

ECOLE POLYTECHNIQUE
PROMOTION X2016
MASTER : Mathématiques appliquées
PARMENTIER Pimprenelle

RAPPORT DE STAGE

Stream graphs multicouches

RAPPORT NON CONFIDENTIEL

Option : Mathématiques appliquées

Champ : Recherche opérationnelle, théorie des graphes

Enseignant référent : Xavier ALLAMIGEON

Tuteur de stage dans l'organisme : Tiphaine VIARD

Co-tuteurs de stage hors organisme : Benjamin RENOUST (Université d'Osaka)

Jean-François BAFFIER (Japan Society for the Promotion of Science)

Dates du stage : 8 avril 2019 - 23 août 2019

Adresse de l'organisme :

RIKEN AIP

Nihonbashi 1-chome

Mitsui Building, 15th floor,

1-4-1 Nihonbashi,

Chuo-ku, Tokyo

103-0027, Japan

Déclaration d'intégrité relative au plagiat

Je soussignée PARMENTIER Pimprenelle certifie sur l'honneur :

1. Que les résultats décrits dans ce rapport sont l'aboutissement de mon travail.
2. Que je suis l'auteur de ce rapport.
3. Que je n'ai pas utilisé des sources ou résultats tiers sans clairement les citer et les référencer selon les règles bibliographiques préconisées.

Mention à recopier

Je déclare que ce travail ne peut être suspecté de plagiat.

Date :

Signature :

Résumé

Ce rapport de stage traite d'un nouvel objet permettant d'étudier des graphes complexes dépendants du temps : les **stream graphs multicouches**.

Dans un premier lieu, nous faisons un état de l'art sur les formalismes existants pour traiter de graphes complexes et de graphes temporels. Nous présentons ainsi les graphes multicouches et les stream graphs qui nous servent à construire les stream graphs multicouches.

Dans un second temps, nous donnons une définition formelle des stream graphs multicouches, et nous montrons en quoi ce nouvel objet est une généralisation des objets existants. Puis nous définissons des mesures permettant d'étudier les stream graphs multicouches, qui prennent en compte l'aspect à la fois multicouches et temporel.

Nous présentons ensuite la bibliothèque Python conçue pour traiter ce type d'objet, ainsi qu'un exemple d'utilisation avec un jeu de données temporel multicouches.

Enfin, nous donnons un ensemble de pistes que nous pourrions explorer pour la suite du stage.

Abstract

This internship report is about a new object that deals with time-dependant complex graphs: the multilayer stream graphs.

First of all, we present the state-of-the-art on the existing formalisms that handle complex graphs and temporal graphs. We present multilayer graphs and stream graphs, on the basis of which we build the multilayer stream graphs.

Then, we give a formal definition of multilayer stream graphs, and we demonstrate that it is a generalisation of existing objects. We define measures to study the multilayer stream graphs, which takes into account both multilayer and temporal aspects.

We also introduce a Python library designed to handle this type of object, as well as an example of use with a multilayer temporal dataset.

Finally, we give a set of tracks that we can explore for the rest of the internship.

Table des matières

1	Etat de l’art : deux formalismes sur les graphes	5
1.1	Graphes	6
1.2	Les graphes multicouches	6
1.2.1	Définition générale	6
1.2.2	Cadres d’utilisation	7
1.2.3	Quelques définitions	8
1.2.4	Pourquoi utiliser des graphes multicouches à la place de graphes classiques ?	8
1.3	Les stream graphs	10
1.3.1	Définition	10
1.3.2	Stream graphs et graphes dynamiques	10
2	Présentation d’un nouvel objet : le stream graph multicouches	11
2.1	Définition	11
2.2	Extraction de sous-graphes	12
2.2.1	Projection par rapport au temps	12
2.2.2	Projection à partir de couches	13
2.3	Mesures	15
2.3.1	Nombre de nœuds et nombre de liens	15
2.3.2	Uniformité et densité	16
3	Application à des données concrètes	18
3.1	Structure de données et organisation du code	18
3.2	CPGE : interactions entre élèves d’un même établissement	19
3.2.1	Représentation graphique	20
3.2.2	Sous-graphes, sous stream graph et sous graphes multicouches	20
3.2.3	Étude des densités intra et inter-couches pour plusieurs aspects	21
4	Conclusions et perspectives	23

Remerciements

En premier lieu, je souhaite remercier ma tutrice Tiphaine VIARD, qui a rendu mon stage possible, ainsi que mes co-tuteurs Jean-François BAFFIER et Benjamin RENOUST pour leur accompagnement pendant la durée du stage. Grâce à leurs conseils et leur suivi, j'ai pu apprendre beaucoup de choses dans le domaine des graphes mais également sur la recherche en général.

Je remercie également les équipes administratives de RIKEN et de l'École Polytechnique pour leur support.

Introduction

Un graphe représente un ensemble d'entités (des nœuds), reliés par des liens. Ces données sont omniprésentes : des villes sont liées par des routes, des personnes par des relations d'amitié, etc. La première formalisation du concept remonterait à Euler, en 1750, qui énonça le problème des ponts de Königsberg [1, 3]. Cet objet a permis de nombreuses avancées dans plusieurs domaines : en algorithmique, en probabilités, en combinatoire, en optimisation ou plus récemment, pour l'analyse de données massives du monde réel (graphes de terrain).

La puissance de la théorie des graphes vient du fait qu'elle permet d'exprimer des problèmes réels variés dans un même formalisme très synthétique, auquel on applique des algorithmes spécifiques permettant d'accéder à de nombreux résultats. Malgré la simplicité apparente du modèle, ces résultats sont subtils, et très utilisés à l'heure actuelle (PageRanking [22], plus court chemin [8], *etc.*).

Ce formalisme peut être étendu pour capturer de la diversité dans les données : les liens peuvent être dirigés, pondérés, étiquetés, *etc.* Cependant, l'extension de ces notions sur ces graphes est loin d'être triviale, ce qui appelle à l'élaboration de nouveaux formalismes. Par ailleurs, de nombreuses données se présentent sous la forme d'interactions plutôt que de relations : cela peut être des ordinateurs qui échangent des paquets, ou encore des chercheurs collaborant sur un sujet. Ces interactions comportent par nature une dimension temporelle. Des données peuvent également présenter des structures particulières : par exemple, des villes sont liées par différents moyens de transport, ou encore nous pouvons nous intéresser à des banques implantées dans différents pays procédant à des transactions.

Notre objectif ici est d'élaborer un formalisme généralisant l'existant afin de représenter des interactions dépendant du temps, tout en prenant en compte la structure des nœuds et des liens : nous appelons cet objet le *stream graph multicouches*¹.

Dans une première partie, nous établissons l'état de l'art en rappelant les définitions pertinentes des graphes, des graphes multicouches et des *stream graphs*, et nous explorons les limites associées à ces formalismes. Ensuite, nous présentons et nous formalisons les *stream graphs multicouches*, ainsi que les définitions associées. Nous explorons en détail les relations entre notre formalisme et les théories existantes, notamment par le biais de projections qui nous définissons.

L'applicabilité de notre formalisme à des données réelles est cruciale et c'est pour cela que nous présentons une implémentation au section 3, ainsi que les résultats obtenus sur un premier jeu de données.

Enfin, nous évoquerons des perspectives ouvertes par ce travail.

1 Etat de l'art : deux formalismes sur les graphes

Dans cette partie, nous présentons les objets qui ont servi à construire les *stream graphs multicouches* : tout d'abord les **graphes**, puis les **graphes multicouches** qui servent à traiter de graphes à structure complexe, et les **stream graphs** qui permettent de modéliser des séquences d'interactions.

¹La traduction du terme anglais scientifique « *stream graph* » n'étant pas définie lors de la rédaction de ce rapport, nous avons décidé de le conserver dans la langue anglaise pour éviter toute ambiguïté.

1.1 Graphes

Un **graphe simple** G est un ensemble de nœuds V ainsi qu'un ensemble d'arêtes $E \subseteq V \otimes V$, chaque élément de E étant une paire non ordonnée de deux éléments de V . Les arêtes représentent une interaction entre deux nœuds. De façon générale, on désigne $V \otimes V$ l'ensemble des paires $\{u, v\}$ non ordonnées d'éléments de V où $u \neq v$.

Nous avons donné ici la version la plus basique des graphes, mais il existe de nombreuses variantes ; on peut prendre une paire ordonnée (les paires ordonnées de V sont écrites $(u, v) \in V \times V$) et on obtient un graphe dirigé, associer une fonction de poids à chaque lien (graphes pondérés), etc.

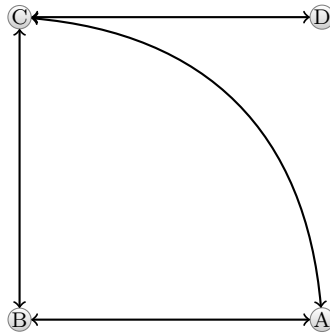


FIGURE 1 – Un exemple de graphe $G = (V, E)$ avec $V = \{A, B, C, D\}$ et $E = \{AB, AC, BC, DC\}$. Il représente des relations entre divers individus de la même espèce. Un lien existe entre deux animaux s'ils ont été en contact.

La théorie des graphes permet d'analyser des jeux de données à un niveau assez fin.

Par exemple, les mesures de **centralité** permettent de détecter quels sont les nœuds les plus « influents ». Il en existe plusieurs types, de plus en plus « fins ». Le *degré* qui est le nombre d'arêtes incidentes à nœud, est la première. Ensuite, on trouve la centralité d'intermédiarité (*betweenness centrality*) [19], dans laquelle on calcule combien de fois le nœud est présent dans les plus courts chemins entre toutes les paires du graphes. Enfin, dans des graphes orientés, on simule des marches aléatoires entre les nœuds du graphe et mesurer vers quels nœuds elles convergent [22]. Un autre exemple est le **partitionnement spectral** [23], permettant de diviser l'ensemble des nœuds en plusieurs sous-ensembles (clusters) distincts, de sorte à minimiser le nombre d'arêtes entre deux sous-ensemble différents (ce nombre est appelé « coupe »).

1.2 Les graphes multicouches

Les graphes permettent de synthétiser et de représenter des relations entre individus. Un seul type de relations entre un seul type d'individus peut être représenté, soit en ignorant leurs différences, soit en ne représentant qu'une partie des relations et des nœuds.

Les graphes multicouches permettent de pallier à cette perte d'informations.

1.2.1 Définition générale

Les graphes multicouches [13] sont utilisés pour décrire des interactions entre des nœuds qui peuvent être de différentes natures et/ou qui peuvent avoir des interactions de natures différentes.

Un **graphe multicouches** est un quadruplet $M = (V_M, E_M, V, \mathcal{L})$.

Sa *structure* est décrite par le k -uplet d'ensembles $\mathcal{L} = L_1, \dots, L_k$. Chaque L_i , appelé *aspect*, est un ensemble d'attributs, appelés *couches élémentaires*. Chaque *couche* correspond à un élément de $L = L_1 \times L_2 \times \dots \times L_k$.

V est l'ensemble des *nœuds*, l'ensemble $V_M \subseteq V \times L$ les *nœuds-couches*. Enfin $E_M \subseteq V_M \otimes V_M$ est l'ensemble des *arêtes*, une arête pouvant relier deux nœuds-couches entre eux.

La figure 2 donne un exemple de graphe multicouches.

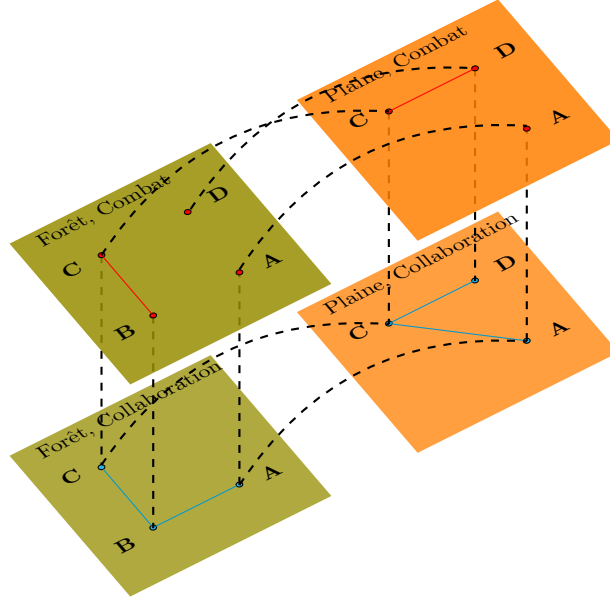


FIGURE 2 – **Représentation d'un graphe multicouches.** Ici, les nœuds sont des animaux d'une même espèce représentés par leurs initiales : $V = \{A, B, C, D\}$. Chaque couche est caractérisée par un milieu naturel et un type de relation : $L = \{\text{Milieu, type de relation}\}$. Les nœuds-couches existent ou non en fonction de la présence ou non d'un animal dans un milieu pour un type de relation : V_M contient par exemple $(A ; \text{Forêt, Combat})$. Enfin, les liens représentent les interactions entre les animaux dans un milieu précis : E_M contient par exemple $((B ; \text{Forêt, Collaboration}), (C ; \text{Forêt, Collaboration}))$.

1.2.2 Cadres d'utilisation

Les graphes multicouches trouvent de nombreuses applications, par exemple en **écologie** [21] (les nœuds sont alors des espèces ou des individus, les couches élémentaires des types d'interaction (parasite, prédateur...), des lieux, des temps, etc). On cherche alors à savoir comment se diffuse une maladie, quelles sont les faiblesses d'un écosystème par exemple en identifiant les liens/nœuds/couches centraux.

En **économie**, les relations entre les banques européennes [9] peuvent être modélisées par un graphe multicouches, les couches étant caractérisées par le type d'interaction (de « prêt »), les nœuds étant les banques et un lien existant entre deux banques de la même couche quand celles-ci procèdent à une interaction du type correspondant.

Un dernier exemple intuitif est celui des **réseaux de transport**, dans lequel les couches sont les différents moyens de transport possible, les nœuds sont des localités et les liens les différentes lignes de transport. Un tel jeu de données peut être trouvé sur le site gouvernemental américain des statistiques de transport, pour les lignes aériennes [16].

1.2.3 Quelques définitions

D'un point de vue pratique, les graphes multicouches peuvent être manipulés à l'aide de **tenseurs d'adjacence** [12], d'ordre 4, construits sur le modèle des matrices d'adjacence. Chaque élément $M_{v,\beta}^{u,\alpha}$ indique s'il existe un lien entre les nœuds-couches (u, α) et (v, β) .

On appelle **arêtes inter-couches** les arêtes qui lient deux nœuds-couches de couches différentes. L'ensemble des arêtes inter-couches s'écrit $E_{\text{inter}} = \{((u, \alpha), (v, \beta)) \in E_M \mid \alpha \neq \beta\}$. Dans l'exemple de la figure 2, les arêtes en pointillés sont les arêtes inter-couches.

On appelle **arêtes intra-couches** les arêtes qui lient deux nœuds-couches de la même couche. L'ensemble correspondant s'écrit $E_{\text{intra}} = E_M \setminus E_{\text{inter}} = \{((u, \alpha), (v, \beta)) \in E_M \mid \alpha = \beta\}$. Dans l'exemple de la figure 2, les arêtes intra-couches sont celles représentées en rouge et en bleu.

On appelle **arêtes de couplage** les arêtes inter-couches entre les mêmes nœuds : $E_{\text{couplage}} = \{((u, \alpha), (v, \beta)) \in E_M \mid u = v\}$

Remarque 1. On remarque que dans beaucoup de situations, nous avons des liens « implicites » entre tous les différents nœuds-couches issus du même nœud, et des liens « explicites » ne pouvant apparaître qu'au sein d'une même couche.

Les **multiplexes** sont des graphes multicouches dans lesquels les seuls liens intercouches sont des liens de couplage. L'exemple de la figure 2 est un multiplexe.

On appelle **graphe agrégé** le graphe que l'on obtient en « superposant » les couches d'un graphe multicouches :

$$Ag(M) = (V, E_G), \quad E_G = (u, v) \mid \exists \alpha, \beta \left(((u, \alpha), (v, \beta)) \in E_M \right)$$

Le **graphe sous-jacent** de M est le graphe obtenu en faisant abstraction de la structure multicouches, c'est à dire dans lequel chaque nœud est un nœud-couche.

$$Sj(M) = (V_M, E_M)$$

1.2.4 Pourquoi utiliser des graphes multicouches à la place de graphes classiques ?

Voici trois exemples d'application qui pourront être élargis au concept des stream graphs multicouches par la suite.

Importance de la structure : Isomorphismes de graphes Deux graphes G_1 et G_2 sont dits isomorphes quand on peut trouver une bijection des sommets du premier graphes vers ceux du second qui préserve les arêtes.

Dans le cadre des multicouches, Kivelä et Porter [10] définissent l'automorphisme de graphe (sans perte de généralité) par une permutation des arêtes, des couches élémentaires et des couches qui préserve les liens.

Kiveä et Porter démontrent qu'il n'est pas équivalent de dire que deux graphes multicouches sont isomorphes et leurs graphes sous-jacent sont équivalents. En effet, il ne faut pas perdre de vue que la structure « arborescente » des couches doit être conservée, et pas seulement l'idée de « partition » des nœuds en différentes couches. Ils présentent un algorithme permettant de vérifier si deux graphes multicouches sont isomorphes.

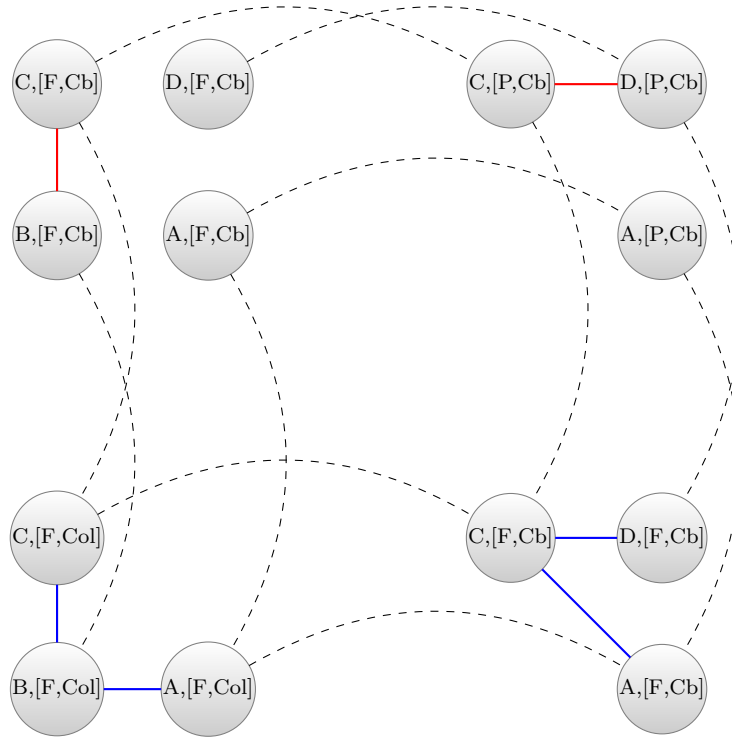


FIGURE 3 – Le graphe agrégé du graphe multicouche de la figure 2. Pour plus de lisibilité, nous avons différencié les arêtes dans leur représentation.

Centralité De Domenico [11] met en avant le fait qu’il est « réducteur » de calculer des centralités sur des graphes agrégés pour trouver quels sont les nœuds ayant les rôles les plus centraux dans la cohésion de l’ensemble d’une structure. Ils définissent donc un nouveau « PageRank » et une nouvelle centralité intermédiaire (betweenness centrality). Le PageRank est calculé à partir du tenseur d’adjacence du graphe, on obtient alors des centralités pour chaque nœud-couche, qu’on traite de la manière la plus adaptée en fonction du problème à résoudre.

La centralité d’intermédierité multicouche est calculée en sommant les centralités intermédiaires des différents nœuds-couches.

De Domenico montre que le fait de conserver l’information multicouche permet de trouver un classement des nœuds en fonction de leur centralité plus proche de la réalité (il compare ses résultats avec des données réelles sur les aéroports par exemple).

Intrication Dans le cas des graphes multiplexes, l’intrication[5] permet de mesurer à quel point différentes couches d’interactions se superposent. Un calcul matriciel permet d’exhiber des couches « centrales ». Cela permet de traiter des flux d’informations, par exemple savoir quel sujet (ou mot clé) génère le plus d’interactions sur un forum ou sur une chaîne de télé, *etc.*

Nous avons présenté ici trois exemples pour lesquels il a été judicieux de conserver l’aspect multicouche des graphes, parce qu’il contient plus d’informations, et qu’il a une structure particulière. Nous avons également montré comment construire des graphes « classiques » à partir de projections sur des graphes multicouches.

1.3 Les stream graphs

Nous présentons maintenant un autre outil récemment créé permettant de traiter de la temporalité dans les graphes : les stream graphs [14].

1.3.1 Définition

Un **stream graph** est un tuple $S = (T, W, V, E)$. T est un intervalle de temps. V est l'ensemble des nœuds, et $W \subseteq T \times V$ est l'ensemble des nœuds de V apparaissant en fonction du temps. Enfin, $E \subseteq T \otimes V \times V$ est l'ensemble des arêtes dont l'existence dépend également du temps.

Étant donné un nœud u , on appelle T_u l'ensemble des temps auxquels u apparaît, de même que T_{uv} décrit les temps d'apparition du lien uv :

$$T_u = \{t, (t, u) \in W\}$$

$$T_{uv} = \{t, (t, uv) \in E\}$$

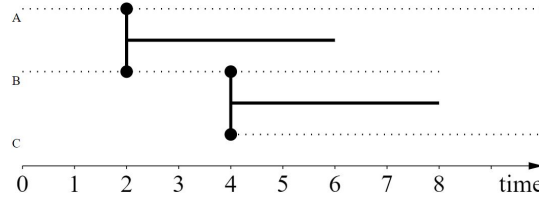


FIGURE 4 – **Exemple de stream graph** $S = (T, V, W, E)$ Dans cet exemple, on représente les interactions entre des animaux durant une expérience de 10 heures ($T = [0, 10]$). Les nœuds sont les trois individus ($V = \{A, B, C\}$), $W = [0, 10] \times A \cup [0, 8] \times B \cup [0, 10] \times C$ représente les moments où les individus étaient présents dans le milieu d'observation. Enfin les liens $E = ([2, 6] \times (A, B)) \cup ([4, 8] \times (B, C))$ représentent les moments où les individus ont interagi.

Dans l'article [14], de nombreuses notions sont ensuite définies de telle sorte à ce que dans le cas où le stream graph serait statique, on retrouve les mêmes notions que pour un graphe classique. Sont ainsi définis, entre autres, le *nombre de nœuds*, la *densité*, l'*uniformité*, la *compacité*, le *degré*, la *centralité intermédiaire* ainsi que les notions correspondant aux *chemins*, et aux *distances*.

1.3.2 Stream graphs et graphes dynamiques

La notion de stream graph est très récente (2017)[14] et ce n'est évidemment pas la première tentative de formalisme pour des graphes dynamiques[15].

Par exemple on peut citer la représentation souvent utilisée en recherche opérationnelle [7, 2], en « snapshot ». Le nombre de nœuds d'un système est multiplié par un nombre de pas de temps discrets $n = \frac{t}{T}$. On crée ensuite un graphe temporel dont les nœuds sont étiquetés (v, t_i) , v étant un nœud et t_i le temps auquel il est considéré. Les liens peuvent être présents entre différents nœuds du même pas de temps, et on ajoute à cela une arête entre tous les couples de nœuds de la forme $(v, t_i), (v, t_{i+1})$. Dans ces cas-là les liens sont très souvent orientés pour indiquer le sens du temps. Ces graphes sont utilisés pour des calculs de flots optimaux. On peut d'ailleurs raccrocher cette représentation à la théorie des graphes multicouches, chaque couche correspondant à un pas de temps[13].

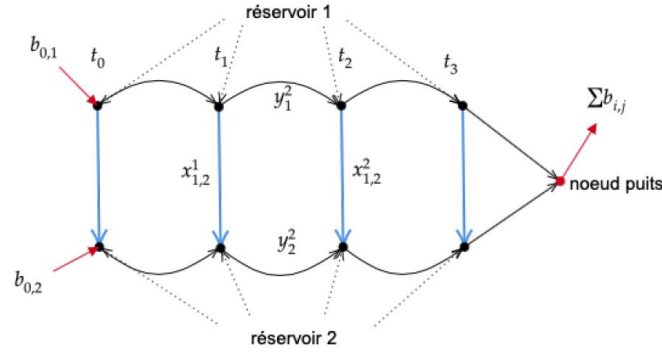


FIGURE 5 – **Exemple de snapshot** : Un « snapshot » représentant un barrage (liens bleus) entre deux réservoirs aux temps t_0, t_1, t_2, t_3 , tiré d'un projet MAP552

Cependant, cela conduit à avoir des graphes de très grande taille et il faut alors faire un choix entre précision dans le découpage en pas de temps et complexité. Ce choix non trivial, est encore un problème ouvert[24, 20]. De plus, dans les jeux de données réelles, les graphes sont souvent creux et leur représentation matricielle est alors très coûteuse inutilement en mémoire, d'autant plus si on augmente le nombre de pas de temps.

La notion de stream graph sert à représenter des données d'*interactions*, plus que des données de *relation*. Elle cherche à capturer à la fois leur composante structurelle et temporelle.

2 Présentation d'un nouvel objet : le stream graph multicouches

Nous avons donc présenté deux formalismes permettant de décrire deux généralisations des graphes. Les **graphes multicouches** servent à différencier plusieurs types de relations et de nœuds, qui appartiennent à des catégories discrètes différentes. Les stream graphs servent à décrire des interactions dépendantes du temps entre divers acteurs, qui peuvent également apparaître ou non au cours du temps.

Notre objectif est de généraliser deux modèles que nous avons présenté pour en créer un nouveau, que nous appelons *stream graph multicouches*, afin de pouvoir décrire des graphes à structures complexes qui dépendent du temps.

En premier lieu, nous définissons de façon formelle cet objet puis nous donnons quelques notions élémentaires, permettant notamment le lien avec les graphes multicouches et les stream graphs.

2.1 Définition

Un **stream graph multicouches** (ou *multilayer stream graph* en anglais) est un tuple $S_M = (T, T_M, V, W_M, E_M, \mathcal{L})$.

T est un intervalle de temps, \mathcal{L} la structure de couches définie comme dans le cadre des graphes multicouches section 1.2.1, et V un ensemble de nœuds. Une couche est un élément de $L = L_1 \times \dots \times L_k$, $\{L_1, \dots, L_k\} = \mathcal{L}$.

$T_M = \{T_\alpha \subseteq T, \alpha \in L\}$ est un ensemble d'intervalles, indexé par les couches de L . Chaque intervalle T_α représente le temps d'apparition de la couche α , avec pour contrainte que l'ensemble des intervalles d'apparition des couches recouvre T : $\cup_{\alpha \in L} T_\alpha = T$.

$W_M \subseteq T \times V \times L$, contient les temps d'existence de chaque nœud-couche, sachant que le temps d'existence d'un nœud-couche est inclus dans le temps d'existence de la couche : si $(t, u, \alpha) \in W_M$, cela signifie que le nœud u apparait dans la couche α au temps t . Enfin, $E_M \subseteq T \times V \times L \times V \times L$ donne les liens entre les nœuds couches et leurs temps d'existence, sachant qu'un lien ne peut exister que pendant l'intersection des temps d'existence des nœuds-couches qu'il relie.

On définit également les temps d'existence des nœuds-couches $T_{u,\alpha} = \{t | (t, u, \alpha) \in W_M\}$ et les temps d'existence des liens $T_{(u,\alpha),(v,\beta)} = \{t | (t, (u, \alpha), (v, \beta)) \in E_M\}$ qui sont des unions d'intervalles.

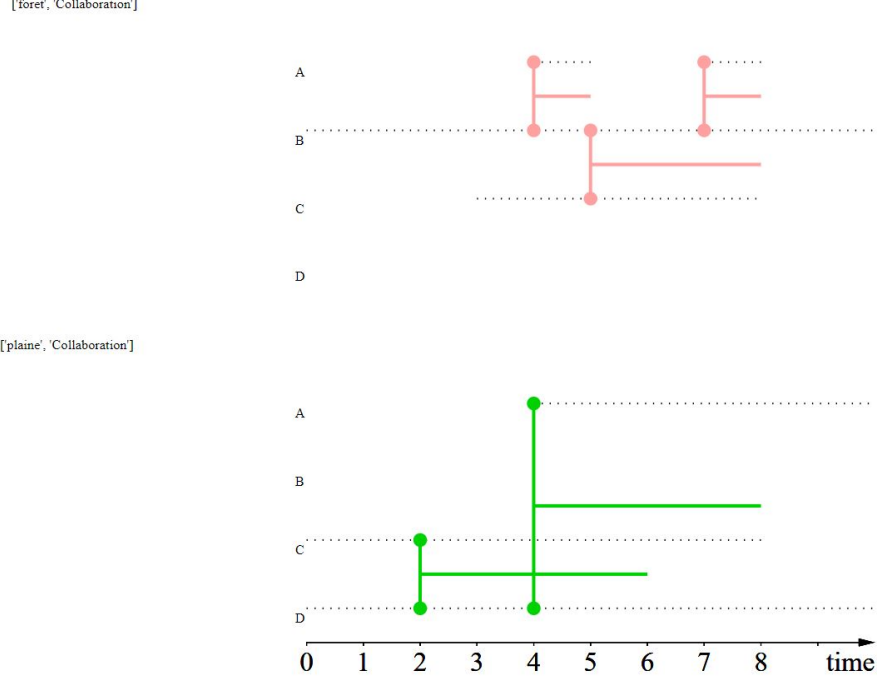


FIGURE 6 – La représentation en stream graph multicouches des couches ["plaine", "Collaboration"] et ["foret", "Collaboration"], générée avec « multiplex-stream » (description du programme en section 3.1)

2.2 Extraction de sous-graphes

2.2.1 Projection par rapport au temps

Nous définissons ici deux moyens d'extraire des graphes multicouches induits par un stream graph multicouches.

Le premier consiste à prendre un instantané au temps t :

Définition 1 (Graphe multicouche au temps t). Le *graphe multicouche au temps t* M_t s'écrit $M_t = (V_{M,t}, E_{M,t}, V, \mathcal{L})$ avec $V_{M,t}$ et $E_{M,t}$ contenant les nœuds-couches et les arrêtes apparaissant au temps t .

$$V_{M,t} = \{(u, \alpha) | (t, u, \alpha) \in W_M\}$$

$$E_{M,t} = \{(u, \alpha, v, \beta) | (t, u, \alpha, v, \beta) \in E_M\}$$

(Exemple de graphe multicouches au temps $t = 4$ et $t = 7$ à la figure 9.)

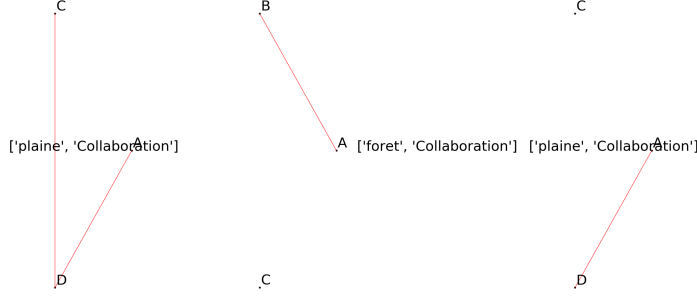


FIGURE 7 – Extraction du stream graph multicouches de la figure 6 au temps $t = 4$

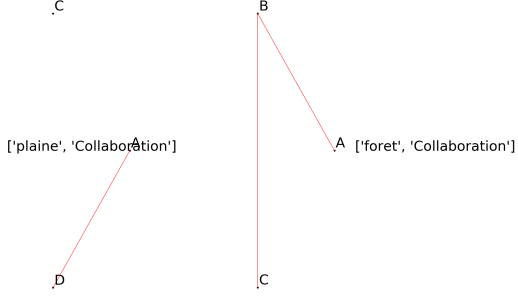


FIGURE 8 – Extraction du stream graph multicouches de la figure 6 au temps $t = 7$

FIGURE 9 – Dessin d'une extraction

Le second consiste à « enregistrer » toutes les interactions pendant un laps de temps.

Définition 2 (Graphe multicouches induit). Le *graphe multicouches induit sur l'intervalle* I $M_I(S_M) = (V_{M,I}, E_{M,I}, V, L)$ de S_M est le graphe multicouche qui rassemble toutes les couches, nœuds couches et nœuds qui apparaissent durant I .

$$V_{M,I} = \bigcup_{t \in I} V_{M,t}$$

$$E_{M,I} = \bigcup_{t \in I} E_{M,t}$$

(Exemple de graphe multicouche induit sur $[0, 10]$ à la figure 10)

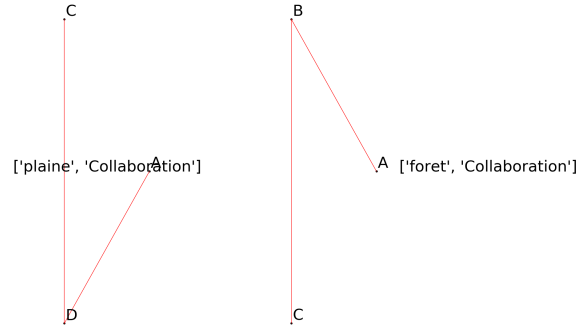


FIGURE 10 – Graphe multicouche induit du stream graph multicouches de la figure 6.

2.2.2 Projection à partir de couches

Comme dans l'article de Kivelä[13] pour les graphes multicouches, nous pouvons classer les arêtes en plusieurs catégories :

Définition 3 (Arêtes de couplage, arêtes intra-couches et inter-couches). Les *arêtes de couplage* sont définies par l'ensemble $E_C = \{(t, u, \alpha, v, \beta) \in E_M | u = v\}$.

Les *arêtes intra-couches* sont définies par l'ensemble $E_I = \{(t, u, \alpha, v, \beta) \in E_M | \alpha = \beta\}$.

Les *arêtes inter-couches* sont définies par l'ensemble $\bar{E}_I = E_M \setminus E_I$.

Nous définissons ici de façon formelle les stream graphs multiplexes.

Définition 4 (Stream graphs multiplexe). Un *stream graph multiplexe* S_{mp} est un stream graph multicouches dans lequel chaque arête est une arête intra-couche ou une arête de couplage. $S_{mp} = (T, T_M, V, W_M, E_M, \mathcal{L})$ avec $E_M \subseteq \{(t, (u, \alpha), (v, \beta)) | u = v \text{ ou } \alpha = \beta\}$

Comme dit précédemment, les graphes multiplexes sont présents dans beaucoup de cas concrets et font l'objet d'études qui leur sont particulièrement dédiées et que nous pouvons adapter aux stream graphs multiplexes.

Il peut être intéressant d'extraire certaines couches du stream graph multicouches, ou au contraire de se focaliser sur des interactions entre couches.

Définition 5 (Stream graph intra-couche). Pour chaque couche $\alpha \in L_1 \times \dots \times L_d$, le *stream graph intra-couche* S^α est le stream graph $S^\alpha = (T_\alpha, V^\alpha, W^\alpha, E^\alpha)$ tel que $T_\alpha \in T_M$ est l'intervalle d'existence de α . V^α est l'ensemble des nœuds couches dans la couche α et W^α représente leurs temps d'apparence dans la couche α . E^α est le sous-ensemble E_M contenant seulement les arêtes entre les nœuds-couches de α .

Définition 6 (Stream graph inter-couches). Étant donné un doublet de couches $\alpha, \beta \in L_1 \times \dots \times L_d$, le *stream graph inter-couches* est le stream graph $S^{(\alpha, \beta)} = (T^{\alpha, \beta}, V^{\alpha, \beta}, W^{\alpha, \beta}, E^{\alpha, \beta})$ tel que : $T^{\alpha, \beta} = T^\alpha \cap T^\beta$ est l'intervalle durant lequel α et β apparaissent simultanément. $V^{\alpha, \beta}$ sont tous les nœuds-couches du stream graph multicouches qui sont dans les couches α et β , $W^{\alpha, \beta}$ décrit leurs intervalles d'existence. Enfin, $E^{\alpha, \beta}$ sont les liens non orientés entre les nœuds couches des couches de α et β avec leurs temps d'existence.

Pour toute couche α , nous définissons $S^{(\alpha)} = S^{(\alpha, \alpha)}$.

L'étude des opérations d'union et d'intersection permet de trouver des propriétés pertinentes.

Définition 7 (Intersection de deux stream graph). L'*intersection* de deux stream graphs S_1 et S_2 est un stream graph et elle est définie de la façon suivante :

$$S' = S_1 \cap S_2 = (T_1 \cap T_2, V_1 \cap V_2, W_1 \cap W_2, E_1 \cap E_2)$$

Définition 8 (Union de deux stream graphs). L'*union* de deux stream graphs S_1 et S_2 est un stream graph et est définie de la façon suivante :

$$S' = S_1 \cup S_2 = (T', V_1 \cup V_2, W_1 \cup W_2, E' \cup E_2)$$

T' est un intervalle :

$$T' = [\min(T_1, T_2), \max(T_1, T_2)]$$

$$E' = E_1 \cup E_2$$

Définition 9 (Stream graph sous-jacent). Le *stream graph sous-jacent* $S_U(S_M)$ de S_M est (T, V_M, W_M, E_M) . C'est un stream graph dans lequel les nœuds sont les nœuds-couches de S_M . Il peut être partitionné selon les différentes couches.

Définition 10 (Stream graph agrégé). Le *stream graph agrégé* $S_A(S_M) = (T, V, W_A, E_A)$ a le même intervalle d'étude T que S_M . Ses nœuds sont l'ensemble V de ceux de S_M . Leurs temps d'existences sont l'union de leurs temps d'existences sur les différentes couches : $T_u = \bigcup_{\alpha \in L} T_{u, \alpha}$ et $W_A = \bigcup_{u \in V} T_u \times \{u\}$. Une arête existe entre deux nœuds de $S_A(S_M)$ si elle existe au même temps entre deux nœuds-couches correspondants de S_M : $E_A = \{(t, u, v) | \exists (\alpha, \beta) \in L^2, (t, (u, \alpha), (v, \beta)) \in E_M\}$

Nous démontrons que le stream graph sous-jacent $S_U(S_M)$ est égal à l'union de tous les stream graph inter et intra couches.

Propriété 1.

$$\bigcup_{(\alpha,\beta) \in L^2} S^{(\alpha,\beta)} = S_U(S_M)$$

Démonstration. On appelle $S = \bigcup_{(\alpha,\beta) \in L^2} S^{(\alpha,\beta)} = (T^*, V^*, W^*, E^*)$. Montrons que $S = S_U(S_M) = (T_U, V_U, W_U, E_U)$.

$T = T_U$ par définition de S_U . Par construction des stream graphs multicouches, pour tout t dans T , il existe au moins une couche α . Comme $T^* = [\min_{\alpha \in L}(T^\alpha), \max_{\alpha \in L}(T^\alpha)]$, t est forcément inclus dans T^* . Chaque T^α est inclus dans T , donc $T^* \subseteq T$, donc $T^* = T_U = T$.

$V_M = V_U$ par définition de V_U . $V^* = \bigcup_{(\alpha,\beta) \in L^2} V^{\alpha,\beta} \subseteq V_M$. Tous les nœuds de toutes les couches sont inclus dans $\bigcup_{(\alpha) \in L^2} V^{\alpha,\alpha} \subseteq V^*$, on obtient donc l'égalité $V_M = V_U = V^*$. Avec le même raisonnement, nous trouvons $W_M = W_U = W^*$.

$E_U = E_M$ et $E^* = \bigcup_{\alpha,\beta \in L^2} (E^{\alpha,\beta})$ par définition. Comme $E^{\alpha,\beta}$ est un sous-ensemble de E_M pour chaque couple de couches α, β , $E^* \subset E_M$. On sait également que pour toute arête $e = (t, u, \alpha, v, \beta) \in E_M$, e appartient à $E^{\alpha,\beta}$. Donc $E_M = E^*$.

On a donc bien une égalité terme à terme entre tous les éléments de S et de $S_U(S_M)$. \square

Nous avons donc exhibé un certain nombre d'extractions que nous pourrions maintenant combiner pour obtenir des graphes, des stream graph ou des graphes multiplexes, en fonction des résultats que nous voudrions obtenir.

2.3 Mesures

Nous avons créé des mesures permettant de mesurer la présence de liens et de nœuds dans les graphes. Nous voulons que ces mesures soient adaptées à l'objet spécifique que nous étudions (et qu'elles prennent en compte l'aspect multicouche et l'aspect temporel de façon simultanée), qu'elles aient des noms qui évoquent intuitivement ce qu'elles expriment et que chacune soit une généralisation des notions utilisées pour les stream graphs, pour les graphes multicouches et pour les graphes.

2.3.1 Nombre de nœuds et nombre de liens

Le *nombre de nœuds dans un graphe* $G = (V, E)$ est $n = |V|$ et le nombre d'arêtes est $m = |E|$.

Le *nombre de nœuds dans un stream graph* $S = (T, V, W, E)$ est défini dans [14] comme $\hat{N}_n^T = \frac{|W|}{|T|} = \sum_{v \in V} n_v$ (ce qui est en fait le nombre moyen de nœuds au cours du temps). n_v est appelé *contribution de v* et est égal à $\hat{N}_v^T = \frac{|T_v|}{|T|}$. Remarquons bien qu'ici cette notion est différente du *nombre de nœuds dans* $|V|$, sauf quand le stream graph S est constant au cours du temps.

Dans les graphes multicouches, une telle notion n'a pas été définie explicitement. Nous avons donc défini le *nombre moyen de nœuds par couche* comme :

$$\hat{N}_n^L(M_l) = \frac{\text{nombre de nœud-couches}}{\text{nombre de couches}} = \frac{|V_M|}{|L|}$$

Remarquons que dans le cas d'un monocouche, nous retrouvons le nombre de nœuds classique, ce qui signifie que nous pouvons choisir cette mesure comme une généralisation du *nombre de nœuds dans un graphe*, comme cela a été fait dans [14] pour les stream graph.

Définition 11 (Contribution des couches). On définit la *contribution* d'une couche comme suit : $n_\alpha = \frac{|T_\alpha|}{|T|}$. Le *nombre de couches dans un stream graph multicouches* est la somme des contributions.

$$\hat{N}_l^T = \sum_{\alpha \in L} \frac{|T_\alpha|}{|T|}$$

Définition 12 (Contribution et quantité de nœud-couches). La *contribution d'un nœud-couche* (v, α) dans un stream graph multicouches est $n_{v,\alpha} = \frac{|T_{v,\alpha}|}{|T|}$. Le *nombre de nœud-couches* est la somme de leurs contributions $\hat{N}_{nl}^T(M) = \sum_{(u,\alpha) \in V_M} n_{u,\alpha} = \frac{|W_M|}{|T|}$.

Dans le cas d'un stream graph monocouche, nous retrouvons bien que le nombre de nœuds est égal au nombre de nœuds-couches. En prenant le stream graph sous-jacent du stream graph multicouches, on retrouve également que le nombre de nœuds est égal au nombre de nœuds couches. Nous avons donc là une généralisation satisfaisante du nombre de nœuds-couches.

Enfin, nous définissons le **nombre de nœuds et de liens dans un stream graph multicouches** de la façon suivante :

Définition 13 (Contribution et nombre de nœuds dans un stream graph multicouches). La *contribution* d'un nœud décrit le taux d'apparition d'un nœud dans les couches : $n_v = \frac{\sum_{\alpha \in L} |T_{v,\alpha}|}{\sum_{\alpha \in L} |T_\alpha|}$.

Le *nombre de nœuds* est la somme de ces contributions :

$$\hat{N}_n^{L,T}(M) = \sum_{v \in V} n_v = \sum_{v \in V} \frac{\sum_{\alpha \in L} |T_{v,\alpha}|}{\sum_{\alpha \in L} |T_\alpha|} = \frac{1}{\sum_{\alpha \in L} |T_\alpha|} \sum_{v \in V} \sum_{\alpha \in L} |T_{v,\alpha}|$$

Cette définition est cohérente avec les notions des multicouches et des stream graph : si le nœud apparait dans toutes les couches tout le temps, nous trouvons $n_v = 1$ comme dans les multicouches. Si tous les temps d'existence sont égaux, on retrouve

$$n_v = \frac{|\{(u, \alpha) \in V_M | u = v\}| \times |T|}{|L| \times |T|}$$

Définition 14 (Nombre de liens dans un stream graph multicouches). Le nombre de lien dans un stream graph multicouches se calcule comme le nombre de liens dans un stream graph.

$$n_l = \frac{\sum_{(u,\alpha)(v,\beta) \in E_M} |T_{(u,\alpha)(v,\beta)}|}{|T|}$$

2.3.2 Uniformité et densité

Dans les stream graphs [14], l'*uniformité entre deux nœuds* u et v est le ratio $\mathbb{U}(u, v) = \frac{|T_u \cap T_v|}{|T_u \cup T_v|}$, c'est à dire la probabilité, prenant un temps t dans $T_u \cup T_v$, que les deux nœuds u et v puissent être reliés entre eux. L'uniformité est définie comme le rapport de tous les temps de co-existence sur les temps d'existence : $\mathbb{U}(S) = \sum_{u,v \in V \otimes V} \frac{|T_u \cap T_v|}{|T_u \cup T_v|}$

L'*uniformité* de deux nœuds-couches (u, α) et (v, β) est définie sur le même modèle :

$$\begin{aligned} \mathbb{U}((u, \alpha), (v, \beta)) &= \mathbb{P}(t \in T_{u,\alpha} \cap T_{v,\beta} | t \in T_{u,\alpha} \cup T_{v,\beta}) \\ &= \frac{|T_{u,\alpha} \cap T_{v,\beta}|}{|T_{u,\alpha} \cup T_{v,\beta}|} \end{aligned}$$

Le *chevauchement* de deux couches α et β est également défini comme suit :

$$\Psi(\alpha, \beta) = \frac{|T_\alpha \cap T_\beta|}{|T_\alpha \cup T_\beta|}$$

L'*uniformité des nœuds-couches* mesure le taux d'apparition simultanée des nœuds-couches dans le stream graph multicouches :

$$\Psi(M) = \frac{\sum_{(u,\alpha),(v,\beta) \in V_M \otimes V_M} |T_{(u,\alpha)} \cap T_{(v,\beta)}|}{\sum_{(u,\alpha),(v,\beta) \in V_M \otimes V_M} |T_{(u,\alpha)} \cup T_{(v,\beta)}|}$$

Cette définition peut être nuancée en considérant que selon la nature du stream graph multicouches, certains nœuds peuvent ou ne peuvent pas apparaître simultanément. En particulier, l'*uniformité dans un stream graph multicouches* est définie ainsi :

$$\Psi(M) = \frac{1}{|L|} \sum_{\alpha \in L} \frac{\sum_{(u,\alpha),(v,\alpha) \in V_M \otimes V_M} |T_{(u,\alpha)} \cap T_{(v,\alpha)}|}{\sum_{(u,\alpha),(v,\alpha) \in V_M \otimes V_M} |T_{(u,\alpha)} \cup T_{(v,\alpha)}|}$$

On ne prend en compte que les temps d'apparition intra-couche, et on moyenne selon le nombre de couche.

Quand $\Psi = 1$, on dit que le stream graph est **uniforme** (autrement dit, $T_v = T_u, \forall (u, v) \in V^2$).

Après avoir mesuré « à quel point le stream graph multicouches peut être connecté » grâce à l'uniformité, nous allons mesurer à quel point il est effectivement connecté grâce à la notion de **densité**.

Dans les graphes classiques (non pondérés et non dirigés), la *densité* est la probabilité, prenant deux nœuds au hasard dans le graphe, qu'une arête existe entre ces nœuds.

$$d(G) = \frac{|E|}{|V \otimes V|} = \frac{2 \times |E|}{|V|(|V| - 1)}$$

Dans les stream graph [14], l'intuition est la même, en prenant un temps aléatoire, et deux nœuds au hasard existant à ce temps.

$$\begin{aligned} \delta_s(G) &= \mathbb{P}((t, u, v) \in E | (t, u), (t, v) \in W) \\ &= \frac{\sum_{(u,v) \in V \otimes V} |T_{uv}|}{\sum_{(u,v) \in V \otimes V} |T_u \cap T_v|} \end{aligned}$$

Nous pouvons ensuite définir la densité dans les stream graphs multicouches. Nous remarquons premièrement que la densité mesure le « taux de connectivité », et qu'une fois encore, en fonction du système étudié, la définition peut changer puisque nous voulons que le cas où la densité est égale à 1 corresponde au cas où on ne peut pas rajouter d'arêtes.

Définition 15 (Densité d'un stream graph multicouches). Nous appelons $C \in T \times V_M \times V_M$ l'ensemble des liens autorisés dans le stream graph multicouches. Si certaines liens sont « automatiques » ou « sous-entendus » comme par exemple les arêtes de couplage, nous ne les comptons pas dans C non plus pour que le cas où la densité soit nulle corresponde au cas où on ne peut pas retirer de lien.

La *densité du stream graph multicouches* s'écrit alors :

$$\delta_M(M) = \frac{\sum_{(u,\alpha),(v,\beta) \in E_M} |T_{(u,\alpha)(v,\beta)}|}{|C|}$$

Par exemple, dans un stream graph multiplexe, les liens possibles sont les liens intra couches et les liens de couplage : $C = \{(t, u, \alpha), (t, v, \beta) \mid t \in T_{u,\alpha} \cap T_{v,\beta}, u = v \text{ ou } \alpha = \beta\}$:

$$\delta_M(M) = \frac{|E_M|}{|C|} = \frac{\sum_{(u,\alpha)(v,\beta) \in (V_M \otimes V_M)} |T_{(u,\alpha)(v,\beta)}|}{\underbrace{\left(\sum_{\alpha \in L} \sum_{(u,v) \in V \otimes V} |T_{u,\alpha} \cap T_{v,\alpha}| \right)}_{\text{arêtes intra-couches}} + \underbrace{\left(\sum_{u \in V} \sum_{(\alpha,\beta) \in L \otimes L} |T_{u,\alpha} \cap T_{u,\beta}| \right)}_{\text{arêtes de couplage}}}$$

Nous avons donc présenté les bases mathématiques des stream graphs multicouches, ayant pour but de représenter des interactions de différentes natures, entre différents acteurs, en fonction du temps. Les mesures de nombre de nœuds et de liens, d'uniformité et de densité permettent d'adapter des notions présentes dans les graphes à notre objet. Les outils d'extraction comme le graphe sous-jacent, le graphe agrégé ou les sous stream graph inter et intra-couches permettent de « retrouver » les graphes classiques auxquels nous sommes habitués.

3 Application à des données concrètes

Nous avons ensuite construit une bibliothèque Python [17] pour mettre en application les notions théoriques que nous avons présentées.

Nous présentons cette bibliothèque ainsi que sa structure. Puis, en utilisant un jeu de données approprié, nous donnons un exemple d'utilisation concrète de la bibliothèque et des notions créées.

3.1 Structure de données et organisation du code

La gestion des stream graphs multicouches implique assez rapidement de gérer un grand nombre d'informations « imbriquées » : chaque couche contient des nœuds, qui apparaissent à des temps précis, ces nœud-couches sont reliés par des liens, eux-mêmes apparaissant dans des intervalles de temps qu'il faut stocker.

Nous choisissons de construire des classes Python pour simplifier la gestion de telles données, et pour permettre de vérifier simplement que le stream graph multicouches reste cohérent quand on le modifie.

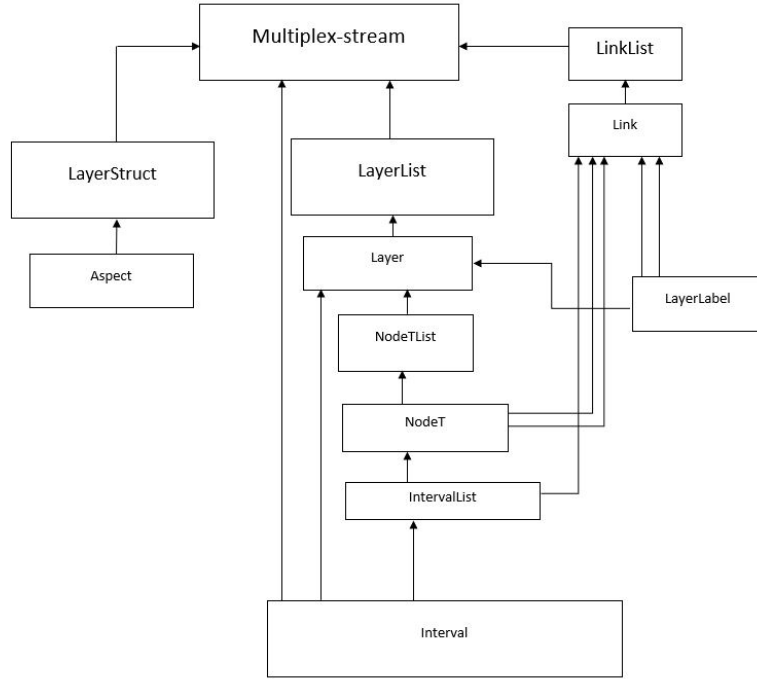


FIGURE 11 – Structure du code en classes

Toutes les listes sont toujours triées et les listes d'intervalles contiennent seulement des intervalles disjoints (quitte à fusionner certains intervalles).

Dans un graphe classique, la *taille*, c'est à dire le nombre d'éléments à parcourir dans le graphe, est donnée par $c_G(G) = |V| + |E|$. Ici, nous avons choisi de stocker des couches, elles-mêmes contenant les nœuds-couches (et un intervalle), qui contiennent des listes d'intervalles. Les liens sont caractérisés par deux nœuds-couches et une liste d'intervalle.

On appelle n_I le nombre d'intervalle d'un ensemble, si celui-ci contient une liste d'intervalles.

Ici la taille du stream graph multicouches est donc : $c_{MS}(M_S) = \sum_{e \in E_M} n_I(e) + \sum_{\alpha \in L} \sum_{v \in V} n_I((v, \alpha))$. Notons que dans le cas où le stream graph multicouches est statique et monocouche, on retrouve la taille d'un graphe classique.

Pour traiter de graphes classiques, nous avons généralement le choix entre utiliser une matrice d'adjacence (ce qui nous permet de savoir s'il existe un lien entre deux nœuds en temps constant) ou d'utiliser des listes d'adjacence (ce qui permet d'économiser de la mémoire).

Ici, le stockage d'un lien est beaucoup plus coûteux puisqu'il faut stocker tous les intervalles d'existence, nous avons plutôt choisi d'utiliser une liste simple de tous les liens, mais stockés de façon ordonnés grâce à la bibliothèque `sorted collection`, qui permet de retrouver ou d'insérer des éléments par dichotomie (complexité logarithmique).

3.2 CPGE : interactions entre élèves d'un même établissement

Nous avons en premier lieu testé notre logiciel avec une jeu de données issu de SocioPatterns [18], recensant les interactions « face à face » entre élèves (enregistrées pendant 5 jours toutes les 20 secondes grâce à des capteurs de proximité), leurs amitiés sur Facebook, et leurs réponses à un questionnaire leur demandant s'ils étaient amis.

Nous avons donc choisi une structure de couches suivante :

```
L=[type de relation,classe,sexe],  
    avec  
    type de relation = [face to face, facebook, contact, amitié],  
    classe = "MP","MP*1","MP*2","2BI01","2BI02","2BI03","PSI*","PC","PC*".
```

Comme les seules interactions dépendantes du temps sont celles de face à face, on peut dire que le jeu de données donne lieu à un graphe « hybride ». Nous avons considéré que les liens Facebook et d'amitié étaient donc constants au cours du temps.

3.2.1 Représentation graphique

Nous avons tout d'abord tenté de représenter le stream graph multicouches en entier (figure 12). La taille du jeu de données le rend difficile à représenter du fait que beaucoup de liens s'entrecroisent. (D'ailleurs, trouver un ordre optimal pour les nœuds qui limite les croisements des liens avec d'autres nœuds est un problème d'optimisation en cours de résolution).

Cependant, grâce à une coloration des liens intercouches, nous pouvons tout de même saisir la structure générale du jeu de données, ainsi que les pauses et les nuits.

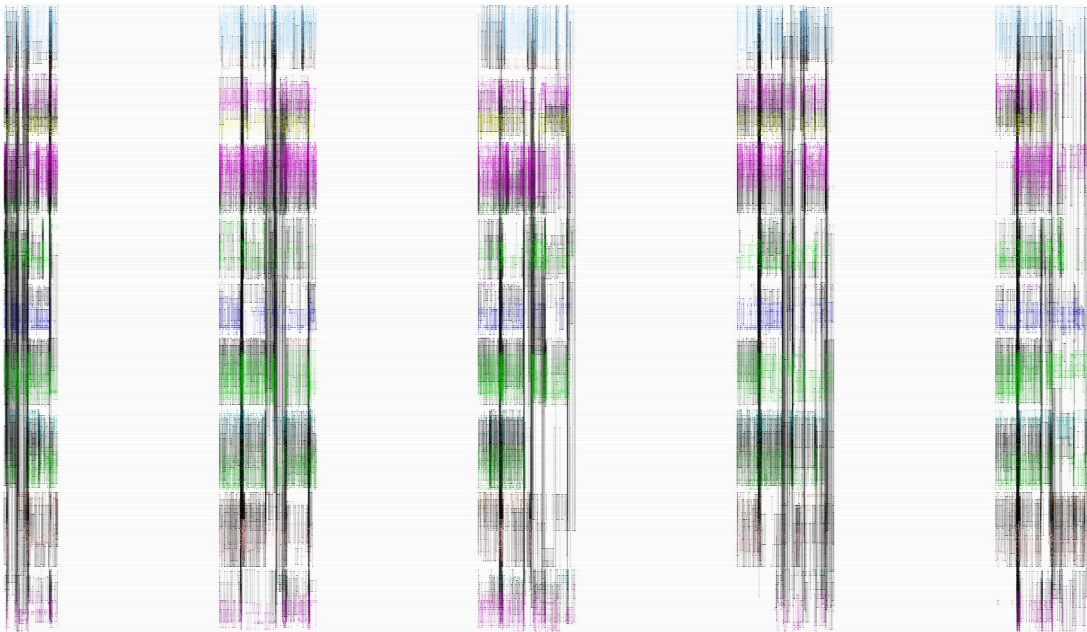


FIGURE 12 – **Visualisation du stream graph multicouches du dataset [18]**. Les liens intra-couches ont été coloriés dans des couleurs aléatoires pour chaque couche. On remarque des apparitions de liens inter-couches assez importants au moment des « pauses » (il y a visiblement une pause le matin, une autre le midi, et une à la fin de la journée). On peut également visualiser simplement l'emploi du temps des élèves (représenté par des vides et des pleins dans le graphe).

3.2.2 Sous-graphes, sous stream graph et sous graphes multicouches

Voici quelques exemples de sous-graphes tels que nous les avons définis section 2.2. Nous avons décidé de nous cantonner aux relations "face to face" et "facebook", mais ces résultats sont généralisables à plus de couches.

Graphe multicouche induit Le graphe multicouche induit des relations de deux classes, filles et garçons, est représenté par la figure 13. Ici, chaque cercle de nœuds (représentés par les points noirs) correspond à une couche différente.

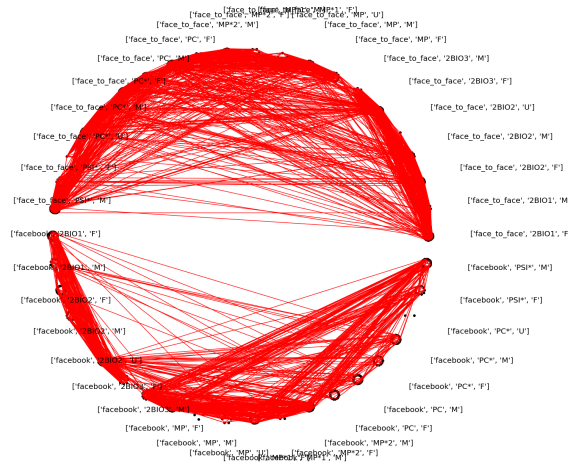


FIGURE 13 – Graphe induit

Nous représentons ensuite les sous-graphes multicouches intra-couches et inter-couches induits des couches de deux classes différentes pour plus de visibilité dans la figure 16.

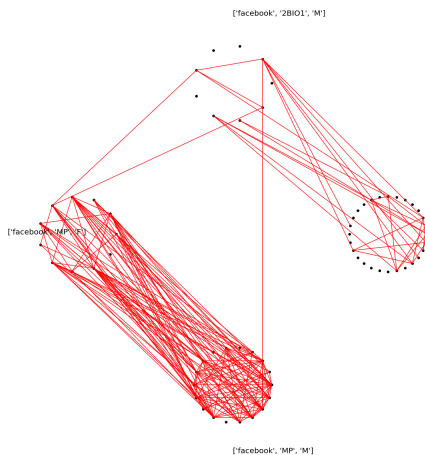


FIGURE 14 – Visualisation du sous graphe multicouches des relations ‘facebook’, entre les élèves de deux classes : les ‘2BIO1’ et les ‘MP’.

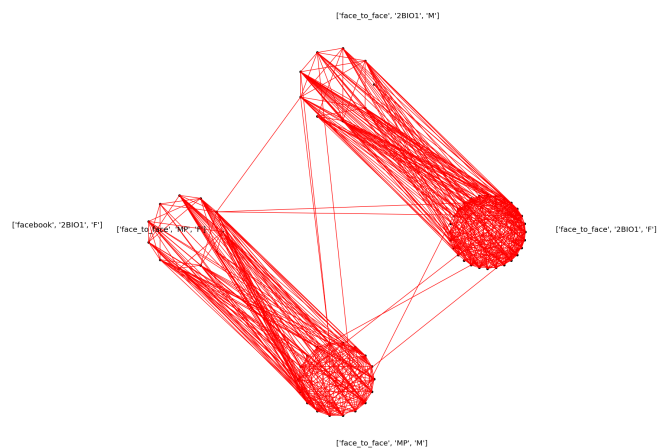


FIGURE 15 – Visualisation du sous graphe multicouches induit des relations ‘face to face’, entre les élèves de deux classes : les ‘2BIO1’ et les ‘MP’.

FIGURE 16 – Comparaison entre différents aspects du graphe multicouches. On peut s’intéresser à la pertinence de séparer filles et garçons en différentes couches. En revanche, la séparation entre différentes classes semble être justifiée.

3.2.3 Étude des densités intra et inter-couches pour plusieurs aspects

On peut alors se demander en regardant ces interactions (figures 13 et 16) , si le partitionnement différenciant les femmes et les hommes est « visible » d’un point de vue des interactions.

Nous avons donc calculé la densité des relations femmes/hommes comparée aux densités des relations au sein des hommes et des femmes, au fil des jours. Cette densité correspond à la probabilité, prenant un temps aléatoire et deux nœuds aléatoires, ces nœuds soient reliés par une arête.

Au sein des hommes, des femmes, ou pour la densité totale, on calcule donc :

$$\delta = \frac{\sum_{u,v \in V \otimes V} |T_{uv}|}{\sum_{u,v \in V \otimes V} |T_u \cap T_v|} = \frac{2|E|}{|T|(|V|(|V| - 1))}$$

et dans le graphe biparti des interactions hommes-femmes, la densité est :

$$\delta_{biparti} = \frac{\sum_{u,v \in V_f \times V_h} |T_{uv}|}{\sum_{u,v \in V_f \times V_h} |T_u \cap T_v|} = \frac{|E|}{|T||V_f||V_h|}$$

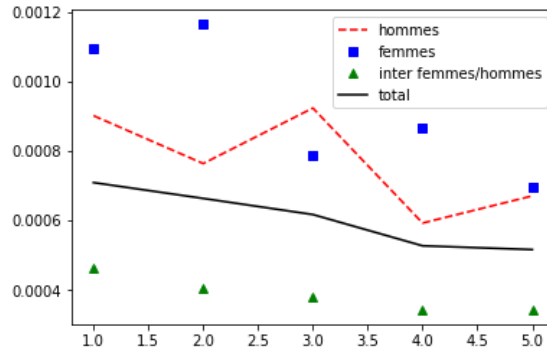


FIGURE 17 – Densité des relations au sein des femmes, des hommes, des relations inter-hommes/femmes, et des relations en général prise chaque jour de la semaine.

Nous voyons qu’une tendance se confirme : les relations sont plus denses au sein d’un même sexe qu’en moyenne, et les relations inter-sexes sont moins denses qu’en moyenne. Nous avons donc tracé les mêmes densités pour des graphes multicouches sous-jacents correspondant aux jours de la semaine (figure 18). Nous obtenons les mêmes tendances mais moins « contrastées » puisqu’une relation de longue durée « pèse » autant qu’une relation de courte durée qui a pu être enregistrée « par hasard », mais qui ne reflète pas une réelle interaction. La composante stream graph dans notre stream graph multicouches nous permet donc d’obtenir un résultat plus « fin » que la projection en graphe agrégé.

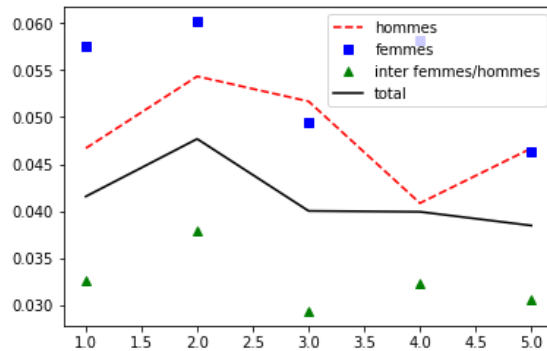


FIGURE 18 – Densité des graphes sous-jacents pour les 5 jours de la semaine.

Nous avons comparé les différentes densités pour les aspects ‘facebook’ et ‘friendship’. Nous avons alors relevé les mêmes tendances, et même nettement plus accentuées pour les relations amicales, où les relations intra-hommes passent en dessous de la moyenne (figure 19)

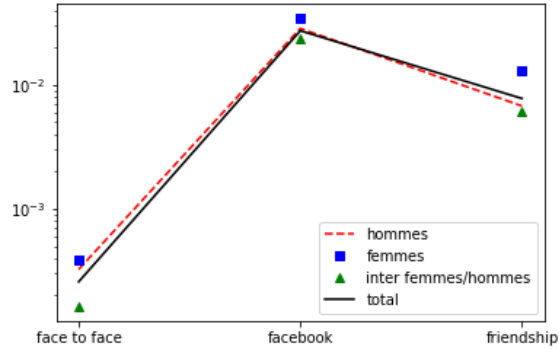


FIGURE 19 – Comparatif des densités pour les relations ‘face to face’, ‘facebook’ et ‘friendship’ en échelle logarithmique, pour les aspects ‘hommes’, ‘femmes’, le graphe inter-couche ‘hommes’/‘femmes’ et le graphe en entier.

Nous avons donc montré que les outils créés pour les stream graphs multicouches peuvent servir pour analyser des interactions entre individus de façon plus nuancée. La densité associée aux sous-stream graph inter-couches et intra-couches sert à caractériser des interactions de différents types entre plusieurs groupes d’individus.

4 Conclusions et perspectives

Nous avons donc présenté notre nouvel objet ainsi que les raisons qui ont mené à sa création. Nous avons formalisé plusieurs notions pour nous permettre d’extraire des graphes d’intérêt et de prendre des mesures, et nous avons montré un exemple d’utilisation de ces notions grâce à un jeu de données.

Pour la suite du stage, nous avons trouvé deux jeux de données très différents pouvant se prêter à notre étude, et permettant des applications diverses. Le premier recense les personnages, dialogues et lieux des films StarWars, ainsi que leurs interactions dans le temps. Le but sera de « trouver » quels sont les personnages, sujets et lieux « principaux » ou « centraux », en adaptant des méthodes d’intrication [5], existantes pour les graphes multicouches, aux stream graphs multicouches. Le second recense tous les vols sur le sol américain entre 1987 et aujourd’hui, pour différentes compagnies aériennes (représentant les différentes couches). Cette fois-ci, il nous faudra adapter le modèle de stream graph multicouches au même objet mais dans lequel les liens sont « instantanés » et coûtent un temps γ à être parcourus. Le but sera ensuite d’étudier la « centralité » des aéroports ou des compagnies aériennes, grâce à des mesures multicouches, des simulations de trajets aléatoires et peut-être une adaptation du PageRanking à notre nouvel objet, en nous inspirant de [11].

Bien que la notion de stream graphs multicouches semble naturellement adaptée en théorie à de nombreux jeux de données, peu d’entre eux sont en fait accessibles à l’heure actuelle. En effet, le formalisme n’existant pas, il n’a probablement pas semblé utile de collecter autant d’informations dans les jeux de données que le type des relations et surtout les temps d’interaction. Mais avec le nombre croissant de « big data » à notre disposition,

il est très probable que nous ayons accès de plus en plus souvent à ce genre de données.

Références

- [1] Théorie des graphes (wikipedia). https://fr.wikipedia.org/wiki/Thorie_des_graphes.
- [2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows : theory, algorithms, and applications*. Prentice Hall, 1993.
- [3] Ziegler G.M. Aigner M. *Raisonnements divins (deuxième édition, version française)*. Spingler, 2004.
- [4] Renoust B., Melançon G., and Viaud M. Mesurer l'intrication sémantique dans une collection de documents. In *EGCFGG*, pages 18–29, France, January 2013.
- [5] Renoust B., Melançon G., and Viaud M. Entanglement in multiplex networks : understanding group cohesion in homophily networks. In *Social Network Analysis-Community Detection and Evolution*, pages 89–117. Springer, 2014.
- [6] M. Berlingerio, M. Coscia, F. Giannotti, A. Monreale, and D. Pedreschi. Foundations of multidimensional network analysis. In *2011 International Conference on Advances in Social Networks Analysis and Mining*, pages 485–489, July 2011.
- [7] Gaubert S. Bonnans J.F. *Recherche Opérationnelle : aspects mathématiques et applications*. Editions Ecole Polytechnique, 2018.
- [8] E.W. Dijkstra. A note on two problems in connexion with graphs. pages 269 – 271, 1959.
- [9] Aldasoro I. and Alves I. Multiplex interbank networks and systemic importance : An application to european data. *Journal of Financial Stability*, 35 :17 – 37, 2018. Network models, stress testing and other tools for financial stability monitoring and macroprudential policy design and implementation.
- [10] M. Kivelä and M. A. Porter. Isomorphisms in multilayer networks. *IEEE Transactions on Network Science and Engineering*, 5(3) :198–211, July 2018.
- [11] De Domenico M., Solé-Ribalta A., Omodei E., Gómez S., and Arenas A. Ranking in interconnected multilayer networks reveals versatile nodes. *Nature Communications*, 6 :6868 EP –, Apr 2015. Article.
- [12] De Domenico M., Solé-Ribalta A. and Cozzo E., Kivelä M., Moreno Y., Porter M.A., Gómez S., and Arenas A. Mathematical formulation of multilayer networks. *Phys. Rev. X*, 3 :041022, Dec 2013.
- [13] Kivela M., Arenas A., Barthélemy M., Gleeson J.P., Moreno Y., and Porter M.A. Multilayer networks. *Journal of Complex Networks*, 2014.
- [14] Latapy M., Viard T., and Magnien C. Stream graphs and link streams for the modeling of interactions over time. 2017.
- [15] Othon M. An introduction to temporal graphs : An algorithmic perspective. *CoRR*, abs/1503.00278, 2015.

- [16] United States Department of Transportation. Bureau of transportation statistics. https://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236.
- [17] Parmentier P., Viard T., J.F. Baffier, and Renoust B. Bibliothèque python sur les multilayer stream graphs. <https://github.com/TiphaineV/multiplex-streams>, 2019.
- [18] Mastrandrea R., Fournet J., and Barrat A. Contact patterns in a high school : A comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PLOS ONE*, 10(9) :1–26, 09 2015.
- [19] Hanneman R.A. and Riddle M. *Introduction to social network methods*. Riverside, CA : University of California, Riverside, 2005.
- [20] Caceres R.S. and Berger-Wolf T. Temporal scale of dynamic networks. In *Temporal Networks*, pages 65–94. Springer, 2013.
- [21] Pilosof S., Porter M.A., Pascual M., and Kéfi S. The multilayer nature of ecological networks. *Nature Ecology & Evolution*, 1 :0101 EP –, Mar 2017. Perspective.
- [22] L.Page S.Brin. The anatomy of a large-scale hypertextual web search engine. 1998.
- [23] von Luxburg U. A tutorial on spectral clustering. *CoRR*, abs/0711.0189, 2007.
- [24] Léo Y., Crespelle C., and Fleury E. Non-altering time scales for aggregation of dynamic networks into series of graphs. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, page 29. ACM, 2015.
- [25] Mourchid Y., Renoust B., Cherifi H., and El Hassouni M. Multilayer network model of movie script. In *Complex Networks and Their Applications VII*, pages 782–796, Cham, 2019. Springer International Publishing.