



Génération de graphes pour la compensation de dettes mutuelles entre entreprises

Rapport de stage L3 maths-info

Arthur Rousseau
Sous la direction de Nazim Fatès

22 août 2021

1 Introduction

L'objet d'étude de ce stage est la compensation de dettes mutuelles entre entreprises. Dans les échanges commerciaux entre entreprises, il est fréquent que les factures ne soient payées que dans un délai de trois mois. Cependant, pendant ces trois mois, de nombreuses factures peuvent ainsi s'accumuler entre les différents acteurs d'une économie, ce qui présente des risques, notamment si une entreprises fait faillite. Aussi, la compensation de dettes mutuelles a pour objectif de réduire la somme totale en suspens dans l'économie. Pour ce faire, un centre financier, par exemple un état, se donne une certaine somme qu'il peut injecter dans l'économie pour compenser de la dette. Le problème est alors de trouver où injecter cette somme pour éliminer un maximum de dette.

La compensation de dettes mutuelles a déjà été employée pour le réseau Sardex en Sardaigne [1, 2]. Celui-ci est une économie complètement indépendante de l'économie italienne, où les échanges sont réalisés seulement à l'aide de l'avoir commercial des entreprises : les entreprises achètent avec l'argent que d'autres leur doivent. On peut voir le phénomène comme une monnaie virtuelle exclusivement sarde, maintenue au cours de l'euro, telle que la somme totale disponible de cette monnaie soit nulle : certaines entreprises sont endettées, d'autres ont une position nette positive et peuvent consommer avec celle-ci. La compensation de dettes intervient alors pour éviter que trop de factures soient impayées simultanément, de manière à maintenir une stabilité économique. On peut aussi citer comme exemple la gestion de la crise de 2008 par la Slovénie, basée sur un système de compensation de dettes mutuelles, qui lui a permis de rebondir [2].

Lors de ce stage, on donne un cadre à l'utilisation des algorithmes de compensation de dettes : on cherche ainsi à modéliser l'économie réelle, de manière à pouvoir travailler et perfectionner les algorithmes sur un grand nombre de jeux de données. En effet, les données disponibles, si elles sont inédites et en quantité importantes, ne permettent pas de faire des études qui ne se réduisent pas à un cas particulier : le jeu de données dont nous disposons n'est en effet qu'un exemple d'économie, en l'occurence en Italie du Nord. De plus, le problème que nous étudions étant NP-complet [3], les algorithmes développés sont forcément des approximations, et faute de preuve mathématique sur les résultats obtenus, les performances sont mesurées empiriquement. Le traitement

d'un cas général nécessite donc d'être capable de générer des économies virtuelles afin d'y tester les algorithmes. Pour ce faire, une étude statistique approfondie a été réalisée, dans le but de caractériser l'essence des échanges économiques par des critères mathématiques simples dont on expliquera l'intérêt, avant de pouvoir générer ceux-ci. En particulier, l'étude économique sera en partie locale, et on testera l'existence de communautés dans la réalité et dans les modèles générés. Les algorithmes de compensation de dettes n'ont pas été étudiés en eux mêmes, mais une partie d'entre eux peut être retrouvée dans un mémoire bibliographique [3] réalisé en préparation du stage.

On dispose d'un jeu de données fourni par l'entreprise Infocert, qui fait de la compensation de dettes, et toutes les transactions impliquent au moins un client Infocert. Ce jeu de données est un ensemble de factures entre différentes entreprises d'Italie du Nord. Ces factures ont été produites sur une année et sont numérotées par le mois auquel elles ont été créées. Les données dont nous disposons sont complètement nouvelles, et aucune étude n'a, à notre connaissance, jamais été réalisée sur un jeu de donnnées de cette taille dans le cadre de la compensation intégrale de dettes. On réalise des mesures sur plusieurs sous-ensembles de ce jeu de données : le premier, appelé INFOCERT, est constitué des factures dont les deux acteurs sont des clients Inforcert. Le deuxième, appelé ELAGUE, est un sous-ensemble de INFOCERT obtenu en élaguant de manière à ne conserver que les entreprises ayant au moins une facture entrante et une facture sortante au cours de l'année. Cet élagage est fait pour essayer de se rapprocher de la réalité économique, dans laquelle tous les acteurs sont fournisseurs et consommateurs. Le troisième, appelé M1, est obtenu en ne conservant que les factures ayant eu lieu lors du mois de janvier dans ELAGUE, pour réduire la taille de l'ensemble étudié. Les tailles de ces ensembles sont :
— INFOCERT : 413 328 entreprises et 3 949 073 factures
— ELAGUE : 61 709 entreprises et 1 879 452 factures
— M1 : 37 188 entreprises et 124 665 factures.
Pour toute la partie informatique, on utilise Python 3 et Jupyter notebook.

On commencera par mathématiser le problème étudié, puis on présentera trois algorithmes de génération parmi ceux développés, ceux qui ont été les plus étudiés et qui représentent le mieux l'évolution de la réflexion. On étudie finalement quelques algorithmes de détection de communautés.

2 Mathématisation du problème

Définition 1. — Un *multigraphe* est une paire $\mathcal{G} = (V, \mathcal{E})$ où V est un ensemble quelconque appelé *ensemble des sommets* et $\mathcal{E}$ est un multiensemble de paires de V, appelé *ensemble des arêtes*. Économiquement, les sommets sont les acteurs (entreprises) et les arêtes sont les factures entre celles-ci. On utilise un multiensemble car il peut y avoir plusieurs factures entre deux mêmes acteurs. On notera $n = |V|$.
— Un multigraphe est appelé *graphe simple* si le multiensemble $\mathcal{E}$ est un ensemble simple, c'est-à-dire s'il ne contient aucun élément en double. Par la suite, on notera E un ensemble simple et G un graphe simple.
— Un multigraphe est dit *orienté* si les éléments de $\mathcal{E}$ sont orientés, autrement dit si $(i, j) \neq (j, i)$ pour $i \neq j$ et $(i, j), (j, i) \in \mathcal{E}$. Dans le cas où les arêtes ne sont pas orientées, le multigraphe est dit *non orienté*. On notera alors les arêtes de façon ensembliste $\{i, j\}$ pour signaler l'absence d'ordre. Dans la suite, tous les multigraphes seront supposés orientés, sauf précision du contraire.
— Un multigraphe est dit *pondéré* s'il est muni d'une fonction de poids $y : \mathcal{E} \to \mathbb{R}_+$. Un multigraphe non pondéré est dit *binaire*. Dans le cadre d'une étude économique, on munit le multigraphe d'une fonction de poids y, appelée *fonction de somme*, qui détermine la somme échangée entre deux acteurs lors d'une transaction.

On appelle *simplifié* d'un multigraphe $\mathcal{G}$ le graphe simple obtenu en fusionnant toutes les arêtes de $\mathcal{G}$ ayant même sommet de départ et d'arrivée. Si le multigraphe est pondéré, le poids de l'arête résultante est la somme

des poids des arêtes fusionnées. On dira aussi qu'on *simplifie* $\mathcal{G}$.

L'objectif est donc de générer un multigraphe ressemblant à multigraphe de dettes, et de choisir des critères de pour juger de la ressemblance des multigraphes générés à la réalité.

Définition 2. Soit un multigraphe de dettes $\mathcal{G} = (V, \mathcal{E}, y)$. On définit :
— La *position nette* d'un sommet $s \in V$ comme la somme algébrique totale due à s :

$$\mathrm{pos}(s, \mathcal{G}) = \sum_{(u,s)\in\mathcal{E}} y(u, s) - \sum_{(s,v)\in\mathcal{E}} y(s, v).$$

Celle-ci est négative si s est globalement endetté et positive sinon. La moyenne des positions nettes est nulle par définition.
— Le *coefficient d'amplification final* comme le rapport entre la somme à débourser si on souhaitait rembourser toute la dette du multigraphe et la somme totale ainsi remboursée :

$$\mathrm{caf}(\mathcal{G}) = \frac{\sum_{\mathrm{pos}(s)<0} |\mathrm{pos}(s)|}{\sum_{(u,v)\in\mathcal{E}} y(u, v)}.$$

Ce coefficient est fortement lié au problème de compensation des dettes.

3 Génération V1 - Méthode uniforme

Le premier algorithme développé pour générer des multigraphes de dettes est assez simple : pour chaque couple de sommets (u, v), on choisit uniformément si l'on met une multiarête de u à v et, le cas échéant, sa multiplicité et les sommes échangées pour chaque arête. Tous les tirages sont faits de manière indépendante.
Pour ce faire, on doit calculer la probabilité d'existence d'une multiarête, et donc le nombre de multiarêtes souhaité, puis étudier la répartition des multiplicités des multiarêtes et des sommes échangées dans chaque facture.

3.1 Rapport arêtes/sommets

Déterminer le nombre d'arêtes d'un multigraphe généré en fonction de son nombre de sommets est nécessaire pour savoir combien d'arêtes générer. De plus, les échanges économiques sont essentiellement linéaires, dans le sens qu'augmenter le nombre d'entreprises n'augmente que de manière linéaire le nombre d'échanges : notre hypothèse est qu'une entreprise ne commerce pas plus sous prétexte qu'il y a plus de partenaires possibles. Ainsi, on introduit le rapport arêtes/sommets.

Définition 3 (Rapport arêtes/sommets)**.** On appelle *rapport arêtes/sommets* d'un multigraphe $\mathcal{G}$ la quantité :

$$r(\mathcal{G}) = \frac{|\mathcal{E}|}{n} \in \mathbb{R}_+.$$

Mesures. Les rapports arêtes/sommets mesurés sur les multigraphes sont :
— INFOCERT : 9,554
— ELAGUE : 30,457
— M1 : 3,352.
On remarque qu'il y a trois fois plus de factures par sommet après élagage, car la plupart des sommets enlevés étaient impliqués dans très peu de factures. Par ailleurs, il y a un facteur proche de 12 entre ELAGUE et M1, ce qui est normal car M1 correspond à un douzième de l'année.

3.2 Répartition des multi-arêtes

La caractéristique des multigraphes est que les arêtes sont multiples : entre deux sommets, il y a plusieurs arêtes orientés de la même manière, chaque arête correspondant à une facture entre l'entreprise de départ et celle d'arrivée. Pour générer des multigraphes, il faut donc pouvoir choisir les multiplicités des arêtes.

Définition 4 (Multiplicité d'une arête)**.** On appelle *multiplicité* d'une arête $(i, j) \in \mathcal{E}$ le nombre d'occurences de (i, j) dans $\mathcal{E}$.

Il est intéressant de tracer les histogrammes pour la multiplicité des arêtes. Celle-ci décroît très rapidement, on trace donc les histogrammes en échelle doublement logarithmique (*loglog*).

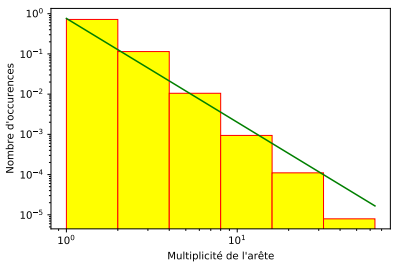


FIGURE 1 – M1

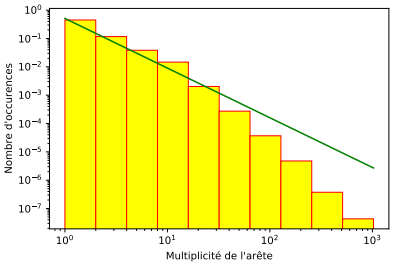


FIGURE 2 – PME

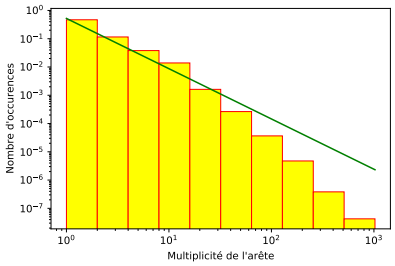


FIGURE 3 – INFOCERT

En faisant cela, on remarque que les courbes, peuvent être approximées par des droites. On peut donc faire l'hypothèse que la multiplicité des arêtes suit une loi de puissance dans un multigraphe de dettes.

3.3 Lois de puissance et répartitions

Dans notre modèle, on fera l'hypothèse que plusieurs paramètres suivent une loi de puissance, comme la multiplicité des arêtes ou les sommes échangées dans les factures. Les lois de puissance sont en effet simples à mettre en oeuvre, ont été très étudiées et se retrouvent dans de nombreux domaines économiques [4, 5, 6].

Définition 5. Soit $(\Omega, \mathcal{A}, \mathbb{P})$ un espace probabilisé et $\gamma > 1$. On dit qu'une variable aléatoire X suit une loi de puissance discrète d'exposant γ si $X(\Omega) \subset \mathbb{N}^\star$ et

$$\forall k \in \mathbb{N}^\star,\ \mathbb{P}(X = k) = \frac{k^{-\gamma}}{\zeta(\gamma)}$$

où ζ est la fonction zêta de Riemann

$$\zeta(\gamma) = \sum_{k \in \mathbb{N}^\star} k^{-\gamma}$$

et sert de constante de normalisation.

On peut remarquer qu'une variable aléatoire suit une loi de puissance en traçant une courbe loglog pour les valeurs qu'elle prend sur un grand nombre d'échantillons. L'aspect de cette courbe se rapproche alors d'une droite.
La recherche de l'exposant γ fait l'objet de plusieurs articles [4, 5, 6]. Dans ceux-ci, il est notamment montré que si γ est la pente de la courbe loglog précédemment tracée, son estimation de cette manière est très imprécise à partir d'échantillons. Pour estimer γ, il vaut mieux utiliser la méthode de vraisemblance maximale (*Maximum Likelihood Estimator, MLE*), qui consiste à maximiser, pour des échantillons indépendants $x = (x_i)_{1 \leqslant i \leqslant N}$, la probabilité que ces échantillons apparaissent. On cherche ainsi à maximiser

$$l(\gamma|x) = \mathbb{P}(x) = \prod_{i=1}^{N} \mathbb{P}(x_i) = \prod_{i=1}^{N} \frac{x_i^{-\gamma}}{\zeta(\gamma)},$$

soit à maximiser

$$\log \mathbb{P}(x) = -N \log \zeta(\gamma) - \gamma \sum_{i=1}^{N} \log x_i.$$

En annulant la dérivée, on obtient

$$\frac{\zeta'(\gamma)}{\zeta(\gamma)} = -\frac{1}{N} \sum_{i=1}^{N} \log x_i,$$

ce qui peut se résoudre par dichotomie en γ à l'aide d'une table des valeurs de $\dfrac{\zeta'}{\zeta}$, qui est croissante et négative.

Mesures. Après calcul, on obtient les exposants suivants pour les sommes échangées lors des factures, la multiplicité des arêtes et les degrés (*cf.* 3.2, 3.4, 5.1) :

	INFOCERT	ELAGUE	M1
$\gamma_{\text{somme échangée}}$ (en €)	1,169	1,169	1,169
$\gamma_{\text{multiplicité arêtes}}$	1,778	1,685	2,789
$\gamma_{\text{degré}}$	1,572	1,294	1,646

On observe une grande stabilité pour les sommes échangées, prévisible car la valeur d'une facture est indépendante du choix des ensembles. On remarque en revanche un grande variation au niveau de l'exposant de la multiplicité des arêtes selon que l'on travaille sur un mois ou un an. En effet, les transactions ayant eu lieu entre deux acteurs sur un mois ont tendance à se répéter sur un an, d'où l'augmentation de la multiplicité des arêtes sur un an et la diminution de l'exposant associé. Les exposants liés aux degrés varient de manière cohérente avec les rapports arêtes/sommets.

Définition 6. Soit $(\Omega, \mathcal{A}, \mathbb{P})$ un espace probabilisé et $\gamma > 1, b_{\min} > 0$. On dit qu'une variable aléatoire X suit une loi de puissance continue d'exposant γ et de borne $b_{\min}$ si $X(\Omega) = [b_{\min}, +\infty[$ et si sa fonction de répartition est de la forme

$$\rho(x) = Cx^{-\gamma},$$

où C est une constante dépendant de γ et $b_{\min}$. La normalisation nous donne $C = (\gamma - 1) b_{\min}^{\gamma-1}$.

On s'autorise à approximer une loi de puissance discrète d'exposant γ par la partie entière d'une loi de puissance continue d'exposant γ et de borne 1, ce qui dans la pratique donne de bons résultats. Ainsi, pour tirer aléatoirement des valeurs selon une loi de puissance discrète, il suffit de pouvoir le faire sur une loi de puissance continue.
Pour tirer une valeur selon une loi de puissance continue d'exposant γ et de borne $b_{\min}$, on cherche une fonction

$$S : [0,1[\to [b_{\min}, +\infty[\text{ telle que } \forall a, b \in [0,1[,\ \mathbb{P}(U \in [a,b]) = \mathbb{P}(X \in [S(a), S(b)]) \text{ et } S(0) = b_{\min},$$

où U suit une loi uniforme sur $[0,1[$ et X suit une loi de puissance continue d'exposant γ et de borne $b_{\min}$. Cela revient à avoir

$$\forall y \in [0,1[,\ y = \int_{b_{\min}}^{S(y)} \rho,$$

où ρ est la fonction de répartition de X. Ainsi,

$$y = \frac{C}{1-\gamma} \left(S(b)^{1-\gamma} - b_{\min}^{1-\gamma} \right),$$

soit

$$S(y) = (1-y)^{\frac{1}{1-\gamma}} b_{\min}.$$

Alors, si on tire aléatoirement un valeur y dans $[0, 1[$ selon une loi uniforme, la valeur $S(y)$ est aléatoire selon une loi de puissance d'exposant γ et de borne $b_{\min}$.

En réalité, les lois de puissance apparaissant dans le contexte économique sont bornées, ce qui n'est pas le cas d'une loi de puissance usuelle. Pour pallier ce problème, on se donne une borne lors du tirage d'une valeur suivant une loi de puissance, et si la valeur tirée dépasse cette borne, on tire une nouvelle valeur. Cela évite de générer des factures de l'ordre du milliard d'euros, peu réalistes.

3.4 Sommes échangées

On s'intéresse ici aux sommes échangées dans les factures : pour cela, on trace les histogrammes en échelle loglog pour les sommes échangées. Les courbes tracées sont en première approximation assimilées à des droites pour des raisons de simplicité, et on fait l'hypothèse que les sommes échangées suivent des lois de puissance pour chaque arête (*cf.* §3.3). On affinera plus tard ce modèle qui présente des différences par rapport à la réalité économique (*cf.* §4.4).

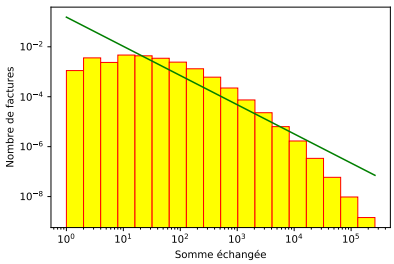


FIGURE 4 – INFOCERT

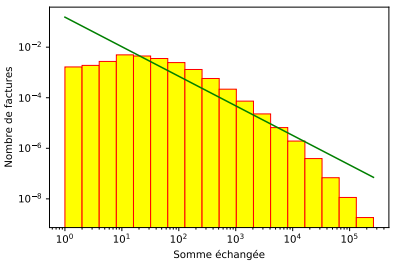


FIGURE 5 – ELAGUE

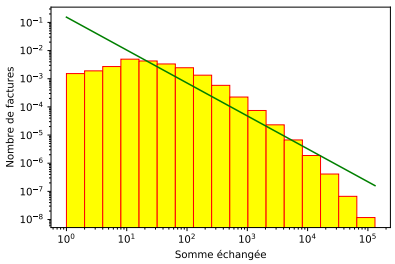


FIGURE 6 – M1

3.5 Création des multiarêtes

Dans l'algorithme de génération V1, on met des multiarêtes entre les sommets car on travaille dans un multigraphe. Pour cela, on choisit aléatoirement la multiplicité m de la multiarête (le nombre d'arêtes dans la multiarête) selon une loi de puissance $\mathcal{M}$, dont l'exposant γ_{mult} a été calculé sur des multigraphes réels par méthode de vraisemblance maximale. Puis, pour chacune des m arêtes, on choisit aléatoirement la somme (en euros) de cette arête selon une loi de puissance $\mathcal{S}$ dont l'exposant γ_{somme} a de même été calculé sur des multigraphes réels. Dans les multigraphes réels, les sommes sont en centimes mais tirer aléatoirement un nombre en centimes directement a posé des problèmes de calcul, les sommes en centimes étant trop importantes pour faire efficacement plusieurs centaines de milliers de tirages. La partie centimes de la somme est donc simplement calculée selon une loi uniforme sur $[\![0, 99]\!]$. Ajouter une multiarête au multigraphe revient alors à ajouter les m arêtes à la liste des arêtes du multigraphe.

3.6 Calibration des paramètres

Les exposants des lois de puissances sont calculés par maximum de vraisemblance (*cf.* §3.3).
L'algorithme prend aussi en entrée un nombre de sommets n, qu'on peut faire varier selon la taille du multigraphe souhaité, ainsi qu'un paramètre donnant la probabilité p de présence d'une multiarête entre deux sommets quelconques. Comme la présence de chaque multiarête est déterminée par une loi de Bernoulli de paramètre p, le nombre de multiarêtes suit une loi binomiale de paramètres $n(n-1)$ et p. Ainsi, l'espérance du nombre

de multiarêtes est $n(n-1)p$ et l'espérance du rapport arêtes/sommets du simplifié du multigraphe généré est exactement $\frac{p}{n-1}$. On choisit donc pour p le rapport arêtes/sommets souhaité pour le multigraphe multiplié par n (*cf.* §[7, 8]).

Remarque. En réalité, le multigraphe résultant n'a pas forcément n sommets : en effet, pour chaque sommet, il y a une probabilité $(1-p)^{2(n-1)}$ pour que celui-ci ne soit relié à aucune arête. Cenpendant, dès que la taille des multigraphes grandit un peu, cette probabilité est très faible et le nombre de tels sommets *isolés* est négligeable.

3.7 Proportion de dièdres et de triangles - Résultats et limites du modèle

L'un des premiers critères pour vérifier si le multigraphe généré est cohérent ou non est la structure topologique du multigraphe, c'est-à-dire la manière dont les sommets et les arêtes sont agencés. Pour mesurer cette structure, on se propose d'étudier les dièdres et les triangles, c'est-à-dire les cycles de tailles 2 et 3 respectivement. On n'étudie pas de cycles de taille supérieure car ceux-ci sont trop coûteux à calculer.

Définition 7. Dans un graphe simple orienté, on définit :
— la *proportion de dièdres* comme le nombre d'arêtes (u, v) faisant partie d'un dièdre, c'est-à-dire telles qu'il existe une arête (v, u), quotienté par le nombre total d'arêtes.
— la *proportion de triangles* comme le nombre d'arêtes (u, v) faisant partie d'un triangle, c'est-à-dire telles qu'il existe un sommet w et des arêtes (v, w) et (w, u), quotienté par le nombre total d'arêtes.

On peut étendre cette définition à un multigraphe, en posant comme proportions de dièdres et de triangles les proportions obtenues avec le simplifié du multigraphe. Ces quantités caractérisent un certain niveau de structure du multigraphe et sont très liées au *coefficient d'agglomération* [9, 10].

On mesure ainsi, sur M1 et sur un multigraphe généré sur le modèle de M1, les proportions de dièdres et de triangles suivantes :

	M1	Multigraphe généré
Proportion de dièdres	$4,0 \cdot 10^{-2}$	$6,1 \cdot 10^{-5}$
Proportion de triangles	$8,8 \cdot 10^{-4}$	$1,8 \cdot 10^{-5}$

Comme on pouvait s'y attendre, le modèle de génération V1 donne des résultats médiocres. En effet, celui-ci ne considère jamais la structure que pourrait avoir un multigraphe de dettes, et les proportions de dièdres et de triangles sont très loin de correspondre à la réalité. De plus, la répartition des degrés suit une loi binomiale au lieu d'une loi de puissance (*cf.* 5.1). Par la suite, on ne cherchera pas à faire correspondre les distributions de degrés à la réalité, cela pourrait être fait à l'aide d'algorithmes d'attachement préférentiel [11].

4 Génération V3 - Chaînes de Markov de mémoire 3

L'algorithme V1, en plus de donner de mauvais résultats, est très coûteux en termes de temps de calcul – l'algorithme fait $n(n-1)$ étapes car on parcourt tous les couples ordonnés possibles sur n sommets. Pour cette raison, on cherche à faire un algorithme beaucoup moins coûteux. Dans cette optique, on pense aux chaînes de Markov qui permettent de faire évoluer des objets avec une très faible mémoire.
Une chaîne de Markov est un objet dynamique qui évolue en fonction de ses positions passées. La mémoire d'une chaîne de Markov est le nombre de positions qu'on retient, et ces positions retenues sont toujours les dernières.
On ne développera pas la théorie des chaînes de Markov, mais le modèle que nous utilisons en est inspiré.
Nous allons construire un multigraphe de dettes en parcourant des « chemins de dettes », et en retenant les trois

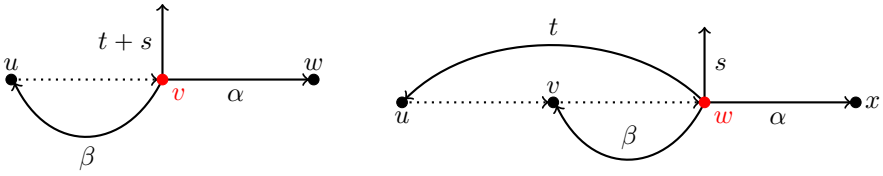


FIGURE 7 – Transitions possibles lors du parcours du multigraphe. Les variables associées aux transitions correspondent aux probabilités de faire cette transition

derniers sommets visités. De plus, notre algorithme est probabiliste : pour chaque état possible, l'état suivant sera aléatoire selon des probabilités que nous allons décrire.

4.1 Transitions

L'algorithme fonctionne selon le principe suivant : initialement, on choisit aléatoirement et uniformément deux sommets distincts, puis on met une multiarête de l'un à l'autre comme pour l'algorithme V1 (*cf.* §3.5), en choisissant d'abord la multiplicité selon $\mathcal{M}$ puis les sommes échangées selon $\mathcal{S}$. On doit ensuite décider où on mettra la multiarête suivante, et c'est ici que les transitions apparaissent (Fig. 7). La chaîne de Markov fonctionne avec un automate à deux états (Fig. 8) : dans l'état II, on regarde pour passer à l'étape suivante les deux derniers sommets en mémoire, alors que dans l'état III on utilise les trois sommets en mémoire. Le choix d'une mémoire de taille trois est lié aux proportions de dièdres et de triangles, qu'on cherche à faire correspondre à la réalité. Les transitions feront intervenir des probabilités α, β, t, s telles que $\alpha+\beta+t+s=1$. Les transitions sont décrites en Figure 7.

Dans l'état II, avec pour derniers sommets en mémoire u, v :

— Avec une probabilité α, on continue le chemin qu'on a commencé : on choisit aléatoirement et uniformément un sommet w distinct de u, v, et la prochaine multiarête sera mise de v à w. w est ajouté à la mémoire, qui devient u, v, w. On passe à l'état III.

— Avec une probabilité β, on ferme un dièdre : la prochaine multiarête sera mise de v à u. La mémoire devient u, v, u et on reste à l'état II.

— Avec une probabilité $t+s$, on termine le chemin qu'on a commencé : on choisit alors deux nouveaux sommets distincts i, j aléatoirement et uniformément et la prochaine multiarête sera mise de i à j. La mémoire devient v, i, j et on reste à l'état II.

Dans l'état III, avec pour sommets en mémoire u, v, w :

— Avec une probabilité α, on continue le chemin qu'on a commencé : on choisit aléatoirement et uniformément un sommet x distinct de u, v, w, et la prochaine multiarête sera mise de w à x. On oublie le sommet u, et les trois sommets en mémoire sont alors v, w, x. On reste à l'état III.

— Avec une probabilité β, on ferme un dièdre : la prochaine multiarête sera mise de w à v. La mémoire devient v, w, v et on passe à l'état II.

— Avec une probabilité t, on ferme un triangle : la prochaine multiarête sera mise de w à u. Les trois sommets en mémoire restent les mêmes, mais leur ordre change : on passe à v, w, u. On reste à l'état III.

— Avec une probabilité s, on termine le chemin qu'on a commencé, et on fait comme pour l'état II.

On obtient ainsi un automate des transitions (Fig. 8), où les états II et III représentent respectivement les situations avec deux et trois sommets en mémoire. L'entrée se fait à l'état II, et la rupture de chemin est signalée par une sortie dans l'automate.

On s'arrête lorsqu'on a atteint le nombre souhaité d'arêtes, calculé préalablement à partir du rapport arêtes/sommets souhaité et du nombre de sommets, donnés en paramètres de l'algorithme.

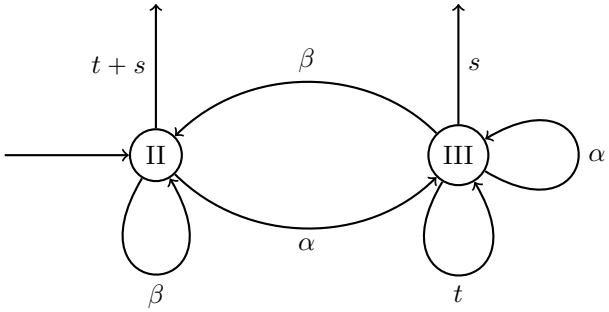


FIGURE 8 – Automate des transitions

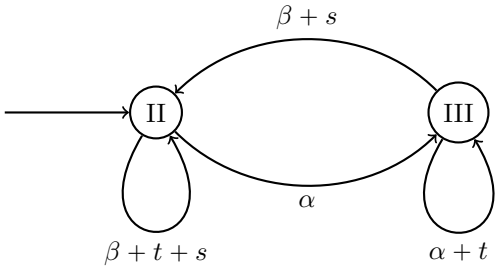


FIGURE 9 – Automate des transitions simplifié pour l'étude probabiliste

4.2 Calibration des paramètres

Les exposants des lois de puissance $\mathcal{M}$ et $\mathcal{S}$ sont les mêmes que dans le modèle de génération V1 (*cf.* §3.6), par maximum de vraisemblance. Il en va de même du nombre souhaité d'arêtes, calibrés à partir du nombre de sommets et du rapport arêtes/sommets souhaité.
La calibration des probabilités α, β, t, s est essentielle pour que les statistiques du multigraphe généré ressemblent à celles de multigraphes réels. On remarque qu'il y a un lien direct entre les paramètres β et t et les proportions de dièdres et de triangles. Nous allons expliciter ce lien.
Pour cela, on va travailler de manière asymptotique en le nombre de sommets, ce qui nous permettra de calibrer nos paramètres en fonction des espérances des proportions de dièdres et de triangles.
On se place dans un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$ tel qu'on puisse parcourir le graphe de manière infinie, les choix de transitions étant indépendants à chaque étape. On note e_i la variable aléatoire décrivant l'état de l'automate des transitions dans lequel on se trouve après i transitions. Comme une rupture de chemin ramène en l'état II (on sort puis on rentre à nouveau), les transitions probabilistes pour e_i peuvent être décrites par un automate simplifié (Fig. 9). Celui-ci est simplifié dans le sens qu'il n'y a plus de sortie de l'automate, et qu'on ne fait plus la différence entre les transitions ayant même états de départ et d'arrivée.

On note $X_i = \begin{pmatrix} \mathbb{P}(e_i = \text{II}) \\ \mathbb{P}(e_i = \text{III}) \end{pmatrix}$. On a alors $X_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ et $X_{i+1} = MX_i$ pour tout $i \in \mathbb{N}$, où $M = \begin{pmatrix} \beta+s+t & \beta+s \\ \alpha & \alpha+t \end{pmatrix}$.
Ainsi, $X_N = M^N X_0$ pour tout $N \in \mathbb{N}$. On remarque que

$$M - I_2 = \begin{pmatrix} \beta+s+t-1 & \beta+s \\ \alpha & \alpha+t-1 \end{pmatrix} = \begin{pmatrix} -\alpha & \beta+s \\ \alpha & -\beta-s \end{pmatrix},$$

qui est non inversible donc 1 est valeur propre de M. De même,

$$M - tI_2 = \begin{pmatrix} \beta+s & \beta+s \\ \alpha & \alpha \end{pmatrix},$$

qui est non inversible donc t est valeur propre de M. Ainsi M est diagonalisable, de valeurs propres 1 et t donc la suite $\left(M^N\right)$ converge, donc la suite (X_N) converge vers une limite X. En passant à la limite, il vient $MX = X$ (X est vecteur propre de valeur de propre 1). Cette limite va nous permettre de déterminer la proportion asymptotique des états II et III parmi les états visités. On note $X = \begin{pmatrix} x \\ y \end{pmatrix}$. On a

$$MX = X \Leftrightarrow \alpha x = (\beta + s)y \Leftrightarrow x = \left(\frac{1-t}{\alpha} - 1\right) y.$$

Ainsi l'espace propre associé à la valeur propre 1 est $\mathbb{R}\begin{pmatrix} \frac{1-t}{\alpha} - 1 \\ 1 \end{pmatrix}$. Alors, comme on a de plus $x + y = 1$, on a $X = \begin{pmatrix} 1 - \frac{\alpha}{1-t} \\ \frac{\alpha}{1-t} \end{pmatrix}$. La convergence étant exponentielle, la limite est rapidement approchée, si bien qu'on approximera X_i par X par la suite pour tout $i \in \mathbb{N}$.

Sous ces conditions, si on note D_i l'évènement « on ferme un dièdre lors de la transition i » et T_i l'évènement « on ferme un triangle lors de la transition i », on a :

$$\mathbb{P}(D_i) = \mathbb{P}(e_i = \text{II})\mathbb{P}(D_i|e_i = \text{II}) + \mathbb{P}(e_i = \text{III})\mathbb{P}(D_i|e_i = \text{III}) = \beta$$

et

$$\mathbb{P}(T_i) = \mathbb{P}(e_i = \text{II})\mathbb{P}(T_i|e_i = \text{II}) + \mathbb{P}(e_i = \text{III})\mathbb{P}(T_i|e_i = \text{III}) = 0 + \frac{\alpha}{1-t}t.$$

Ainsi, si on néglige les fermetures de dièdres et de triangles non liées à des transitions dans l'automate (les cas où l'on passe par les mêmes sommets à deux moments différents sont négligeables), la proportion de dièdres dans le multigraphe est 2β (chaque dièdre contient 2 multiarêtes) et la proportion de triangles est $3t\frac{\alpha}{1-t}$ (*cf.* §3.7). Ces égalités permettent, connaissant les proportions de dièdres et de triangles dans un graphe de dettes réel, et ayant fixé le paramètres α, de calibrer les paramètres β et t.

4.3 Résultats obtenus et limites du modèle

L'algorithme de génération V3 donne de bons résultats topologiques : les proportions de dièdres et de triangles sont proches de la réalité :

	M1	Multigraphe généré
Proportion de dièdres	$4,04 \cdot 10^{-2}$	$4,06 \cdot 10^{-2}$
Proportion de triangles	$8,8 \cdot 10^{-4}$	$10,3 \cdot 10^{-4}$

Plusieurs inconvénients apparaissent cependant :

— Tout d'abord, aucune calibration automatique de α n'a été faite : en effet, on n'a pas trouvé de lien simple entre la valeur de α et les statistiques des multigraphes étudiés. Les valeurs utilisées pour celui-ci sont fixées empiriquement, dans la plupart de nos tests on a pris $\alpha = 0,79$. On note cependant les proportions de dièdres et de triangles semblent invariantes si on change α, après plusieurs α testés. On sent cependant que ce paramètre a une grande importance dans la génération, d'où l'importance de sa calibration.

— De plus, si la structure topologique non pondérée du multigraphe est satisfaisante, les paramètres liés aux sommes échangées ne sont pas cohérents. En particulier, le coefficient d'amplification final (*cf.* §2) est de l'ordre de 1,9 au lieu de 1,3, ce qui est une différence significative dans le cadre de la compensation de dettes. De plus, il y a un facteur 0,03 entre la variance des positions nettes dans un graphe réel et dans un graphe généré.

Algorithme 1 : Génération des sommes

Entrées : Exposant γ, seuil bas $b_{\min}$, seuil médian b_{med}, seuil haut $b_{\max}$, exposant p_w, somme m

Résultat : Somme $S > 0$

Choisir y aléatoirement et uniformément dans $[0, 1[$;

Poser $z = \left\lfloor (1-y)^{\frac{1}{1-\gamma}} \right\rfloor$; `// Ce calcul découle de §3.3`

;

si $z > b_{max}$ **alors**

  Relancer l'algorithme depuis le début ; `// On ne veut pas de trop grandes valeurs`

sinon

  Choisir r aléatoirement et uniformément dans $[0, 1[$;

  si $r < \exp\left(-\left(\frac{z}{b_{min}}\right)^{\gamma}\right)$ **alors**

    Relancer l'algorithme depuis le début ; `// On veut réduire le nombre de petites valeurs`

  sinon si $z > b_{med}$ **alors**

    Choisir t aléatoirement et uniformément dans $[0, 1[$;

    Renvoyer $S = \frac{z^{p_w \cdot t}}{m}$; `// On veut réduire les grandes valeurs`

  sinon

    Renvoyer $S = \frac{z}{m}$

fin

4.4 Répartition des sommes

Dans un multigraphe de dettes, une loi de puissance présente des inconvénients pour modéliser les sommes échangées (Fig. 6-5). En effet, la loi de puissance surreprésente les petites valeurs, qui n'ont aucune signification économique : les factures de moins de 100€ sont sous-représentées dans le monde des entreprises. De plus, la loi de puissance génère quelques très grandes valeurs, trop importantes pour être économiquement réalistes. On cherche donc à remédier à ces deux travers. Pour ce faire, on propose un nouvel algorithme de génération des sommes (Alg. 1), qui, bien qu'inspiré d'une loi de puissance, s'en éloigne significativement.

4.5 Calibration, résultats obtenus et limites du modèle

L'algorithme avec la génération des sommes améliorées est appelé algorithme de génération V5. La calibration de γ reste faite par principe de maximum de vraisemblance en supposant une loi de puissance. La valeur de m est choisie pour que la moyenne des sommes échangées soit la même que dans un multigraphe réel. Les différents paramètres sont calibrés « à la main », auncune étude mathématique n'ayant été réalisée. Celui-ci donne cependant de bien meilleurs résultats sur le plan des sommes échangées (*cf.* §3.4) : ainsi, le coefficient d'amplification final du graphe généré est 1,362 contre 1,335 sur un graphe réel, et il n'y a plus qu'un rapport 0,13 entre les deux variances des positions nettes. Les statistiques obtenues concernant les sommes sont bien meilleures que précédemment, même si l'absence de calibration automatique nous empêche de coller complètement à ceux-ci. Cette absence de calibration pour les seuils $b_{\min}, b_{\max}, b_{\text{med}}$ ainsi que pour les paramètres p_w et α est l'inconvénient majeur de cet algorithme qu'il faudrait résoudre avec une étude mathématique poussée, pour ainsi valider ou invalider le modèle proposé pour la distribution des sommes.

Par ailleurs, la répartition des degrés est mauvaise : en effet, les degrés ne suivent pas une loi de puissance sur le multigraphe généré. Pour résoudre ce problème, on pourrait ensuite permuter des arêtes du multigraphe de manière à changer la distribution des degrés, ou chercher des critères de distribution locaux des sommes pour mieux correpondre à la réalité.

5 Détection de communautés

Les critères évoqués jusqu'ici pour étudier les multigraphes de dettes sont assez simples : ce sont des paramètres globaux qui, s'ils contiennent un peu de la structure du graphe, n'en donnent pas une vision très précise. C'est pourquoi on se propose d'introduire la détection de communautés comme outil de comparaison entre les multigraphes générés et réels. Les algorithmes de détection de communautés, outre leur intérêt en propre pour trouver de la structure dans une écconomie, peuvent servir dans notre cas à différencier les différents algorithmes de génération. En effet, les multigraphes générés, s'ils ressemblent à des multigraphes réels, doivent réagir de la même manière que ces derniers à la recherche de communautés. La détection de communautés est ainsi un outil pour vérifier si un modèle de génération est réaliste ou non.
Dans toute cette partie, on travaille avec des graphes simples non orientés. Ce choix est fait pour des raisons pratiques - la plupart des algorithmes existant fonctionne sur des graphes simples non orientés - et économiques - on s'intéresse ici aux liens entre les différents acteurs, et ces liens sont seulement liés aux factures entre eux, sans considération pour l'orientation de ces factures. La détection de communautés dans des multigraphes de dettes se fait donc après simplification du multigraphe et sommation des arêtes opposées.
La détection de communautés consiste à trouver un ou plusieurs sous-ensembles de sommets qui sont fortement connectés les uns aux autres. La définition de « fortement connectés » est bien sûr ambigüe, et les critères utilisés dépendent des propriétés qu'on cherche à obtenir sur ces communautés. De nombreux algorithmes de détection de communautés existent, on s'est notamment interessé lors de ce stage à l'algorithme de dépliage rapide des communautés [12], l'algorithme SCoDA [13] ou l'algorithme SLPA [14]. On ne détaillera que ce dernier pour des raisons de concision.

5.1 Degrés

Définition 8 (Degré)**.** Soit $G = (V, E, y)$ un graphe simple orienté pondéré et $s \in V$. On appelle :
— *degré* de s dans G l'entier $\deg(s) = \sum_{\{u,s\}\in E} 1$. C'est le nombre d'arêtes impliquant s.
— *degré pondéré* de s dans G la valeur $\deg_p(s) = \sum_{\{u,s\}\in E} y(\{u,s\})$. On l'appelle aussi *flux total* en s dans d'autres contextes.

Lorsqu'on trace les histogrammes en échelle loglog pour les degrés non pondérés, les courbes tracées semblent être des droites. On fait cette hypothèse par la suite, ce qui signifie que les degrés suivent des lois de puissance (*cf.* §3.3).

5.2 Approximation par les sommets à haut degré, algorithme du chef d'équipe

On propose ici un algorithme ressemblant à K-means pour former des communautés dans un graphe. On appelle celui-ci *algorithme du chef d'équipe* (Alg. 2). Celui-ci consiste à choisir $k \in \mathbb{N}^\star$ centres pour les communautés, puis à faire croître les communautés à partir de ces centres en y ajoutant les sommets qui en sont le plus proche.

Algorithme 2 : Algorithme du chef d'équipe

Entrées : Graphe pondéré $G = (V, E, y)$, nombre de communautés k, critère de choix des centres Γ, critère d'expansion des communautés E

Résultat : Partition de G en communautés $C_1, \cdots, C_k$

Choisir k centres $c_1, \cdots, c_k$ pour les communautés selon Γ et poser $C_i = \{c_i\}$ pour tout $i \in [\![1, k]\!]$;

Tant que Un sommet n'a pas de communauté **faire**

| Choisir un sommet s sans communauté et une communauté C_i selon E et ajouter s à C_i

fin

Renvoyer les communautés obtenues

Les temps de calcul et les résultats obtenus dépendent fortement des critères choisis. Quelques critères vont être étudiés, pour les centres et pour l'expansion.

5.2.1 Critère de choix des centres

Pour choisir les centres des communautés, on va se baser sur le degré, l'idée étant de choisir pour centre une entreprise ayant soit beaucoup de partenaires, soit de gros échanges (les deux étant souvent liés). On choisit ainsi comme centres les sommets ayant le plus haut degré ou le plus haut degré pondéré, selon le critère qu'on juge le plus réaliste.

5.2.2 Critère d'expansion

On propose deux critères pour l'expansion des communautés :

1. Le premier critère consiste à choisir les communautés C_i successivement, et à choisir pour sommet le sommet s le plus fortement relié à cette communauté, c'est-à-dire qui maximise $\sum_{u \in C_i} y(\{u, s\})$. On ajoute alors s à C_i. Les communautés gagnent ainsi des éléments l'une après l'autre, ce qui donne des communautés très équilibrées. On n'ajoute cependant pas à une communauté un sommet qui n'est pas relié à celle-ci : si une communauté se trouve déconnectée de tous les sommets non assignés, elle est terminée.
2. Le second critère est plus complexe : on choisit, pour chaque communauté C_i le sommet s_i qui est le plus fortement relié à celle-ci. On note $p_i = \sum_{u \in C_i} y(\{u, s_i\})$ le poids qui relie s_i à C_i. Puis, on cherche j maximisant $\frac{p_i}{\varphi(|C_i|)}$, où $\varphi : \mathbb{N}^\star \to \mathbb{R}_+^\star$ est une fonction permettant de pondérer par la taille de la communauté. Alors on ajoute s_j à C_j. La fonction φ est prise monomiale pour des raisons de calcul. On peut ainsi faire des communautés plus ou moins équilibrées selon l'exposant de φ.

Le second critère donne de meilleurs résultats que le premier, mais rend les calculs beaucoup plus lents. En effet, il y a un facteur k entre les complexités des deux algorithmes, ou k est le nombre de communautés.

5.3 SLPA

L'algorithme SLPA (*Speaker-Listener Label Propagation Algorithm*), proposé en 2011 par J. Jierui et B. K. Szymanski [14], détecte des communautés par propagation d'étiquette : chaque sommet dispose d'une ou plusieurs étiquettes, qu'il communique à ses voisins. Il permet d'obtenir des communautés de tailles diverses. Cependant, les résultats obtenus avec cet algorithme sont peu concluants, les communautés formées étant beaucoup plus faiblement connectées que le graphe total : par exemple, les coefficients d'amplification finaux sont trop faibles. De nombreux essais ont été réalisés avec cet algorithme lors du stage, sans résultats concluants.

5.4 Application à la comparaison des algorithmes de génération

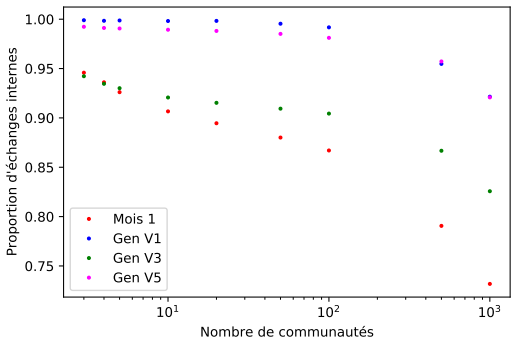


FIGURE 10 – Application de l'algorithme du chef d'équipe aux différents graphes générés

Dans l'exemple de la figure 5.4, on a utilisé un critère de choix des centres maximisant le nombre de voisins, et le premier critère d'expansion. En abscisses on représnte le nombre de centres, et en ordonnées la proportion des sommes échangées qui le sont entre deux membres d'une même communauté. On voit alors que l'algorithme de génération V5, s'il est plus réaliste sur le plan des sommes échangées, est beaucoup plus distant de la réalité que l'algorithme V3 dans sa réaction à l'algorithme du chef d'équipe. On n'a pas trouvé d'explication à ce phénomène, qui pourrait faire l'objet d'études plus poussées.

Conclusion

Lors de ce stage, on a élaboré différents algorithmes pour générer des multigraphes de dettes, avec des améliorations successives permettant de gagner en réalisme. On a tout d'abord proposé un modèle de génération uniforme utilisant des lois de puissance, puis on a étudié un modèle markovien d'ordre 3, de manière à jouer sur la structure topologique, pour enfin modifier la génération des sommes. On a finalement étudié de la détection de communautés pour estimer les performances de ces différents algorithmes. Plusieurs pistes existent pour continuer l'élaboration du modèle :

— Une étude mathématique poussée permettrait de calibrer mécaniquement les paramètres, qui sont pour l'instant calibrés empiriquement.
— Une recherche d'un modèle ergodique, tendant vers un équilibre asymptotique peut être envisagée. Il permettrait d'améliorer de manière locale les propriétés globales du multigraphe. Une tentative a été effectuée durant de stage, mais n'a pas donné de résultats concluants
— Les algorithmes présentés fonctionnent sur des résultats asymptotiques, et sont donc peu efficaces pour générer des multigraphes de dettes de petite taille
— Enfin, les algorithmes de détection de communautés présentés ne donnant que des résultats médiocres sur les multigraphes de dettes, il pourrait être utile de trouver des algorithmes plus efficaces sur ces derniers.

Un autre point important à souligner est la représentativité des données étudiées : si celles-ci correspondent bien à une économie réelle, on ne peut cependant pas savoir à priori si d'autres économies réelles seraient similaires. Il serait donc intéressant de travailler sur d'autres économies pour vérifier nos algorithmes de génération.
Ce stage était une première expérience de recherche, et ce fut une expérience étonnante de nouveauté. En effet, on passe du statut d'étudiant assimilant un cours à celui de chercheur créant ses propres outils pour parvenir à ses propres fins.

Remerciements

Merci à toute l'équipe MOCQUA du Loria de m'avoir accueilli pendant ces deux mois.
Merci à Irène Marcovici pour ses conseils mathématiques toujours bienvenus.
Merci à Massimo Amato, Lucio Gobbi et Daniele Citterio pour nous avoir fourni les jeux de données et expliqué les subtilités économiques de la compensation de dettes.
Enfin, un grand merci à Nazim Fatès pour m'avoir accompagné durant ce stage, me donnant des conseils informatiques, philosophiques ou étymologiques tout au long de ces deux mois. Sans lui ce stage n'aurait pas été possible.

Bibliographie

[1] G. Iosifidis, Y. Charette, E. M. Airoldi, G. Littera, L. Tassiulas, and N. A. Christakis. Cyclic motifs in the Sardex monetary network. *Nature human behaviour*, 2(11) :822–829, 2018.

[2] T. Fleischman, P. Dini, and G. Littera. Liquidity-saving through obligation-clearing and mutual credit : an effective monetary innovation for SMEs in times of crisis. *Journal of Risk and Financial Management*, 13(12) :295, 2020.

[3] A. Rousseau. Mémoire maths-info. 2021.

[4] M. L. Goldstein ans S. A. Morris and G. G. Yen. Problems with fitting to the power-law distribution. *The European Physical Journal B-Condensed Matter and Complex Systems*, 41(2) :255–258, 2004.

[5] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM review*, 51(4) :661–703, 2009.

[6] Y. S. Yogesh and A. ClausetS. *Power-Law Distributions and Binned Empirical data.* PhD thesis, University of Colorado at Boulder, 2012.

[7] X. Chen, W. Zang, and Q. Zhao. Densities, matchings, and fractional edge-colorings. *SIAM Journal on Optimization*, 29(1) :240–261, 2019.

[8] M. K. Goldberg. Clusters in a multigraph with elevated density. *The electronic journal of combinatorics*, pages R10–R10, 2007.

[9] G. Fagiolo. Clustering in complex directed networks. *Physical Review E*, 76(2) :7–24, 2007.

[10] S. N. Soffer and A. Vazquez. Network clustering coefficient without degree-correlation biases. *Physical Review E*, 71(5), 2005.

[11] P. L. Krapivsky, S. Redner, and F. Leyvraz. Connectivity of growing random networks. *Physical review letters*, 85(21), 2000.

[12] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics : theory and experiment*, 2008(10) :10008, 2008.

[13] A. Hollocou, J. Maudet, T. Bonald, and M. Lelarge. A linear streaming algorithm for community detection in very large networks. *arXiv preprint*, 1703(02955), 2017.

[14] J. Jierui and B. K. Szymanski. Towards linear time overlapping community detection in social networks. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2012.