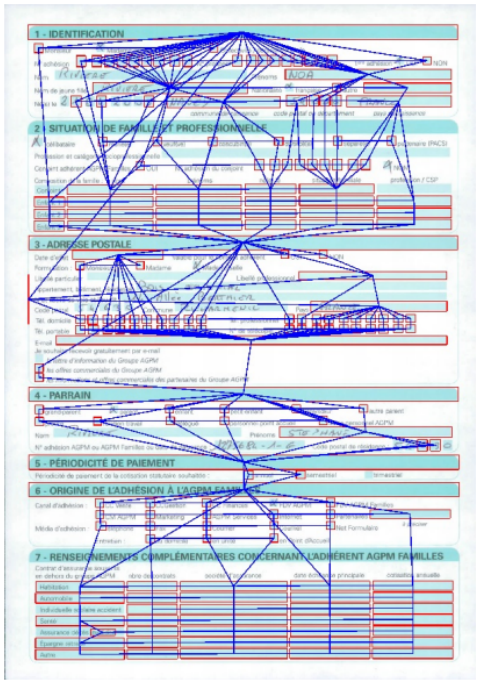
(3). Il n'est pas nécessaire d'y avoir déjà publié ;-))

(4). Nom, année et URLs si possible

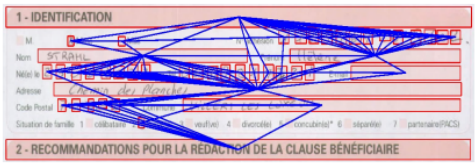
(a) Graphe d'un document complet



(b) Graphe d'un document complet



(c) Zoom sur un sous-graphe



(d) Zoom sur un sous-graphe

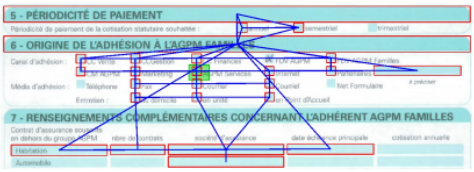


FIGURE G.1. Maroua

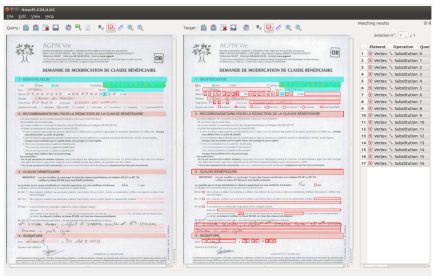


FIGURE G.2. Itesoft

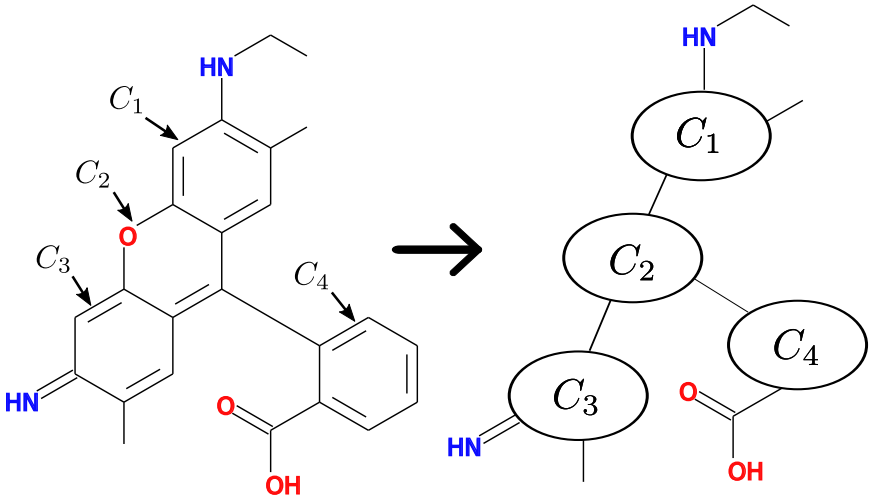


FIGURE G.3. Représentation explicite des cycles d'une molécules dans le cadre des noyaux sur graphes.

Quelques remarques en vrac pour vous aider à vous souvenir du contenu de la réunion

- graphes plutôt statiques (mention d'une perspective sur les aspects dynamiques), orientés ou pas, étiquettés ou pas, pas de limitation vraiment
- isomorphismes, métriques entre graphes, classification
- Méthodologies RO vs GNN
- Recherche de motifs (connus) dans les graphes selon différentes modalités
- Applications : chemoinformatique, document (image et textuel)
- Une thématique plus fondamentale en lien avec l'explicabilité des modèles (BG)

Modèles prédictifs de données sous forme de graphes

La première thématique consiste à traiter des problèmes de prédictions sur des données en entrées sous forme de graphes. Par exemple, à partir d'un ensemble de molécules (codées sous forme de graphes) où chaque molécule est associée à une propriété (par exemple : soigne le cancer ou non), on souhaite être capable de prédire si une nouvelle molécule est susceptible de soigner le cancer. Il faut donc extraire des caractéristiques pertinentes des graphes codant les molécules, afin de pouvoir déduire la propriété de la nouvelle molécule, par comparaison des caractéristiques calculées. Cette problématique peut s'étendre au niveau noeud, où l'on cherche à prédire les propriétés de chacun des noeuds d'un graphe.

Contrairement au machine learning classique travaillant sur des espaces vectoriels, il y a plusieurs difficultés :

- Les données graphes sont discrètes (difficulté pour les problèmes d'optimisation),
- les graphes sont généralement de taille variable (pas de taille prédéfinie et connue à l'avance)
- L'ordre de numérotation des noeuds n'a pas de signification au niveau graphe. Il faut donc définir des méthodes invariantes aux permutation des noeuds dans le cas de la prédiction au niveau graphe, et équivariante lorsque que l'on cherche à prédire des propriétés de noeuds. Cette problématique induit la difficulté l'équivalence entre deux graphes via le test d'isomorphisme.

En conséquence, il faut trouver des moyens pour être capable de mettre en oeuvre des modèles prédictifs. Au sein de notre groupe de recherche, on a 3 axes (cf. Figure G.4) :

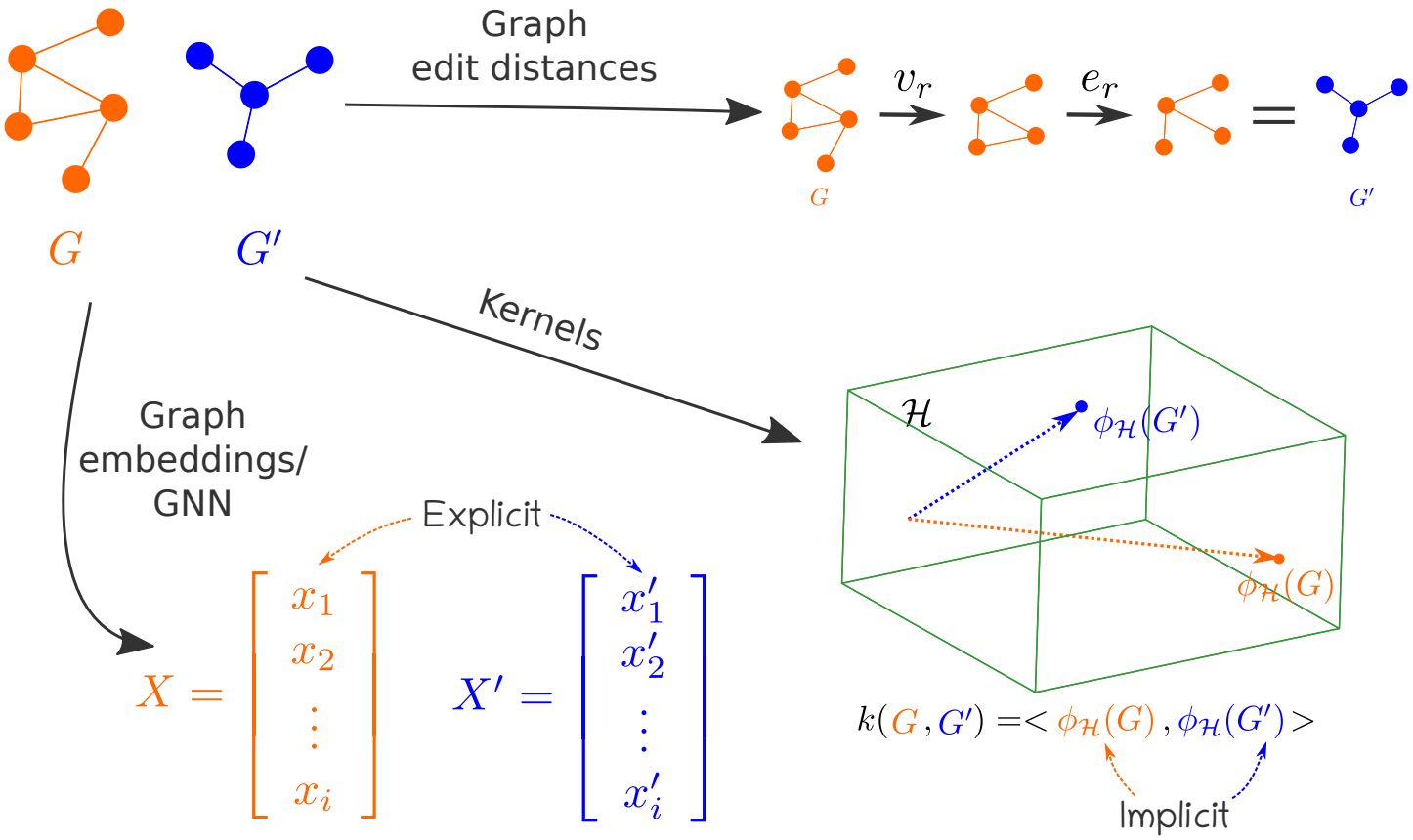


FIGURE G.4. Différentes approches pour connecter Graphes et ML

- La distance d'édition (GED) : elle consiste à définir une mesure de dissimilarité entre graphes via des opérations élémentaires d'édition (ajout,suppression, substitution de noeuds et d'arêtes). Le problème peut être vu comme un problème d'appariement de noeuds entre les graphes, et est NP-Difficile. On attaque le problème via différentes approches :

  1. formulations de problème linéaire discret sous contraintes, de l'approximation via des méthodes d'optimisation continue linéaire ou quadratique.
  2. Formulation du problème comme une forme quadratique à minimiser sous contraintes [23]. Des algortihmes type Franck-Wolfe sont dans ce cas particulièrement adaptés. Notons que ce type de méthodes nécéssite de savoir résoudre un *Linear Sum Assignment Problem with Editions* [24] (LSAPE) que nous avons défini formellement et pour lequel nous avons proposé des méthodes de résolution efficaces.
  3. Un axe parallèle est l'apprentissage des coûts associés à chacune des opérations afin d'adapter au mieux la distance à un problème particulier. Les problèmes actuels auxquels on s'intéresse sur cet axe de recherche est l'apprentissage des couts via des réseaux de neurones. Cela impose nottament de rendre la résolution du problème LSAPE différentiable ce que nous avons fait récément [27] (ICPR 2022).

  Ces travaux sur la distance d'édition nous ont permis, dans le cadre du projet AGAC de proposer une méthode efficace de calcul du graph médian d'un ensemble de graphes [21].

- Plongement de graphes et Noyaux sur graphes : Ces méthodes consistent à projeter les graphes dans un espace où les méthodes d'apprentissage peuvent calculer une solution optimale. Le plongement de graphes sont basés sur un ensemble de descripteurs définis a priori permettant de calculer des représentations pertinentes des graphes pour un problème donné. C'est ce qui a été fait sur des do-

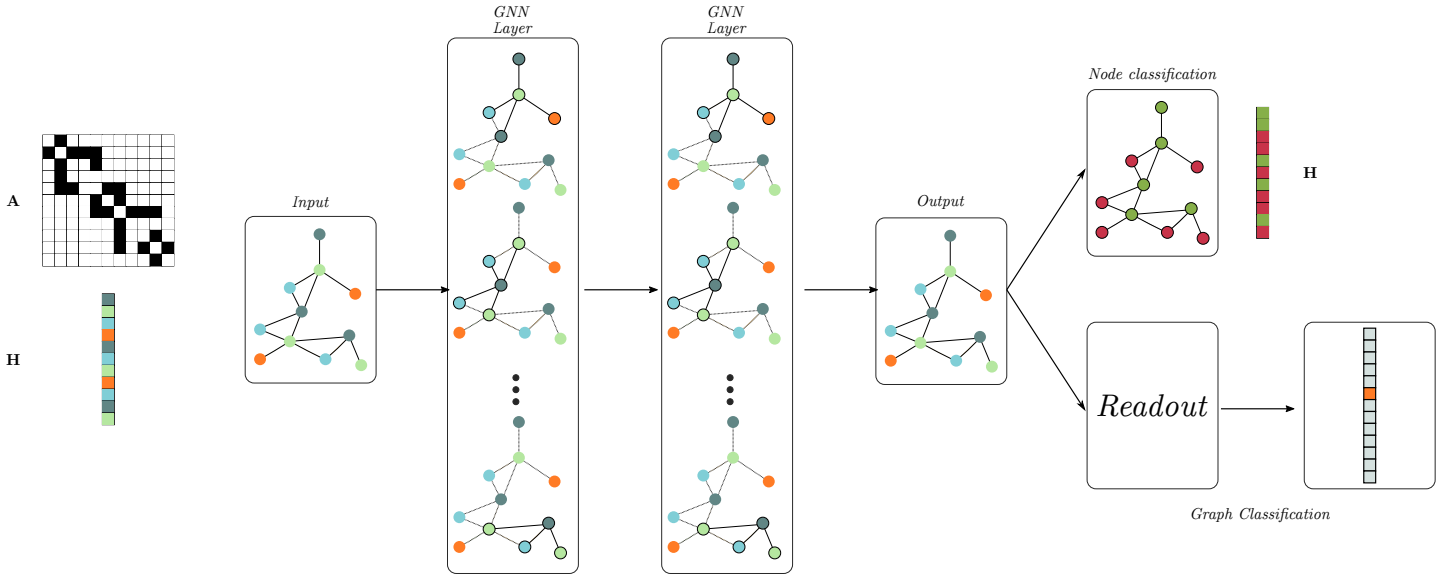


FIGURE G.5. Schéma global du fonctionnement d'un GNN

cuments par Seb et Pierre (je crois). Les noyaux sur graphes peuvent être vus comme une extension des méthodes de plongement dans un espace implicite, ce qui limite les méthodes utilisables aux méthodes dites "à noyaux". Cette approche a été explorée dans le cadre de la thèse de Benoit sur des applications à la Chemoinformatique [45] (Fig. G.3). Cette thématique me semble quand même être mise de coté actuellement.

- Réseaux de neurones sur graphes (GNN, cf. Fig. G.5) : La limitation des méthodes de plongement ou de noyaux est que l'on doit définir a priori les caractéristiques des graphes à extraire pour chaque tâche à traiter. Les réseaux de neurones sur graphes permettent eux de calculer une représentation "optimale" sans que cette dernière ne soit complètement expliciter par un expert. Ce changement paradigme a amené la révolution que l'on connait en machine learning. Pour définir un réseaux de neurones sur graphes, il faut définir deux opérations :

  1. Comment l'information est échangée entre les noeuds
  2. Comment l'ensemble des informations reçue par un noeud est synthétisée

  Plusieurs méthodes ont été définies pour répondre à ces questions. Le LITIS y a contribué en démontrant une équivalence entre méthodes définies à partir du voisinage des noeuds et celles définies à partir de filtre sur le spectre du Laplacien (thèse de Guillaume Renton et travaux de Muhammet Balcilar). Nous avons également proposé une méthode pour prendre en compte les étiquettes d'arêtes (Thèse de Guillaume Renton). Pour le court terme, nous avons une ANR avec Caen et Tours où l'on souhaite :

  1. Apprendre les filtres spectraux du Laplacien au lieu de les définir a priori
  2. Apprendre les étapes de synthétisation de l'information au niveau graphe
  3. Appliquer les GNN sur des graphes spatio-temporels.

C'est sur ces derniers axes de recherche que l' on souhaiterait se focaliser en priorité.

Génération de graphes

Via l'ANR APi menée par Paul Honeine et dans laquelle Benoit participe, on s'intéresse à générer des graphes à partir d'un point particulier (dans un espace associé de noyau sur graphe par exemple, cf Fig. G.6). L'intérêt ici est de générer des graphes ayant des propriétés particulières et d'apporter de l'interprétabilité des méthodes utilisées. Cette thématique a donné lieu à la thèse de Linlin Jia (soutenue en juillet) qui a utilisé des noyaux sur graphes et de la distance d'édition. Cette thèse est poursuivie par celle de Clément Gledel, qui explore des méthodes de génération de graphes basées sur des réseaux de neurones.

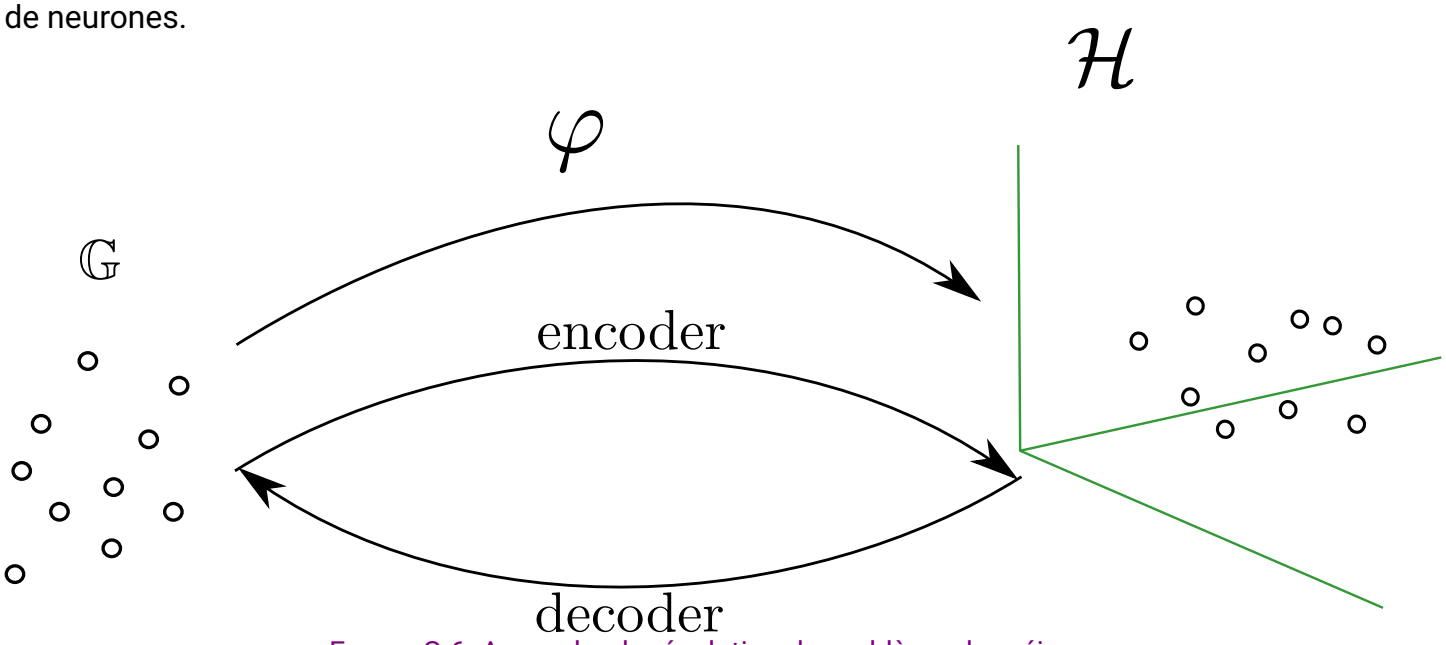


FIGURE G.6. Approche de résolution du problème de préimage

Apprentissage des coûts

Une tache annexe ménée avec l'aide de Florian Yger, Luc Brun et Sébastien Bougleux consiste à définir une méthode pour apprendre les couts de la GED donné un problème particulier. Une première approche linéaire a été proposée dans le cadre de la thèse de Linlin [49]. Les travaux se poursuivent avec Luc, Sébastien, Florian et plus récemment Guillaume Renton pour concevoir une architecture de réseaux de neurones permettant d'optimiser les couts de la GED. On recoupe ainsi les premiers objectifs de la thèse de Guillaume. Dans le cadre de cette collaboration, on a eu un stage de Master 2 financé par Normastic.

Chapitre H

Entretien Nicolas Bedon (LITIS - C&A), Brigitte Vallée (GREYC - AMACC) 7 décembre 2021

Les graphes dans l'axe AlgoComb de la Fédération NormaSTIC

Le document contient une synthèse des thèmes liés aux graphes sur lesquels travaillent les trois équipes (CA, AMACC, TIBS) émargeant à l'axe AlgoComb de NormaStic. Nous accompagnons d'astérisques les thèmes dont l'axe transverse « Interactions avec les Graphes » a déjà pris connaissance. Mais le document contient aussi beaucoup d'autres sujets que l'axe « Interactions avec les Graphes » ne connait pas bien encore. Les fiches synthétiques des trois équipes sont en fin de document.

Les auteurs du présent document (Nicolas Bedon et Brigitte Vallée) ont tenté d'établir une organisation générale entre les thèmes. Mais le document ne prétend pas être une véritable synthèse car les auteurs ont choisi de ne pas retravailler les textes qu'ils ont reçus de leurs équipes et de les garder tels qu'ils leur ont été envoyés. Certains paragraphes sont donc beaucoup plus détaillés que d'autres, certains s'appuient sur des références bibliographiques ou des figures (mais pas tous), certains décrivent des travaux déjà commencés ou quelquefois terminés, tandis que d'autres décrivent par exemple des propositions de thèse.

Nous pensons cependant que ce document, même sous sa forme actuelle très imparfaite, peut se révéler utile, non seulement à l'axe transverse « Interactions avec les Graphes » mais aussi aux membres de l'axe AlgoComb eux-mêmes.

H.1 Graphes généraux

De nombreux problèmes d'optimisation combinatoires sont désormais classiques en théorie des graphes. La plupart sont montrés NP-difficiles (coloration, domination, ensembles indépendants), mais divers algorithmes donnent des solutions satisfaisantes dans les cas courants (que l'on imagine proche des situations concrètes, sans en avoir de preuve, bien que cela puisse être argumenté...).

Nous nous intéressons à des questions qui se formuleraient ainsi : « si mon graphe a telle propriété, que puis-je en déduire sur tel paramètre ? ». La propriété sur le graphe définit une sous famille de graphes sur laquelle nous travaillons. La preuve de la propriété permet généralement de déduire un algorithme.

H.1.1 Conjecture de Bruce Reed (*)

CONTACTS ET COMMENTAIRES: Paul Dorbec

Dans un graphe, un ensemble dominant est un ensemble de sommets tels que tout sommet qui n'est pas dans l'ensemble est voisin d'un sommet de l'ensemble. En 1996, Bruce Reed [75] a prouvé que dans un graphe à n sommets dont tous les sommets ont au moins trois voisins, la taille minimum d'un ensemble dominant était au plus $\frac{3n}{8}$. Il a conjecturé que si le graphe est connexe et que tous les sommets sont de degré égal à trois, alors $\lceil \frac{n}{3} \rceil$ sommets suffisent à dominer le graphe. Ce résultat n'est malheureusement pas vrai dans le cas général, mais tout résultat dans ce sens est devenu un challenge, sur lequel Paul Dorbec travaille depuis plusieurs années.

Pour prouver un tel résultat, la stratégie de preuve permet généralement d'en déduire un algorithme,

simplement en interprétant les choix faits. Une approche adoptée pour attaquer ce problème consiste à identifier de petits sous-graphes que l'on peut dominer avec un nombre de sommets suffisamment petit, et les retirer du graphes. Si on trouve un ensemble de sous-graphes qui permet de décomposer tout le graphe, nous avons une preuve, et un algorithme par déconstruction du graphe. Cette approche a déjà été adoptée avec succès dans [41].

FIGURE H.1. Graphe de Peterson généralisé G_{14} et un ensemble dominant minimum (en rouge).

H.1.2 Représentation des graphes dans le modèle RAM

CONTACTS ET COMMENTAIRES: Etienne Grandjean

Etienne Grandjean étudie actuellement les représentations « naturelles » des graphes et des structures similaires (arbres, circuits, hypergraphes) dans le modèle de calcul des machines RAM. Le but est de représenter de façon fidèle dans le modèle RAM les algorithmes de la littérature portant sur les graphes et leurs variantes, y compris les algorithmes « dynamiques » (algorithmes d'énumération, par exemple) de façon à rendre compte formellement de leur complexité fine : algorithmes en temps linéaire, algorithmes d'énumération à délai constant après pré-calcul linéaire, etc.

Dans les années 2000, Etienne Grandjean a démontré avec ses doctorants que de nombreux problèmes NP-complets classiques sur les graphes (ainsi que les problèmes associés de comptage et d'énumération de leurs solutions) ont exactement la même complexité : coloration d'un graphe avec 3 couleurs, existence d'un noyau dans un graphe, etc : ces problèmes se réduisent les uns aux autres par des réductions en temps linéaire.

H.1.3 Graphes de Cayley

CONTACTS ET COMMENTAIRES: Ali Akhavi

Dans le cadre général de la recherche de bornes inférieures de complexité, on étudie les diamètres des graphes de Cayley, car un minorant de ce diamètre fournit en général une borne inférieure pour la complexité d'une classe d'algorithmes. (dans un cadre aussi général, le problème que résolvent les algorithmes considérés sera un problème adhoc)

H.1.4 Hypergraphes

CONTACTS ET COMMENTAIRES: Loick Lhote, Julien David (Proposition de thèse)

Initiée par Knuth dans les années 60 puis repensée par Flajolet avec les objets et les méthodes de la combinatoire analytique, l'analyse probabiliste d'algorithmes attribue un poids à toutes les entrées à travers une distribution de probabilité. Ainsi, si la distribution des entrées est fidèle à celle qui est mise en jeu dans une application, les résultats de l'analyse pourront prédire efficacement les résultats expérimentaux.

Les hypergraphes sont des structures de données qui généralisent les graphes. L'équipe AMACC considère les hypergraphes et les algorithmes associés comme des objets d'étude dont il convient de décrire le comportement asymptotique lorsque la taille des hypergraphes aléatoires augmente. Les modèles aléatoires considérés sont inspirés d'applications issues de la fouille de données orientée motifs et se veulent les plus réalistes possibles. L'équipe s'intéresse plus particulièrement à la complexité probabiliste du problème de génération des traverses minimales d'un hypergraphe (problème THG - Transversal Hypergraph Generation).

Calculer toutes les traverses minimales d'un hypergraphe est un problème qui apparaîit dans de nombreux domaines (intelligence artificielle, logique, biologie, fouille de données, télécommunications, sécurité, ... Comme un hypergraphe peut avoir un nombre exponentiel de traverses minimales en le nombre de sommets, le problème de générer les traverses minimales ne peut être résolu en temps polynomial dans le pire des cas. Une question encore ouverte est de savoir si le problème peut être résolu (dans le pire des cas) en temps output-polynomial , c'est à dire polynomial en la taille de l'entrée et de la sortie. L'algorithme de Fredman et Khachiyan détient aujourd'hui la meilleure complexité prouvée connue et est output quasi-polynomial.

Il existe plusieurs analyses probabilistes des structures d'hypergraphes aléatoires sous des distributions très particulières. Cependant, il existe assez peu d'analyses probabilistes du problème de génération des traverses minimales. Récemment dans [38], nous nous sommes intéressés au nombre moyen de traverses minimales et nous avons prouvé que l'algorithme introduit par Hébert, Bretto et Crémilleux en 2007 était asymptotiquement presque sûrement output polynomial.

À terme, l'équipe souhaite mener une étude systématique des objets et algorithmes en lien avec le problème de génération des traverses minimales ou plus généralement, en lien avec la structure d'hypergraphe, mais toujours dans l'optique des applications possibles en fouille de données. Trois axes de travail se dégagent alors :

- Proposer des modèles aléatoires d'hypergraphes capables de capturer la complexité des applications en fouille de données mais qui laissent la possibilité de faire des analyses probabilistes ensuite ;
- Analyser le comportement probabiliste des objets informatiques en lien avec les hypergraphes aléatoires et en mesurer l'impact dans les applications en fouille de données (comme les traverses minimales et leurs applications) ;
- Analyser le comportement probabilistes des algorithmes agissant sur les hypergraphes aléatoires et comparer leurs efficacités dans le contexte applicatif de la fouille de données.

H.1.5 Graphes cordaux (*)

CONTACTS ET COMMENTAIRES: Julien Clément, Julien Courtiel (Proposition de thèse)

Un graphe cordal est un graphe que l'on peut obtenir à partir d'un sommet en ajoutant successivement des sommets, de telle sorte qu'à chaque ajout d'un sommet, son voisinage dans le graphe forme une clique. Un ordre d'ajout des sommets selon ce schéma est appelé ordre simplicial. Les graphes cordaux constituent une famille importante de la théorie des graphes notamment parce qu'ils apparaissent dans de nombreux domaines scientifiques transverses, comme en algorithmique des graphes (notion de treewidth), en statistique (réseaux bayésiens) ou en bioinformatique (phylogénie parfaite). Étrangement, peu d'études sur les graphes cordaux ont été menées par le prisme de la combinatoire

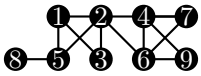


FIGURE H.2. Exemple de 1-2-arbre, étiqueté par un ordre simplicial

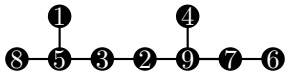


FIGURE H.3. L'arbre de Cayley correspondant

énumérative. En particulier, malgré leur large utilisation, les générateurs aléatoires pour les graphes cordaux emploient une approche heuristique non mathématique : la distribution de sortie, par exemple, est mal pas maîtrisée.

Cette thèse aura pour objectif d'analyser, compter, générer les graphes cordaux. Du point de vue méthodologique de la combinatoire analytique, le comptage des graphes cordaux s'avèrera intéressant et complexe. En effet, les séries génératrices des graphes cordaux se caractérisant par des équations aux dérivées partielles (ce qui est une nouveauté), et on peut espérer que les travaux conduits pendant la thèse s'appliqueront à d'autres familles d'objets combinatoires.

Nous nous sommes déjà intéressés aux 1-2-arbres, qui sont les graphes cordaux particuliers obtenus à partir d'une clique de taille au plus 3, en ajoutant successivement des sommets dont le voisinage consiste en un sommet ou une arête (une clique de taille au plus deux) (cf Figure H.2). Par construction, tout graphe de largeur arborescente (*treewidth*) au plus 2 est sous-graphe d'un 1-2-arbre. En comptant ces objets, nous avons identifié une similarité avec les arbres de Cayley (comptés selon une numérotation des sommets de 1 à n, cf Figure H.3, et pu mettre en évidence une bijection explicite entre les deux familles.

Un intérêt de ce genre de résultat est de transposer les propriétés des arbres de Cayley, très étudiés, sur les 1-2-arbres. Un autre intérêt est dans la génération. Une fois qu'une famille de graphe est bien caractérisée, il est alors possible de générer de façon aléatoire uniforme un objet de cette famille, en particulier.

H.2 Graphes dans les caractérisations et analyses d'algorithmes

H.2.1 Graphes et traces : un exemple

CONTACTS ET COMMENTAIRES: Giovanna Guaiana

Les graphes sont utilisés comme outils permettant de modéliser des propriétés de systèmes algébriques. Les caractéristiques des graphes modèles permettent de déterminer des classes particulières de ces systèmes, et étudier les graphes en tant que tels aide à la caractérisation de ces systèmes.

Les monoïdes de traces ou monoïdes partiellement commutatifs libres sont des structures algébriques fondées sur l'application de certaines commutations sur les lettres de leur alphabet et donc sur leur mots. Plus précisément un ensemble de commutations fixé sur les lettres d'un alphabet produit un monoïde de traces, celui dont les éléments sont des classes d'équivalence de mots qui différent entre elles par l'échange successif de lettres adjacentes, selon les commutations fixées. Cet ensemble de commutations se représente usuellement par un graphe non direct. Les sommets sont les lettres de l'alphabet fini et il y a un arc entre la lettre a et la lettre b si a et b commutent. Ce graphe est donc utilisé pour représenter le monoïde de traces correspondant. Selon les problèmes qu'on traite sur ces

monoïdes (propriétés combinatoires, étude des familles de langages de traces, etc.), il peut être utile de représenter un monoïde de trace par le graphe de non commutations qui est le graphe complémentaire du premier. Certains graphes déterminent une structure algébrique particulièrement étudié.

Par exemple le graphe sur $\{a, b, c\}$ où tous les sommets sont reliés représente le monoïde commutatif sur 3 lettres.

Le graphe sur $\{a, b, c, d\}$ où a–c et a–d b–c b–d représente le produit direct de deux monoïdes libres, l'un sur $\{a, b\}$ l'autre sur $\{c, d\}$.

Un résultat sur les graphes a été trouvé par un membre de l'équipe CA en collaboration avec des chercheurs extérieurs, dans le cadre d'un problème sur la fermeture par commutations de familles significatives de langages de mots.

Si G est un graphe direct, les conditions suivantes sont équivalentes :

- ✿ le graphe complément de chaque composante connexe d'un graphe direct G est transitif ;
- ✿ le graphe G ne contient pas comme sous-graphe ni Paw ni P_3 (cf. Figure H.4).

FIGURE H.4. Les graphes Paw (à gauche) et P_3 (à droite).

H.2.2 Graphes et cryptographie : un exemple

CONTACTS ET COMMENTAIRES: Ayoub Otmani, Magali Bardet

Décider si deux sous-espaces linéaires finis de $\mathbb{F}_n$ où $\mathbb{F}$ est un corps (non nécessairement fini) sont identiques à une permutation des coordonnées près est connu sous le nom de *permutation-équivalence de codes*. C'est un cas spécial du *problème d'isomorphisme* : pour une métrique donnée, existe t-il une isométrie linéaire entre deux sous-espaces linéaires. Ces problèmes se posent souvent en théorie des codes quand $\mathbb{F}$ est muni de la distance de Hamming. Le problème d'isomorphisme est important dans la classification des codes car les codes isomorphes présentent des caractéristiques similaires comme leurs longueurs, dimensions, distributions de poids et capacités de correction. Le problème de permutation-équivalence de codes a été utilisé en cryptographie par McEliece dans [63] comme outil pour améliorer la sûreté de schéma cryptographiques à clef publique. Bien que la sûreté de schéma de McEliece ne lui est pas directement relié, il existe des schémas qui peuvent être entièrement cryptanalysés quand l'attaquant le résoud (voir par exemple [65, 12]). En conséquence, une meilleure compréhension de sa difficulté calculatoire permet une meilleure évaluation de la sécurité de primitives cryptographiques basées sur des codes.

Sa difficulté calculatoire a été étudiée dans [73] où il est montré que si le problème sur le corps a 2 éléments est NP-complet, alors la hiérarchie polynomiale s'effondre. En conséquence il n'est probablement pas NP-complet. Une réduction polynomiale en temps du problème d'isomorphisme de graphes à la permutation-équivalence de codes est également donnée dans l'article. Cependant, à la lumière de récents résultats montrant que le problème d'isomorphisme de graphes pourrait ne pas être NP-complet, cette réduction devient moins intéressante. Une complexité dans le pire des cas a été donnée dans [10] montrant que le problème de permutation-équivalence de codes peut être résolu en $(2 + o(1))^n$ opérations.

Une autre question très importante connectée au problème d'isomorphisme est la caractérisation du groupe d'isométries laissant globalement invariant un espace linéaire donné. Ce groupe est appelé groupe d'automorphisme. L'ensemble des isomorphismes entre deux sous-espaces linéaire est un *coset* du groupe d'automorphismes. En particulier le nombre de solutions (quand elles existent) au problème d'isomorphismes est exactement la taille du groupe d'automorphismes. Comme les permutations sont toujours des isométries pour chaque distance sur $\mathbb{F}_n$, l'ensemble des permutations qui stabilise globalement un espace linéaire est appelé le groupe de permutations. Leon [57] a introduit une méthode pour déterminer entièrement le groupe de permutation d'un code linéaire sur un corps fini $\mathbb{F}_q$. L'algorithme nécessite le calcul de tous les mots du code d'un poids donné. La complexité en temps est donc $nq^{O(k)}$ pour des sous-espaces vectoriels de dimension k en supposant que les complexités en temps des opérations sur le corps sont constantes. Les techniques développées dans l'algorithme de Leon peuvent être utilisées pour résoudre le problème de permutation-équivalence pour les codes linéaires de petites dimensions sur de petits corps. Cet algorithme a par exemple été implanté dans le logiciel Magma [22].

Plus tard, Sendrier [78] a considéré l'enveloppe des codes qui est l'intersection d'un sous-espace vectoriel avec son complément orthogonal relativement au produit Euclidien interne. La motivation sous-jacente est que les sous-espaces linéaires (permutation-) équivalents ont des enveloppes permutation-équivalentes. Bien que le problème des permutations entre les enveloppes puisse avoir plus d'une solution, et que de plus des codes non équivalents peuvent avoir des enveloppes équivalentes, l'avantage en pratique est que les enveloppes peuvent avoir de très petites dimensions [79]. En fait les enveloppes se réduisent souvent à $\{0\}$ ce qui, paradoxalement, n'aide pas à résoudre le problème. En conséquence, l'approche de Sendier est intéressante quand l'enveloppe est très petite mais non triviale, et que son groupe de permutation est idéalement trivial. La technique développé dans [78] commence par extraire de chaque position le code à tester, et à calculer pour chacun d'eux l'enumérateur de poids pour son enveloppe. Tous ces énumérateurs de poids fournissent une « signature » compacte telle que des codes permutation-équivalents ont des signatures équivalentes. Le problème de permutation équivalence est ainsi réduit à l'équivalence de deux signatures. La complexité en temps de [78] est au moins exponentielle dans la dimension de l'enveloppe. De plus, pour considérer les cas où plusieurs solutions peuvent exister -sans considérer que deux codes non équivalents peuvent avoir des signatures équivalentes- différentes extractions successives doivent être réalisées, ce qui complique l'analyse de l'algorithme. Cependant (voir par exemple [81]) la complexité empirique en temps est $O(n^3 + 2^h n^2 \log n)$ avec h la dimension de l'enveloppe, ce qui signifie que l'algorithme devient polynomial quand il est restreint aux enveloppes de dimensions bornées.

Une nouvelle approche algébrique a été récemment introduite dans [77] pour résoudre le problème de permutation équivalence des codes. Elle construit un système polynomial quadratique dont les solutions binaires sont les permutations. Les techniques sur les bases de Gröbner permettent la résolution de l'équivalence des codes mais le coût devient rapidement prohibitif quand la longueur des codes augmente. Plusieurs améliorations ont été proposées permettant la résolution efficace de la permutation-équivalence quand les codes ont de petites enveloppes ou sont définis sur de grands corps.

Ces travaux tendent à prouver que la difficulté de la permutation-équivalence des codes n'est pas bien comprise. D'un point de vue pratique, la croyance majoritaire est que le problème est facile (voir par exemple [80]) pour les codes à très petites enveloppes, comme c'est le cas des codes linéaires aléatoires. D'un autre coté, aucune borne théorique rigoureuse validant les observations concrètes n'est connue, en particulier il n'y a pas de preuve montrant que le problème de permutation-équivalence des

codes est polynomial en temps pour les enveloppes triviales. De plus, certains travaux ont été centrés sur la conception d'algorithmes en pratique [57, 78, 25, 43] mais aucun ne semble capable de décider efficacement si deux codes de taux mettons un demi et de longueur supérieure à 200 sont équivalents ou non. La situation empire quand les corps sont grands, pour ne pas mentionner ceux de caractéristique 0, parce que ces méthodes supposent de petits corps.

Nous avons montré dans [14] qu'en supposant qu'on dispose d'une sous-routine décidant si deux graphes pondérés non dirigés sont isomorphes, nous pouvons décider de manière déterministe la permutation-équivalence des codes sur un corps arbitraire $\mathbb{F}$ en supposant triviales les enveloppes des espaces vectoriels. Notre réduction est très efficace, car elle consiste à calculer l'inverse d'une matrice carrée d'ordre n avec n la longueur des codes considérés. Nous exploitons le fait que la somme directe d'un sous-espace vectoriel $U \subset \mathbb{F}^n$ et son orthogonal $U^\perp$ est égal à l'espace vectoriel $\mathbb{F}^n = U \oplus U^\perp$ quand $U \cap U^\perp = \{0\}$. Il est alors possible de décomposer chaque vecteur e_i de la base canonique $\{e_i : i \in [[1, n]]\}$ comme $e_i = \sigma_U(e_i) + \sigma_{U\perp}(e_i)$ avec $\sigma_U(e_i)$ (resp. $\sigma_{U^\perp}(e_i)$) est la projection de e_i dans U (resp. $U^\perp$). Définissons la matrice carrée $\Sigma^{U^\perp}$ permettant d'écrire $\Sigma_U + \Sigma_{U^\perp} = I_n$. Nous prouvons que Σ_U et $\Sigma_{U^\perp}$ sont symétriques et que pour chaque matrice de permutation X nous avons $X^T(\Sigma_U + \Sigma_{U^\perp})X = X^T\Sigma_U X + X^T\Sigma_{U^\perp}X$. Par l'unicité de la décomposition, $\Sigma_{UX} = X^T\Sigma_U X$ and $\Sigma_{U^\perp X} = X^T\Sigma_{U^\perp}X$. Ces relations expriment la propriété que les graphes représentés par Σ_{UX} et Σ_U vues comme des matrices d'adjacences sont isomorphes.

L'avantage de cette réduction est double. D'un point de vue pratique, la permutation-équivalence pourrait bénéficier de (nombreux) outils efficaces existants de résolution d'isomorphismes entre graphes. Nos expérimentations avec des codes binaires de longueurs allant jusqu'à 20000 montrent qu'on peut décider de l'équivalence de deux codes en moins de deux secondes (Intel Core I7, 2.5 GHz, 16 GB ram, Magma [22] utilisant la version 2.6r7 des paquetages `nauty` et `Traces` [64]). L'autre avantage est que, à la lumière de résultats récents [9], notre réduction valide dans un certain sens la croyance que les codes à enveloppes triviales forment une classe d'instances faciles pour le problème de la permutation-équivalence des codes.

Finalement, nous proposons deux solutions pour traiter le cas des codes ayant une intersection non triviale avec leur complément orthogonal. La première s'applique plus particulièrement aux codes définis sur une extension de $\mathbb{F}_{p^m}$ où $m > 1$ et p est premier. Nous équipons l'espace $\mathbb{F}_{p^m}^n$ avec un produit interne Hermitien en utilisant un automorphisme de $\mathbb{F}_{p^m}$. Cela permet l'adoption d'un point de vue différent car un code peut avoir une enveloppe non triviale pour un produit interne mais a une enveloppe triviale pour un autre. Nous pouvons ainsi généraliser notre réduction à n'importe quel produit interne Hermitien. Cette généralisation permet d'inclure tout sauf une exponentiellement petite fraction des codes linéaires.

? Algorithme Gitbisect et sa modélisation par des graphes (*)

CONTACTS ET COMMENTAIRES: Paul Dorbec, Julien Courtiel (Thèse de Romain Lecoq)

Dans le cadre de la thèse de Romain Lecoq, nous cherchons à analyser l'algorithme gitbisect. L'outil de version git permet de gérer les versions successives d'un logiciel (*commits*). L'ensemble des versions, les différentes branches et les liens de parenté entre les versions, définissent un graphe orienté (par le temps et le lien de parenté). Le problème consiste à chercher un point de régression dans le dépôt, ou, pour simplifier, un commit qui introduit un bug dans le dépôt. L'hypothèse de travail est que le bug est

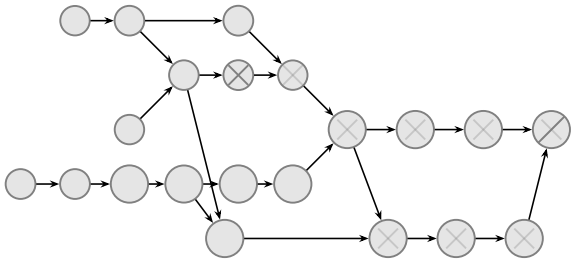


FIGURE H.5. Exemple d'un graphe représentant un dépôt. La croix noire représente le premier commit d'introduction du bug, tous les commits qui en découlent portent le bug.

présent dans la dernière version du dépôt, et que les versions qui présentent le bug ont toutes un ancêtre commun, la version issue de l'introduction du bug (cf. Figure H.5. Pour identifier le commit défectueux, on applique des tests aux différentes versions, jusqu'à trouver la version défectueuse dont aucun parent n'est défectueux.
Le problème consistant à trouver le plus petit nombre de tests nécessaires pour identifier le bug où qu'il soit est prouvé NP-difficile [31].

L'algorithme gitbisect s'inspire de la recherche dichotomique pour proposer une méthode de recherche assez intuitive, et qui semble assez efficace. Nous avons montré que, dans le cas général, cette méthode peut être très lente (utiliser un nombre de tests exponentiel en le nombre de requêtes optimal). Nous avons aussi montré, à l'inverse, que si, dans le dépôt, les opérations de fusion de branches (*merge*) se limitent à la fusion de deux branches (i.e. le graphe orienté a degré entrant au plus 2), alors gitbisect est une bonne approximation de l'optimal (au plus $\equiv 1.71$ fois le nombre de requêtes optimal) et que la borne est atteinte sur certains graphes). Nous avons aussi construit un autre algorithme qui améliore un peu ce facteur d'approximation (pour atteindre environ 1.4 fois l'optimal au pire).

H.2.3 Graphes dynamiques : adaptation des problèmes classiques (*)

CONTACTS ET COMMENTAIRES: Matthieu Dien, Paul Dorbec, projet RIN commun avec l'équipe RI2C (LITIS, le Havre)

Dans le cas de ce projet RIN, nous avons identifié une problématique que nous souhaitons étudier finement sur les graphes dynamiques. Un graphe dynamique est défini sur un ensemble de sommets statique, et un ensemble d'arêtes, chacune munie d'un étiquetage correspondant aux intervalles de temps (dates) où l'arête est présente (la disparition d'un sommet peut simplement être modélisée par son isolement sur la fin de la période).

Certains problèmes sont très pertinents dans les graphes dynamiques. Par exemple, la notion de trajet dans un graphe dynamique consiste à trouver un ensemble de paires (arête, date) qui constitue un chemin d'un sommet à un autre. Cela représente par exemple le choix d'un itinéraire avec escale, et on peut voir ce problème comme une généralisation de la notion de chemin. Sur cette base, de nouvelles définitions de la connexité ont été proposées.

Sur d'autres problèmes, il semble difficile de définir des définitions qui enrichissent les problématiques classiques. Par exemple, pour adapter la définition d'un ensemble indépendant aux graphes dynamiques, on peut vouloir que l'ensemble soit indépendant à tout instant. Le problème revient alors à

chercher un ensemble indépendant dans le graphe comportant toutes les arêtes qui existent à un moment ou à un autre dans le graphe, revenant au problème d'ensemble indépendant dans les graphes statiques. On observera que le problème de l'ensemble dominant n'admet pas ce même cadre général, puisqu'un ensemble peut être dominant à chacune des dates sans l'être sur le graphe correspondant à l'intersection des différentes dates.

Au sein de ce projet, nous souhaitons (entre autres) catégoriser les problèmes et les différentes adaptations possibles, pour identifier les questions intéressantes.

H.3 Graphes comme structures de données pour l'algorithmique

H.3.1 DAG

CONTACTS ET COMMENTAIRES: Julien Clément

Au cours de la dernière décennie, de nombreuses améliorations ont été réalisées afin de caractériser les graphes acycliques orientés (ou DAG) de manière combinatoire, ouvrant la voie à l'analyse d'objets induits par les procédés de compactage. Nous adoptons ici le point de vue conjoint de la combinatoire et des probabilités, et nous nous intéressons au dénombrement, à la génération aléatoire ainsi qu'à l'analyse probabiliste de ces structures à l'aide de ces outils. En particulier, nous concentrons sur l'étude des diagrammes de décision binaire qui constituent une structure de données particulièrement efficace en pratique pour représenter une formule logique.

H.3.2 Automates

CONTACTS ET COMMENTAIRES: C&A/Théorie des langages

Les automates [53] sont des multi-graphes orientés (plusieurs arêtes $n \to n'$ sont possibles) dans lesquels les arcs sont étiquetés par des lettres. Ils sont un des outils centraux de la théorie des langages formels. Parmi les très nombreuses utilisations des automates, citons :

- la modélisation de processus, protocoles ou systèmes finis (voir Figure H.6), la spécification de propriétés et la vérification formelle de propriétés des processus ;
- la représentation et la recherche de motifs (par exemple, dans le traitement du texte ou les compilateurs) ;
- l'algorithmique de la logique formelle : la théorie des langages basée sur les automates est au cœur de très nombreux résultats (décision, caractérisations, transformations, etc) et algorithmes en logique.

Ces différents aspects des automates sont développés dans l'équipe C&A, essentiellement par des développements théoriques. Par exemple, nous nous intéressons à l'algorithmique des transformations entre expressions rationnelles et automates, ou à la notion de déterminisme dans les automates. Nous développons également des extensions de la notion habituelle d'automates, comme par exemple :

- les automates d'arbres ;
- les automates de structures infinies ;

- les automates d'ordres partiels, qui étendent les automates de mots finis en permettant aux éléments constitutifs de l'entrée d'être partiellement ordonnés plutôt que totallement.

Toutes ces extensions ont des applications en modélisation des processus ou en logique formelle. À l'intersection de ces deux domaines se trouve la vérification formelle de processus, protocoles ou programmes.

FIGURE H.6. Les automates sont des modèles naturels de systèmes finis. À gauche, un automate modélisant un distributeur à café. Il accepte les pièces de 10 et 20 centimes, le café coûte 30 centimes. À droite, le même automate où les actions sont abstraites par des lettres (a : annuler, 1 : 10, 2 : 20, c : café). Le langage de l'automate de droite peut être représenté par l'expression rationnelle $(a + (1 + 2 + 11)a + (111 + 21 + 12)c)^*(111 + 21 + 12)$.

Automates de pomsets

CONTACTS ET COMMENTAIRES: Nicolas Bedon

Les mots sont des suites totallement ordonnées de lettres; quand chaque lettre modélise une instruction d'un programme, ou un évènement, les mots représentent donc naturellement des exécutions de processus *séquentiels*, et les automates les prenant en entrée les processus *séquentiels* eux-mêmes. Les pomsets sont une généralisation des mots : les lettres sont partiellement ordonnées plutôt que totalement. Ils modélisent ainsi des exécutions de processus concurrents, et les automates les prenant en entrée les processus concurrents eux-mêmes. En programmation, la séquentialité n'est la plupart du temps pas anarchique : elle est structurée par le langage de programmation. De même, la concurrence est souvent structurée par les primitives mises à disposition par le langage. Le modèle Fork/Join est très utilisé dans les langages et en algorithmique; il apparaît naturellement dans les descriptions des algorithmes appliquant la stratégie du « diviser pour régner ». Les graphes représentant l'ordre des lettres dans les pomsets correspondants ont une forme particulière, dite « sans N », ou de manière équivalente [85], « série-parallèle ».

Nous nous intéressons aux extensions des automates prenant en entrée des pomsets série-parallèles.

Automates de pomsets finis Lodaya et Weil [58, 59, 60] ont étendus les automates sur les mots finis aux pomsets finis sans N avec deux familles de transitions, les transitions *fork* et les transitions *join*, permettant respectivement de débuter et de terminer des flots d'exécutions parallèles. Ils ont étendus des résultats classiques de la théorie des automates de mots finis à leurs automates de pomsets finis. Nous nous sommes également intéressés à ces questions [16].

Motivés par l'étude des algèbres de Kleene concurrentes, Kappé & al. [50, 51] ont également étendus les automates sur les mots finis aux pomsets finis sans N, par l'ajout d'une unique famille de transition qui permet à la fois de débuter le parallelisme mais aussi de spécifier le comportement de l'automate

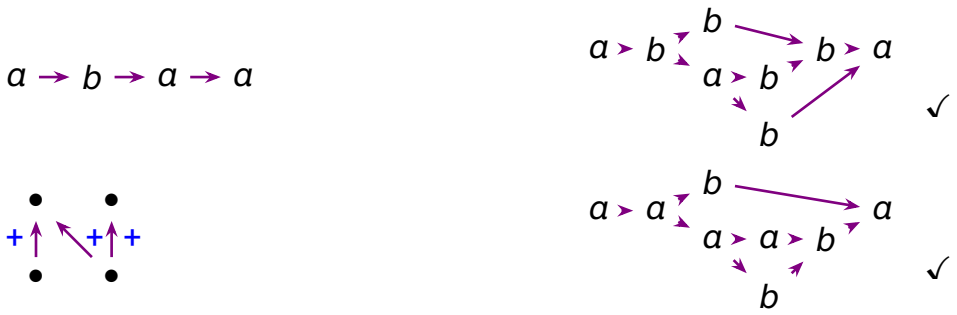


FIGURE H.7. En haut à gauche, un mot. À droite, deux pomsets. La relation d'ordre entre leurs éléments est représentée par un graphe acyclique. Un pomset « sans N » ne contient pas quatre éléments dont l'ordre relatif est donné par le graphe en bas à gauche, où le symbole + désigne la fermeture reflexive de la relation donnée par les flèches. Le pomset en bas à droite est sans N, celui au dessus ne l'est pas.

à la fin du parallelisme.

Nous nous sommes interessés à un premier rapprochement de ces deux familles d'automates de pomsets finis, et en avons en particulier lié les deux familles dans certains cas importants [17]. Dans le cas le plus général, les deux familles sont équivalentes : tout langage pouvant être décrit par un automate d'une des deux familles peut également être décrit dans l'autre. Dans ce cadre, nous avons en particulier donné plusieurs caractérisations des automates de Lodaya & Weil pour le cas des langages séries-parallèles.

Automates de pomsets transfinis Dans une autre direction, nous nous sommes intéressés [18, 6, 7, 4, 5] à l'extension des automates de Lodaya & Weil au cas des pomsets sans N transfinis. Si les automates de pomsets finis modélisent des processus concurrents dont l'exécution termine en un nombre fini d'étapes, le cas transfini modélise ceux dont l'exécution ne termine idéalement pas (comme les processus systèmes démons), ou dans lesquels une boucle infinie est détectée, un signal envoyé au processus pour le faire sortir de sa boucle infinie afin qu'il reprenne son exécution dans un autre état. L'extension ajoute aux automates de Lodaya & Weil des transitions employées à l'infini, comme Büchi [28, 29] l'a fait pour obtenir des algorithmes de décision de la logique monadique du second ordre (MSO) sur les ordinaux dénombrables. En l'utilisant, nous avons en particulier obtenu des algorithmes de décision de MSO sur des ordres transfinis sans N, généralisant ainsi des résultats de Büchi. Les travaux de Büchi ont été la source de nombreux autres travaux fondateurs en logique formelle et en vérification de processus, et certaines de leurs généralisations restent à développer.

Théorie des langages et théorie des graphes

Plusieurs aspects et outils de la théorie des langages de mots finis, comme par exemple son approche par de la logique, dont MSO, ont été entendus aux graphes finis [35]. Ces extensions de la théorie des langages finis ont des applications en théorie structurelle des graphes finis. Elles passent souvent par l'utilisation de la logique pour exprimer des problèmes sur les graphes, spécifier des classes de graphes ou des transformations.

H.3.3 Trie

Les *tries* sont étudiés de part et d'autre de la Normandie, avec des points de vues différents.

Trie, recherche de motifs et indexation (*)

CONTACTS ET COMMENTAIRES: TIBS/Algorithmique du texte

Pour la recherche de motifs, on utilise fréquemment des arbres (graphes sans cycles). Par exemple, l'algorithme de recherche des occurrences d'un ensemble fini de mots d'Aho-Corasick [1] utilise un *trie* (mot provenant du terme anglais *information re**trie**val*) qui permet de représenter l'ensemble fini de mots en factorisant les préfixes communs (voir Figure H.8).

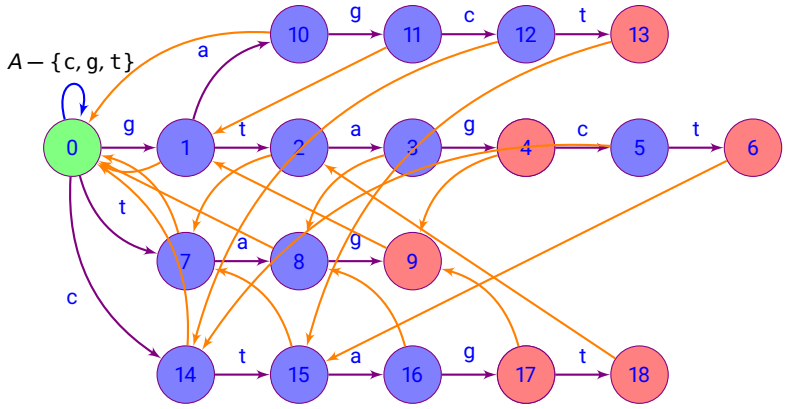


FIGURE H.8. Machine de Aho-Corasick pour l'ensemble fini de mots $X = \{\texttt{gtagct}, \texttt{tag}, \texttt{gagct}, \texttt{ctagt}\}$. Les sommets représentent les préfixes des mots de X, les arcs noirs permettent de passer d'un préfixe u à un préfixe ua. Pour rechercher efficacement les mots de X dans un texte on rajoute un lien de suppléance (en orange) pour tous les sommets différent de la racine. Ce lien associe un sommet u à son plus long suffixe qui est un préfixe d'un mot de X. On rajoute également une boucle sur la racine (en bleu) pour toutes les lettres de l'alphabet qui ne sont pas préfixe d'un mot de X. Les sommets roses correspondent à des mots qui ont au moins un mot de X parmi leurs suffixes.

Lorsqu'il s'agit d'indexer tous les facteurs d'une séquence, ce type de *trie* s'avère trop gourmand en espace. Il est possible de réduire l'espace nécessaire en considérant l'arbre compact des suffixes [88] obtenu à partir du *trie* des suffixes en ne gardant que les nœuds internes de degré sortant au moins égal à 2 et les feuilles, en ayant pris soin d'ajouter un terminateur à la fin de la séquence de manière à identifier les feuilles et les suffixes de la séquence (voir Figure H.9).

Les *tries* et les automates sont largement utilisés pour la recherche de motifs [36].

Trie et théorie de l'information

CONTACTS ET COMMENTAIRES: Brigitte Vallée, et laboratoire sud-américain SINFIN

L'équipe AMACC étudie et utilise les tries avec un point de vue de théorie de l'information. Quand il est construit sur des mots fournis par une source, le trie, avec sa profondeur moyenne, permet de quantifier les propriétés aléatoires de la source. Quand il est construit sur une source d'entropie positive, la profondeur moyenne fait naturellement intervenir l'entropie. Que se passe-t-il quand la source est d'entropie nulle ? Dans ce cas, les sud-américains du laboratoire SINFIN ont introduit et définit une fonction qui généralise l'entropie, la fonction de poids. Le groupe travaille actuellement dans deux directions : (a) pour une source d'entropie nulle, relier la profondeur moyenne du trie au poids de la source ; (b) construire des sources (naturelles...) d'entropie nulle, où l'on peut observer des comportements très variés de la profondeur moyenne du trie.

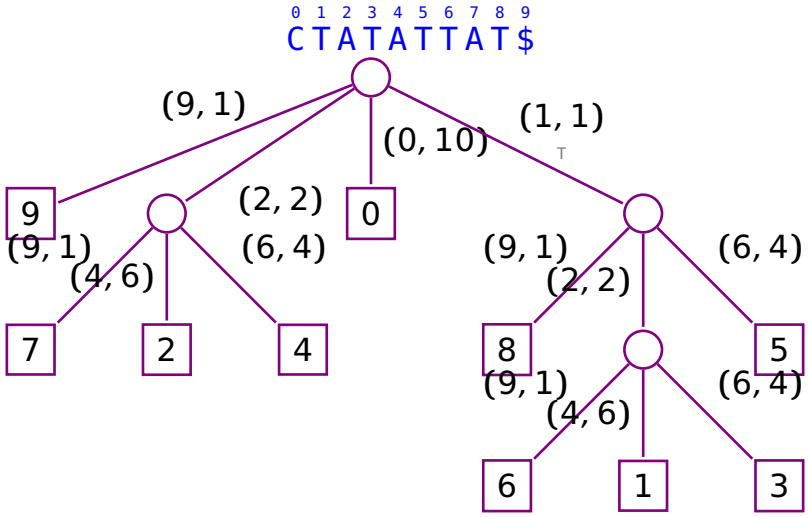


FIGURE H.9. L'arbre des suffixes de CTATATTAT$. Les nœuds internes sont représentés par des cercles alors que les feuilles sont représentées par des carrés et contiennent la position de début des suffixes auxquels elles sont associées. Tous les nœuds internes ont au moins deux descendants. Les arcs sont étiquetés par des couples (i, ℓ) correspondant au facteur commençant en position i de longueur ℓ. Tout chemin de la racine à une feuille correspond à un suffixe du texte. Par exemple le chemin de la racine à la feuille 4 correspond au suffixe qui commence en position 4, soit ATTAT$ qu'on peut lire sur l'arbre avec l'étiquette (2,2) = AT et l'étiquette (6, 4) = TAT$.

H.3.4 Oracle des facteurs

CONTACTS ET COMMENTAIRES: TIBS/Algorithmique du texte

L'arbre compact des suffixes (Figure 3.10) s'avère encore trop gourmand en espace pour de longues séquences. Il existe plusieurs méthodes pour encore réduire l'espace nécessaire. L'une d'elle s'appelle l'oracle des facteurs [3], c'est un automate fini déterministe qui reconnait un peu plus que l'ensemble des facteurs d'une séquence (voir Figure H.10). Des membres de l'équipe ont développé une méthode rapide pour calculer de longues répétitions dans la séquence d'origine sur cette structure [54].

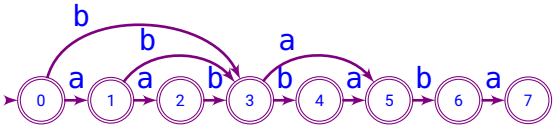


FIGURE H.10. L'oracle des facteurs de aabbaba accepte tous les facteurs de aabbaba mais aussi abab qui ne l'est pas. Cet oracle peut être représenté par ses transitions externes (sans les étiquettes) $(0, 3)$, $(1, 3)$ et $(3, 5)$ en plus du mot aabbaba.

L'oracle des facteurs a été utilisés pour détecter des répétitions dans les séquences génomiques [56], pour la compression de texte sans perte [55] et pour la génération de musique [8].

H.3.5 Graphes de chevauchement et graphes de de Bruijn

CONTACTS ET COMMENTAIRES: TIBS/Algorithmique du texte

Les génomes ne peuvent pas être séquencés en un seul morceau. Ils sont donc fragmentés, les fragments sont ensuite amplifiés et enfin séquencés. À l'aide des multiples copies des fragments générés

il faut alors reconstruire le génome d'origine. Pour cela deux types de graphes sont principalement utilisés : les graphes de chevauchement et les graphes de de Bruijn. Pour les graphes de chevauchement, les sommets sont les fragments et il existe un arc entre deux sommets u et v si un suffixe de u est égal à un préfixe de v. L'arc est généralement valué par la longueur du chevauchement. Pour les graphes de de Bruijn (d'ordre k), les sommets sont les k-mers des fragments et il existe un arc entre deux sommets u et v si le suffixe de u de longueur $k-1$ est égal au préfixe de v. de longueur $k-1$. L'arc peut être étiqueté par la dernière lettre de v (voir Figure H.11).

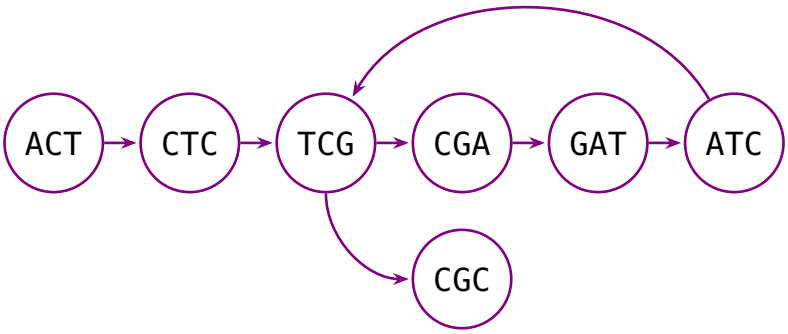


FIGURE H.11. Graphe de de Bruijn d'ordre 3 de $S = \{\texttt{ACTCGA}, \texttt{TCGAT}, \texttt{CGATC}, \texttt{TCGC}\}$. Chaque sommet du graphe correspond à un élément de S. Il existe une arête de TCG vers CGA car le suffixe de TCG dont la longueur est 2 est égal au préfixe de même longueur de CGA, c'est-à-dire à CG.

Nous avons établi un lien entre les arbres compacts des suffixes et les graphes de de Bruijn [33].

Les graphes de de Bruijn ont également été utilisés pour la correction de données de séquençage de troisième génération produisant des fragments longs et bruités [67, 61, 66].

H.3.6 Transformée de Burrows-Wheeler (*)

CONTACTS ET COMMENTAIRES: TIBS/Algorithmique du texte

L'une des problématiques qui a émergé avec l'accumulation des données (notamment avec le séquençage à haut débit) consiste à trouver des alternatives économique à ces graphes qui sont trop volumineux. La transformée de Burrows-Wheeler en est une. En effet, couplée à un tableau des suffixes échantillonné, il est possible de retrouver sensiblement les mêmes informations que dans l'arbre des suffixes compact avec un espace moindre que la séquence d'origine. Cette structure s'appelle un FM-index (voir Figure H.12).

Fiches synthétiques

i	SA	F										L	B^a	B^c	B^t
0	10	$	a	c	a	a	a	c	a	t	a	t	0	0	1
1	2	a	a	a	c	a	t	a	t	$	a	c	0	1	0
2	3	a	a	c	a	t	a	t	$	a	c	a	1	0	0
3	0	a	c	a	a	a	c	a	t	a	t	$	0	0	0
4	4	a	c	a	t	a	t	$	a	c	a	a	1	0	0
5	8	a	t	$	a	c	a	a	a	c	a	t	0	0	1
6	6	a	t	a	t	$	a	c	a	a	a	c	0	1	0
7	1	c	a	a	a	c	a	t	a	t	$	a	1	0	0
8	5	c	a	t	a	t	$	a	c	a	a	a	1	0	0
9	9	t	$	a	c	a	a	a	c	a	t	a	1	0	0
10	7	t	a	t	$	a	c	a	a	a	c	a	1	0	0
												BWT			

FIGURE H.12. La transformée de Burrows-Wheeler (BWT) de `acaaacatat$` correspond à la dernière colonne (L) de la matrice formée par les rotations cycliques de `acaaacatat$` triée dans l'ordre lexicographique. Cette dernière colonne est inversible (elle permet de retrouver la séquence d'origine) et peut être représentée par des vecteurs de bits (à droite). La colonne SA (*Suffix Array*) correspond aux positions de début des rotations cycliques. Elle peut être échantillonnée. Les vecteurs de bits et le SA échantillonné permettent de calculer les mêmes informations que l'arbre compact des suffixes de la même séquence.

Fiche synthétique TIBS/Algorithmique du texte

Mots clés : trie, arbres des suffixes, automate, graphe de de Bruijn, transformée de Burrows-Wheeler, indexation, recherche de mot

Problématiques scientifiques : L'algorithmique du texte vise à développer des structures de données et des algorithmes efficaces pour manipuler des suites de symboles (appelées mots, textes ou séquences). L'aspect sémantique sous-jacent à ces suites de symboles est rarement pris en compte. En algorithmique du texte, plusieurs types de graphes sont utilisés, par exemple :

- les *tries* (Section H.3.3) ;
- l'oracle des facteurs (Section H.3.4) ;
- les graphes de chevauchement et de de Bruijn (Section H.3.5) ;
- la transformée de Burrows-Wheeler (Section H.3.6).

Personnes de la fédération impliquées (principaux et permanents) :

Les membres suivants de TIBS participent à des travaux comme ceux décrits précédemment : Richard Groult, Yannick Guesnet, Thierry Lecroq Arnaud Lefebvre, Laurent Mouchard et Élise Prieur-Gaston.

GdR(s) concerné(s) :

Les membres présentent régulièrement leurs travaux aux journées du groupe de travail SeqBIM com-

muns aux GDR IM et BIM.

Quelques revues et conférences que vous visez :

Les principales revues visées sont *Theoretical Computer Science* pour les aspects d'algorithmique et *Bioinformatics* pour les aspects applicatifs en bioinformatique.

Les conférences du domaine sont CPM (*Combinatorial Pattern Matching*), SPIRE (*String Processing and Information Retrieval*) et PSC (*Prague Stringology Conference*).

Fiche synthétique C&A

Mots clés : Informatique fondamentale, décidabilité, théorie des langages, combinatoire des mots, codes, sécurité des données, cryptologie.

Problématiques scientifiques : Modélisation du calcul et de l'information, codage, cryptologie

Domaines applicatifs (exemples) : La plupart des résultats que nous obtenons sont théoriques et nous ne développons pas directement d'applications. Néanmoins, ils sont liés aux domaines applicatifs suivants : transmission, stockage et protection de l'information, représentation des documents, compilation, spécification et vérification, calcul formel, logique formelle.

Personnes de la fédération impliquées (principaux et permanents) : L'équipe C&A.

Labos/Industries impliqués en dehors de la fédération :

GdR(s) concerné(s) : Informatique Mathématique (Groupes DAAL [(1)], SDA2, C2, Calcul Formel), Sécurité Informatique (Groupe C2)

Quelques revues et conférences que vous visez :

Conférences : EuroCRYPT, MFCS, STACS, ICALP, FST&TCS, CIAA, DLT, FPSAC, LATA, PQCrypto, PKC, ISIT

Journaux : Discret. Math. Theor. Comput. Sci., Theoretical Computer Science, Journal of Symbolic Computation, Logical Methods in Computer Science, Int. J. Found. Comput. Sci., Fundam. Inform., Journal of Automata, Languages and Combinatorics, J. Discrete Algorithms, Linear and Multilinear Algebra, Advances in Applied Mathematics, Annales de l'Institut Henri Poincaré, Journal of Integer Sequences, Bull. of European Assoc. for Theor. Comput. Sci. (EATCS Bulletin), IEEE Trans. Information Theory, Des. Codes Cryptography

Ressources logicielles que vous auriez développées : pas de logiciel à proprement parler, mais des réponses à appels (NIST) à standardisation d'algorithmes en cryptographie.

Projets en cours ou passés [(2)] :

- SWAP (ANR), 2021-2025
- PQTLS (ANR PEPR Technologies quantiques), 2021– ??
- CBCrypt (ANR), 2017–2022
- ARTIQ (Région Normandie), 2018–2021
- MinMacs (Région Normandie), 2019–2022
- MOUSTIC (Région Normandie, FEDER), 2016–2020
- MIDI (Région Normandie), 2014–2015
- IThèm (PHC Maghreb), 2014–2016

Fiche synthétique AMACC

(1). Anciennement ALGA
(2). Nom, année et URLs si possible

Mots clés : Algorithmes, Modèles de Calcul, Systèmes dynamiques, Structures aléatoires, Analyse d'algorithmes, Combinatoire, Complexité.

Problématiques scientifiques : L'équipe AmacC s'intéresse à deux concepts génériques, l'algorithme et la complexité, qui sous-tendent toutes ses activités, avec des points de vue complémentaires. Elle étudie les modèles de calcul et la notion de complexité, dans le pire des cas, via les classes de complexité. Elle travaille aussi dans un cadre aléatoire, étudie des modèles probabilistes, qu'elle utilise notamment pour l'analyse de complexité en moyenne des algorithmes ou des structures de données associées.

Domaines applicatifs (exemples) : Les algoritmes étudiés sont ceux ceux qui interviennent en algorithmique des graphes, en algorithmique du texte ou encore en algorithmique arithmétique.

Personnes de la fédération impliquées : Toute l'équipe AMACC

Labos impliqués en dehors de la fédération : Le laboratoire CNRS international SINFIN entre la France et l'Argentine – Le laboratoire IRIF.

GdR où l'équipe s'implique. Le GdR IM (Informatique Mathématique) avec les groupes ALEA, DAAL (anciennement ALGA), Graphes, SDA2, SeqBim, C2, Calcul Formel ;

Revues et conférences dans lesquelles vous avez publié.

Conférences :

CPM (Combinatorial Pattern Matching)
ICALP (International Colloquium on Automata, Languages and Programming)
Conference on Computability in Europe
MFCS (Mathematical Foundations of Computer Science)
ISSAC (International Symposium on Symbolic and Algebraic Computation)
AofA (International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms)
LICS (Logic in Computer Science)
AUTOMATA (International Workshop on Cellular Automata and Discrete Complex Systems)
STACS (Symposium on Theoretical Aspects of Computer Science)
LATIN (Latin American Symposium in Theoretical Informatics

Journaux :

Theoretical Computer Science,
Journal of Symbolic Computation
Combinatorics, Probability and Computing
Theory of Computing Systems
Mathematical Structures in Computer Science
Annales de l'Institut Henri Poincaré (Combinatorics, Physics and their Interactions)
Communications in Mathematical Physics
The Electronic Journal of Combinatorics
Discrete Mathematics
Journal of Computer and System Sciences
IEEE Transactions on Information Theory
SIAM Journal on Computing

SIAM Journal on Discrete Mathematics

Ressources logicielles développées : Usain Boltz (2020) : Logiciel de génération aléatoire efficace de grandes structures de données (URL :https ://gitlab.com/ParComb/usain-boltz)

Projets en cours ou récents [(3)] :

International

- Laboratoire INFINIS puis SINFIN : (2018– ??) AMACC équipe de ce laboratoire
- Projets STIC AmSud : AleaEnAmSud (2017–2019) puis RAPA2 (2020–2022) (co-portés par le GREYC)

National

- Projet ANR PING/ACK (2019–2020) (participation)
- Projet ANR CODYS (2019–2023) (participation)
- Projet ANR JCJC C-SyDiSi (2019–2023) (porté)
- PEPS CNRS Aztek (2018–2019) (porté)
- PEPS CNRS HYDRATA (2016–2017) (porté)
- W2Next (2018–2019), avec l'entreprise W2Next (co-porté)

Régional

- Projet RIN ALENOR (2019–2022). Partenaires : GREYC, LITIS, LMRS. (porté)
- Projet EUR MinMacs, (2019–2022) participation

(3). Nom, année et URLs si possible

Bibliographie

[1] A. V. Aho and M. J. Corasick. Efficient string matching : an aid to bibliographic search. *Commun. ACM*, 18(6) :333–340, 1975.

[2] Cyril Allauzen, Maxime Crochemore, and Mathieu Raffinot. Factor oracle : A new structure for pattern matching. In Jan Pavelka, Gerard Tel, and Miroslav Bartosek, editors, *SOFSEM '99, Theory and Practice of Informatics, 26th Conference on Current Trends in Theory and Practice of Informatics, Milovy, Czech Republic, November 27 - December 4, 1999, Proceedings*, volume 1725 of *Lecture Notes in Computer Science*, pages 295–310. Springer, 1999.

[3] Cyril Allauzen, Maxime Crochemore, and Mathieu Raffinot. Factor oracle : A new structure for pattern matching. In Jan Pavelka, Gerard Tel, and Miroslav Bartosek, editors, *SOFSEM '99, Theory and Practice of Informatics, 26th Conference on Current Trends in Theory and Practice of Informatics, Milovy, Czech Republic, November 27 - December 4, 1999, Proceedings*, volume 1725 of *Lecture Notes in Computer Science*, pages 295–310. Springer, 1999.

[4] Amazigh Amrane and Nicolas Bedon. Logic and rational languages of scattered and countable series-parallel posets. *CoRR*, abs/1912.10240, 2019.

[5] Amazigh Amrane and Nicolas Bedon. Logic and rational languages of scattered and countable series-parallel posets. In Carlos Martín-Vide, Alexander Okhotin, and Dana Shapira, editors, *Language and Automata Theory and Applications - 13th International Conference, LATA 2019, St. Petersburg, Russia, March 26-29, 2019, Proceedings*, volume 11417 of *Lect. Notes in Comput. Sci.*, pages 275–287, St. Petersburg,Russian Federation, 2019. Springer.

[6] Amazigh Amrane and Nicolas Bedon. Equational theories of scattered and countable series-parallel posets. In Nataša Jonoska and Dmytro Savchuk, editors, *Developments in Language Theory*, LNCS, pages 1–13, Cham, 2020. Springer International Publishing.

[7] Amazigh Amrane and Nicolas Bedon. Logic and rational languages of scattered and countable series-parallel posets. *Theoretical Computer Science*, 809 :538–562, 2020.

[8] Gérard Assayag and Shlomo Dubnov. Using factor oracles for machine improvisation. *Soft Comput.*, 8(9) :604–610, 2004.

[9] László Babai. Graph isomorphism in quasipolynomial time. *CoRR*, abs/1512.03547, 2015.

[10] László Babai, Paolo Codenotti, Joshua A. Grochow, and Youming Qiao. *Code Equivalence and Group Isomorphism*, pages 1395–1408. 2011.

[11] Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzere, Sébastien Adam, and Paul Honeine. Analyzing the expressive power of graph neural networks in a spectral perspective. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[12] Magali Bardet, Julia Chaulet, Vlad Dragoi, Ayoub Otmani, and Jean-Pierre Tillich. Cryptanalysis of the mceliece public key cryptosystem based on polar codes. In Tsuyoshi Takagi, editor, *Post-Quantum Cryptography*, pages 118–143, Cham, 2016. Springer International Publishing.

[13] Magali Bardet, Ayoub Otmani, and Mohamed Saeed-Taha. Permutation code equivalence is not harder than graph isomorphism when hulls are trivial. In *IEEE International Symposium on Information Theory, ISIT 2019, Paris, France, July 7–12, 2019*, pages 2464–2468. IEEE, 2019.

[14] Magali Bardet, Ayoub Otmani, and Mohamed Saeed-Taha. Permutation code equivalence is not harder than graph isomorphism when hulls are trivial. In *IEEE International Symposium on Information Theory, ISIT 2019, Paris, France, July 7-12, 2019*, pages 2464–2468. IEEE, 2019.

[15] Nicolas Bedon. Logic and branching automata. *Logical Methods in Computer Sciences*, 11(4 :2) :1–38, October 2015.

[16] Nicolas Bedon. Logic and branching automata. *Logical Methods in Computer Sciences*, 11(4 :2) :1–38, October 2015.

[17] Nicolas Bedon. Branching Automata and Pomset Automata. In Mikołaj Bojańczy and Chandra Chekuri, editors, *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021)*, volume 213 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 37 :1–37 :13, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[18] Nicolas Bedon and Chloé Rispal. Series-parallel languages on scattered and countable posets. *Theoretical Computer Science*, 412(22) :2356–2369, 2011.

[19] Sandeep Bhadra and Afonso Ferreira. Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. In *International Conference on Ad-Hoc Networks and Wireless*, pages 259–270. Springer, 2003.

[20] David Blumenthal, SÃ©bastien Bougleux, Johann Gamper, and Luc Brun. Gedlib : A c++ library for graph edit distance computation. In Donatello Conte, Jean-Yves Ramel, and Pasquale Foggia, editors, *12th IAPR TC15 Workshop on Graph-Based Representation in Pattern Recognition (GbR)*, volume 11510 of *LNCS*, pages 14–24, Tours, June 2019. IAPR TC15, Springer.

[21] David B. Blumenthal, Nicolas Boria, Sébastien Bougleux, Luc Brun, Johann Gamper, and Benoit Gaüzère. Scalable generalized median graph estimation and its manifold use in bioinformatics, clustering, classification, and indexing. *Information Systems*, 100 :101766, 2021.

[22] Wieb Bosma, John Cannon, and Catherine Playoust. The magma algebra system i : The user language. *Journal of Symbolic Computation*, 24(3) :235–265, 1997.

[23] Sébastien Bougleux, Luc Brun, Vincenzo Carletti, Pasquale Foggia, Benoit Gaüzère, and Mario Vento. Graph edit distance as a quadratic assignment problem. *Pattern Recognition Letters*, 87 :38–46, 2017. Advances in Graph-based Pattern Recognition.

[24] Sébastien Bougleux, Benoit Gaüzère, David B. Blumenthal, and Luc Brun. Fast linear sum assignment with error-correction and no cost constraints. *Pattern Recognition Letters*, 134 :37–45, 2020. Applications of Graph-based Techniques to Pattern Recognition.

[25] Iliya G. Bouyukliev. *About the code equivalence*, pages 126–151. 2007.

[26] R.W. Brockett and P. Maragos. Evolution equations for continuous-scale morphological filtering. *IEEE Transactions on Signal Processing*, 42(12) :3377–3386, 1994.

[27] Luc Brun, Benoit Gaüzère, Sébastien Bougleux, and Florian Yger. A new Sinkhorn algorithm with Deletion and Insertion operations. Research report, GREYC, UMR 6072, November 2021.

[28] J. Richard Büchi. Transfinite automata recursions and weak second order theory of ordinals. In *Proc. Int. Congress Logic, Methodology, and Philosophy of Science (1964)*, pages 2–23, Jerusalem, 1965. North-Holland Publ. Co. Invited address.

[29] J. Richard Büchi and Dirk Siefkes. *The monadic second order theory of all countable ordinals*, volume 328 of *Lecture notes in mathematics*. Springer, 1973.

[30] Binh-Minh Bui-Xuan, Afonso Ferreira, and Aubin Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(02) :267–285, 2003.

[31] R. Carmo, J. Donadelli, Y. Kohayakawa, and E. Laber. Searching in random partially ordered sets. *Theoret. Comput. Sci.*, 321(1) :41–57, 2004.

[32] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5) :387–408, 2012.

[33] Bastien Cazaux, Thierry Lecroq, and Eric Rivals. Linking indexing data structures to de Bruijn graphs : Construction and update. *J. Comput. Syst. Sci.*, 104 :165–183, 2019.

[34] Christian Couder. Fighting regressions with git bisect. `https://git-scm.com/docs/git-bisect-lk2009`, 2009.

[35] Bruno Courcelle and J. Engelfriet. *Graph structure and monadic second-order logic, a language theoretic approach*. Cambridge university Press, 2011.

[36] Maxime Crochemore, Christophe Hancart, and Thierry Lecroq. *Algorithms on strings*. Cambridge University Press, 2007.

[37] Julien David, Loïck Lhote, Arnaud Mary, and François Rioult. An average study of hypergraphs and their minimal transversals. *Theor. Comput. Sci.*, 596 :124–141, 2015.

[38] Julien David, Loïck Lhote, Arnaud Mary, and François Rioult. An average study of hypergraphs and their minimal transversals. *Theor. Comput. Sci.*, 596 :124–141, 2015.

[39] Géraldine Del Mondo, Peng Peng, Jérôme Gensel, Christophe Claramunt, and Feng Lu. Leveraging spatio-temporal graphs and knowledge graphs : Perspectives in the field of maritime transportation. *ISPRS International Journal of Geo-Information*, 10(8), 2021.

[40] V Diekert and G Rozenberg. *The Book of Traces*. WORLD SCIENTIFIC, 1995.

[41] Paul Dorbec, Michael A. Henning, Mickael Montassier, and Justin Southey. Independent domination in cubic graphs. *J. Graph Theory*, 80(4) :329–349, 2015.

[42] François-Xavier Dupé. *Restauration et reconnaissance des formes dans des images avec bruit de Poisson : application à l'analyse des neurones en microscopies de fluorescence*. Theses, Université de Caen, January 2010.

[43] Thomas Feulner. The automorphism groups of linear codes and canonical representatives of their semilinear isometry classes. *Advances in Mathematics of Communications*, 3(4) :363–383, 2009.

[44] Benoit Gaüzère. *Application des méthodes à noyaux sur graphes pour la prédiction des propriétés des molécules*. Theses, Université de Caen, November 2013.

[45] Benoit Ga"uz'ere, Pierre-Anthony Grenier, Luc Brun, and Didier Villemin. Treelet kernel incorporating cyclic, stereo and inter pattern information in chemoinformatics. *Pattern Recognition*, 48(2) :356–367, February 2015.

[46] Pierre-Anthony Grenier. *Modélisation de la stéréochimie : une application à la chémoinformatique*. Theses, Université de Caen Normandie, November 2015.

[47] Frédéric Guinand, François Guerin, and Mark Bastourous. Alignment of Three Robots without Communication nor Localization*. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 647–654, Bari, Italy, October 2019. IEEE.

[48] John. E. Hopcroft, Rajeev Motwani, and Jeffrey. D. Ullman. *Introduction to automata theory, languages, and computation*. Pearson, 3 edition, 2006.

[49] Linlin Jia, Benoit Gaüzère, and Paul Honeine. A graph pre-image method based on graph edit distances. In *Proceedings of IAPR Joint International Workshops on Statistical techniques in Pattern Recognition (SPR 2020) and Structural and Syntactic Pattern Recognition (SSPR 2020).*, 2021.

[50] Tobias Kappé, Paul Brunet, Bas Luttik, Alexandra Silva, and Fabio Zanasi. Brzozowski goes concurrent - A Kleene theorem for pomset languages. In *Proc. CONCUR 2017 : 28th International Conference on Concurrency Theory*, volume 28, pages 21 :1–21 :15, 01 2017.

[51] Tobias Kappé, Paul Brunet, Bas Luttik, Alexandra Silva, and Fabio Zanasi. On series-parallel pomset languages : Rationality, context-freeness and automata. *Journal of Logical and Algebraic Methods in Programming*, 103, 12 2018.

[52] David Kempe, Jon Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64(4) :820–842, 2002.

[53] Stephen C. Kleene. Representation of events in nerve nets and finite automata. In Shannon and McCarthy, editors, *Automata studies*, pages 3–41, Princeton, New Jersey, 1956. Princeton University Press.

[54] A. Lefebvre and T. Lecroq. Computing repeated factors with a factor oracle. In L. Brankovic and J. Ryan, editors, *Proceedings of the 11th Australasian Workshop On Combinatorial Algorithms*, pages 145–158, Hunter Valley, Australia, 2000.

[55] Arnaud Lefebvre and Thierry Lecroq. Compror : On-line lossless data compression with a factor oracle. *Inf. Process. Lett.*, 83(1) :1–6, 2002.

[56] Arnaud Lefebvre, Thierry Lecroq, Hélène Dauchel, and Joël Alexandre. Forrepeats : detects repeats on entire chromosomes and between genomes. *Bioinform.*, 19(3) :319–326, 2003.

[57] J. Leon. Computing automorphism groups of error-correcting codes. *IEEE Transactions on Information Theory*, 28(3) :496–511, 1982.

[58] Kamal Lodaya and Pascal Weil. A Kleene iteration for parallelism. In V. Arvind and R. Ramanujam, editors, *Foundations of Software Technology and Theoretical Computer Science*, volume 1530 of *Lect. Notes in Comput. Sci.*, pages 355–367. Springer-Verlag, 1998.

[59] Kamal Lodaya and Pascal Weil. Series-parallel posets : algebra, automata and languages. In M. Morvan, Ch. Meinel, and D. Krob, editors, *STACS'98*, volume 1373 of *Lect. Notes in Comput. Sci.*, pages 555–565. Springer-Verlag, 1998.

[60] Kamal Lodaya and Pascal Weil. Series-parallel languages and the bounded-width property. *Theoretical Computer Science*, 237(1–2) :347–380, 2000.

[61] Camille Marchet, Pierre Morisse, Lolita Lecompte, Arnaud Lefebvre, Thierry Lecroq, Pierre Peterlongo, and Antoine Limasset. ELECTOR : evaluator for long reads correction methods. *NAR Genomics and Bioinformatics*, 2(1) :1–12, 2020.

[62] R. J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. *Deep Space Network Progress Report*, 44 :114–116, January 1978.

[63] R. J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. *Deep Space Network Progress Report*, 44 :114–116, January 1978.

[64] Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, ii. *Journal of Symbolic Computation*, 60 :94–112, 2014.

[65] Lorenz Minder and Amin Shokrollahi. Cryptanalysis of the sidelnikov cryptosystem. In Moni Naor, editor, *Advances in Cryptology - EUROCRYPT 2007*, pages 347–360, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[66] P Morisse, C Marchet, A Limasset, T Lecroq, and A Lefebvre. Scalable long read self-correction and assembly polishing with multiple sequence alignment. *Sci. Rep.*, 11(1) :761, 2021.

[67] Pierre Morisse, Thierry Lecroq, and Arnaud Lefebvre. Hybrid correction of highly noisy long reads using a variable-order de Bruijn graph. *Bioinform.*, 34(24) :4213–4222, 2018.

[68] Pierre Morisse, Thierry Lecroq, and Arnaud Lefebvre. Hybrid correction of highly noisy long reads using a variable-order de Bruijn graph. *Bioinform.*, 34(24) :4213–4222, 2018.

[69] Kamaldeep S Oberoi, Géraldine Del Mondo, Yohan Dupuis, and Pascal Vasseur. Towards a qualitative spatial model for road traffic in urban environment. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1724–1729. IEEE, Oct 2017.

[70] Kamaldeep Singh Oberoi. *Spatio-temporal modeling of urban road traffic*. Theses, Normandie Université, November 2019.

[71] Kamaldeep Singh Oberoi, Géraldine Del Mondo, Yohan Dupuis, and Pascal Vasseur. Modeling Road Traffic Takes Time. In *10th International Conference on Geographic Information Science (GIScience 2018)*, volume 114 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 52 :1–52 :7, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[72] Louise Penz. *Interconnexion d'un système de transport point à point et d'un réseau de transport en commun (en cours)*. Theses, Université du Havre.

[73] E. Petrank and R.M. Roth. Is code equivalence easy to decide? *IEEE Transactions on Information Theory*, 43(5) :1602–1604, 1997.

[74] Laura Pinson, Géraldine Del Mondo, and Pierrick Tranouez. Representation of interdependencies between urban networks by a multi-layer graph. In *14th International Conference on Spatial Information Theory (COSIT 2019)*, volume 142, Regensburg, Germany, Sep 2019. Sabine Timpf, Christoph Schlieder, Markus Kattenbeck, Bernd Ludwig, and Kathleen Stewart eds.

[75] Bruce Reed. Paths, stars and the number three. *Combinatorics, Probability and Computing*, 5(3) :277–295, 1996.

[76] Guillaume Renton. *Analyzing and improving graph neural networks*. Theses, Normandie Université, July 2021.

[77] Mohamed Ahmed Saeed-Taha. *Algebraic Approach for Code Equivalence*. PhD thesis, Normandy University, 2017.

[78] N. Sendrier. Finding the permutation between equivalent linear codes : the support splitting algorithm. *IEEE Transactions on Information Theory*, 46(4) :1193–1203, 2000.

[79] Nicolas Sendrier. On the dimension of the hull. *SIAM Journal on Discrete Mathematics*, 10(2) :282–293, 1997.

[80] Nicolas Sendrier. *Cryptosystèmes à clé publique basés sur les codes correcteurs d'erreurs*. PhD thesis, Université Paris 6, 2002. Mémoire d'habilitation à diriger des recherches.

[81] Nicolas Sendrier and Dimitris E. Simos. The hardness of code equivalence over and its application to code-based cryptography. In *Post-Quantum Cryptography*, volume 7932 of *Lect. Notes in Comput. Sci.*, pages 203–216. Springer, 2013.

[82] Frédéric Suard. *Méthodes à noyaux pour la détection de piétons*. Theses, INSA de Rouen, December 2006.

[83] Michele Tirico. *Morphogenesis of complex networks. An application in urban growth*. phdthesis, Normandie Université, 2020.

[84] Michele Tirico, Stefan Balev, Antoine Dutot, and Damien Olivier. Morphogenesis of complex networks : A reaction diffusion framework for spatial graphs. In Luca Maria Aiello, Chantal Cherifi, Hocine Cherifi, Renaud Lambiotte, Pietro Lió, and Luis M. Rocha, editors, *Complex Networks and Their Applications VII*, Studies in Computational Intelligence, pages 769–781. Springer International Publishing, 2019.

[85] Jacobo Valdes, Robert E. Tarjan, and Eugene L. Lawler. The recognition of series parallel digraphs. *SIAM J. Comput.*, 11 :298–313, 1982.

[86] Mathilde Vernet. *Modèles et algorithmes pour les graphes dynamiques*. Theses, Normandie Université, October 2020.

[87] Mathilde Vernet, Yoann Pigné, and Éric Sanlaville. A study of connectivity on dynamic graphs : computing persistent connected components. *4OR*, 2022.

[88] P. Weiner. Linear pattern matching algorithm. In *Proceedings of the 14th Annual IEEE Symposium on Switching and Automata Theory*, pages 1–11, Washington, DC, 1973.