



HAL
open science

Nouvelles visualisations 3D collaboratives de graphes pour l'exploration de relations intra et inter communautés

Piriziwè Kobina

► **To cite this version:**

Piriziwè Kobina. Nouvelles visualisations 3D collaboratives de graphes pour l'exploration de relations intra et inter communautés. Synthèse d'image et réalité virtuelle [cs.GR]. Ecole nationale supérieure Mines-Télécom Atlantique, 2022. Français. NNT : 2022IMTA0299 . tel-03766450

HAL Id: tel-03766450

<https://theses.hal.science/tel-03766450v1>

Submitted on 1 Sep 2022

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

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

THÈSE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPÉRIEURE
MINES-TÉLÉCOM ATLANTIQUE BRETAGNE
PAYS-DE-LA-LOIRE - IMT ATLANTIQUE

ÉCOLE DOCTORALE N° 601
*Mathématiques et Sciences et Technologies
de l'Information et de la Communication*
Spécialité : *Informatique*

Par

Piriziwè KOBINA

**Nouvelles visualisations 3D de graphes pour l'exploration
des relations intra et inter communautés**

Thèse présentée et soutenue à IMT ATLANTIQUE - CAMPUS DE BREST, le 13 Juillet 2022
Unité de recherche : Lab-STICC UMR 6285 CNRS
Thèse N° : 2022IMTA0299

Rapporteurs avant soutenance :

Christophe HURTER Professeur, École Nationale d'Aviation Civile - Toulouse
Gilles VENTURINI Professeur, École Polytechnique de l'Université de Tours

Composition du Jury :

Président :	Bruno ARNALDI	Professeur, INSA de Rennes
Examineurs :	Anne-Gwenn BOSSER	Maître de conférences, ENIB
	Alma CANTU	Lecturer, Newcastle University
	Guillaume GADEK	Ingénieur de recherche, Airbus Defence and Space
	Christophe HURTER	Professeur, École Nationale d'Aviation Civile - Toulouse
Dir. de thèse :	Gilles VENTURINI	Professeur, École Polytechnique de l'Université de Tours
	Thierry DUVAL	Professeur, IMT Atlantique
Co-encadrant :	Laurent BRISSON	Maître de conférences, IMT Atlantique

REMERCIEMENTS

Il y a tant de personnes qui ont contribué à cette thèse directement ou indirectement et auxquelles je suis redevable. Je m'excuse de ne pas pouvoir tous les nommer.

Avant tout, je tiens à exprimer ma profonde gratitude à mon directeur de thèse, Thierry DUVAL, et à mon encadrant, Laurent BRISSON, pour leur soutien, leur patience et leurs encouragements. Travailler avec eux a été un grand plaisir pour moi et c'est un privilège absolu que je ne saurais assez apprécier.

Mes remerciements vont également à la région Bretagne et au Pôle d'Excellence Cyber (PEC) pour le financement de cette thèse. Je tiens à remercier Inna LYUBAREVA, notre experte économiste, avec qui j'ai travaillé afin d'avoir un cas concret pour instancier nos méthodes de visualisation de graphes. Je remercie également Bruno ARNALDI et Frédéric MAJORCZYK, membres de mon comité de suivi de thèse, pour leurs conseils pendant ces années de thèse.

Je tiens à remercier Gilles VENTURINI et Christophe HURTER pour avoir accepté de relire ce manuscrit et pour leurs critiques à travers leur rapport. Mes remerciements vont également à tous les membres du jury pour leur disponibilité.

Le travail et la recherche ne sont pas amusants sans de bons collègues. J'étais heureux de faire partie d'un environnement très stimulant de personnes créatives et brillantes. Ainsi, je tiens à remercier l'ensemble du département INFO, du département LUSSI, les membres de l'équipe INUIT, le personnel administratif du Lab-STICC et d'IMT Atlantique.

Enfin, je tiens à remercier mes amis pour leurs encouragements, ma famille notamment ma mère, Pyalo EDJAMFEILE, et ma conjointe, Nazia KOBINA, pour leur patience, leur soutien, leur amour et leurs encouragements.

SOMMAIRE

1	Introduction	9
1.1	Contexte	10
1.2	Questions de recherche	13
1.3	Contributions	13
1.4	Organisation du manuscrit	14
I	État de l'art	17
2	Théorie des graphes et application à l'analyse de réseaux sociaux	19
2.1	Théorie des graphes	19
2.1.1	Définition d'un graphe	21
2.1.2	Types de graphes	22
2.1.3	Représentation d'un graphe	25
2.1.4	Propriétés d'un graphe	27
2.1.5	Propriétés d'un nœud	30
2.1.6	Rôles d'un nœud dans un graphe	35
2.1.7	Détection de clusters	37
2.1.7.1	Algorithme de Louvain	37
2.1.7.2	Algorithme de Leiden	38
2.1.7.3	Algorithme de Walktrap	41
2.1.8	Modèles de génération de graphes artificiels	42
2.1.8.1	Modèle d'Erdős-Rényi	42
2.1.8.2	Modèle de blocs stochastiques (SBM)	43
2.2	Application à l'analyse des réseaux sociaux	44
2.2.1	Définition des objectifs d'analyse	45
2.2.2	Données et modélisation	45
2.2.3	Résultats	46
2.2.4	Discussion	51

2.3	Conclusion	53
3	Visualisation de graphes	55
3.1	Spatialisation de nœuds	56
3.1.1	Critères de lisibilité d'un graphe	56
3.1.2	Algorithmes de force	57
3.1.3	Algorithmes utilisant la réduction de dimension	61
3.1.3.1	Réduction de dimension	61
3.1.3.2	Algorithmes basés sur la réduction de dimension	62
3.1.4	Synthèse sur la spatialisation des nœuds	64
3.2	Mise en évidence de la centralité des nœuds dans un graphe	66
3.3	Visualisation de clusters dans un graphe	72
3.4	Algorithmes de regroupement d'arêtes	77
3.5	Conclusion	81
II	Contributions	83
4	Visualisation exocentrique de graphes : mise en évidence des propriétés topologiques	85
4.1	Travaux relatifs à la visualisation de graphes mettant en évidence la centralité	86
4.2	Visualisation radiale 3D	88
4.2.1	Concept	88
4.2.2	Projection de la vue uniforme	89
4.2.3	Projection de l'emphase sur le centre	91
4.2.4	Projection de l'emphase sur la périphérie	92
4.3	Regroupement d'arêtes	96
4.4	Mise en évidence des connecteurs	99
4.5	Évaluations	102
4.5.1	Évaluation 1	102
4.5.1.1	Tâches	103
4.5.1.2	Données expérimentales	103
4.5.1.3	Interactions avec la visualisation de graphes	104
4.5.1.4	Hypothèses	104
4.5.1.5	Dispositif expérimental	105

4.5.1.6	Participants	108
4.5.1.7	Résultats	108
4.5.1.8	Discussion	113
4.5.2	Évaluation 2	114
4.5.2.1	Tâches	115
4.5.2.2	Données expérimentales	115
4.5.2.3	Interactions avec la visualisation de graphes	115
4.5.2.4	Hypothèses	117
4.5.2.5	Dispositif expérimental	117
4.5.2.6	Participants	118
4.5.2.7	Résultats	118
4.5.2.8	Discussion	124
4.6	Conclusion	127
5	Visualisation égocentrique de graphes	129
5.1	Travaux relatifs à la visualisation égocentrique	130
5.2	Métaphore d'un espace de visualisation rétractable	134
5.2.1	Concept	134
5.2.2	Synthèse	136
5.2.3	Discussion	138
6	Application de nos visualisations à l'analyse d'un réseau social	141
6.1	Analyse des relations intercommunautés	142
6.2	Analyse des relations intracommunauté	145
7	Conclusion et perspectives	149
	Conclusion	149
	Conclusion	149
	Perspectives	152
	Contributions	153
	Bibliographie	155
	Annexes	174

INTRODUCTION

De nombreux phénomènes du monde réel, tels que le trafic Internet, les interactions entre protéines, les chaînes alimentaires, les transactions financières et les personnes et leurs liens sociaux, peuvent être modélisés sous la forme de systèmes d'entités interconnectées qui définissent ainsi des réseaux. Un modèle mathématique est souvent utilisé pour représenter cet ensemble d'entités et leurs interactions : celui des graphes [42]. Un graphe peut modéliser un système d'entités où un nœud représente une entité et où une arête représente une connexion entre deux entités. Les graphes sont fondamentaux dans de nombreux domaines de recherche tels que l'informatique, la chimie, l'économie et l'analyse des réseaux sociaux.

Plusieurs travaux [132, 118] s'intéressent à la compréhension et à la description de ces types de graphes à l'aide de la théorie des graphes, une branche des mathématiques concernant l'étude des propriétés et des mesures sur les graphes. Les propriétés et mesures, telles que le nombre de nœuds et d'arêtes, le degré d'un nœud et les mesures de centralité, permettent de caractériser des graphes. Le nombre de nœuds et d'arêtes indique la taille et la densité d'un graphe, alors que le degré d'un nœud indique le nombre de nœuds voisins auxquels il est connecté. Les mesures de centralité donnent quant à elles la position topologique d'un nœud dans un graphe.

En raison de l'importance croissante des études sur les graphes et de la complexité sans cesse croissante de ces graphes, leur visualisation s'est également avérée très utile pour explorer des structures dans ces différentes études. La visualisation de graphes peut alors être considérée comme un outil d'accompagnement dans ces études. Au-delà des simples fins de présentation, une représentation visuelle appropriée peut contribuer de manière significative à la compréhension des propriétés structurelles d'un graphe étudié. Mais il existe souvent des contraintes et des critères de lisibilité spécifiques au contexte. La visualisation doit représenter la structure du graphe de manière appropriée pour améliorer son exploration et la compréhension des connexions entre les nœuds. Le domaine de représentation des graphes s'intéresse aux méthodes de calcul automatique des positions

géométriques des nœuds et arêtes dans un graphe. Il s'intéresse également aux méthodes de représentation des arêtes et aux méthodes permettant de mettre en évidence certaines propriétés topologiques des nœuds dans le graphe. Les méthodes de représentation des arêtes ont pour objectif principal la réduction d'occlusions dues à la densité des arêtes. Les méthodes mettant en évidence les propriétés topologiques visent à transmettre visuellement ces informations afin de faciliter leur comparaison. Plusieurs algorithmes de représentation de graphes existent et le choix d'un algorithme dépend de l'application particulière.

Les travaux de cette thèse pourraient avoir des applications aussi bien civiles que militaires. Sur le plan civil, il s'agirait d'étudier par exemple un réseau issu d'une plateforme de financement participatif afin d'identifier des structures sociales et des interactions sociales les moins évidentes. Il s'agirait d'identifier également les profils des contributeurs et les rôles des contributeurs au sein d'un groupe (influenceurs, suiveurs, etc.). Si un analyste s'intéresse à une caractéristique des contributeurs, un niveau d'expertise par exemple, il pourrait être aussi intéressé par l'identification des contributeurs experts et les contributeurs entourés d'experts. Sur le plan militaire, il s'agirait d'étudier par exemple un réseau terroriste afin d'identifier la structure de cellules (centralisées ou non) et les individus-types (centraux, connecteurs entre des cellules). Il s'agirait d'identifier également les profils des membres (comptable, militaire, prisonnier, etc.), les rôles des membres (recruteur, formateur, etc.) et un groupe qu'on pourrait infiltrer sans éveiller de soupçons.

1.1 Contexte

L'étude des propriétés des graphes et de leurs mesures (les mesures de centralité, par exemple) permet de révéler certains aspects sémantiques derrière leur topologie. Il peut s'agir d'identifier et de caractériser des nœuds qui ont des propriétés particulières (les nœuds qui sont connectés à beaucoup d'autres nœuds, les nœuds qui ont une position centrale par rapport aux autres nœuds, etc.) [148, 163]. Il peut également s'agir d'identifier des zones fortement connectées ou de comprendre le comportement de groupes dans le graphe. Une façon de pouvoir étudier le comportement de groupes dans un graphe consiste à le découper en groupes de nœuds.

Plusieurs travaux [8, 144] s'intéressent à l'identification des groupes de nœuds fortement connectés entre eux et moins connectés avec les nœuds des autres groupes. Cette identification de groupes est connue sous le nom de *détection de communautés ou clusters*.

Un cluster est un regroupement donné par un algorithme de détection de clusters. Sur les graphes, ce cluster est souvent un groupe de nœuds. Dans le domaine de l'analyse des réseaux sociaux, un tel groupe de nœuds est appelé une communauté. D'autres travaux [31, 24] s'intéressent à l'étude des caractéristiques des groupes de nœuds identifiés à l'aide d'un algorithme de détection de clusters. Ainsi, l'analyse d'un graphe peut être résumée en trois niveaux : macroscopique, mésoscopique et microscopique.

L'analyse macroscopique consiste à étudier les propriétés qui décrivent la structure globale du graphe [125]. Il s'agit par exemple de décrire la manière dont les nœuds sont connectés entre eux et de vérifier s'il existe un chemin entre deux nœuds du graphe. Ainsi, l'analyse macroscopique d'un réseau issu d'une plateforme de financement participatif consisterait par exemple à identifier des structures sociales et des interactions sociales les moins évidentes dans le réseau. Si un analyste s'intéresse à une caractéristique des contributeurs, un niveau d'expertise par exemple, il pourrait être aussi intéressé par l'identification des contributeurs experts et les contributeurs entourés d'experts. L'analyse macroscopique d'un réseau terroriste consisterait à identifier par exemple la structure de cellules (centralisées ou non).

L'analyse microscopique d'un graphe consiste quant à elle à étudier les propriétés des nœuds à travers leur connectivité avec les autres nœuds [163, 68]. Ainsi, l'analyse microscopique du réseau issu d'une plateforme de financement participatif consisterait à identifier par exemple les contributeurs les plus connectés et la position des contributeurs dans le réseau (influenceurs, par exemple). L'analyse microscopique du réseau terroriste consisterait quant à elle à identifier par exemple des individus-types (centraux, par exemple).

L'analyse mésoscopique d'un graphe est un niveau intermédiaire qui repose sur un découpage du graphe en groupes de nœuds [125]. Ainsi, l'analyse mésoscopique du réseau issu d'une plateforme de financement participatif consisterait à identifier par exemple les rôles des contributeurs dans un groupe (connecteurs, par exemple). Celle du réseau terroriste consisterait à identifier par exemple des connecteurs entre des cellules et un groupe qu'on pourrait infiltrer sans éveiller de soupçons.

La visualisation de graphes permet d'explorer des structures dans les graphes afin de comprendre la connectivité entre les nœuds. Elle permet également de présenter visuellement le résultat de certaines propriétés étudiées : par exemple, des visualisations radiales [13, 16] permettent de comparer visuellement la centralité des nœuds, qui est un résultat d'une étape d'analyse précédente. Cependant, ces visualisations radiales manquent d'espace d'affichage lorsqu'un graphe contient plusieurs milliers de nœuds. Elles ne sont

pas adaptées pour visualiser les rôles des nœuds (les connecteurs, par exemple). Elles ne sont pas non plus adaptées pour visualiser la centralité et une autre propriété (topologique ou métier) simultanément ou la centralité et un rôle des nœuds (les connecteurs).

Il existe plusieurs algorithmes de positionnement des nœuds dans l'espace [50, 147]. Une approche fréquemment utilisée en raison de sa simplicité et de ses résultats attrayants est celle basée sur le paradigme de force [50, 83]. L'agencement géométrique des nœuds et arêtes d'un graphe est déterminé à l'aide d'une mesure de qualité, qui évalue à quel point la représentation visuelle du graphe reflète la connectivité des nœuds. Certaines des méthodes de représentation des graphes ont des équivalents dans d'autres disciplines scientifiques telles que l'analyse de données ou les statistiques. Bien qu'elles aient été conçues pour des tâches plus générales d'analyse de données [97, 82], nombre de ces méthodes ont été adaptées aux graphes [15, 64] et peuvent fournir des représentations visuelles significatives mettant en évidence certaines structures dans un graphe : ce sont des algorithmes utilisant une approche de réduction de dimension.

Afin de faciliter l'exploration et la compréhension d'un graphe, certains des algorithmes de force visent entre autres à réduire le nombre de croisements d'arêtes [50]. Cependant, certains nœuds peuvent être très éloignés de leurs voisins à cause des forces d'attraction et de répulsion. Les algorithmes utilisant une approche de réduction de dimension ont quant à eux pour but principal de réduire la différence entre la distance euclidienne et la longueur du plus court chemin entre deux nœuds du graphe [69, 15]. Certains de ces algorithmes permettent de mettre en évidence les connectivités locales d'un graphe [98], tandis que d'autres permettent de mettre en évidence la structure globale du graphe [142].

Il existe également des méthodes permettant de visualiser les groupes de nœuds détectés à l'aide d'un algorithme de détection de clusters [70] et des méthodes permettant de visualiser à la fois les groupes de nœuds et les connecteurs [30]. Cependant, les méthodes permettant de visualiser seulement les groupes de nœuds ne sont pas adaptées pour visualiser les connecteurs et la centralité dans un groupe de nœuds. Les méthodes permettant de visualiser à la fois les groupes de nœuds et les connecteurs ne sont quant à elles pas adaptées pour visualiser la centralité dans un groupe de nœuds.

Plusieurs méthodes sont également proposées afin de réduire les problèmes d'occlusion de nœuds et d'arêtes qu'on peut observer au niveau de la visualisation d'un graphe contenant plusieurs milliers d'arêtes. Certaines méthodes [41, 80] regroupent les arêtes afin d'améliorer l'exploration du graphe et d'identifier par exemple la connectivité entre groupes de nœuds créés par le regroupement d'arêtes. D'autres exploitent la 3D pour

bien distribuer les nœuds et pour améliorer la perception de la connectivité entre les nœuds [151, 107]. Les méthodes qui exploitent la 3D sont limitées lorsqu'elles sont associées à un affichage 2D classique, car les indices de profondeur en 3D peuvent rendre la compréhension d'un grand graphe plus complexe [161]. Des travaux ont ensuite suggéré qu'une immersion dans les données peut atténuer cette complexité [60, 29]. Cependant, un utilisateur peut se retrouver empêtré dans les nœuds et les arêtes. Cela n'améliore pas l'exploration d'un graphe en immersion.

1.2 Questions de recherche

L'objectif de cette thèse consiste à étudier l'apport de la visualisation 3D immersive par rapport à la visualisation 2D classique pour améliorer l'exploration des relations entre les nœuds d'un groupe et entre des groupes de nœuds. Pour cela, nous nous intéressons à explorer des travaux relatifs à l'étude des propriétés et mesures sur les graphes, à la spatialisation des nœuds, à la représentation des arêtes et à la visualisation de la centralité des nœuds et les connecteurs afin de pouvoir répondre aux questions suivantes :

- comment optimiser la distribution des nœuds dans l'espace ?
- comment réduire le problème d'occlusions dû à la densité des arêtes ?
- comment mettre en évidence la centralité des nœuds dans un graphe pour aider à comparer leur valeur ?
- comment mettre en évidence les connecteurs dans un groupe à la suite d'un découpage d'un graphe en groupes de nœuds ?
- comment visualiser simultanément la centralité et les connecteurs dans un groupe de nœuds ?
- comment passer de manière continue d'un niveau d'analyse à un autre ?
- comment améliorer l'exploration d'un graphe en environnement immersif ?

1.3 Contributions

Pour répondre à ces questions, nous avons proposé deux méthodes de visualisation de graphes : une visualisation exocentrique qui permet de mettre en évidence la centralité des nœuds dans un graphe et une visualisation égocentrique particulièrement adaptée à l'exploration d'un graphe en immersion. Une visualisation exocentrique consiste à observer un objet d'un point de vue extérieur [44], tandis qu'une visualisation égocentrique consiste

à observer un environnement depuis l'intérieur [44].

Notre visualisation exocentrique consiste à projeter des visualisations radiales 2D sur différentes surfaces 3D tout en gardant la visualisation radiale 2D en vue de dessus. Nous avons proposé un mécanisme permettant de passer de manière continue d'une surface à l'autre. Nous avons ensuite comparé l'efficacité de notre visualisation exocentrique par rapport à des visualisations radiales 2D et évalué l'utilisabilité de la 3D pour la visualisation de graphes. Les résultats de cette évaluation ont montré que notre visualisation exocentrique est significativement plus performante que les visualisations radiales 2D quand on s'intéresse au centre ou à la périphérie d'un graphe.

Nous avons également proposé de combiner un mode de visualisation d'arêtes regroupées avec un mode de visualisation d'arêtes détaillées. Ce mécanisme permet à un utilisateur de pouvoir passer d'un mode à un autre en faisant varier l'opacité des arêtes.

Dans le cas d'un graphe découpé en groupes de nœuds, il peut être utile de visualiser simultanément la centralité des nœuds et les connecteurs d'un groupe de nœuds sans avoir à afficher les autres groupes. Nous avons proposé que chaque connecteur soit entouré d'un ensemble d'anneaux où chaque anneau représente un groupe voisin et où l'épaisseur d'un anneau représente le nombre de nœuds voisins du connecteur dans le groupe. Nous avons également proposé un mécanisme permettant de décoller les connecteurs de la surface afin d'améliorer leur visibilité.

Enfin, nous avons proposé une visualisation égocentrique, qui est particulièrement adaptée à l'exploration d'un graphe en immersion. Elle consiste à créer une zone vide autour d'un utilisateur afin d'éviter qu'il soit gêné par des nœuds et des arêtes. Elle offre également à l'utilisateur une possibilité de rétracter l'espace d'affichage du graphe afin d'avoir une vue d'ensemble du graphe sans quitter le mode égocentrique. Cependant, nous n'avons pas encore pu mener d'expérimentation utilisateur afin de valider l'intérêt de cette visualisation égocentrique.

1.4 Organisation du manuscrit

La première partie de cette thèse présente un état de l'art sur la théorie des graphes et son application à l'analyse des réseaux sociaux et sur les méthodes de visualisation de graphes.

Le chapitre 2 va donc traiter de la théorie des graphes et son application à l'analyse des réseaux. Dans ce chapitre, nous présentons non seulement les différentes propriétés et

mesures sur un graphe, mais aussi leurs usages. Nous illustrons ensuite une application de la théorie des graphes à l'analyse des réseaux.

Le chapitre **3** va traiter des méthodes de visualisation de graphes. Nous présentons les algorithmes de spatialisation de nœuds, notamment les algorithmes de force et les algorithmes utilisant une approche de réduction de dimension. Nous présentons ensuite des méthodes qui permettent de mettre en évidence la centralité des nœuds, puis des méthodes permettant de visualiser les groupes de nœuds dans un graphe partitionné. Nous présentons enfin les algorithmes qui permettent de regrouper des arêtes afin de réduire le recouvrement visuel dans un graphe densément connecté.

La deuxième partie de cette thèse présente nos contributions sur la visualisation de la centralité et des connecteurs, sur une visualisation égocentrique et sur une application de nos propositions à l'analyse d'un réseau social.

Ainsi, dans le chapitre **4** nous rappelons certains travaux relatifs à la mise en évidence de la centralité des nœuds et leurs limitations en termes de manque d'espace. Nous présentons ensuite notre visualisation exocentrique et notre mécanisme de combinaison de mode de visualisation d'arêtes regroupées avec un mode de visualisation d'arêtes détaillées. La visualisation exocentrique va permettre de disposer de plus d'espace afin de bien distribuer les nœuds. Elle va également mettre en évidence la centralité des nœuds. Nous présentons ensuite notre méthode mettant en évidence les connecteurs d'un groupe de nœuds puis les résultats d'évaluations de notre visualisation exocentrique. Rappelons que la visualisation des connecteurs peut être utile lorsqu'un analyste s'intéresse à un groupe de nœuds sans avoir à afficher les autres groupes du graphe.

Afin d'améliorer l'exploration d'un graphe dans un environnement immersif et d'éviter qu'un utilisateur se retrouve empêtré dans les nœuds et les arêtes, nous présentons dans le chapitre **5** notre métaphore de visualisation égocentrique qui consiste à créer une zone vide autour de l'utilisateur. Cette visualisation égocentrique offre également à l'utilisateur une possibilité de rétracter l'espace d'affichage du graphe afin de passer de manière flexible d'une vue détaillée égocentrique à une vue d'ensemble du graphe tout en restant égocentrique.

Dans le chapitre **6** nous illustrons nos contributions en les appliquant à l'analyse des réseaux sociaux, au travers de l'étude des données issues d'une plateforme de financement participatif.

Le chapitre **7** est dédié à une synthèse sur nos travaux et aux perspectives de nos futurs travaux potentiels.

PREMIÈRE PARTIE

État de l'art

THÉORIE DES GRAPHS ET APPLICATION À L'ANALYSE DE RÉSEAUX SOCIAUX

L'analyse des graphes est basée sur la théorie des graphes, une branche des mathématiques qui s'intéresse à l'étude des propriétés des graphes. La théorie des graphes a de nombreuses applications, telles que l'informatique, la biologie et l'analyse des réseaux sociaux. En complément des outils et algorithmes utilisés dans ces disciplines, une représentation visuelle des graphes peut aider à explorer intuitivement les données ou à illustrer des résultats d'analyse.

Un graphe décrit des interactions entre entités. L'ensemble d'entités en interaction est appelé un *réseau* que l'on peut modéliser sous forme d'un *graphe* où un nœud représente une entité et où une arête représente une interaction. L'étude des propriétés et des mesures sur les graphes permet de comprendre et de décrire leur structure. Un graphe peut être décrit au niveau macroscopique (structure topologique générale du graphe), au niveau mésoscopique (structure topologique interne des groupes) à la suite d'un découpage du graphe en groupes à l'aide d'un algorithme de détection de clusters, ou au niveau microscopique, c'est-à-dire l'étude des propriétés des nœuds et arêtes.

Afin de comprendre la nature des informations que nous serons amenés à représenter visuellement, nous proposons de présenter dans un premier temps les différentes propriétés et mesures sur un graphe, ainsi que leurs usages. Nous présentons ensuite une application de la théorie des graphes à l'analyse des réseaux sociaux qui permettra d'illustrer dans le chapitre 6 une application de nos contributions.

2.1 Théorie des graphes

Les graphes permettent de modéliser les interactions entre les éléments d'un ensemble, telles que les connexions sociales et les réseaux de relations biologiques, en se ramenant à l'étude de nœuds et d'arêtes. La théorie des graphes concerne donc l'étude des propriétés

de ces types d'ensembles, et comment ils peuvent être utilisés pour modéliser et résoudre une multitude de problèmes.

L'idée de base des graphes a été introduite pour la première fois par le mathématicien suisse Leonhard Euler [42] avec l'étude du problème des sept ponts de la ville de Königsberg (voir la figure 2.1).



FIGURE 2.1 – Le problème des sept ponts de Königsberg étudié par Euler [42] (source des images : Wikipédia). (a) représente la carte de la ville de Königsberg au temps d'Euler mettant en évidence la rivière Pregel et les sept ponts. (b) est la modélisation de cette carte sous forme de graphe où les nœuds représentent les îles et les deux parties continentales et où les arêtes représentent les ponts.

En effet, la ville de Königsberg en Prusse (aujourd'hui Kaliningrad, en Russie) est située de part et d'autre de la rivière Pregel, et comprend deux grandes îles - Kneiphof et Lomse - qui sont reliées l'une à l'autre par un pont et aux deux parties continentales de la ville par six ponts. Le problème était de concevoir une promenade à travers la ville qui traverserait chacun de ces ponts une et une seule fois. Euler, reconnaissant que les contraintes pertinentes sont les deux îles, les parties continentales et les sept ponts, dessine la première représentation visuelle connue d'un graphe.

Cette abstraction d'un problème concret concernant une ville et des ponts à un graphe rend le problème mathématiquement traitable, car cette représentation abstraite ne comprend que les informations importantes pour résoudre le problème. Euler a en fait montré que ce problème spécifique n'avait pas de solution. Cependant, la difficulté à laquelle il a été confronté a été le développement d'une technique d'analyse appropriée, et des tests ultérieurs ont établi cette affirmation avec une rigueur mathématique [7].

Les graphes ont un large éventail d'applications utiles, par exemple :

- l'identification des communautés dans les réseaux, tels que les médias sociaux (recommandations d'amis, de connexion), ou ces derniers temps pour une éventuelle détection de propagation de COVID-19 dans la communauté par le biais des

- contacts ;
- le classement des liens hypertextes dans les moteurs de recherche ;
- l'utilisation du GPS pour trouver le chemin le plus court vers une destination ;
- l'étude des molécules et des atomes en chimie ;
- le séquençage de l'ADN.

2.1.1 Définition d'un graphe

Un graphe est une représentation symbolique d'un réseau et de sa connectivité. Il implique une abstraction de la réalité de sorte qu'elle soit simplifiée comme un ensemble de nœuds liés (figure 2.2, par exemple). Ainsi, un graphe $G = (V, E)$ est composé d'un ensemble V de nœuds qui représentent les entités qui interagissent entre elles et d'un ensemble E d'arêtes qui indiquent la présence d'une interaction ou connexion entre deux nœuds. Le nombre de nœuds $|V|$ est l'**ordre** du graphe tandis que son nombre d'arêtes $|E|$ est appelé **taille** du graphe.

Un nœud peut avoir certaines propriétés propres telles que le poids, la taille, la position et tout autre attribut, et des propriétés basées sur la structure du graphe telles que le nombre de voisins, la communauté à laquelle il appartient. Une arête peut également avoir des propriétés qui impliquent les deux nœuds reliés par l'arête.



FIGURE 2.2 – Graphe non orienté représentant les métadonnées de milliers de documents d’archives, documentant le réseau social de centaines d’acteurs de la Société des Nations [59] (Image extraite de <http://www.martingrandjean.ch/archive-reseau-visualisation-donnees-sciences-humaines/>, licence CC BY 3.0 CH).

Il existe plusieurs types de graphes qui décrivent différents types de problèmes et les contraintes qu’ils comportent. Nous présentons dans la section suivante les principaux types de graphes.

2.1.2 Types de graphes

Il existe différents types de graphes qui se distinguent par rapport à des propriétés topologiques sur les nœuds et les arêtes, sur la présence ou non d’attributs et la façon dont ils tiennent compte du temps.

Graphes orientés et graphes non orientés

Un graphe est dit orienté lorsque la connexion entre deux nœuds est unidirectionnelle. Par exemple, Twitter ou Instagram sont principalement des réseaux orientés, car un individu peut suivre un autre sans que ce dernier le suive en retour. Par contre, un graphe est dit non orienté si la relation entre deux nœuds est mutuelle. Par exemple, être ami(e) avec un individu sur Facebook ou LinkedIn crée automatiquement une connexion mutuelle.

Graphes pondérés

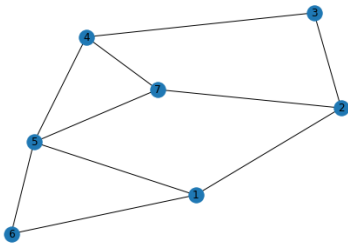
De nombreux graphes peuvent avoir des arêtes contenant un poids associé pour représenter une implication du monde réel telle que le coût, la distance, la quantité, etc. Plusieurs poids peuvent être associés à chaque arête, notamment la distance, le temps de trajet ou le coût monétaire. De tels graphes pondérés sont couramment utilisés pour programmer les GPS et les moteurs de recherche de planification de voyage qui comparent le temps et les coûts des vols, par exemple. Un graphe pondéré peut également être orienté ou non.

Graphes simples

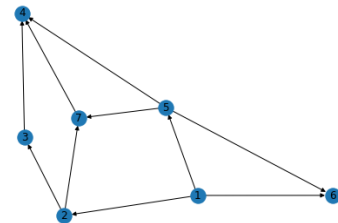
Un graphe simple est un graphe qui ne comporte pas plus d'une arête entre deux nœuds et qui ne comporte aucune boucle (exemple : les figures 2.3a et 2.3b). Une boucle est une arête qui relie un nœud à lui-même. Un graphe simple peut être orienté ou non, pondéré ou non. Un graphe simple de n nœuds possède au plus $n(n - 1)/2$ arêtes s'il n'est pas orienté et $n(n - 1)$ arêtes s'il est orienté.

Multigraphes

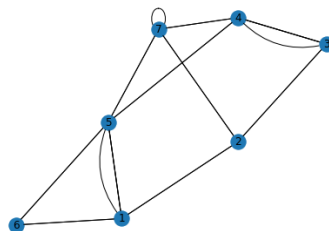
Un multigraphe (figure 2.3c) est un graphe qui peut avoir des boucles et plus d'une arête entre une paire de nœuds [124].



(a) Graphe simple non orienté (7 nœuds et 10 arêtes)



(b) Graphe simple orienté (7 nœuds et 10 arêtes)



(c) Multigraphe (7 nœuds et 13 arêtes)

FIGURE 2.3 – Illustration de quelques types de graphes.

Graphes bipartites

Un graphe bipartite $G = (V, E)$ est un graphe dont les nœuds peuvent être divisés en deux ensembles disjoints (voir figure 2.4), V_1 et V_2 , où chaque arête du graphe relie les nœuds entre V_1 et V_2 . En d'autres termes, $G = (V, E)$ est tel que $V = V_1 \cup V_2$ avec $V_1 \cap V_2 = \emptyset$ et les arêtes $e = \{u, v\} \in E$ sont telles que $u \in V_1, v \in V_2$. Par exemple, le réseau acheteur-produit d'une plateforme commerciale en ligne.

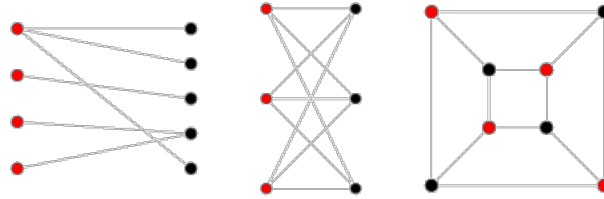


FIGURE 2.4 – Exemples de graphes bipartites (Image extraite de Weisstein [153]).

Graphes complets

Un graphe est dit complet si et seulement si chaque paire de nœuds est reliée par une arête unique. Un graphe complet à n nœuds est appelé une clique K_n et possède $n(n - 1)/2$ arêtes (voir la figure 2.5).

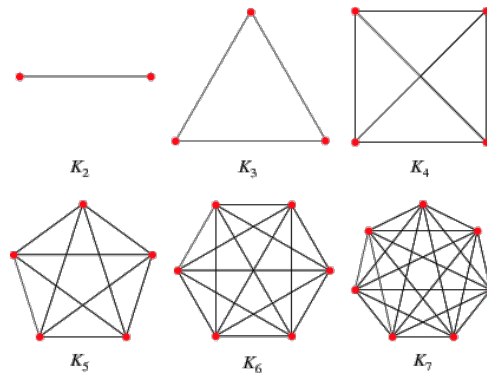


FIGURE 2.5 – Exemples de graphes complets (Image extraite de Weisstein [154]).

Graphes temporels

Les graphes temporels, également appelés graphes dynamiques ou évolutifs [45, 129], peuvent être décrits de manière informelle comme des graphes qui subissent des changements avec le temps, tels que des ajouts ou des suppressions de nœuds et d'arêtes (voir

figure 2.6). Le réseau social Twitter peut être une bonne illustration : d'abord, un nouveau nœud est créé quand un utilisateur rejoint la plateforme. Une arête de suivi est ensuite créée lorsque cet utilisateur en suit un autre. Enfin, le nœud est mis à jour lorsque l'utilisateur met son profil à jour.

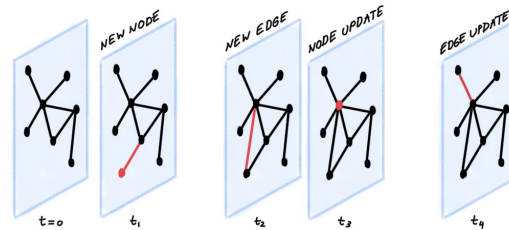


FIGURE 2.6 – Un graphe dynamique évolue dans le temps et son évolution peut être vue comme une séquence d'événements minutés. Dans la figure ci-dessus, les différents événements se produisent entre t_1 et t_4 . (Image extraite de Bronstein et al. [20])

Les graphes que nous étudions dans le cadre de cette thèse ne sont ni orientés ni pondérés, mais ils sont connexes, c'est-à-dire qu'il existe un chemin composé d'une ou plusieurs arêtes entre chaque paire de nœuds.

2.1.3 Représentation d'un graphe

Un graphe peut être représenté de plusieurs manières à des fins différentes. Ainsi, il peut être représenté sous forme d'une liste d'adjacence, d'une liste d'arêtes ou sous forme d'une matrice d'adjacence. Une liste d'adjacence offre un moyen efficace de représenter les graphes creux (*sparse graphs*) tandis qu'une liste d'arêtes est appropriée pour traiter les graphes dynamiques dont les nœuds et les arêtes sont ajoutés ou supprimés régulièrement. La matrice d'adjacence facilite, quant à elle, les calculs algébriques.

Si nous considérons le graphe non orienté de la figure 2.7a (7 nœuds et 10 arêtes), nous pouvons le représenter sous la forme d'une **liste d'adjacence** (voir la figure 2.7b). Ce type de représentation facilite la navigation dans le graphe et est préféré par de nombreux algorithmes de théorie des graphes tels que l'algorithme de Dijkstra [33] pour la recherche du plus court chemin.

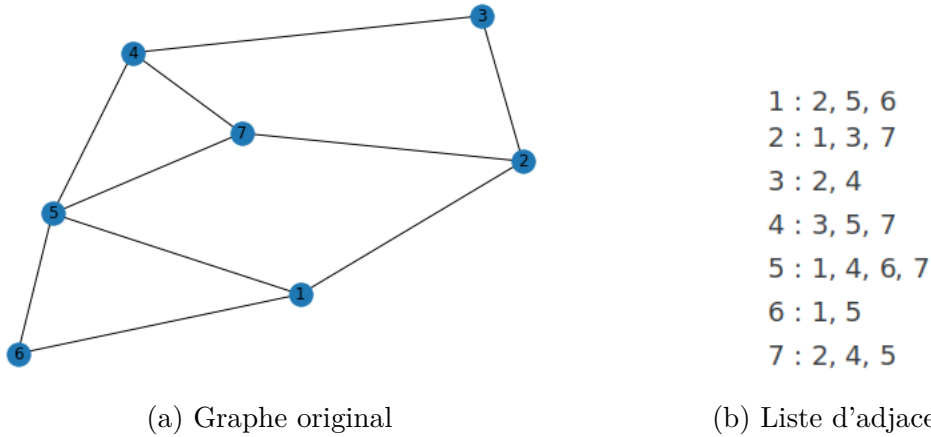


FIGURE 2.7 – Représentation de (a) sous la forme d'une liste d'adjacence (b) où $(k : i, j)$ indique que les nœuds i et j sont adjacents au nœuds k .

Ce graphe peut également être représenté sous la forme d'une **liste d'arêtes** $\{(i, j)\}$: $\{(1, 2), (1, 5), (1, 6), (2, 3), (2, 7), (3, 4), (4, 5), (4, 7), (5, 6), (5, 7)\}$. Les listes d'arêtes et d'adjacence sont souvent utilisées pour stockage informatique, car elles nécessitent beaucoup moins de mémoire ($O(|E|)$) que les matrices d'adjacence ($O(|V|^2)$). Une **matrice d'adjacence** est une matrice carrée A de $n \times n$, n étant le nombre de nœuds du graphe et $A_{ij} = 1$ signifie qu'il existe une arête entre les nœuds i et j . Notre graphe considéré peut également être représenté sous la forme suivante :

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Par ailleurs, les graphes ont des propriétés qui sont très utiles pour caractériser l'information qu'ils contiennent. Il est donc important de comprendre que le but de toute analyse de graphe est de travailler avec la complexité du graphe pour extraire des informations significatives qu'un analyste n'aurait pas si chaque composant était examiné séparément. Ainsi, les propriétés d'un graphe, et en particulier les propriétés topologiques, peuvent aider un analyste à identifier les sous-structures pertinentes au sein du graphe.

Comprendre ces différentes propriétés et leur usage permet de faire des propositions sur la façon de visualiser certaines de ces propriétés. Pour cela, nous présentons d'abord les propriétés d'un graphe dans la section 2.1.4 puis celles des nœuds dans la section 2.1.5. Nous présentons ensuite dans la section 2.1.6 quelques rôles déterminants d'un nœud dans un graphe.

2.1.4 Propriétés d'un graphe

L'analyse statistique d'un graphe commence par l'exploration des données dont le but est par exemple de :

- comprendre et décrire la topologie du graphe ;
- caractériser les propriétés des nœuds ;
- comparer les nœuds en se basant sur des mesures caractérisant leur position dans le graphe ;
- comparer plusieurs graphes les uns avec les autres.

Les propriétés des graphes sont essentiellement utilisées pour caractériser les graphes en fonction de leurs structures. Dans cette section, nous allons présenter certaines propriétés de base qui sont communes à tous les graphes [93, 108], telles que la densité et la distribution de degrés.

Densité

La densité d'un graphe $G = (V, E)$, comprise entre 0 et 1, mesure le nombre d'arêtes existantes par rapport au nombre d'arêtes possibles et elle est définie par :

$$den(G) = \frac{2|E|}{|V|(|V| - 1)} \quad (2.1)$$

où $|E|$ est le nombre d'arêtes et $|V|$ le nombre de nœuds dans le graphe. Plus la valeur de la densité est proche de 1, plus le graphe ressemble à un graphe complet (figure 2.5) où chaque nœud est connecté à tous les autres nœuds.

Distribution de degrés

La distribution de degrés $P(k)$ est la fraction des nœuds dans un graphe ayant un degré k (voir la figure 2.8). Elle est définie comme suit :

$$P(k) = \frac{n_k}{|V|} \quad (2.2)$$

où n_k est le nombre de nœuds de degré k et $|V|$ le nombre de nœuds dans le graphe.

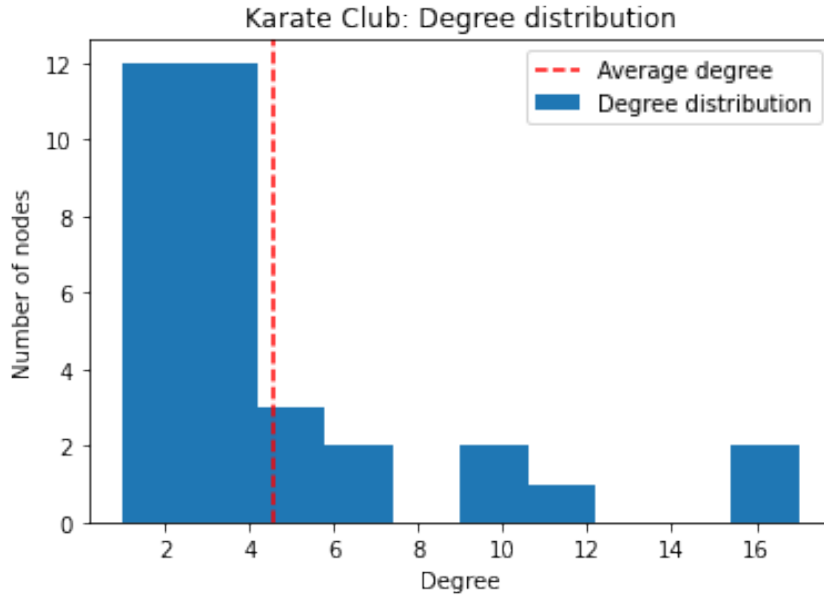


FIGURE 2.8 – Distribution de degrés du graphe « Karate Club » (34 nœuds et 78 arêtes) [160]. La majorité des membres du club n'ont pas beaucoup de connexions (la plupart ont 2 ou 3 connexions).

La distribution de degrés permet de savoir si les nœuds sont connectés de manière homogène ou hétérogène et, quantitativement, comment les arêtes s'étendent sur le graphe. Il a été montré que pour de nombreux graphes du monde réel (« *scale-free networks* »), la distribution de degrés suit une loi de puissance [2] (voir l'équation 2.3), c'est-à-dire une distribution inégalitaire de degrés. Le point important concernant ces graphes est la probabilité accrue de nœuds avec un grand degré. Ces nœuds représentent des « hubs » qui ont un impact significatif sur la connectivité globale du graphe, puisqu'un ensemble connecté de hubs peut fournir des courts chemins entre les nœuds d'une grande partie du graphe. Nous détaillerons la notion de hub dans la section 2.1.6.

$$P(k) \propto k^{-\alpha} \quad (2.3)$$

où k représente le degré d'un nœud dans le graphe et α est le coefficient de la loi de puissance (avec $2 \leq \alpha \leq 3$ [116])

Par ailleurs, en analyse des réseaux sociaux, la distribution de degrés est utile pour analyser et comprendre la structure et la formation des réseaux sociaux. Elle peut éga-

lement aider à la conception d'algorithmes de graphes. En particulier, la distribution de degrés en loi de puissance a été longtemps utilisée pour modéliser la structure des réseaux sociaux en ligne [155], et constitue la base des algorithmes et des heuristiques dans les applications de graphes telles que la maximisation de l'influence [27, 146] et la recherche sociale [95, 122].

Diamètre

Le diamètre d'un graphe est la plus grande distance entre deux nœuds i et j du graphe. Il est défini par :

$$\text{diam}(G) = \max\{d_{i,j}; i, j \in V\} \quad (2.4)$$

Un petit diamètre indique que le graphe est compact. Par exemple, lorsqu'on étudie le flux de l'information dans un graphe, un petit diamètre indique que l'information peut circuler rapidement dans le graphe.

Connexité

Un graphe $G = (V, E)$ est dit **connexe** s'il possède une composante connexe unique, c'est-à-dire pour tout $i, j \in V$, il existe un chemin de i à j dans G . Notons qu'une **composante connexe** de G est un sous-ensemble $H = \{v_1, \dots, v_k\} \subset V$ tel que pour tout $v_i, v_j \in H$, il existe un chemin dans G de v_i à v_j .

Distance

La distance $d_{i,j}$ entre deux nœuds i et j est la longueur de leur plus court chemin. La figure 2.9 illustre un exemple du plus court chemin entre les nœuds **A** et **B**.

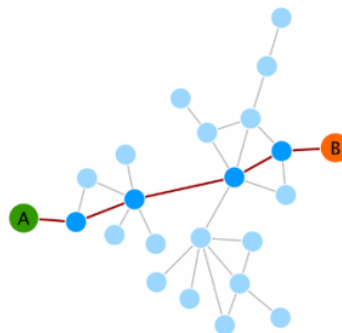


FIGURE 2.9 – Exemple du plus court chemin entre les nœuds A et B d'un graphe.

Coefficient de clustering

Il existe différentes définitions du coefficient de clustering. Le *coefficient de clustering global* ou la *transitivité*, introduit par Barrat et Weigt [3] et généralement compris entre 0 et 1, donne une indication générale du regroupement dans le graphe. Il est basé sur des triplets de nœuds connectés. Un tel triplet est un ensemble de 3 nœuds tel que le sous-graphe induit par ces 3 nœuds est connexe. Le coefficient de clustering global CCF_g d'un graphe G est donc défini comme suit :

$$CCF_g = 3 \times \frac{\text{nombre de triangles dans le graphe}}{\text{nombre de triplets de nœuds connectés dans le graphe}} \quad (2.5)$$

où un triangle est un sous-graphe complet à 3 nœuds de G .

Une autre alternative de la définition du coefficient de clustering global est la moyenne pondérée du coefficient de clustering local qui sera présenté dans la section suivante (section 2.1.5). Le coefficient de clustering montre que, lorsque deux personnes se connaissent, il y a une forte probabilité que ces personnes aient des amis communs. Des faits suggèrent que dans la plupart des graphes du monde réel, notamment les réseaux sociaux, les nœuds ont tendance à créer des groupes très soudés caractérisés par une densité relativement élevée [72, 152]. Le coefficient de clustering a de nombreuses applications importantes, telles que la détection de spam [11], la recommandation de liens [26, 135], les performances de certains réseaux de neurones [85], etc.

2.1.5 Propriétés d'un nœud

Nous présentons, dans cette section, quelques propriétés les plus utilisées telles que le degré, le coefficient de clustering (local) et les mesures de centralité.

Degré

Dans un graphe $G = (V, E)$, le degré deg_i d'un nœud $i \in V$ est le nombre de voisins de i dans G (voir la figure 2.10). Les voisins de i sont les nœuds $j \in V$ tels que $\{i, j\} \in E$.

L'étude du degré dans un graphe peut être importante pour plusieurs raisons. D'une part, dans un réseau social, les individus qui ont des liens avec beaucoup d'autres peuvent avoir plus d'influence et plus d'accès à l'information que ceux qui ont moins de liens. D'autre part, le degré peut être un risque imminent pour un nœud d'attraper ce qui se propage dans le graphe (exemple : un virus ou une information).

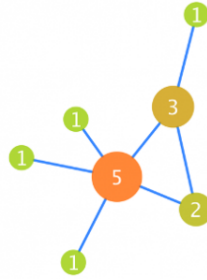


FIGURE 2.10 – Illustration du degré : le degré de chaque nœud est écrit sur le nœud.

Coefficient de clustering local

Le coefficient de clustering local, introduit par Watts et Strogatz [152], indique à quel point les voisins d'un nœud i sont connectés entre eux. Il permet d'analyser l'environnement local d'un nœud en incluant les interactions de ses voisins. Ainsi, si les voisins du nœud i sont tous connectés, alors il a un fort coefficient de clustering. Le coefficient de clustering d'un nœud i est défini comme suit :

$$CCF_i = \frac{\text{nombre de triangles connectés au nœud } i}{\text{nombre de triplets centrés sur } i} \quad (2.6)$$

où un triplet centré sur i est un ensemble de deux arêtes connectées au nœud i . Tout comme le coefficient de clustering global, on a $0 \leq CCF_i \leq 1$.

Mesures de centralité

Les mesures de centralité caractérisent la position topologique des nœuds dans un graphe. Elles révèlent les parties du graphe qui nécessitent l'attention d'un analyste. Il existe de nombreuses mesures de centralité et chacune fonctionne différemment. Il est donc utile de comprendre leur fonctionnement pour trouver une bonne manière de visualiser chaque mesure. Parmi les mesures de centralité les plus couramment utilisées, nous avons la *centralité de degré*, la *centralité d'intermédiation*, la *centralité de proximité* et la *centralité de vecteur propre*.

- **Centralité de degré** (Degree centrality)

La centralité de degré [48] quantifie le nombre de voisins d'un nœud dans un graphe. Intuitivement, un nœud très connecté aux autres est très important dans le graphe (voir

la figure 2.11a). La centralité de degré d'un nœud i est définie par :

$$C_D(i) = \frac{deg_i}{|V| - 1} \quad (2.7)$$

où deg_i et $|V|$ sont respectivement le degré (nombre de voisins) du nœud i et le nombre de nœuds dans le graphe.

Dans l'analyse des réseaux sociaux, par exemple, un analyste peut utiliser la centralité de degré lorsqu'il s'intéresse aux individus très connectés, aux individus susceptibles de détenir le plus d'informations ou aux individus qui peuvent rapidement se connecter à un réseau plus large.

- **Centralité d'intermédiarité** (Betweenness centrality)

La centralité d'intermédiarité [48] mesure la fréquence à laquelle un nœud i est un point de passage sur les plus courts chemins entre deux autres nœuds du graphe (voir la figure 2.11b). La centralité d'intermédiarité du nœud i est définie comme suit :

$$C_I(i) = \sum_{(j \neq i \neq k) \in V} \frac{d_{j,k}^i}{d_{j,k}} \quad (2.8)$$

où $d_{j,k}^i$ est le nombre de plus courts chemins de j à k passant par i et $d_{j,k}$ le nombre total de plus courts chemins de j à k .

Freeman [48] souligne que l'équation 2.8 est dépendante de la taille du graphe et par conséquent, cela n'est pas pertinent pour certaines applications. Il propose alors une définition relative qui est souvent utilisée pour comparer la centralité des nœuds de différents graphes :

$$C'_I(i) = \frac{2 \times C_I(i)}{|V|^2 - 3|V| + 2} \quad (2.9)$$

où $|V|$ est le nombre de nœuds dans le graphe. La centralité d'intermédiarité d'un nœud i est élevée s'il est un point de passage sur un grand nombre de chemins les plus courts entre deux autres nœuds.

La centralité d'intermédiarité peut s'appliquer à de nombreux problèmes liés à la propagation de l'information dans les réseaux sociaux [86, 105] ou à la détection de communautés dans les réseaux sociaux [115, 43]. Dans un réseau d'échange d'informations, par exemple, le nœud qui a la plus grande valeur d'intermédiarité dispose d'une plus grande influence sur le transfert d'informations. Elle peut également être utilisée pour prédire la force des liens dans les réseaux égocentrés [140]. Notons qu'un réseau égocentré se compose

d'un nœud focal (« égo ») et des nœuds auxquels l'égo est directement connecté (ceux-ci sont appelés « alters ») plus les connexions entre les alters.

- **Centralité de proximité** (Closeness centrality)

La centralité de proximité [48] d'un nœud i est l'inverse de la somme des distances à tous les autres nœuds du graphe (voir la figure 2.11c). Dans un réseau social, par exemple, cette mesure peut exprimer la notion d'effort, c'est-à-dire qu'un individu ayant une forte valeur de proximité peut atteindre la plupart des individus du réseau avec un minimum d'effort. La centralité de proximité est définie comme suit :

$$C_P(i) = \frac{|V| - 1}{\sum_{(i \neq j) \in V} d_{i,j}} \quad (2.10)$$

où $|V|$ et $d_{i,j}$ sont respectivement le nombre de nœuds dans un graphe et la distance entre les nœuds i et j . La centralité de proximité d'un nœud i est élevée quand sa distance à tous les autres nœuds est faible.

- **Centralité de vecteur propre** (Eigenvector centrality)

La centralité de vecteur propre ou centralité spectrale [9] est une extension de base de la centralité de degré, qui définit la centralité d'un nœud i comme étant proportionnelle à celle de ses voisins (voir la figure 2.11d). Les centralités de **PageRank** [17] et de **Katz** [84] sont des variantes de cette mesure. Le PageRank classique, défini par Google, a été conçu pour classer les pages web dans « Google recherche ». De nos jours, Le PageRank et ses extensions sont des mesures populaires pour l'analyse des réseaux sociaux [157, 119].

La centralité de vecteur propre x_i d'un nœud i est définie par :

$$x_i = \frac{1}{\lambda} \sum_{j \in V} A_{ij} x_j \quad (2.11)$$

où $\lambda \neq 0$ est une constante et A_{ij} un élément de la matrice d'adjacence d'un graphe $G = (V, E)$, c'est-à-dire $A_{ij} = \begin{cases} 1 & \text{s'il existe une arête entre deux nœuds } i \text{ et } j \\ 0 & \text{sinon} \end{cases}$

Ainsi, un nœud est important s'il est connecté à d'autres nœuds importants [68]. En considérant par exemple deux nœuds qui ont un même degré dans un graphe d'amis, celui qui est connecté à plus de nœuds centraux devrait être plus central. Cette situation peut être illustrée avec la figure 2.11d où les nœuds ayant le même nombre de voisins n'ont pas nécessairement la même couleur dans la carte thermique.

Cependant, cette mesure peut être problématique pour les graphes orientés (voir la

figure 2.12). Considérons par exemple qu'un nouvel article de recherche est publié et qu'il fait référence à une poignée d'articles existants. Il ne contribuerait à aucun de ces articles référencés dans un graphe de citations, car il n'est cité par aucun autre article et n'a aucune centralité de vecteur propre. En d'autres termes, la centralité de vecteur propre ne prendrait pas en compte les nœuds qui ont un degré entrant égal à zéro dans les graphes orientés tels que les réseaux de citations.

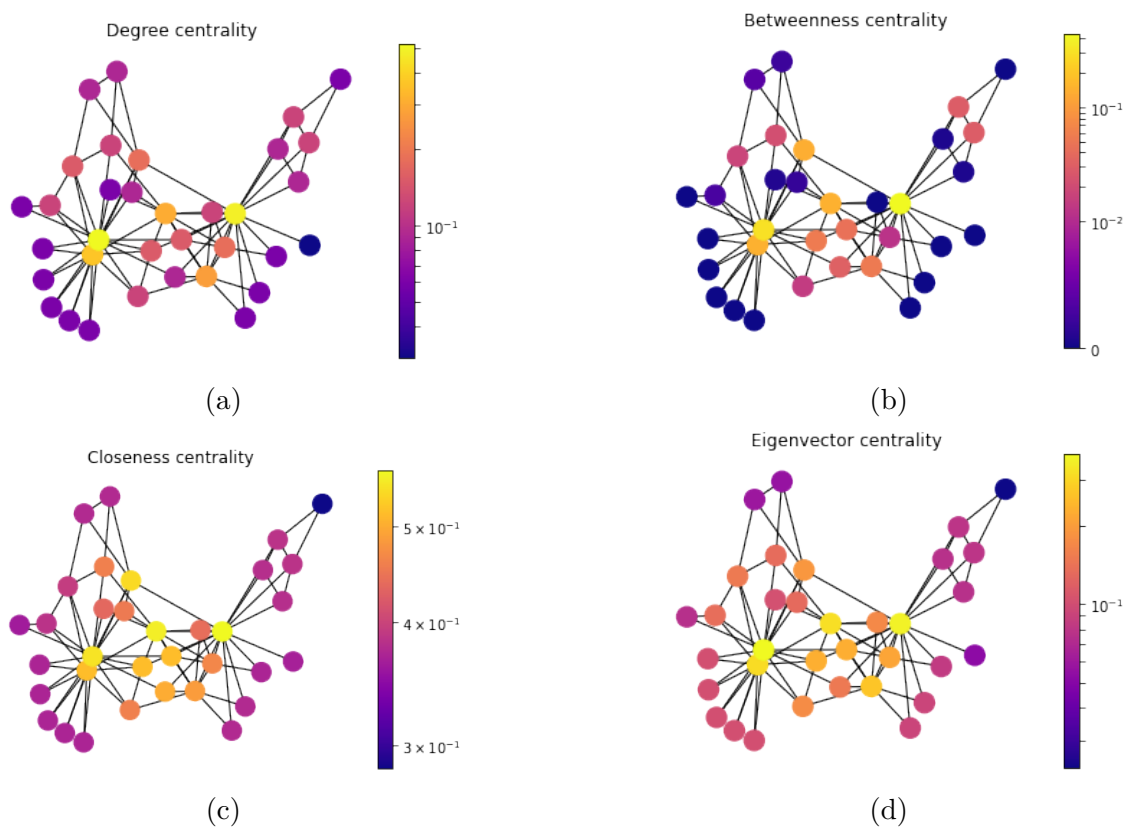


FIGURE 2.11 – Illustration de quelques mesures de centralité par une carte thermique avec le graphe « Karate Club » (34 nœuds et 78 arêtes) [160]. Les couleurs vont du bleu (moins central) au jaune (plus central). (a) : la centralité de degré, (b) : la centralité d'intermédiation, (c) : la centralité de proximité et (d) : la centralité de vecteur propre.

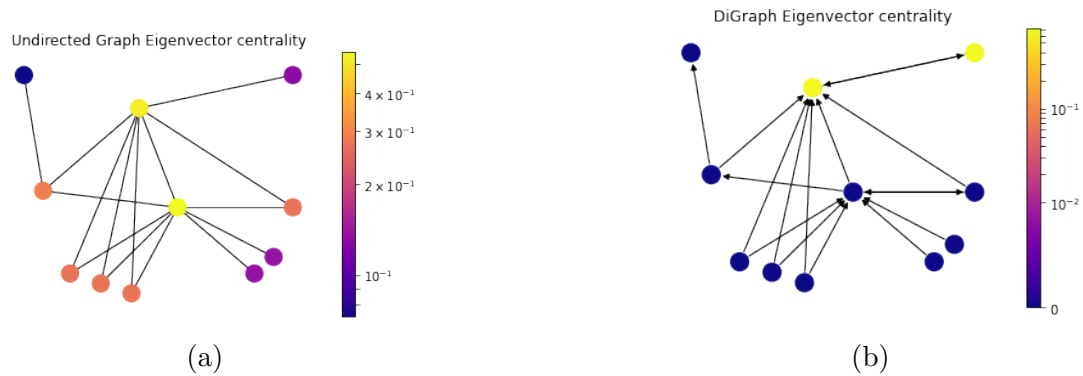


FIGURE 2.12 – Comparaison de la centralité de vecteur propre pour deux types de graphe. Les couleurs vont du bleu (moins central) au jaune (plus central). (a) : graphe non orienté, (b) : graphe orienté.

Les propriétés d'un graphe et des nœuds présentées jusqu'alors permettent d'avoir une idée de la structure générale d'un graphe : existence des zones denses, identification des nœuds centraux, etc. Certaines propriétés, telles que la centralité d'intermédiarité, permettent de définir ou d'assigner certains rôles à certains nœuds dans le graphe, alors que d'autres rôles reposent sur un découpage du graphe en clusters, en appliquant un algorithme de détection de clusters, notion qui sera présentée dans la section 2.1.7. Rappelons qu'un cluster est un groupe de nœuds en interaction partageant souvent des valeurs ou comportements communs. Du point de vue des travaux de cette thèse, il est utile de comprendre l'usage de certains rôles des nœuds afin de proposer une manière de les visualiser efficacement dans un graphe partitionné. Dans la section suivante, nous présentons certains rôles des nœuds qui reposent sur le partitionnement d'un graphe en clusters.

2.1.6 Rôles d'un nœud dans un graphe

La plupart des graphes actuels comportent plusieurs milliers de nœuds et ils sont complexes à analyser et à visualiser. L'une des méthodes fréquemment utilisées pour les analyser consiste à trouver des groupes de nœuds à partir d'un algorithme de détection de clusters. Ainsi, un analyste de graphes peut facilement se focaliser sur un cluster donné afin de mieux explorer les interactions entre ses nœuds. Le niveau d'interaction entre les nœuds d'un cluster par rapport aux autres peut conduire à l'apparition de plusieurs rôles qui peuvent être caractérisés comme des positions dans le graphe.

Ainsi, Guimera et Amaral. [62] proposent de définir sept rôles à partir de deux métriques : le *z-score* et le *coefficient de participation*. Le *z-score* mesure à quel point un

noeud est connecté aux autres noeuds du cluster, alors que le coefficient de participation P quantifie la connectivité de ce noeud par rapport aux autres clusters.

Les noeuds ayant un z-score $\geq 2,5$ sont considérés comme des « hubs » tandis que ceux qui ont un z-score $< 2,5$ sont considérés comme des « non-hubs ». Notons qu'un hub est un noeud dont le degré est particulièrement grand et qui est caractérisé par une connectivité stratégique facilitant l'intégration du graphe [23]. Il est connecté à la plupart des noeuds dans le graphe et par conséquent, en le supprimant, le graphe peut être composé de plusieurs composantes connexes.

- **Rôle 1** : noeuds ultra-périphériques (*Ultra-peripheral nodes*). Les noeuds ultra-périphériques sont des noeuds qui n'ont aucune connexion avec les clusters voisins ($P \approx 0$).
- **Rôle 2** : noeuds périphériques (*Peripheral nodes*). Un noeud périphérique est un noeud ayant au moins 60% de voisins dans son cluster ($P < 0,62$).
- **Rôle 3** : connecteurs non-hub (*Non-hub connectors*). Un connecteur non-hub est un noeud qui a plusieurs connexions vers d'autres clusters ($0,62 < P < 0,8$). Notons qu'un connecteur est un noeud qui connecte son cluster aux clusters voisins.
- **Rôle 4** : noeuds « Non-hub kinless » (*Non-hub kinless nodes*). Il s'agit des noeuds dont les arêtes sont réparties de manière homogène entre tous les clusters ($P > 0,8$).
- **Rôle 5** : hubs provinciaux (*Provincial hubs*). Un hub provincial est un hub avec la majorité des voisins dans son cluster ($P \leq 0,3$).
- **Rôle 6** : hubs connecteurs (*Connector hubs*). Un hub connecteur est un hub ayant de nombreuses connexions avec la plupart des autres clusters ($P < 0,75$).
- **Rôle 7** : « kinless hubs ». Il s'agit des hubs dont les arêtes sont réparties de manière homogène entre tous les clusters ($P > 0,75$).

En conclusion, l'identification des rôles des noeuds dans un graphe est utile, car cela peut fournir une compréhension plus approfondie du graphe. En marketing, par exemple, trouver les personnes qui feront en sorte que l'information sur un nouveau produit parvienne au plus grand nombre d'individus peut entraîner une augmentation des ventes.

Dans le cadre des travaux de cette thèse, par ailleurs, nous aurons à proposer une façon de mettre en évidence certains rôles des noeuds dans un graphe partitionné afin d'améliorer l'exploration et la compréhension des connexions dans un cluster et entre certains clusters. Pour cela, il est utile de comprendre le fonctionnement des algorithmes de détection de clusters, que nous présentons dans la section suivante.

2.1.7 Détection de clusters

L'analyse des graphes comportant plusieurs milliers de nœuds et d'arêtes passe souvent par un résumé de l'information. Par exemple, à travers un partitionnement des nœuds d'un graphe, on va chercher à regrouper les nœuds en classes homogènes, c'est-à-dire les nœuds dans une même classe se comportent de façon similaire au sein du graphe. Une classe est appelée un *cluster* ou une *communauté* (dans le domaine de l'analyse des réseaux sociaux).

Les techniques de détection de clusters sont utiles aux algorithmes des réseaux sociaux pour découvrir des personnes qui ont des intérêts communs et les garder étroitement connectées. Ces techniques peuvent être globalement classées en deux types d'approches : l'approche *ascendante* ou *agglomérative* et l'approche *descendante* ou *divisive*.

Les algorithmes appliquant une approche agglomérative prennent un graphe en entrée sans arêtes [57]. Les arêtes sont ensuite ajoutées une par une au graphe, les arêtes de poids fort étant prioritaires par rapport aux arêtes de poids faible, jusqu'à ce que tous les nœuds appartiennent au même cluster.

Les algorithmes appliquant une approche divisive reposent quant à eux sur un processus de suppression d'arêtes du graphe de manière itérative, les arêtes de poids fort étant supprimées avant celles de poids faible [57] jusqu'à ce que chaque nœud soit placé dans un cluster unique. À chaque itération, le calcul du poids des arêtes est répété, car le poids des arêtes restantes change après la suppression d'une arête.

Il existe de nombreux algorithmes de détection de clusters et certains déterminent automatiquement le nombre de clusters, contrairement à d'autres où ce nombre est un paramètre en entrée. Dans cette section, nous présentons trois algorithmes parmi les plus populaires. Plusieurs algorithmes de détection de clusters ont été caractérisés et présentés dans les travaux de Dao [31] afin de fournir une recommandation pour le choix des algorithmes appropriés de détection de clusters. Dans le cadre de cette thèse, il n'est donc pas envisageable de changer le choix d'un algorithme fait par un expert, mais plutôt de lui permettre d'explorer les résultats de différents algorithmes et éventuellement lui permettre de comparer visuellement les différentes approches de détection de clusters.

2.1.7.1 Algorithme de Louvain

L'algorithme de Louvain, appliquant une approche agglomérative, a été proposé comme une méthode rapide d'identification de clusters pour les grands graphes [8]. Cette ap-

proche est basée sur la modularité, qui tente de maximiser la différence entre le nombre réel d'arêtes dans le cluster et le nombre attendu d'arêtes dans le cluster. Cependant, l'optimisation de la modularité d'un graphe est une tâche complexe à réaliser [114], d'où la nécessité d'utiliser des heuristiques. L'algorithme de Louvain est divisé en deux phases qui se répètent de façon itérative (voir figure 2.13) :

1. déplacement local des nœuds ;
2. agrégation du graphe.

Il commence avec un graphe pondéré de n nœuds. Dans la première étape, l'algorithme attribue un cluster différent à chaque nœud du graphe. Ensuite, pour chaque nœud, on considère les voisins et on évalue le gain de modularité en retirant le nœud particulier du cluster actuel et en le plaçant dans le cluster du voisin. Le nœud sera placé dans le cluster du voisin si le gain est positif et maximisé. Dans le cas contraire, il restera dans le même cluster. Ce processus est appliqué de manière répétée et pour tous les nœuds jusqu'à ce qu'il n'y ait plus d'amélioration.

Dans la deuxième étape, l'algorithme construit un nouveau graphe en considérant les clusters trouvés dans la première étape comme des nœuds. Une fois cette étape terminée, l'algorithme réapplique la première étape au graphe résultant. Ces étapes sont répétées jusqu'à ce qu'il n'y ait plus de changement dans le graphe et que la modularité maximale soit obtenue.

En définitive, l'algorithme de Louvain est très populaire en raison de la facilité de mise en œuvre et aussi de sa complexité de $O(n \log n)$ en temps d'exécution (n étant le nombre de nœuds dans un graphe).

2.1.7.2 Algorithme de Leiden

Récemment, Traag et al. [144] ont montré que l'algorithme de Louvain a tendance à détecter des clusters mal connectés en interne (voir la figure 2.14). En fait, dans l'algorithme de Louvain, le déplacement d'un nœud qui fait office de connecteur entre deux composants d'un cluster vers un nouveau cluster peut déconnecter l'ancien cluster. Toujours selon eux, l'algorithme de Louvain a tendance à détecter des clusters très faiblement connectés. Ils proposent alors l'algorithme de *Leiden*, qui applique également une approche agglomérative et qui garantit que les clusters sont bien connectés.

En plus des étapes utilisées dans l'algorithme de Louvain, l'algorithme de Leiden utilise une étape supplémentaire qui tente d'affiner les partitions détectées :

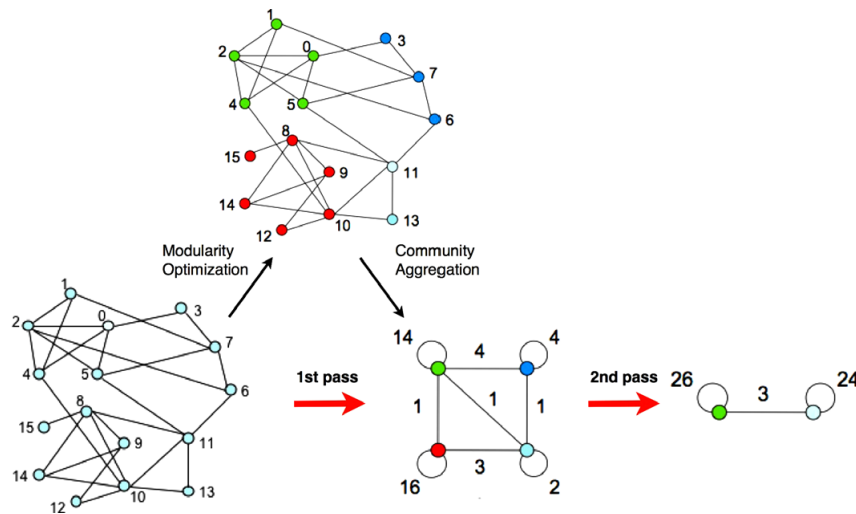


FIGURE 2.13 – Illustration de l’algorithme de Louvain. Chaque passe est constituée de deux phases : la première où la modularité est optimisée en ne permettant que des changements locaux de communautés ; la seconde où les communautés trouvées sont agrégées afin de construire un nouveau réseau de communautés. Les passes sont répétées jusqu’à ce qu’aucune augmentation de la modularité ne soit possible (Image extraite de Blondel et al. [8]).

1. déplacement local des nœuds ;
2. raffinement des partitions ;
3. agrégation du graphe basée sur les partitions raffinées.

Dans la première étape, l’algorithme de Leiden suit une approche différente de celle de l’algorithme de Louvain. En effet, au lieu de revisiter tous les nœuds du graphe après la première visite de tous les nœuds, l’algorithme de Leiden ne revisite que les nœuds dont le voisinage a changé.

Ensuite, dans l’étape de raffinement, l’algorithme tente d’identifier des partitions raffinées à partir des partitions proposées par la première étape. Les clusters proposés par la première étape peuvent ensuite être divisés en plusieurs partitions dans la deuxième étape. L’étape de raffinement ne suit pas une approche averse et peut fusionner un nœud avec un cluster choisi au hasard, ce qui augmente la fonction de qualité. Ce caractère aléatoire permet de détecter l’espace de partition de manière plus large.

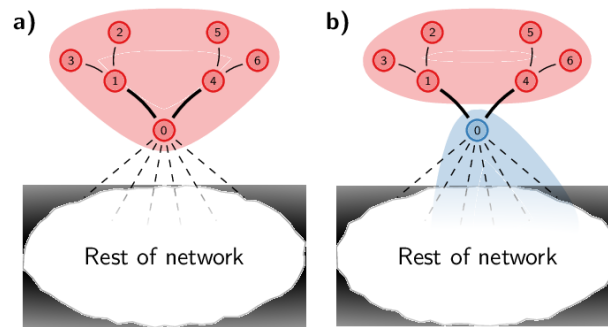


FIGURE 2.14 – Louvain : cluster déconnecté. Considérons la partition illustrée en (a). Lorsque le nœud 0 est déplacé vers un cluster différent, le cluster rouge devient déconnecté en interne, comme illustré en (b). Cependant, les nœuds 1 à 6 sont toujours affectés de manière optimale localement, et donc ces nœuds resteront dans le cluster rouge (Image extraite de Traag et al. [144], licence CC BY 4.0).

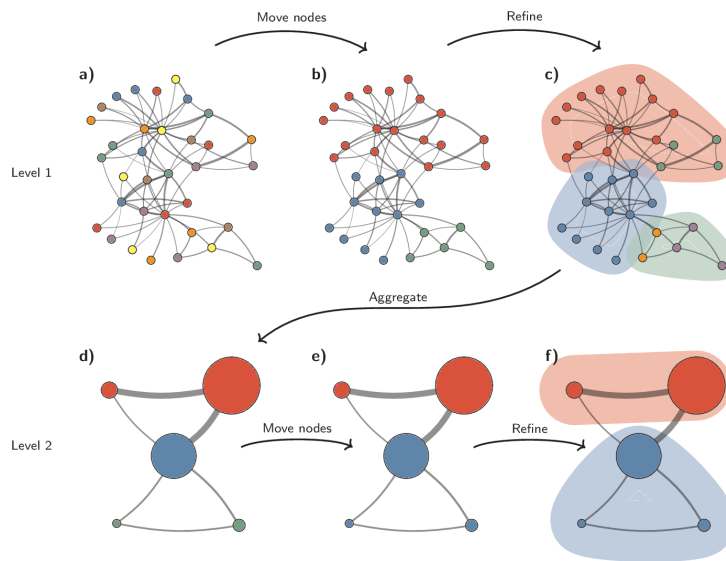


FIGURE 2.15 – Algorithme de Leiden. L'algorithme commence par un singleton (a), (b) déplacement des nœuds individuels d'un cluster à l'autre, (c) raffinement, (d) agrégation sur la base de la partition raffinée, (e) déplacement des nœuds individuels dans le graphe agrégé, (f) dans ce cas, le raffinement ne modifie pas la partition. Ces étapes sont répétées jusqu'à ce qu'aucune autre amélioration ne puisse être apportée (Image extraite de Traag et al. [144], licence CC BY 4.0).

2.1.7.3 Algorithme de Walktrap

Walktrap est un autre algorithme de détection de clusters, appliquant également une approche agglomérative, dans laquelle la distance entre les nœuds est mesurée par des marches aléatoires dans le graphe. [121] C'est un algorithme efficace qui fonctionne en $O(mn^2)$ de complexité en temps et en $O(n^2)$ de complexité spatiale dans le pire des cas. Mais dans la plupart des scénarios du monde réel, Walktrap fonctionne en $O(n^2 \log n)$ de complexité en temps et en $O(n^2)$ de complexité spatiale.

L'intuition de base est que les marches aléatoires sur un graphe ont tendance à être piégées dans des zones densément connectées correspondant donc à des clusters. Par ailleurs, l'algorithme utilise le résultat des marches aléatoires d'une manière ascendante pour fusionner des clusters séparés.

La qualité des partitions peut être évaluée en utilisant n'importe quel critère de qualité disponible [31]. Il peut s'agir, par exemple, de la modularité comme dans l'algorithme de Louvain ou de toute autre mesure.

Détection de clusters

La détection de clusters est très utile pour comprendre et évaluer la structure des graphes comportant plusieurs milliers de nœuds et d'arêtes. Cette approche exploite les propriétés des arêtes dans un graphe afin de proposer un découpage particulier du graphe.

De nos jours, de nombreux algorithmes de détection de clusters ont été proposés et chacun d'entre eux présente des avantages et des inconvénients qui dépendent de la nature du graphe ainsi que du domaine d'application : selon les algorithmes choisis les clusters et le découpage ont des structures différentes qui peuvent être plus ou moins adaptées selon le domaine métier ou ce que l'on cherche à mettre en évidence lors de la l'analyse ; certains algorithmes déterminent automatiquement le nombre de clusters au lieu de l'exiger comme paramètre.

Dans le cadre des travaux de cette thèse, nous évaluerons l'efficacité de nos propositions de visualisation par rapport à certaines visualisations classiques avec des graphes ayant des caractéristiques topologiques équivalentes et une structure de clusters. L'existence

d'une structure de clusters pourrait permettre de définir des tâches liées aux zones denses dans un graphe. Toutefois, il est difficile de trouver de tels graphes dans les bases de données en ligne. Nous générerons donc des graphes avec des caractéristiques topologiques équivalentes à partir d'un modèle artificiel. Nous présentons dans la section suivante quelques modèles de génération de graphes parmi les plus populaires.

2.1.8 Modèles de génération de graphes artificiels

Un modèle de génération de graphes artificiels est une collection de graphes et une loi de probabilité sur cette collection. Les modèles de génération de graphes artificiels (ou aléatoires) permettent de modéliser des distributions complexes et de générer des échantillons de données convaincants qui peuvent être à peine différenciés des vraies données. En raison de ses nombreuses applications [61, 145], le développement de modèles génératifs de graphes a permis d'aboutir à des modèles célèbres tels que les graphes aléatoires (*random graphs*), le modèle de blocs stochastiques, qui génèrent des graphes sur la base d'hypothèses structurelles. Dans cette section, nous présentons le modèle d'Erdős-Rényi et le modèle de blocs stochastiques, deux modèles de graphes artificiels parmi les plus populaires.

2.1.8.1 Modèle d'Erdős-Rényi

Le modèle d'Erdős et Rényi [40] est l'une des méthodes de génération de graphes les plus populaires. L'idée principale de ce modèle est de fixer un seuil de probabilité uniforme pour une arête entre deux nœuds ($p \in (0, 1)$). Il existe deux variantes de ce modèle, mais la principale peut être définie comme suit [56] :

$$P(A_{ij} = 1) = r, \forall i, j \in V, i \neq j \quad (2.12)$$

où A_{ij} est la matrice d'adjacence d'un graphe et r la probabilité d'une arête entre deux nœuds arbitraires i et j . Plus r est élevée, plus le graphe sera densément connecté (voir la figure 2.16).

Bien que le modèle d'Erdős et Rényi soit simple, il ne génère que des graphes homogènes alors que la plupart des graphes du monde réel sont assez hétérogènes. En effet, ce modèle ne permet pas de générer des graphes plus complexes, c'est-à-dire des graphes contenant une structure de clusters ou une distribution de degrés hétérogène (une distribution de degré qui suit une loi de puissance).

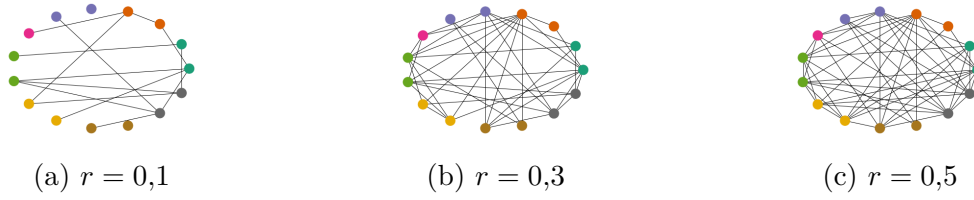


FIGURE 2.16 – Exemple de graphes générés avec le modèle d’Erdős et Rényi en utilisant des valeurs de probabilité différentes. Plus r est élevée, plus le graphe est densément connecté.

2.1.8.2 Modèle de blocs stochastiques (SBM)

Le modèle de blocs stochastiques est un modèle probabiliste basé sur la structure de clusters dans les graphes (voir la figure 2.17). Ce modèle partitionne les nœuds en blocs de tailles arbitraires et place les arêtes entre les paires de nœuds de manière indépendante, avec une probabilité qui dépend des blocs [136, 128]. Il présente de multiples intérêts :

- la forte capacité à modéliser des graphes hétérogènes avec des hubs et des clusters. Le SBM s’ajuste correctement à un grand nombre de graphes issus d’applications [104].
- l’interprétabilité des paramètres du modèle est un grand avantage pour les applications.
- la procédure d’estimation fournit automatiquement un clustering des nœuds, ce qui augmente l’interprétabilité du modèle.

Le grand inconvénient du SBM est que l’estimation des paramètres est difficile, car la structure de dépendance du modèle est complexe.

L’idée du SBM est que chaque nœud a un profil de connexion décrit par sa variable latente. Les variables latentes permettent de modéliser des structures sous-jacentes et la dépendance entre les observations [10]. Par conséquent, on peut obtenir des graphes hétérogènes, car la probabilité d’une arête entre deux nœuds dépend des variables latentes associées à ces nœuds.

Ainsi, le SBM peut être défini par :

1. chaque nœud i d’un $G = (V, E)$ appartient à un certain cluster C_x , où $x \in (0, \dots, \gamma)$, γ étant le nombre total de clusters dans le graphe ;
2. pour chaque nœud $i \in C_x$ et $j \in C_y$, il existe une probabilité pour une arête telle que :

$$P(A_{ij} = 1) = C_{x,y} \tag{2.13}$$

où A_{ij} est la matrice d'adjacence du graphe et $C(\gamma \times \gamma)$ est la matrice de probabilité pour tous les clusters du graphe.

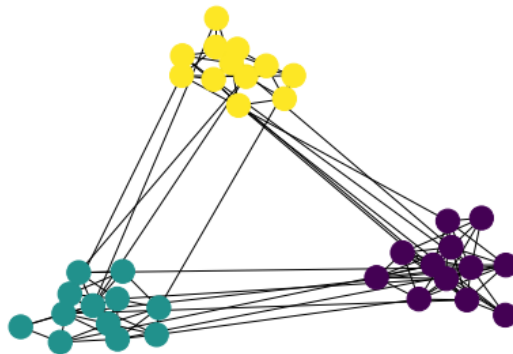


FIGURE 2.17 – Exemple de graphe généré avec le modèle de blocs stochastiques. L'algorithme de Louvain [8] a été appliqué sur le graphe afin mettre en évidence les clusters présents

Modèles de génération de graphes

Le modèle de génération de graphe d'Erdoős-Rényi est simple, mais il ne produit que des graphes homogènes, alors que la majorité des graphes réels sont assez hétérogènes [67]. Le modèle de blocs stochastiques produit, quant à lui, des graphes contenant une structure de clusters et une distribution de degré qui suit une loi de puissance. Il peut donc être plus approprié pour générer des graphes pouvant servir à simuler une application réelle d'analyse de graphes.

2.2 Application à l'analyse des réseaux sociaux

L'analyse des réseaux sociaux consiste à détecter, décrire et explorer des structures et des relations au sein d'un groupe d'individus [112, 118]. Elle utilise le concept de la théorie des graphes pour explorer le comportement des individus au niveau macroscopique (les propriétés du graphe), au niveau mésoscopique (les propriétés partagées par des groupes de nœuds) et au niveau microscopique (les propriétés des nœuds et arêtes).

Dans cette section, nous illustrons un cas d'application de la théorie des graphes à l'analyse des réseaux sociaux. Ce cas découle des travaux d'analyse des données issues d'une plateforme de financement participatif [109]. C'est une plateforme qui réunit des contributeurs de tout genre et des porteurs de projets. Les images et les données utilisées dans cette section nous ont été fournies par les auteurs de ces travaux.

2.2.1 Définition des objectifs d'analyse

Le cas que nous illustrons a principalement réuni deux types d'experts qui ont des objectifs différents : un économiste et un data scientist. Les principaux objectifs de l'économiste sont de déterminer s'il existe une structure communautaire sous-jacente à la plateforme de financement participatif et de caractériser l'influence de ces communautés et de certains individus sur la performance des campagnes de levée de fonds.

L'étude des caractéristiques du réseau consiste d'abord à avoir une vision d'ensemble du réseau et à identifier ensuite, entre autres, des structures sociales présentes. L'économiste cherche d'abord à valider l'hypothèse que la plateforme de financement participatif permet de créer des communautés. Il souhaite ensuite identifier d'une part les profils des contributeurs et le lien entre ces profils, d'autre part. Un profil est défini par la combinaison des caractéristiques socio-économiques (montant dépensé, date d'arrivée sur la plateforme, etc.) avec des propriétés topologiques (les mesures de centralité, par exemple). En ce qui concerne l'impact du réseau du financement participatif sur les campagnes de levée de fonds, l'économiste souhaite vérifier, par exemple, si l'activité sociale d'un profil de contributeur a un impact sur le succès des campagnes.

Le data scientist aide quant à lui à modéliser le graphe permettant de représenter les interactions sur la plateforme de financement participatif, à choisir les propriétés topologiques à considérer lors de l'étude et à conseiller l'économiste sur le choix de l'algorithme de détection de clusters à utiliser.

2.2.2 Données et modélisation

L'analyse est basée sur un ensemble de données anonymisées des cinq premières années du fonctionnement de la plateforme.

Pour modéliser un graphe de co-contributions où les nœuds représentent les contributeurs, l'économiste et le data scientist proposent d'abord de fixer à trois projets communs le seuil minimal permettant d'établir une connexion entre deux contributeurs. Le data

scientist modélise ensuite un graphe non orienté et non pondéré de co-contributions où un nœud représente un contributeur et une arête représente le financement d'un minimum de trois projets communs.

L'analyse du graphe ainsi modélisé est utile pour étudier l'intensité de l'activité d'un contributeur, identifier les contributeurs influents, les communautés et les principaux intermédiaires au sein des communautés. Elle est également utile pour identifier et mettre en évidence les différents profils de contributeurs. La distribution de degrés a permis d'étudier l'intensité des activités des contributeurs. Plus le degré est grand, plus l'activité est intense, c'est-à-dire plus le contributeur a un comportement de financement similaire à d'autres contributeurs, étant donné que chaque arête met en évidence que 2 contributeurs ont financé au moins 3 projets communs. La centralité d'intermédierité est quant à elle utilisée pour identifier les contributeurs influents dans le réseau. L'algorithme de Louvain [8], présenté dans la section 2.1.7.1, a permis de partitionner le graphe en clusters, tandis que le coefficient de partition a permis d'identifier les connecteurs dans les clusters.

2.2.3 Résultats

Analyse topologique

L'analyse du graphe aide à comprendre la nature du réseau social de contributeurs, comme la densité. Elle est également utile pour identifier les influenceurs et les communautés du réseau. Le graphe de co-contributions contient 2 081 nœuds et 4 749 arêtes (voir la figure 2.18). La densité du graphe est égale à 0,002, suggérant donc que le graphe n'est pas complet. Rappelons que la densité d'un graphe mesure le nombre d'arêtes existantes par rapport au nombre d'arêtes possibles. Le diamètre du graphe, c'est-à-dire la plus grande distance entre deux contributeurs du réseau, est égal à 13 et la distance moyenne entre deux nœuds est égale à 3,98. Ces valeurs de diamètre et de la distance moyenne suggèrent la présence de contributeurs influents dans le réseau. L'étude des degrés montre une distribution hétérogène (elle suit une loi de puissance), ce qui suggère que la majorité des contributeurs soit impliquée dans peu de financements de projets communs.

L'analyse topologique du graphe soutient une présence des contributeurs influents. L'analyse de la centralité d'intermédierité confirme encore une telle conclusion (voir la figure 2.19). Le tableau 2.1 fournit une liste des top 5 des contributeurs influents du réseau, avec respectivement les valeurs de la centralité d'intermédierité et du z-score correspondantes.

En outre, l'économiste a émis une hypothèse qu'il existe une structure communautaire

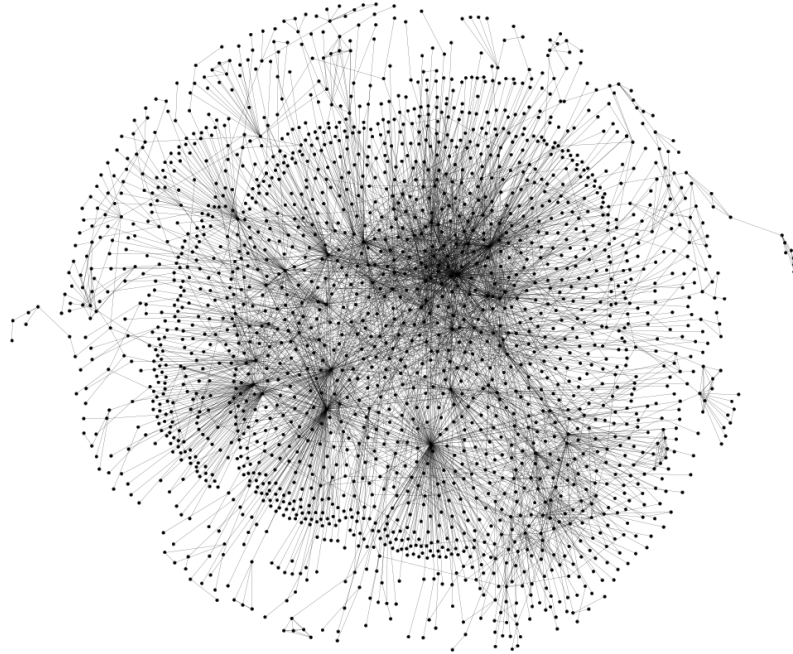


FIGURE 2.18 – Visualisation du graphe de co-contributions d'une plateforme de financement participatif (2 081 nœuds et 4 749 arêtes). Un nœud représente un contributeur et une arête représente le financement d'un minimum de trois projets communs.

Contributeur	Degré	Centralité d'intermédiarité	z-score
34 481	199	462 130	12,40
13 887	192	334 889	12,07
75 549	141	213 404	8,94
731	98	201 661	7,87
192 404	133	190 394	5,79

TABLE 2.1 – Top 5 des contributeurs influents dans le réseau.

sous-jacente à la plateforme. Afin de vérifier cette hypothèse, le data scientist a partitionné le graphe de co-contributions en 23 clusters, à l'aide de l'algorithme de Louvain [8]. La figure 2.20 illustre les différents clusters avec des colorations différentes. Deux nœuds ayant une même couleur appartiennent à un même cluster.

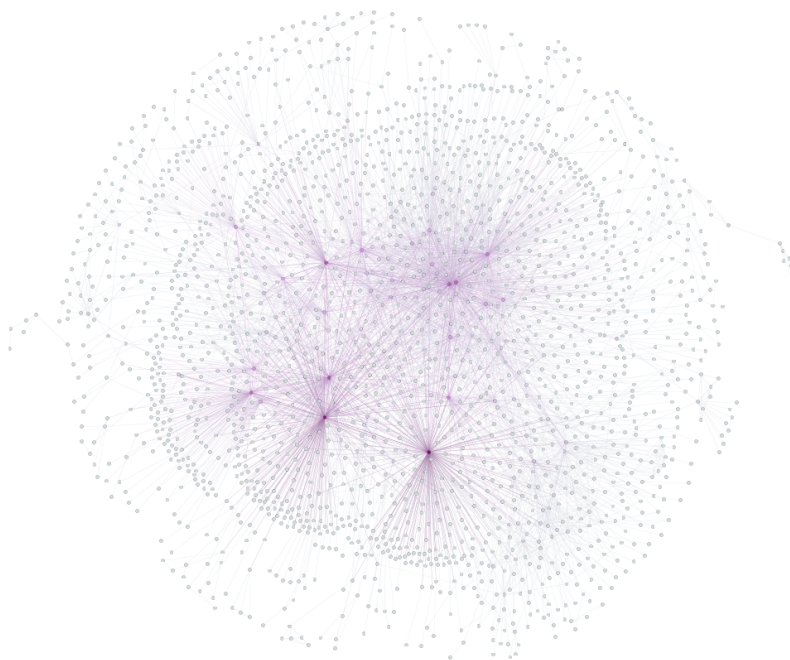


FIGURE 2.19 – Mise en évidence de la centralité d'intermédiarité des nœuds dans le graphe. Les nœuds les plus centraux sont en violet tandis que les moins centraux sont en gris.

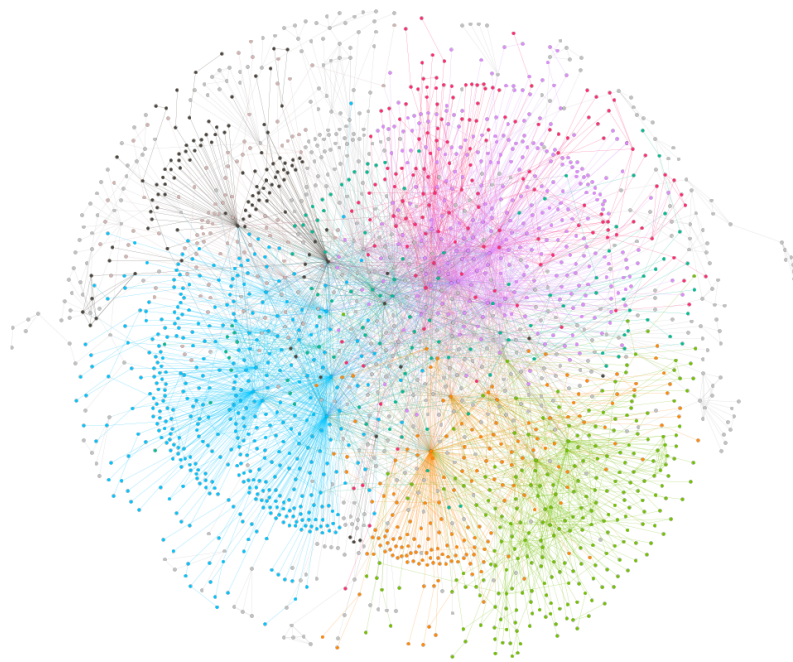


FIGURE 2.20 – Visualisation des 23 clusters détectés en appliquant l'algorithme de Louvain. Les nœuds ayant une même couleur appartiennent à un même cluster.

Identification des profils

Comme indiqué précédemment, la définition des profils des contributeurs repose sur leur activité sociale (les propriétés topologiques) et leur activité de contribution aux projets. Les caractéristiques de l'activité socio-économique sont, entre autres, le montant moyen de contributions, le nombre de projets financés et la date de contribution par rapport à la date de levée de fond.

Ainsi, à partir de ces deux types de caractéristiques, des méthodes d'analyse de données sont mobilisées pour mettre en évidence cinq groupes de contributeurs (voir la figure 2.21). Le premier groupe (*les sponsors*) est constitué de contributeurs les plus centraux. Le deuxième groupe (*les suiveurs*) est constitué de contributeurs qui arrivent tardivement dans les projets, mais qui financent un nombre de projets proche de la moyenne, tandis que le troisième groupe (*les précurseurs*) représente les contributeurs qui arrivent précocement dans les projets. Le quatrième groupe (*les spécialistes collaboratifs*) est, quant à lui, caractérisé par un coefficient de clustering très significatif par rapport au coefficient de clustering moyen dans le graphe. Enfin, le dernier groupe (*les spécialistes*) caractérisé par un coefficient de clustering moins important, mais les contributeurs de ce groupe partagent les mêmes intérêts que leurs voisins.

En conclusion, l'identification des profils des contributeurs a permis à l'économiste de valider l'hypothèse que l'activité d'un contributeur a un impact sur le succès des campagnes de levée de fonds.

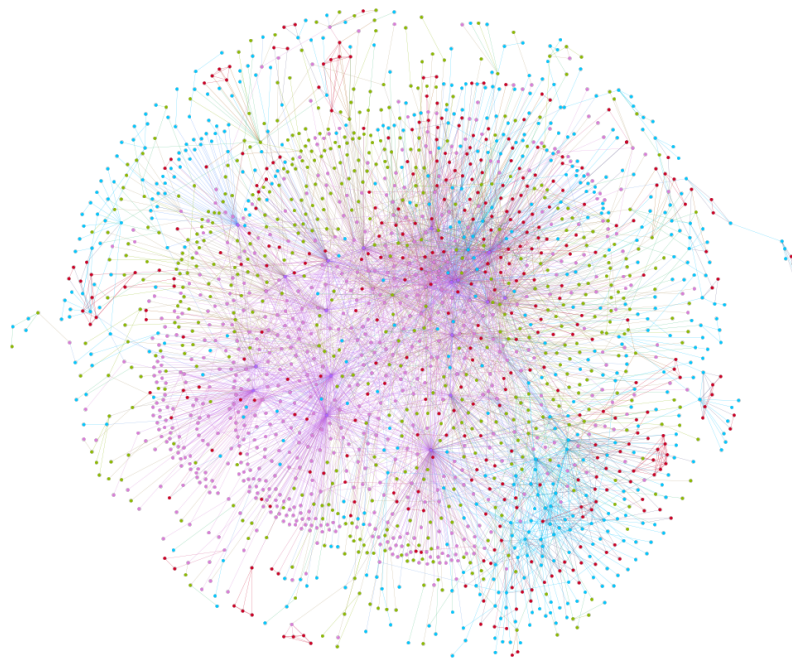


FIGURE 2.21 – Mise en évidence des 5 profils des contributeurs. Cinq profils sont identifiés : les *Sponsors* sont en bleu, les *Suiveurs* en orange, les *Précurseurs* en vert, les *Spécialistes collaboratifs* en rouge et enfin les *Spécialistes* en violet.

2.2.4 Discussion

L'analyse du graphe de co-contributions a permis de montrer l'hétérogénéité du réseau de financement participatif et la présence de contributeurs influents qui facilitent la diffusion des informations dans le réseau. La présence des influenceurs est par ailleurs confirmée par une analyse de la centralité. L'analyse du graphe a également révélé la présence de communautés grâce à l'algorithme de Louvain [8]. La présence de communautés fournit une preuve de l'autocategorisation des contributeurs dans le réseau. La combinaison des caractéristiques socio-économiques avec certaines propriétés topologiques a, quant à elle, permis d'identifier cinq profils des contributeurs.

L'analyse exploratoire des données s'est avérée utile pour identifier des tendances ou des relations entre les contributeurs du réseau afin de générer des idées ou des hypothèses. L'utilisation de multiples méthodes d'analyse et de visualisation a été un élément clé de

cette phase d'analyse pour fournir des informations plus intuitives sur la structure du réseau (identification des zones denses du graphe, par exemple).

Cependant, l'outil de visualisation utilisé (Gephi [4]) n'est pas pratique pour visualiser ou comparer certaines propriétés topologiques des nœuds. Pour comparer par exemple la centralité d'intermédiarité des nœuds le data scientist a utilisé des couleurs. Il peut également manipuler la taille des nœuds pour pouvoir comparer leur centralité. Toutes ces manipulations vont nécessiter un effort cognitif important pour pouvoir analyser le graphe de co-contributions. Cet outil n'est également pas pratique pour visualiser simultanément plusieurs propriétés topologiques ou une propriété topologique et une caractéristique socio-économique (propriété métier). Pour mettre en évidence la centralité d'intermédiarité en plus des profils des contributeurs, le data scientist peut être obligé d'avoir deux graphes de couleurs différentes côte à côte (voir la figure 2.22). Il peut également être obligé d'avoir deux graphes de couleurs différentes pour pouvoir comparer par exemple le coefficient de clustering et le degré des nœuds (voir la figure 2.23). L'outil de visualisation utilisé peut par ailleurs être peu pratique pour comparer visuellement la centralité des nœuds dans un cluster ou pour identifier les nœuds connecteurs. En effet, pour pouvoir visualiser un cluster, l'analyste serait obligé de l'extraire du graphe pour ensuite le visualiser séparément. Cependant, il perd les informations sur la connectivité de ce cluster avec les autres clusters, c'est-à-dire les informations sur les nœuds connecteurs.

En définitive, l'analyse d'un graphe est utile pour découvrir les nœuds influents, les clusters et les principaux intermédiaires au sein des clusters. Pour cela, il est utile de disposer d'un outil de visualisation capable de mettre en évidence simultanément certaines propriétés et mesures des graphes pour faciliter l'exploration et la comparaison de la structure du graphe.

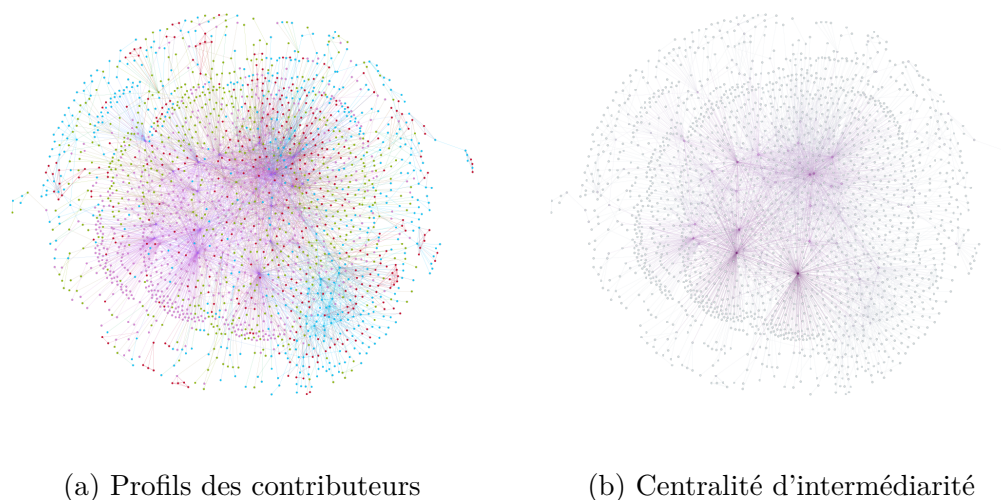


FIGURE 2.22 – Visualisation d'une métrique métier et d'une caractéristique topologique.

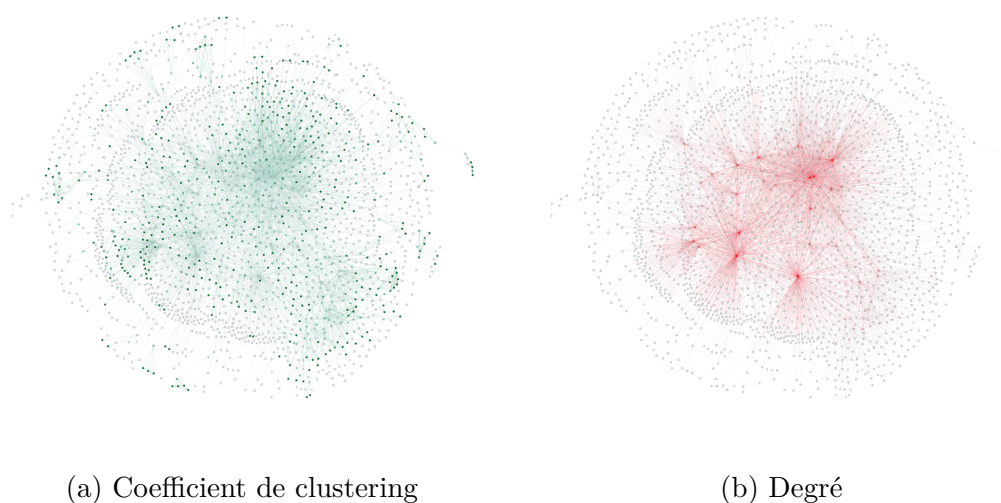


FIGURE 2.23 – Comparaison de deux caractéristiques topologiques.

2.3 Conclusion

La théorie des graphes est un concept mathématique qui permet de modéliser de nombreux problèmes sous la forme d'un graphe où les nœuds représentent les entités et où les arêtes représentent les interactions entre ces entités. Elle a plusieurs applications utiles, en particulier l'analyse des réseaux sociaux.

Les graphes peuvent aider à répondre à différentes questions, telles que la présence de

groupes, l'identification de ces groupes à l'aide d'un algorithme de détection de clusters et l'identification des individus les plus influents dans un groupe. Pour cela, une visualisation de graphes est utile.

Premièrement, une vision d'ensemble d'un graphe peut permettre d'obtenir une intuition pour ensuite identifier des zones denses, mais aussi des individus particuliers (les individus centraux, par exemple) puis émettre des hypothèses. Ainsi, une visualisation peut être utile pour améliorer la compréhension des connexions entre les nœuds d'un graphe. Deuxièmement, une visualisation peut permettre de présenter certains résultats. En effet, l'analyse topologique d'un graphe peut suggérer par exemple la présence d'influenceurs. Une visualisation peut être utile pour présenter visuellement la position de ces influenceurs dans le graphe.

Ainsi, dans le chapitre 3 nous présenterons les algorithmes de visualisation de graphes et de clusters les plus populaires dont l'un des objectifs est de donner un moyen intuitif d'explorer les graphes.

VISUALISATION DE GRAPHERS

L'exploration d'un graphe consiste à fournir des représentations visuelles des nœuds et arêtes et des stratégies d'interactions pour explorer ce graphe [12]. Grâce à ce rendu visuel, il est plus facile de comprendre la connectivité des nœuds du graphe. Ainsi, un outil de visualisation bien conçu pourrait permettre d'identifier facilement certaines structures dans un graphe, telles que les zones denses et les nœuds très fortement connectés [77]. La représentation visuelle d'un graphe permet donc :

- une assimilation rapide des informations ;
- une découverte des informations en interagissant avec les données ;
- une compréhension facile d'un problème en ayant un aperçu complet de la façon dont tout est connecté.

Le problème général de représentation des graphes, par exemple l'attribution de coordonnées aux nœuds, la représentation des arêtes, la mise en évidence des mesures de centralité et autres éléments du graphe, a été largement étudié [83, 50, 16]. Plusieurs classes d'algorithmes de représentation de graphes ont été développées et le choix d'un algorithme spécifique dépend à la fois du type de graphe à visualiser et des exigences du rendu visuel. Par exemple, les graphes orientés contenant une structure hiérarchique inhérente sont représentés à l'aide des algorithmes hiérarchiques [52, 32].

De nombreux autres algorithmes ont également été conçus pour représenter des graphes. Parmi eux, les algorithmes de force et les algorithmes utilisant la réduction de dimension sont les plus couramment utilisés. Les algorithmes de force [83, 50] reposent sur une minimisation par descente de gradient d'une fonction d'énergie. Les algorithmes utilisant une approche de réduction de dimension [51, 15] reposent quant à eux généralement sur une minimisation de la différence entre la distance du plus court chemin dans le graphe et la distance euclidienne [159].

Dans ce chapitre, nous nous concentrons d'abord sur les algorithmes de spatialisation de nœuds, notamment les algorithmes de force et les algorithmes utilisant une réduction de dimension. Nous nous intéressons ensuite aux algorithmes qui permettent de mettre en

évidence la centralité des nœuds puis aux algorithmes permettant de visualiser les clusters dans un graphe. Nous présentons enfin les algorithmes qui permettent de regrouper des arêtes afin de réduire le recouvrement visuel dans un graphe densément connecté.

3.1 Spatialisation de nœuds

Les algorithmes de spatialisation des nœuds basés sur le paradigme de la force consistent à modéliser un graphe en tant qu'un système physique où les nœuds sont attirés et repoussés en fonction d'une certaine force [91]. Les algorithmes utilisant une approche de réduction de dimension exploitent quant à eux des propriétés mathématiques pour déterminer la position des nœuds tout en minimisant la différence entre la distance du plus court chemin et la distance euclidienne. Tous ces algorithmes essaient de respecter certains critères de lisibilité d'un graphe que Dunne et Shneiderman nomment les métriques de lisibilité [35], mais certains sont moins pratiques pour les graphes contenant plusieurs milliers de nœuds et d'arêtes. Ainsi, des algorithmes adaptés à de tels graphes ont été proposés [63].

Dans cette section, nous présentons d'abord les critères de lisibilité d'un graphe les plus utilisés. Nous discutons ensuite des algorithmes de force et de leurs extensions adaptées aux graphes de plusieurs milliers de nœuds puis des algorithmes qui utilisent une approche de réduction de dimension pour calculer la position des nœuds dans un graphe.

3.1.1 Critères de lisibilité d'un graphe

Les critères de lisibilité importants sont donnés par Di Battista et al. [32] et ils consistent à :

- minimiser les croisements d'arêtes : cela permet d'améliorer la lisibilité et de mieux suivre le chemin entre deux nœuds. Les croisements d'arêtes peuvent dissimuler des informations importantes et rendre un graphe moins accessible [78] ;
- homogénéiser les longueurs des arêtes : cela empêche le graphe de se déformer (pour un graphe ayant une structure régulière) ;
- distribuer les nœuds de façon rationnelle sur la surface de représentation : cela évite que les nœuds et arêtes apparaissent trop regroupés et l'ambiguïté quant à l'endroit où se termine une arête [6].

La plupart des algorithmes mettent l'accent sur une distribution rationnelle des nœuds et

une homogénéité des longueurs des arêtes. D'autres s'efforcent explicitement de minimiser les croisements d'arêtes [25].

3.1.2 Algorithmes de force

Les algorithmes de force sont parmi les plus fréquemment utilisés [28] et sont également implémentés dans certains outils de visualisation (Gephi¹ [4], par exemple).

Cette classe d'algorithmes modélise d'abord un graphe comme un système de particules qui exercent des forces les unes sur les autres. Les ressorts sont utilisés pour modéliser les arêtes dans le graphe. Le système est ensuite placé dans une configuration initiale aléatoire et libéré, le laissant atteindre un état stable où la force sur chaque particule est nulle.

L'algorithme de Fruchterman et Reingold [50] est une variante bien connue des algorithmes de force. Dans cet algorithme, une force d'attraction entre deux nœuds adjacents i et j est définie par $f_a = \frac{d_{i,j}^2}{k}$ et une force de répulsion entre chaque paire de nœuds i et j est définie par $f_r = -\frac{k^2}{d_{i,j}}$ où $d_{i,j}$ est la distance entre les nœuds i et j , $k = C\sqrt{\frac{\text{espace disponible}}{\text{nombre de nœuds}}}$ est la distance optimale entre les nœuds i et j et C est une constante déterminée expérimentalement. La convergence de l'algorithme n'est pas garantie, donc Fruchterman et Reingold suggèrent que 50 itérations sont suffisantes pour avoir un rendu visuel optimal [91]. Le temps d'exécution de cet algorithme est de $O(|V|^2 + |E|)$: toutes les paires de nœuds doivent être prises en compte pour calculer les forces répulsives et toutes les arêtes sont prises en compte pour calculer les forces attractives. Afin d'éviter les changements excessifs, en particulier dans les dernières étapes de l'itération lorsque le positionnement est proche d'un état stable, l'algorithme utilise une valeur de déplacement maximal dépendant du temps qui diminue avec le temps.

Kamada et Kawai [83] ont introduit une variante différente des algorithmes de force, dans laquelle la distance euclidienne entre deux nœuds doit se rapprocher de la distance théorique entre ces nœuds, c'est-à-dire la longueur de leur plus court chemin (section 2.1.4). Ils considèrent que le résultat de l'algorithme est le minimum d'un problème d'optimisation dans lequel une fonction d'énergie encode les propriétés souhaitées d'un graphe. Ainsi, les solutions qui aboutissent à des minima locaux plutôt que globaux ne sont pas optimales [28]. L'idée ici est de minimiser directement l'énergie du positionnement du graphe au lieu de réduire les forces qui agissent sur les nœuds. La fonction d'énergie d'un

1. Une plateforme de visualisation de graphes

graphe comportant n nœuds est définie par [91] :

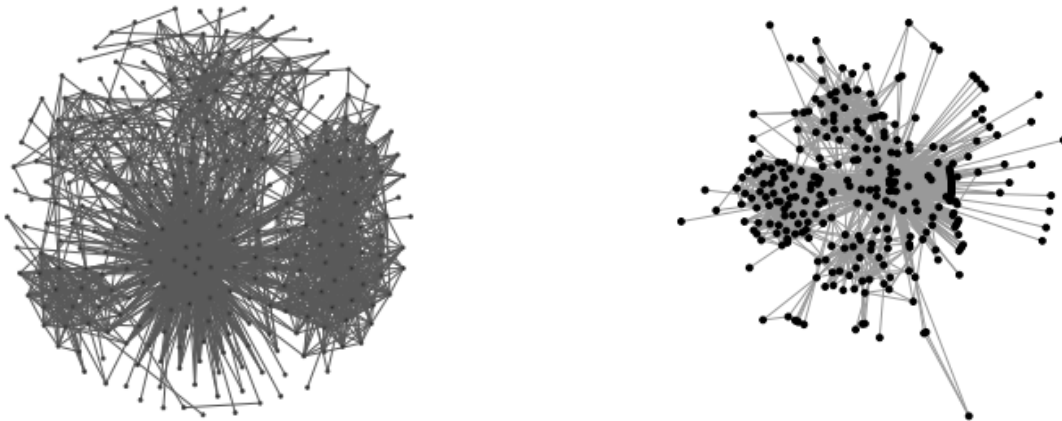
$$E = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{1}{2} k_{i,j} (|p_i - p_j| - l_{i,j})^2 \quad (3.1)$$

où

- $k_{i,j} = K/d_{i,j}^2$ est la force entre les nœuds i et j , où K est une constante et $d_{i,j}$ est la distance (le plus court chemin) entre les nœuds i et j ;
- p_i et p_j sont respectivement les positions des nœuds i et j ;
- $l_{i,j} = \frac{L_0}{\max_{i < j} d_{i,j}} \times d_{i,j}$ est la distance idéale entre les nœuds i et j , où L_0 est la longueur d'un côté de la zone d'affichage et $\max_{i < j} d_{i,j}$ est le diamètre du graphe, c'est-à-dire la plus grande distance entre deux nœuds (section 2.1.4).

Contrairement à d'autres algorithmes, l'algorithme de Kamada et Kawai ne vise pas à minimiser les croisements d'arêtes ; il peut rapprocher voire superposer certains nœuds qui ne sont pas forcément connectés [91]. Cet algorithme ne considère pas les forces d'attraction et de répulsion séparément, mais plutôt conjointement pour minimiser la fonction d'énergie [28]. Cependant, son temps d'exécution est assez élevé ($O(|V|^3)$), car il nécessite de calculer les plus courts chemins de toutes les paires de nœuds et analyser tous les nœuds du graphe, puis ne déplacer qu'un seul nœud.

La figure 3.1 illustre les résultats des algorithmes de Fruchterman et Reingold [50] et de Kamada et Kawai [83] sur un graphe d'interactions protéiques.



(a) Fruchterman et Reingold (Gephi [4])

(b) Kamada et Kawai (Ondex [92])

FIGURE 3.1 – Layout d'un graphe d'interactions protéiques (Images extraites de Gibson et al. [55]).

En raison du coût de calcul élevé des algorithmes de force, de nombreuses extensions

ont été proposées afin de représenter les graphes contenant plusieurs milliers de nœuds et d'arêtes [63, 75]. Une méthode populaire consiste à utiliser une approche multi-niveaux [65, 147]. L'idée ici est de d'abord réduire récursivement la taille du graphe en trouvant des appariements qui fusionnent les nœuds et les arêtes, tout en conservant la structure topologique du graphe initial, jusqu'à obtenir un graphe suffisamment petit. Ensuite, les nœuds du graphe le plus grossier créé reçoivent une position initiale et la spatialisation est successivement optimisée sur tous les graphes intermédiaires jusqu'au graphe original (voir un exemple dans la figure 3.2).

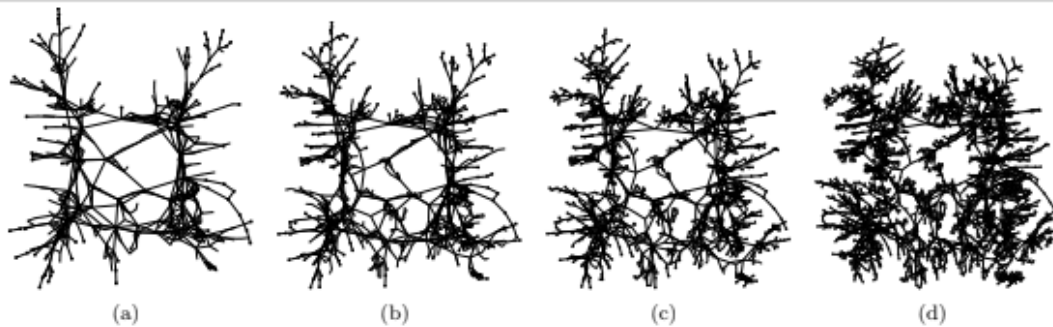


FIGURE 3.2 – 4 étapes de l'algorithme multi-niveau de Hu [75] produit dans Gephi [4]. Chaque image (de la gauche vers la droite) montre la propagation du layout à travers chaque version moins grossière du graphe, c'est-à-dire (a) est la version la plus grossière du graphe, tandis que (d) est le layout final du graphe (Images extraites de Gibson et al. [55]).

Malgré quelques améliorations ajoutées par les nouvelles méthodes, le temps de calcul et le problème de convergence persisteraient pour des graphes de plusieurs millions de nœuds et d'arêtes. Ainsi, pour répondre à ces problématiques, Brinkmann et al. [18] proposent un nouvel algorithme capable de calculer la position des nœuds d'un graphe de 4 millions de nœuds et 120 millions d'arêtes en 14 minutes en exploitant les performances de la carte graphique (voir la figure 3.3). Cet algorithme permet d'identifier des groupes présents dans le graphe, mais il est moins pratique pour l'identification de la connexion entre deux nœuds.

Les algorithmes de force déterminent la position des nœuds en utilisant un système physique avec des forces d'attraction et de répulsion entre les nœuds d'un graphe. Bien que ces algorithmes soient simples et faciles à mettre en œuvre [5], ils présentent une complexité élevée de calcul des forces entre les paires de nœuds. Cependant, préserver des distances entre les paires de nœuds très éloignés peut entraîner une charge importante à

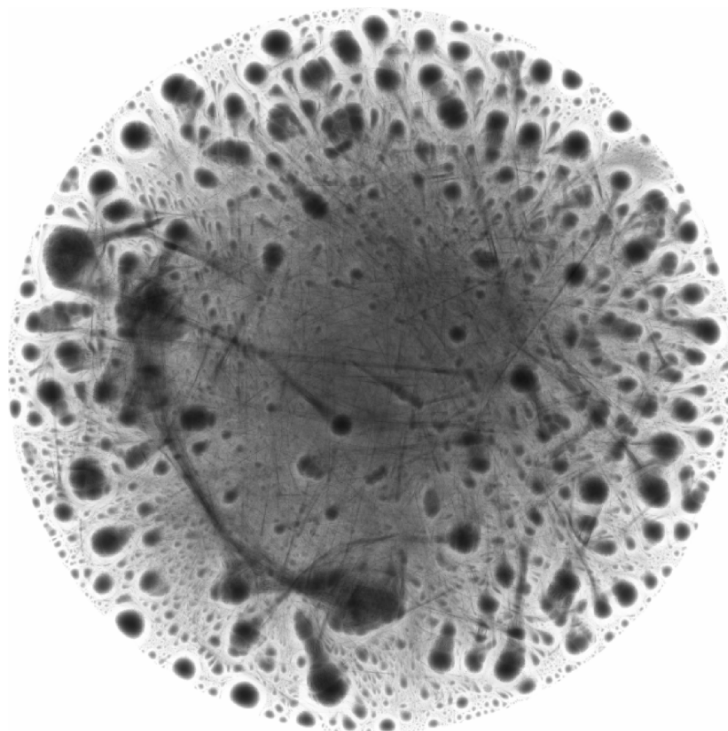


FIGURE 3.3 – Visualisation d’un graphe (4 602 225 nœuds et 123 329 543 arêtes) en exploitant les performances de la carte graphique (Image extraite de Brinkmann et al. [18] (© 2017, IEEE)).

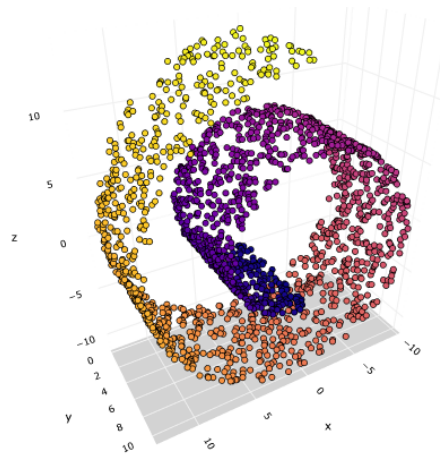
la fonction d’optimisation. Ainsi, ces algorithmes sont moins pratiques pour préserver les structures locales des graphes (par exemple la transitivité, c’est-à-dire la probabilité que les nœuds voisins d’un nœud soient connectés) et peuvent facilement converger vers des minima locaux [28].

Comme alternative, certains travaux proposent d’adapter des méthodes de réduction de dimension, telles la « Multidimensional scaling » (MDS) [97], l’analyse en composante principale (ACP) [82] et la « t-distributed stochastic neighbor embedding » (t-SNE) [110], à la représentation des graphes [15, 69, 96]. Du point de vue des travaux de cette thèse, nous exploiterons des graphes pouvant contenir plusieurs milliers de nœuds et d’arêtes. Nous pensons donc que les algorithmes qui utilisent une approche de réduction de dimension peuvent conduire à des visualisations significatives permettant d’améliorer l’exploration des graphes et la mise en évidence de certaines propriétés topologiques utiles.

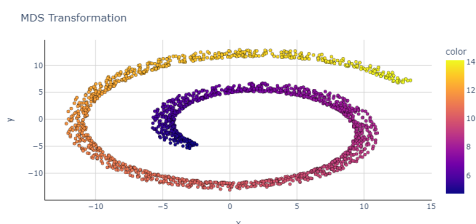
3.1.3 Algorithmes utilisant la réduction de dimension

3.1.3.1 Réduction de dimension

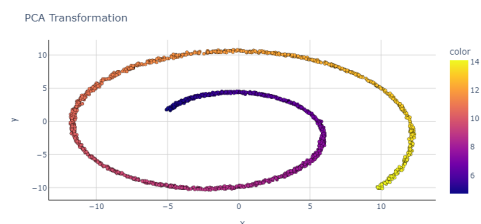
La réduction de dimension consiste à prendre des données exprimées dans un espace de grande dimension et à les projeter sur un espace de dimension inférieure [162] (voir la figure 3.4). En tant qu'un moyen fondamental de visualisation, la réduction de dimension a été utilisée dans plusieurs domaines et sur un ensemble de données en constante augmentation [120]. Les méthodes classiques comprennent la MDS [97] et l'ACP [82]. La MDS est une méthode de réduction de dimension qui tente de préserver la distance entre les paires de points (la distance euclidienne, par exemple), contrairement à l'ACP qui cherche à préserver la variance des données.



(a) Représentation des données en 3D



(b) MDS (dimension=2)



(c) ACP (dimension=2)

FIGURE 3.4 – Comparaison entre la MDS et l'ACP : (a) donnée d'origine en 3 dimensions, (b) réduction de la dimension à l'aide de la MDS et (c) réduction de la dimension à l'aide de l'ACP. La MDS cherche à préserver au mieux les structures globale et locale des données, alors que l'ACP tente de préserver plutôt la structure globale des données.

La MDS vise à préserver la structure de voisinage local qui est analogue au concept de représentation des graphes [159], alors que l'ACP ne préserve que la structure globale.

Ainsi, adapter les méthodes de réduction de dimension à la représentation des graphes peut produire des résultats permettant de mettre en évidence les structures globale et locale d'un graphe.

3.1.3.2 Algorithmes basés sur la réduction de dimension

Les algorithmes utilisant une approche de réduction de dimension visent généralement à préserver les structures des graphes [58], c'est-à-dire minimiser la différence entre la distance euclidienne et la distance du plus court chemin entre deux nœuds. De nombreux travaux ont poursuivi cette ligne de pensée et ont illustré comment utiliser la réduction de dimension pour représenter des graphes [159, 96]. Les algorithmes basés sur la réduction de dimension peuvent être classés en deux : les algorithmes basés sur la projection et les algorithmes basés sur la distance (modèle MDS).

Les algorithmes basés sur la projection comportent deux étapes. D'abord, l'intégration d'un graphe dans un espace à haute dimension, puis la projection des vecteurs dans un espace de dimension inférieure. L'algorithme « high dimension embedding » (HDE) proposé par Harel et Koren [69] utilise l'ACP pour projeter le graphe. Koren [94] améliore ensuite le HDE en remplaçant l'ACP par une optimisation de l'espace de dimension inférieure. Hachul et Jünger [64] comparent le temps d'exécution et la qualité de représentation du HDE par rapport à certains algorithmes de force. Les résultats de leur évaluation montrent que le HDE est le plus rapide. Pour comparer la qualité de représentation, Hachul et Jünger mesurent le nombre de croisements d'arêtes et le nombre de recouvrements d'arêtes de chaque algorithme. Les résultats de leur évaluation montrent que le HDE est parmi les meilleurs algorithmes en termes du nombre de croisements d'arêtes, mais en termes du nombre de recouvrements d'arêtes, il n'obtient pas de bons scores, comparativement aux autres algorithmes.

Les algorithmes basés sur la distance utilisent quant à eux la distance théorique, c'est-à-dire la distance des plus chemins entre les paires de nœuds. Ils visent à minimiser la différence entre la distance théorique et la distance euclidienne entre les paires de nœuds [15]. Il existe deux approches pour résoudre ce problème : l'approche (*distance scaling*) [98] et l'approche (*classical scaling*) [142]. L'approche distance scaling permet de mettre en évidence les détails locaux d'un graphe, alors que l'approche classical scaling est plutôt adaptée pour montrer la structure globale du graphe [14].

L'approche distance scaling (figure 3.5a) est la plus courante et l'idée est de calculer directement une approximation de la différence entre la distance théorique et la distance

euclidienne pour chaque paire de nœuds. La somme des carrés de la différence est appelée le « stress » de la représentation et le but est de minimiser ce stress grâce à une procédure d’optimisation. L’équation 3.2 représente la fonction du stress à minimiser [14] :

$$\text{stress}(X) = \sum_{i \neq j} w_{i,j} (d_{i,j} - \|X_i - X_j\|)^2 \quad (3.2)$$

où $d_{i,j}$ est la distance entre les nœuds i et j , X_i et X_j sont respectivement les positions des nœuds i et j et $w_{i,j}$ est un facteur de pondération. Un choix typique est $w_{i,j} = d_{i,j}^{-2}$, donc cette contrainte d’énergie mesure la différence relative entre la longueur réelle et la longueur idéale de l’arête.

L’approche distance scaling considère le plus court chemin entre une paire de nœuds comme étant la distance idéale. Cela lui permet de bien distribuer les nœuds dans la représentation et par conséquent de montrer les structures locales d’un graphe (la connectivité du voisinage d’un nœud, par exemple).

L’approche classical scaling essaie également de minimiser la différence entre la distance euclidienne et la distance théorique entre les paires de nœuds d’un graphe tout en mettant en évidence la structure globale du graphe (voir la figure 3.5b). Pour représenter la structure globale, elle se concentre sur les longs chemins entre les nœuds [58], contrairement à l’approche distance scaling qui se concentre plutôt sur les plus courts chemins grâce au facteur de pondération ($w = d^{-2}$ dans l’équation 3.2).

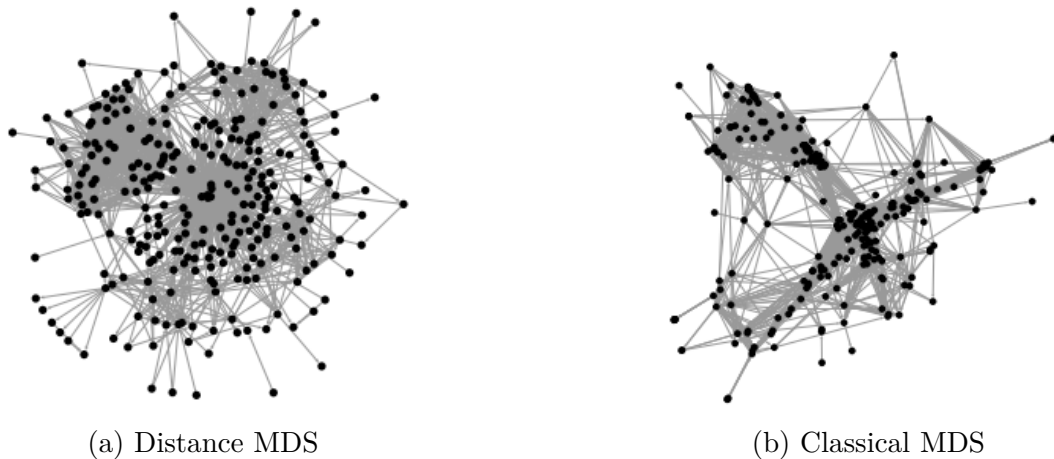


FIGURE 3.5 – Layout d’un graphe d’interactions protéiques. (a) l’approche distance scaling met en évidence les détails locaux, alors que (b) l’approche classical montre la structure globale du graphe. (Image extraite de Gibson et al. [55]).

Cependant, comparativement aux algorithmes de force, ces deux approches n'améliorent pas le temps d'exécution ($O(|V|^2 \log |V| + |V||E|)$) car elles nécessitent toujours le calcul des chemins les plus courts entre toutes les paires de nœuds. Brandes et Pich [15] proposent alors une méthode d'approximation connue sous le nom de *Pivot MDS* où certains nœuds sont désignés comme pivots. Le premier pivot est choisi au hasard et les suivants sont choisis en étant le nœud ayant la plus grande distance par rapport à tous les pivots déjà sélectionnés. Les non-pivots sont positionnés en fonction de leur distance par rapport aux pivots. Par contre, les pivots peuvent utiliser les distances à la fois par rapport aux autres pivots et aux non-pivots pour déterminer leur position. Brandes et Pich [15] réalisent cependant que le choix des nœuds pivots est un problème clé et proposent une seconde méthode, *Progressive MDS*, où le nombre de pivots peut être défini progressivement.

Algorithmes basés sur la réduction de dimension

Les algorithmes de réduction de dimension sont une famille d'algorithmes qui reposent sur des propriétés mathématiques pour représenter des graphes tout en cherchant à minimiser la différence entre la distance euclidienne et la distance théorique entre deux nœuds. La distance scaling permet d'améliorer la représentation des structures locales d'un graphe, tandis que la classical scaling est plus adaptée pour mettre en évidence la structure globale du graphe [14]. Le HDE [69] est quant à lui l'un des algorithmes les plus rapides, malgré certains problèmes de qualité [64]. Brandes et Pich [14] soulignent par ailleurs que le HDE et le PivotMDS ont des temps d'exécution similaires, mais la qualité du résultat du PivotMDS semble être meilleure que celle du résultat de la HDE.

3.1.4 Synthèse sur la spatialisation des nœuds

Plusieurs algorithmes ont été proposés pour répondre au problème de représentation des graphes. Initialement, la plupart étaient les algorithmes de force qui sont toujours restés populaires. Cependant, ils font face à un certain nombre d'inconvénients techniques lorsqu'il s'agit de visualiser des graphes contenant plusieurs milliers de nœuds et d'arêtes. D'abord, quand bien même les performances matérielles ont été améliorées, le temps de

calcul reste toujours élevé pour calculer la position des nœuds. Deuxièmement, ils sont moins pratiques que les algorithmes basés sur la réduction de dimension pour préserver les détails locaux des graphes [110]. En effet, à cause des forces d'attraction et de répulsion, certains nœuds peuvent être éloignés de leurs voisins directs.

Afin de résoudre ces problèmes, de nouvelles méthodes d'approximation et heuristiques ont été proposées et adaptées aux grands graphes. Celles-ci comprennent les algorithmes multi-niveaux qui rendent les approches de force plus efficaces pour les grands graphes et les algorithmes basés sur la réduction de dimension qui visent à préserver la structure globale des graphes et les structures de voisinages des nœuds. Cette deuxième famille d'algorithmes pourrait bien être utile au domaine d'analyse de graphes qui peut s'intéresser parfois aux structures internes des graphes. En effet, les algorithmes basés sur la réduction de dimension pourraient faciliter la visualisation de la connectivité entre les nœuds d'un graphe et entre les structures du graphe.

Le tableau 3.1 présente un résumé comparatif de quelques algorithmes de spatialisation des nœuds selon certains critères de lisibilité d'un graphe (section 3.1.1), la taille du graphe et la qualité de la représentation. Globalement, les algorithmes utilisant une approche de réduction de dimension (le HDE, le PivotMDS) et les approches multi-niveaux sont parmi les algorithmes les plus rapides et les mieux adaptés aux grands graphes [76].

Algorithmes	Croise-ments	Distri-bution	Nombre de nœuds	Complexité	Qualité
Fruchterman et Rein-gold [50]	✓	✓	$\leq 1\ 000$	$O(V ^2 + E)$	bonne
Kamada et Kawai [83]	✗	✓	≥ 30	$O(V ^3)$	bonne
PivotMDS [15]	✗	✗	$\geq 100\ 000$	$O(k E + k^2n)$	bonne
classical MDS [15]	✗	✗	$\geq 1\ 000$	$O(V ^2 \log V + V E)$	moyenne
distance MDS [15]	✗	✗	$\geq 1\ 000$	$O(V ^2 \log V + V E)$	bonne
HDE [69]	✗	✗	$\geq 1\ 000\ 000$	$O(k E + k^2n)$	pauvre
Approches Multi-niveaux [63, 75]	✓	✓	$\geq 10\ 000$ en $< 30s$	$O(V \log V + E)$	bonne

TABLE 3.1 – Résumé comparatif de quelques algorithmes de spatialisation des nœuds selon certains critères de lisibilité d'un graphe (section 3.1.1, p.56). Une coche (✓) d'un critère indique qu'il a été pris en compte dans la conception de l'algorithme et une croix (✗) indique le contraire. $k < |V|$ est le nombre de nœuds pivots pour PivotMDS [15] et le nombre de nœuds centraux pour la HDE [69].

3.2 Mise en évidence de la centralité des nœuds dans un graphe

Alors que la plupart des techniques de visualisation de graphes se concentrent sur l’affichage des nœuds et de leurs attributs, une autre application importante de la visualisation de graphes est la mise en évidence de certaines propriétés topologiques des nœuds [16, 123].

Ainsi, une technique de visualisation idéale pourrait permettre de communiquer efficacement des informations topologiques des nœuds et faciliter l’identification des structures sous-jacentes grâce à la représentation visuelle. Elle pourrait par exemple :

- permettre de mettre en évidence certaines propriétés topologiques des nœuds dans un graphe (exemple : la centralité, les connecteurs, les hubs et le coefficient de clustering) ;
- être adaptée aux grands graphes (limiter les recouvrements de nœuds et d’arêtes) ;
- fournir des stratégies d’interactions qui permettent de mieux explorer un graphe.

Historiquement, la question de lisibilité d’un graphe reposait principalement sur des critères esthétiques présentés dans la section 3.1.1 tels que minimiser les croisements d’arêtes et homogénéiser les longueurs des arêtes. Mais d’autres facteurs sont également pris en compte dans les travaux les plus récents. Par exemple, certains travaux s’intéressent à la compréhension et à la description des interactions en analysant la topologie d’un graphe [132, 39]. D’autres s’intéressent à l’identification et à la caractérisation des nœuds influents dans un graphe [148] et comment leurs voisins sont connectés entre eux [163].

Pour cela, Brandes et al. [13] et Brandes et Pich [16] proposent des méthodes radiales qui intègrent la centralité des nœuds dans un modèle de présentation d’un graphe. La méthode proposée par Brandes et al. [13] est basée sur une extension de l’algorithme de Kamada et Kawai [83], tandis que celle proposée par Brandes et Pich [16] est basée sur une extension de l’algorithme de minimisation des contraintes MDS présenté dans la section 3.1.3. Dans ces méthodes, la valeur d’une mesure de centralité de chaque nœud est représentée par le rayon d’un cercle sur lequel doit être placé ce nœud. Cette valeur est normalisée et les nœuds les moins centraux sont placés à la périphérie du graphe sur un cercle de rayon égal à 1 tandis que le nœud le plus central est associé à un cercle de rayon égal à 0 et se retrouve donc au centre du graphe. Le rayon r_i du cercle associé à un

nœud i est calculé comme suit [16] :

$$r_i = \frac{\text{diam}(G)}{2} * \left(1 - \frac{C_i - \min_{u \in V} C_u}{\max_{u \in V} C_u - \min_{u \in V} C_u + C(G)}\right) \quad (3.3)$$

où :

- $\text{diam}(G)$ est le diamètre d'un graphe G . Rappelons que le diamètre d'un graphe est la plus grande distance entre deux nœuds (section 2.1.4). $\frac{\text{diam}(G)}{2}$ sert à ramener tous les rayons à la même échelle ;
- C_i est la valeur d'une mesure de centralité d'un nœud i ;
- $\min_{u \in V} C_u$ et $\max_{u \in V} C_u$ sont respectivement la valeur minimale et la valeur maximale d'une mesure de centralité des nœuds dans le graphe G ;
- $C(G)$ est un petit décalage supérieur à zéro s'il existe plusieurs nœuds avec une centralité maximale.

Ainsi, la distance au centre encode une mesure de centralité qui est un résultat d'une étape d'analyse préalable.

Brandes et Pich [16] proposent ensuite de mettre en place un mécanisme dit d'emphase permettant de mieux visualiser les nœuds concernés en augmentant la distance entre eux. Par contre, les nœuds non concernés par la mise en emphase seront plus proches. Ainsi, si l'emphase est mise au centre, les nœuds à la périphérie seront plus proches et inversement si l'emphase est mise à la périphérie. La figure 3.6 illustre une mise en évidence de la centralité d'intermédiarité.

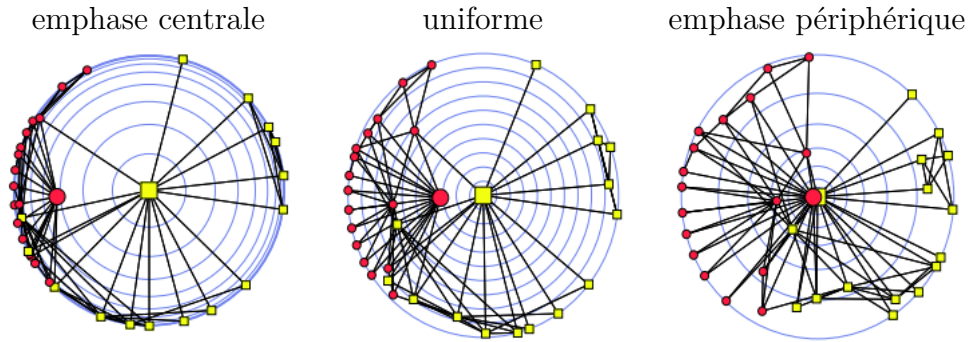


FIGURE 3.6 – Visualisation radiale de la centralité d'intermédiarité du graphe « Karate club » [160] (34 nœuds et 78 arêtes). Le centre et la périphérie sont mis en évidence par une transformation respective des rayons $r'_i = 1 - (1 - r_i)^3$ et $r'_i = r_i^3$ tels que $(0 \leq r_i \leq 1$ et $0 \leq r'_i \leq 1)$ (Images extraites de Brandes et Pich [16]).

Raj et Whitaker [123] proposent quant à eux une méthode anisotrope radiale basée

également sur une extension de l'algorithme de MDS pour mettre en évidence la centralité d'intermédiarité des nœuds dans un graphe (voir la figure 3.7). Ils proposent d'utiliser plutôt des courbes d'isocentralité au lieu des cercles concentriques utilisés dans l'approche radiale de Brandes et Pich [16], mais aussi la taille des nœuds pour représenter la centralité. L'utilisation de ces courbes permet d'offrir plus de flexibilité pour préserver la structure du graphe, comparativement à l'approche radiale de Brandes et Pich. Cependant, il serait alors plus complexe d'identifier certains nœuds qui ont une même valeur de centralité. En effet, on serait obligé de suivre une courbe irrégulière pour pouvoir identifier par exemple deux nœuds qui ont une même valeur de centralité d'intermédiarité. De plus, visuellement certains nœuds pourraient avoir la même taille, mais de valeurs de centralité différentes.

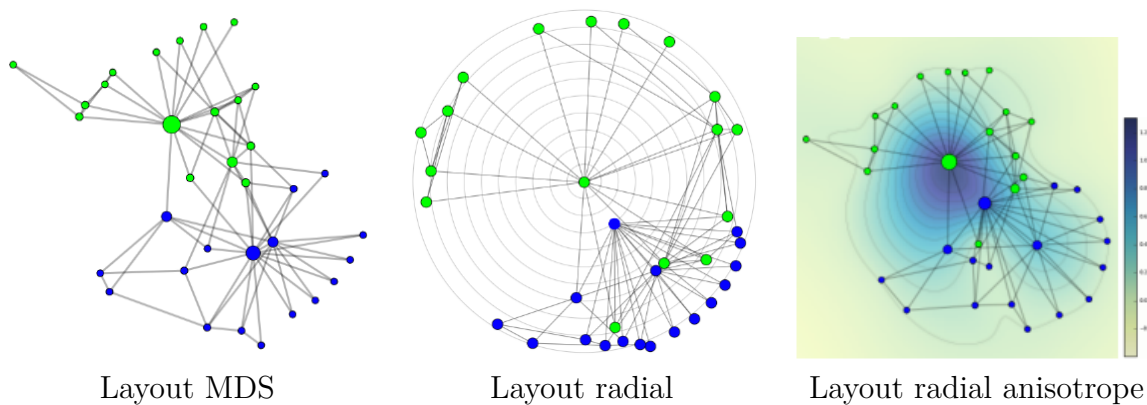


FIGURE 3.7 – Mise en évidence de la centralité d'intermédiarité du graphe « Karate club » [160] (34 nœuds et 78 arêtes) (Images extraites de Raj et Whitaker [123]).

Pour pouvoir comparer simultanément différentes mesures de centralité dans un graphe, Dwyer et al. [36] proposent trois méthodes de visualisation : 1) les coordonnées parallèles 3D, 2) la comparaison basée sur l'orbite et 3) la comparaison basée sur la hiérarchie. La différence entre ces trois méthodes réside principalement au niveau de la mise en correspondance des valeurs de centralité avec la position des nœuds.

La méthode des coordonnées parallèles 3D (voir la figure 3.8) permet de placer les nœuds sur des lignes verticales où chaque ligne représente une mesure de centralité. Une ligne verticale comporte plusieurs lignes horizontales dont l'ensemble constitue un plan. Elle représente également une plage des valeurs de centralité, de la valeur la plus faible à la plus élevée. Les nœuds ayant une même valeur de centralité sont placés sur une même ligne horizontale pour mieux les séparer. Il existe ensuite des arêtes interplans entre deux plans adjacents lorsque la différence de centralité est supérieure à un seuil donné, de sorte

que les mêmes nœuds du graphe soient connectés. Cela permet d'identifier visuellement les nœuds entre deux plans différents.

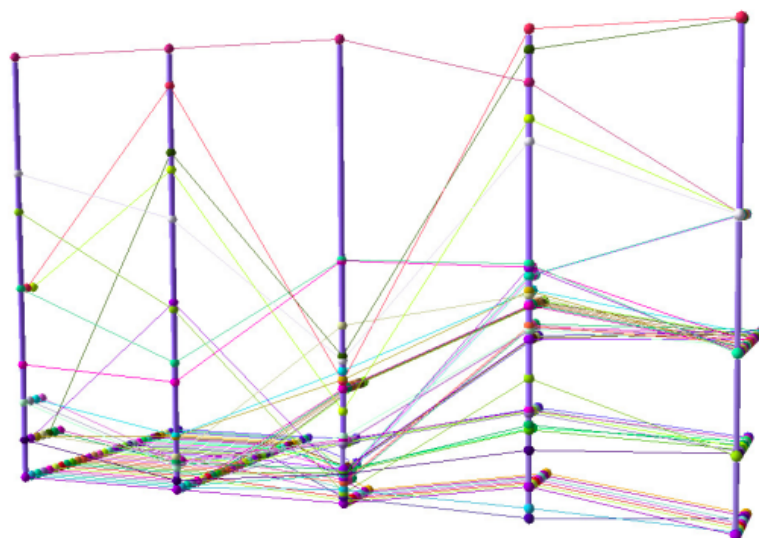


FIGURE 3.8 – Les coordonnées parallèles 3D montrant 5 mesures de centralité des nœuds (degré, excentricité, vecteur propre, proximité et intermédierité). Chaque mesure de centralité est représentée par une ligne verticale et les nœuds ayant une même valeur de centralité sont placés sur une même ligne horizontale (Image extraite de Dwyer et al. [36]).

En ce qui concerne l'approche orbitale (voir la figure 3.9), les nœuds sont placés sur des cercles concentriques (orbites) en fonction de leur valeur de centralité. Ainsi, chaque nœud se voit attribuer une contrainte orbitale avec un rayon $r = f(C(u))$. Tout comme l'approche des coordonnées parallèles 3D, chaque mesure de centralité est représentée par un plan comportant des cercles concentriques et des arêtes interplans qui connectent les mêmes nœuds du graphe.

Enfin, les nœuds sont placés sur des lignes horizontales pour l'approche hiérarchique (voir la figure 3.10). Comme pour les méthodes précédentes, une représentation dans chaque plan correspond à une mesure de centralité. Pour représenter chaque graphe dans un plan sous forme de hiérarchie, les nœuds sont d'abord divisés en couches en fonction de leur valeur de centralité. Les nœuds des couches supérieures ont des valeurs de centralité plus élevées que ceux des couches inférieures. Chaque nœud d'une couche est ensuite placé sur une ligne horizontale suivant un ordre, par exemple de gauche à droite, en fonction de leur valeur de centralité.

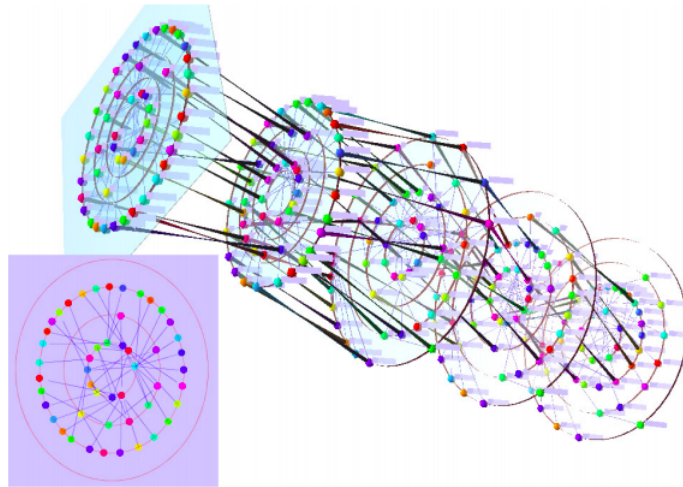


FIGURE 3.9 – Visualisation orbitale montrant 5 mesures de centralité des nœuds (degré, excentricité, vecteur propre, proximité et intermédiarité). Chaque plan de cercles concentriques représente une mesure de centralité. Dans la petite fenêtre (à gauche), le plan mis en évidence est affiché en 2D (Image extraite de Dwyer et al. [36]).

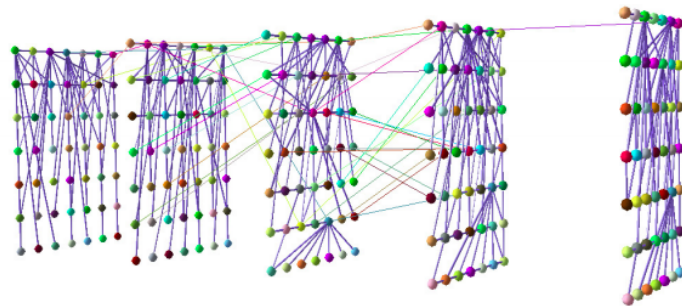


FIGURE 3.10 – Visualisation hiérarchique montrant 5 mesures de centralité des nœuds (degré, excentricité, vecteur propre, proximité et intermédiarité). Chaque plan hiérarchique représente une mesure de centralité (Image extraite de Dwyer et al. [36]).

Les approches orbitale et hiérarchique facilitent particulièrement le suivi de l'évolution des valeurs de centralité des nœuds et la structure du graphe, contrairement à la méthode des coordonnées parallèles 3D où les arêtes entre les nœuds sont absentes. Cependant, il est visuellement plus facile d'identifier le nœud central avec la méthode des coordonnées parallèles 3D, comparativement aux méthodes orbitale et hiérarchique.

Synthèse

Les approches de Dwyer et al. [36] permettent de comparer plusieurs mesures de centralité, mais il est visuellement difficile d'identifier les nœuds centraux, contrairement aux approches radiales [13, 16]. En ce qui concerne la méthode des coordonnées parallèles 3D, l'absence des arêtes entre les nœuds supprime une information importante pour un analyste de graphes. Néanmoins, il est moins complexe d'identifier les nœuds centraux, comparativement aux méthodes orbitale et hiérarchique.

Avec la méthode anisotrope de Raj et Whitaker [123], la taille des nœuds encode la centralité d'intermédiarité, alors que celle-ci est visuellement représentée avec la méthode radiale proposée par Brandes et Pich [16]. Par ailleurs, il est plus complexe d'identifier certains nœuds ayant une même valeur de centralité. Dans ce cas, il faudra suivre une courbe pour pouvoir les identifier. Ainsi, la représentation visuelle qu'offre la méthode radiale de Brandes et Pich facilite une comparaison de la centralité des nœuds.

Dwyer et al. [36] ont ajouté une troisième dimension aux coordonnées parallèles 2D pour pouvoir placer horizontalement les nœuds qui ont une même valeur de centralité. Cependant, il est complexe de suivre l'évolution des valeurs de centralité des nœuds et la structure du graphe si ce dernier est de grande taille. Toutes les méthodes 2D présentées peuvent donc être moins pratiques pour mettre en valeur la centralité des nœuds dans un grand graphe.

Pour conclure, les différentes méthodes présentées permettent de mettre en évidence la centralité des nœuds dans un graphe sans ambiguïté lorsque celui-ci est de taille relativement réduite. Cependant, d'autres caractéristiques topologiques des nœuds ne sont pas traitées, car certains analystes de graphes peuvent s'intéresser à observer simultanément, par exemple la centralité et le coefficient de clustering (propriétés nœuds, section 2.1.5). Par ailleurs, Dwyer et al. [36] ont montré que l'ajout d'une troisième dimension permet d'améliorer la lisibilité d'un graphe. Par conséquent, la 3D pourrait permettre de mieux visualiser des graphes contenant plusieurs milliers de nœuds et d'arêtes.

Une comparaison des différentes approches discutées dans cette section est présentée dans le tableau 3.2. Cette comparaison porte sur la mise en évidence de certaines propriétés des nœuds dans un graphe, telles que la centralité et le coefficient de clustering, et l'adaptation à un grand graphe. Globalement, toutes ces techniques permettent de mettre en évidence la centralité des nœuds. Cependant, seule la méthode des coordonnées parallèles 3D [36] semble être adaptée à un graphe pouvant contenir au moins plusieurs milliers de nœuds.

Techniques	Centralité des nœuds	Coefficient de clustering	Bien adaptée aux grands graphes
Coordonnées parallèles 3D [36]	✓possibilité de comparer 5 mesures simultanément	✗	✓
Comparaison basée sur l'orbite [36]	✓possibilité de comparer 5 mesures simultanément	✗	✗
Comparaison basée sur la hiérarchie [36]	✓possibilité de comparer 5 mesures simultanément	✗	✗
Visualisation radiale [16]	✓	✗	✗
Visualisation radiale anisotrope [123]	✓identification des nœuds de même valeur de centralité complexe	✗	✗

TABLE 3.2 – Tableau comparatif des différentes techniques de mise en évidence des propriétés topologiques des nœuds dans un graphe (section 2.1.5). Une coche (✓) d'un critère indique qu'il est pris en compte par la technique et une croix (✗) indique le contraire.

3.3 Visualisation de clusters dans un graphe

Un grand graphe complexe est un graphe composé d'au moins plusieurs dizaines de milliers de nœuds et qui a des caractéristiques topologiques non triviales (exemple : une faible densité, un petit diamètre, un coefficient de clustering élevé et une distribution de degrés hétérogène) [31]. Par conséquent, essayer de comprendre la structure d'un tel graphe à partir de sa visualisation peut faire appel à un effort cognitif important. Ainsi, plusieurs stratégies de partitionnement (détection de clusters, section 2.1.7) peuvent être utilisées pour identifier automatiquement des groupes de nœuds (clusters). D'une part, la visualisation de ces groupes est utile pour analyser et évaluer les algorithmes de partitionnement. D'autre part, la visualisation des groupes peut aider à identifier les groupes très connectés. Elle peut également aider à comprendre la connectivité entre les différents groupes et fournir une orientation de l'analyse. Par exemple, un analyste peut directement s'intéresser à un groupe grâce à la visualisation. Donc, les objectifs de la visualisation des

clusters peuvent être :

- comprendre et décrire la structure des interactions entre les nœuds en analysant la topologie du graphe ;
- identifier certains nœuds particuliers tels que les nœuds centraux, les connecteurs, les hubs (voir les propriétés des nœuds dans la section 2.1.5 et le rôle des nœuds dans la section 2.1.6). Rappelons que le concept de connecteurs repose sur un découpage d'un graphe en clusters à l'aide d'un algorithme de détection de clusters (section 2.1.7).
- identifier les groupes de nœuds densément connectés.

Ainsi, une méthode de visualisation des clusters devrait pouvoir répondre aux critères suivants [49] :

- la taille de chaque cluster doit être proportionnelle au nombre de nœuds qu'il contient ;
- la représentation de chaque cluster doit être compacte ;
- limiter les chevauchements entre les clusters ;
- éviter les recouvrements de nœuds.

Dans cette section, nous présentons des techniques de visualisation des clusters dont l'une permet de mettre en évidence les connecteurs dans un graphe.

Cruz [30] propose une technique de visualisation des clusters pour mettre en évidence les nœuds connecteurs (voir les figures 3.11a et 3.11b). Cette technique est basée sur l'une des extensions de l'algorithme de MDS (SMACOF² [81]) dont les améliorations reposent principalement sur l'approche de minimisation de contraintes de Kamada et Kawai [83]. Pour mettre en évidence les nœuds connecteurs, Cruz propose d'ajouter d'abord un cluster au graphe déjà partitionné. Ce nouveau cluster va contenir tous les connecteurs. Ainsi, si le graphe d'origine comporte k clusters, on aura $k + 1$ clusters à visualiser. Il place ensuite les connecteurs dans un cercle (*Interaction Zone* sur la figure 3.11a) de rayon 0.5. Il place enfin les nœuds internes de chaque cluster dans un autre cercle (*Inner Nodes* sur la figure 3.11a) dont le centre est fonction de l'origine de la représentation et du centre de masse des nœuds connecteurs du cluster.

Held et al. [70] proposent quant à eux trois méthodes de visualisation des graphes dynamiques partitionnés. Rappelons que d'un graphe dynamique peut être défini comme un graphe qui change avec le temps, tel qu'un ajout ou une suppression de nœuds et d'arêtes. Pour un graphe dynamique partitionné, on peut également avoir l'ajout ou la

2. Scaling by MAjorizing a COmplicated Function

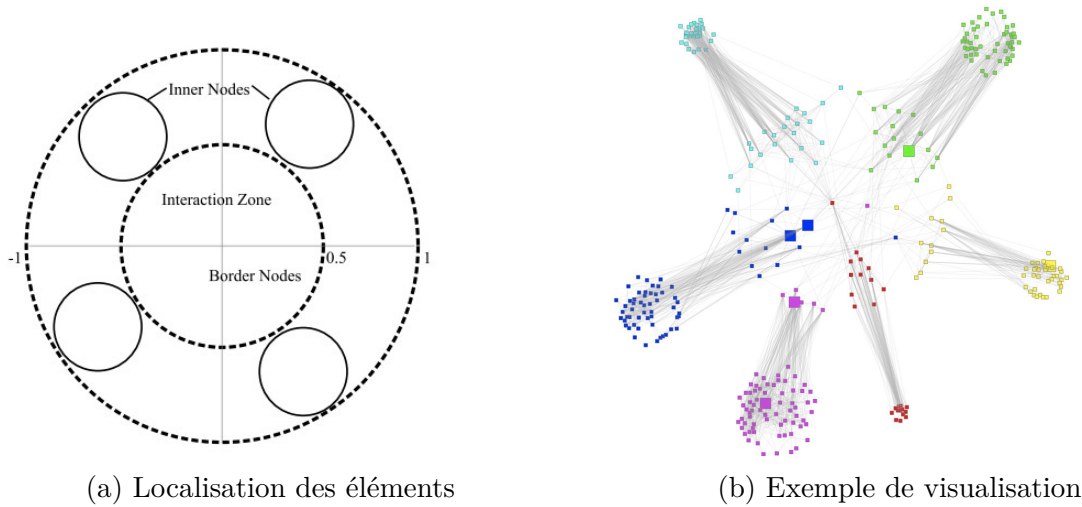


FIGURE 3.11 – Modèle de visualisation de clusters de Cruz [30] (Images extraites de Cruz [30]).

suppression de clusters. Alors que deux méthodes sont basées sur l'amélioration d'un algorithme de force, la troisième utilise une approche de type « diviser pour régner » :

1. Force-directed clustered layout.
2. ClusterNode layout.
3. Divide-and-Conquer layout.

L'objectif du Force-directed clustered layout (figure 3.12a) est de séparer visuellement les clusters les uns des autres. Held et al. [70] exploitent donc le poids des arêtes pour manipuler leur longueur à l'intérieur des clusters et entre les clusters. Ainsi, les arêtes entre deux nœuds d'un même cluster obtiennent un poids plus élevé, tandis que les arêtes entre deux nœuds de clusters différents obtiennent un poids inférieur. Par conséquent, les clusters sont bien séparés, car les arêtes interclusters sont plus longues.

Ensuite, contrairement au Force-directed clustered layout, ClusterNode layout (figure 3.12b) est basé sur l'algorithme de Frishman et Ayellet [49], un algorithme de représentation dynamique pour les graphes partitionnés. Held et al. [70] ajoutent d'abord un nœud factice invisible à chaque cluster, puis le connecte à tous les autres nœuds du cluster. Ils calculent ensuite la position des nœuds du graphe à partir de l'algorithme de SpringBox³. Sur la base du nœud factice, les nœuds d'un cluster sont proches les uns des autres en raison des forces d'attraction supplémentaires entre eux.

3. Un algorithme basé sur l'algorithme de force de Fruchterman et Reingold [50]

Enfin, la troisième méthode (figure 3.12c) combine le principe de « diviser pour régner » avec un algorithme de force. Le principe de « diviser pour régner » a été proposé par Wang et Miyamoto [149] pour représenter les graphes partitionnés. Held et al. [70] proposent donc de construire d’abord un méta graphe pondéré où un nœud représente un cluster et son poids représente le nombre de nœuds qui composent le cluster. Ils calculent ensuite la position des nœuds de chaque cluster séparément, puis la position des nœuds du méta graphe. Les nœuds de chaque cluster sont enfin positionnés relativement à la position du méta nœud représentant le cluster.

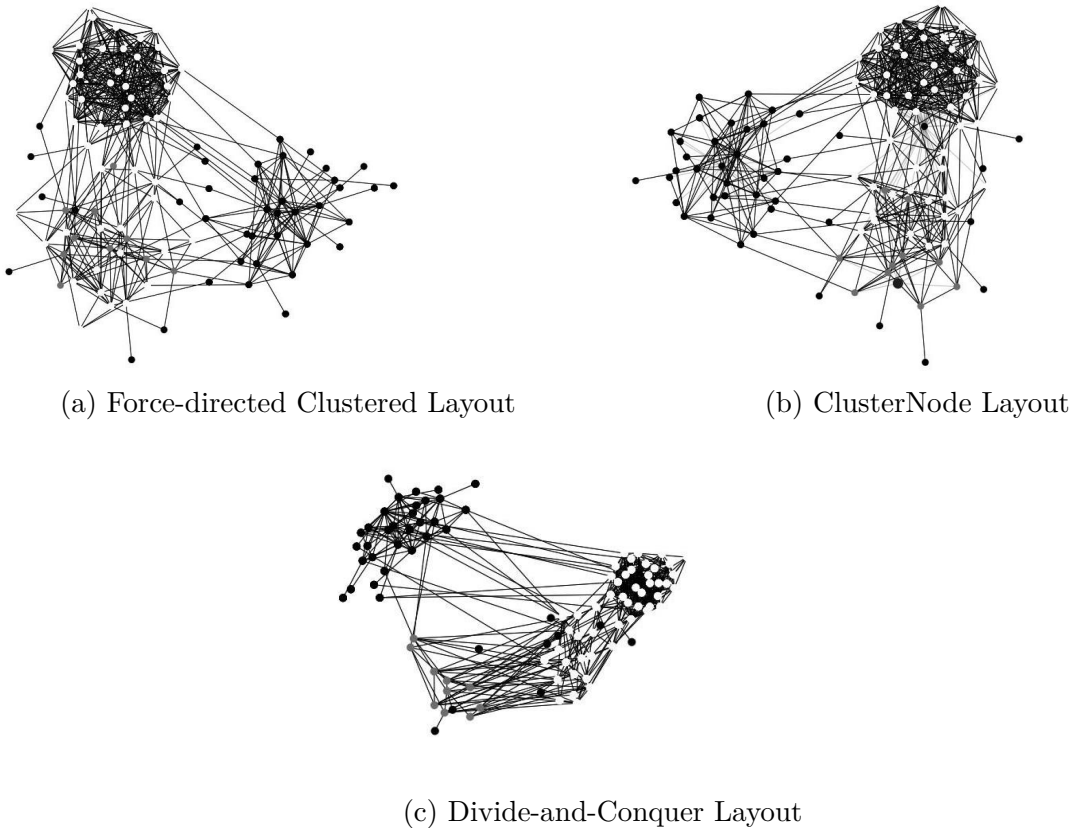


FIGURE 3.12 – Modèles de visualisation de clusters de Held et al. [70] (Images extraites de Held et al. [70]).

Pour évaluer la qualité de ces trois méthodes, Held et al. [70] proposent d’utiliser les critères suivants sur 6 graphes générés :

- le recouvrement de clusters : il décrit la surface utilisée par deux clusters en même temps ;
- la distance minimale des clusters : elle est la distance minimale entre deux clusters

du graphe. Si deux clusters se chevauchent, cette valeur est égale à 0 ;

- les arêtes de croisement de clusters : c'est le nombre d'arêtes intraclusters qui traversent d'autres clusters.

Selon les résultats de leur évaluation, l'approche *Divide-and-Conquer* est la plus adaptée des trois méthodes, car la position de chaque cluster est calculée indépendamment des autres. Ces résultats montrent que la méthode *Divide-and-Conquer* minimise le recouvrement de clusters et le nombre d'arêtes intraclusters traversant d'autres clusters.

En définitive, le modèle de Cruz [30] et les modèles de Held et al. [70] permettent de visualiser les graphes partitionnés. Le modèle de Cruz [30] fournit une excellente séparation visuelle des clusters et permet par la même occasion de mettre en évidence les nœuds connecteurs. Néanmoins, il y a des recouvrements de nœuds et d'arêtes au sein des clusters denses. Les résultats d'évaluation de Held et al. [70] ont quant à eux montré que l'approche *Divide-and-Conquer* est la mieux adaptée pour visualiser les graphes dynamiques partitionnés puisqu'elle sépare bien les clusters, comparativement aux approches *ClusterNode* et *Force-directed Clustered*. Cependant, les trois méthodes de Held et al. [70] ne mettent en évidence aucune caractéristique topologique des nœuds dans un graphe, comparativement au modèle de Cruz [30].

Le tableau 3.3 présente une comparaison des différents modèles de visualisation de clusters. Cette comparaison porte non seulement sur les critères évoqués dans l'introduction de cette section, mais aussi sur la mise en évidence des connecteurs. Dans l'ensemble, le modèle proposé par Cruz [30] est beaucoup plus adapté pour visualiser les graphes partitionnés, car elle remplit plusieurs critères, comparativement aux approches de Held et al. [70]. Toutefois, l'approche « Diviser pour régner » [70] semble également une bonne alternative de visualisation de graphes partitionnés.

Techniques	Taille des clusters proportionnelle au nombre de nœuds	Compacité des clusters	Limiter les chevauchements des clusters	Limiter le recouvrement des nœuds	Connecteurs
Modèle de Cruz [30]	✓	✓	✓	✗	✓
Force-directed Clustered layout [70]	✗	✗	✗	✗	✗
ClusterNode layout [70]	✗	✗	✓	✗	✗
Divide-and-Conquer layout [70]	✓	✓	✓	✗	✗

TABLE 3.3 – Tableau comparatif des différentes techniques de visualisation de clusters. Une coche (✓) d'un critère indique qu'il est respecté dans le rendu visuel de la technique et une croix (✗) indique le contraire.

3.4 Algorithmes de regroupement d'arêtes

Les arêtes sont une représentation pour mettre en évidence la connectivité entre les nœuds dans la visualisation de graphes. Cependant, lorsqu'un graphe contient plusieurs milliers d'arêtes, la visualisation présente souvent beaucoup d'occlusions dues aux nombreux croisements d'arêtes comme illustré dans la figure 3.13a. Par conséquent :

- les arêtes peuvent facilement submerger l'affichage ;
- les différentes structures présentes dans le graphe peuvent être obscures ou complètement cachées ;
- les connectivités importantes peuvent être facilement ignorées ;
- le graphe peut être presque impossible à analyser.

Plusieurs approches ont été proposées pour résoudre ce problème d'occlusion des arêtes qui est courant dans les graphes denses [74, 41]. Un état de l'art de quelques méthodes de regroupement d'arêtes a été présenté par Lhuillier et al. [106].

Holten propose d'abord l'utilisation d'un regroupement hiérarchique d'arêtes (HEB⁴) pour les graphes hiérarchiques [73]. Cette technique est basée sur l'idée de combiner des arêtes proches ensemble, de la même manière que nous pouvons combiner les fils électriques

4. Hierarchical Edge Bundling

ou d'autres câbles de réseau en faisceaux et les séparer à nouveau à la fin. Cependant, cette méthode est moins adaptée aux graphes non hiérarchiques. Holten et Van Wijk étendent ensuite HEB en proposant la méthode Force-Directed Edge Bundling (FDEB) (voir la figure 3.13b) [74]. Il s'agit d'une méthode de regroupement d'arêtes qui utilise une approche d'auto-organisation et modélise les arêtes comme des ressorts flexibles qui peuvent s'attirer les uns les autres. Contrairement à l'approche HEB, aucune hiérarchie n'est utilisée et le résultat présente une réduction significative de l'encombrement visuel.

Gansner et al. [54] tirent parti du clustering agglomératif [8] pour améliorer les effets du regroupement des arêtes pour les grands graphes. Le clustering agglomératif est une méthode de classification ascendante qui commence par des singletons et fusionne des clusters au fur et à mesure que l'on monte dans la hiérarchie. Ersoy et al. [41] proposent quant à eux SBEB⁵ un algorithme de regroupement d'arêtes basé sur le clustering agglomératif et sur des méthodes de traitement d'images (voir la figure 3.13c). Le clustering fournit d'abord des groupes d'arêtes ayant une forte similarité géométrique. Une enveloppe fine est ensuite calculée pour chaque groupe d'arêtes. Le squelette de l'enveloppe est enfin utilisé pour orienter le regroupement d'arêtes. Un squelette est une courbe localement centrée par rapport à la forme de l'enveloppe.

Hurter et al. [80] proposent une autre méthode de regroupement d'arêtes qui repose également sur des méthodes de traitement d'images (KDEEB⁶) (voir la figure 3.13d). Elle consiste à discrétiser d'abord chaque arête en un ensemble de points, puis à transformer une représentation de graphe en une carte de densité en utilisant une estimation de densité par noyau. Cette carte est ensuite utilisée pour créer des regroupements d'arêtes en déplaçant itérativement des points des arêtes convoluées vers les maxima locaux de la carte.

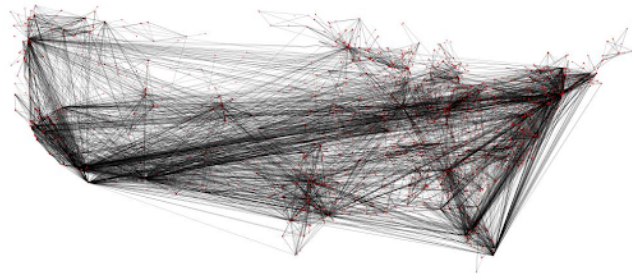
Ces algorithmes de regroupement d'arêtes peuvent générer des résultats visuels permettant de mettre en évidence des structures présentes dans un graphe. Cependant, ils peuvent prendre du temps pour générer des regroupements d'arêtes pour de grands graphes [164]. Par conséquent, ils peuvent être moins adaptés aux grands graphes, et en particulier à certaines applications interactives [156]. Notons qu'une application interactive peut être définie comme une application qui répond immédiatement aux entrées d'un utilisateur.

Ainsi, pour résoudre le problème de temps de calcul, Zhu et al. [164] proposent un

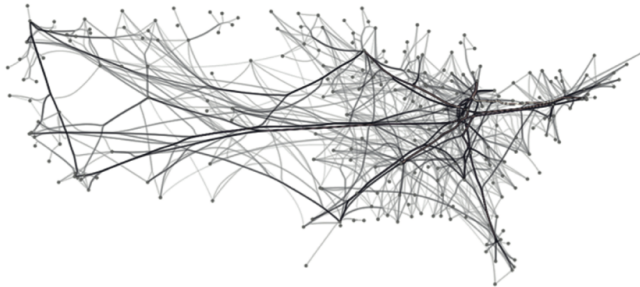
5. Skeleton-Based Edge Bundling

6. Kernel Density Estimation Edge Bundling

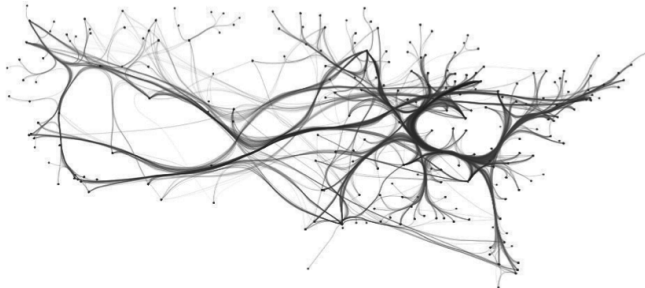
FDEB parallélisé (GPUFDEB) qui améliore le FDEB [74] et qui réalise un partitionnement équilibré des données et des calculs pour s'adapter au calcul sur GPU. Wu et al. [156] proposent quant à eux d'utiliser des textures GPU pour encoder un graphe. Ils utilisent ensuite le GPU pour effectuer un regroupement d'arêtes à partir du modèle FDEB [74].



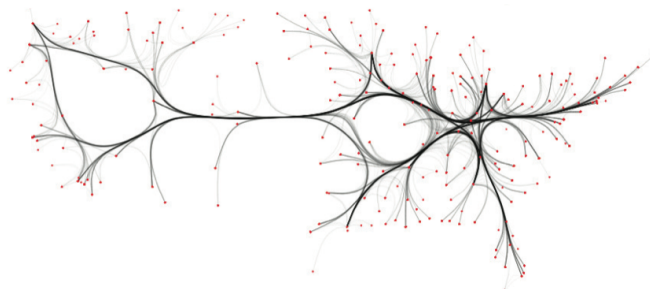
(a) Graphe original avec arêtes non regroupées



(b) Force-Directed Edge Bundling (FDEB)



(c) Skeleton-Based Edge Bundling (SBEB)



(d) Kernel Density Estimation Edge Bundling (KDEEB)

FIGURE 3.13 – Algorithmes de regroupement d’arêtes appliqués sur (a) le graphe d’US airlines (235 nœuds et 2099 arêtes) (Images extraites de Hurter et al. [80]).

3.5 Conclusion

Nombreux sont les domaines qui s'intéressent à l'analyse des données représentées sous forme de graphes. L'un des problèmes courants dans ces tâches d'analyse est la manière de représenter correctement ces graphes afin de faciliter leur exploration. Pour cela, plusieurs algorithmes ont été proposés. Certains s'intéressent à la spatialisation des nœuds, tandis que d'autres s'intéressent à la représentation (le regroupement) des arêtes. Les algorithmes de spatialisation sont basés sur différents paradigmes (la physique et l'analyse des données). Certains algorithmes s'adaptent mieux aux graphes de taille relativement réduite, tandis que d'autres sont plus pratiques pour des graphes reflétant les données du monde réel. Cependant, les algorithmes basés sur la réduction de dimension semblent être plus pratiques pour mettre en évidence les structures locales d'un graphe. Les algorithmes de regroupement d'arêtes permettent quant à eux de réduire les occlusions dues à la densité des arêtes afin de bien identifier par exemple la connectivité entre les groupes de nœuds.

En ce qui concerne la mise en évidence des propriétés topologiques des nœuds dans un graphe (section 3.2), la méthode des coordonnées parallèles 3D et les approches orbitale et hiérarchique permettent de comparer plusieurs mesures de centralité. Les méthodes radiales permettent quant à elles de comparer une mesure de centralité et d'observer la structure du graphe. Cependant, seule la méthode des coordonnées parallèles 3D peut être pratique pour visualiser un grand graphe. Par ailleurs, un analyste peut s'intéresser à plusieurs caractéristiques des nœuds à la fois. Ainsi, il serait utile de réussir à mettre en évidence une des caractéristiques topologiques en exploitant la spatialisation des nœuds afin d'utiliser des variables rétinienne (par exemple la couleur, la forme et la taille) pour d'autres caractéristiques (topologique ou métier).

La visualisation des clusters permet quant à elle d'identifier les clusters, d'observer la connectivité entre les clusters, d'observer la structure interne d'un cluster et le rôle des nœuds particuliers (hubs et connecteurs). Les méthodes de visualisation de clusters présentées dans la section 3.3 permettent d'identifier visuellement les clusters. Celle proposée par Cruz [30] permet en plus de visualiser les connecteurs. Cependant, toutes ces méthodes de visualisation de clusters sont peu pratiques pour comparer ou mettre en évidence d'autres caractéristiques des nœuds dans un cluster.

En définitive, un algorithme de spatialisation fournit une vision d'ensemble d'un graphe permettant d'identifier les zones denses du graphe, les nœuds qui ne sont pas en zone dense et chercher à comprendre leurs propriétés. Pour un graphe contenant plusieurs milliers

de nœuds et d'arêtes, cela nécessite un découpage en clusters. La visualisation de ces clusters permet d'identifier et de comprendre la connectivité entre clusters, d'identifier des nœuds qui ont des propriétés particulières (les connecteurs et les hubs) et de comparer des nœuds par rapport à certaines de leurs caractéristiques topologiques. Parmi les méthodes étudiées, certaines permettent de comparer la centralité des nœuds, tandis que d'autres mettent en évidence le rôle de certains nœuds quand il s'agit d'un graphe partitionné. Cependant, elles manquent d'espace pour afficher des graphes contenant plusieurs milliers de nœuds. On note par conséquent une présence d'occlusion de nœuds et d'arêtes et une difficulté à observer certaines arêtes. Ainsi, il s'avère utile de trouver des métaphores de visualisation qui pourraient aider à résoudre ces problèmes.

DEUXIÈME PARTIE

Contributions

Conformément à l'analyse de l'état de l'art qui a été présentée, nos contributions vont donc viser à :

- donner une vision d'ensemble d'un graphe ;
- permettre d'identifier les zones denses d'un graphe ;
- permettre d'identifier facilement des nœuds par rapport à plusieurs de leurs propriétés topologiques ou métiers ;
- permettre d'identifier les nœuds qui ne sont pas en zone dense et comprendre leurs propriétés ;
- permettre d'identifier les interconnexions entre zones denses ;
- permettre d'identifier des nœuds qui ont des propriétés intéressantes (exemple : forte centralité, degré et connecteurs).

Par conséquent, nous avons exploité la 3D pour proposer une visualisation exocentrique de graphes. Elle consiste à projeter des représentations radiales 2D sur des surfaces 3D. Cette métaphore de visualisation sera présentée en détail dans le chapitre 4

Afin de réduire le recouvrement des arêtes, nous avons ensuite exploité les stratégies de regroupement des arêtes proposées par Holten et Van Wijk [74] et par Hurter et al [80].

Nous avons également proposé une visualisation égocentrique de graphes qui consiste à placer, par exemple, un analyste de graphes au centre du rendu visuel d'un graphe. Cette méthode de visualisation sera détaillée dans le chapitre 5.

Nous avons aussi mis en place certaines stratégies d'interactions nécessaires à l'analyse d'un graphe tels que le zoom, la sélection d'un nœud et l'affichage de certains attributs lors d'un survol d'un nœud.

Nous avons enfin conduit deux expérimentations afin d'évaluer l'efficacité de notre proposition. La première a pour but de comparer l'efficacité de notre proposition 3D à celle des représentations 2D. La seconde vise à comparer l'efficacité de notre proposition en environnement virtuel par rapport à la 3D et à la 2D desktop. Les résultats de la première évaluation ont permis de montrer que la 3D est plus efficace que la 2D pour les tâches liées aux nœuds centraux et aux nœuds en périphérie d'un graphe. Les résultats de la seconde évaluation ont, quant à eux, permis de montrer que la 3D immersive est bien plus efficace que la 2D pour les tâches liées aux nœuds centraux et aux zones denses en passant d'un graphe de 250 nœuds à un graphe de 500 nœuds.

Nous présentons d'abord notre proposition radiale 3D dans le chapitre 4. Nous présentons ensuite notre métaphore égocentrique dans le chapitre 5 et enfin une application de nos propositions à l'analyse des réseaux sociaux dans le chapitre 6.

VISUALISATION EXOCENTRIQUE DE GRAPHES : MISE EN ÉVIDENCE DES PROPRIÉTÉS TOPOLOGIQUES

Les deux principaux besoins d'un analyste de graphes sont d'une part d'explorer un graphe pour lui donner l'intuition des phénomènes à analyser et d'autre part de mettre en avant des résultats. Pour cela, il pourrait avoir besoin d'une vision d'ensemble du graphe, de pouvoir se focaliser sur une zone (zone dense, par exemple) et d'identifier facilement des nœuds par rapport à plusieurs de leurs caractéristiques topologiques ou métiers.

Les approches classiques de représentation de graphes en 2D sont limitées, même si celles qui associent la centralité des nœuds à une position spécifique de l'espace permettent facilement de comparer la centralité des nœuds dans un graphe [16, 123]. Cependant, ces dernières manquent d'espace d'affichage pour les grands graphes. On note par conséquent l'existence d'occlusions et une difficulté à distinguer certaines connectivités (arêtes). Un analyste pourrait alors en effet rencontrer des difficultés pour pouvoir observer dans de bonnes conditions :

- la connectivité d'un nœud avec ses voisins ;
- les interconnexions entre des groupes de nœuds ;
- le rôle de certains nœuds (les connecteurs, par exemple) dans un groupe quand il s'agit d'un graphe partitionné.

L'intérêt de visualiser un graphe en transmettant au mieux des informations topologiques sur les nœuds en utilisant leur position dans l'espace, c'est de pouvoir comparer facilement ces informations grâce à la représentation visuelle. Les informations topologiques concernent les propriétés topologiques des nœuds telles que les mesures de centralité (présentées dans la section 2.1.5 et les rôles des nœuds (les connecteurs, les hubs) (présentés dans la section 2.1.6) lorsqu'il s'agit d'un graphe découpé en clusters à l'aide d'un algorithme de détection de clusters. Rappelons qu'une mesure de centralité caracté-

rise la position topologique des nœuds dans un graphe. Dans le cadre des travaux de cette thèse, nous nous intéressons particulièrement à la mise en évidence de la centralité des nœuds et des connecteurs. La mise en évidence des connecteurs permet de voir un cluster et les interconnexions avec les autres sans avoir à les afficher.

De ce fait, nous proposons une méthode de visualisation exocentrique qui consiste à transmettre des informations topologiques des nœuds dans un graphe et à faciliter l'exploration :

- des connectivités entre les nœuds ;
- les interconnexions entre des groupes de nœuds ;
- d'un groupe de nœuds (dans un graphe partitionné) ;
- des connecteurs dans un groupe (dans un graphe partitionné).

Nous rappelons d'abord certains travaux relatifs à la mise en évidence de la centralité des nœuds (présentés dans l'état de l'art, section 3.2). Nous proposons ensuite une visualisation radiale 3D et notre stratégie de regroupement d'arêtes. Nous présentons ensuite notre méthode permettant de mettre en évidence les connecteurs d'un groupe, puis les résultats des expérimentations que nous avons menées pour évaluer l'intérêt de notre visualisation radiale 3D.

4.1 Travaux relatifs à la visualisation de graphes mettant en évidence la centralité

Brandes et al. [13] et Brandes et Pich [16] (voir figure 3.6, p.67) proposent des approches radiales 2D qui permettent de mettre en évidence la centralité des nœuds dans un graphe. Dans ces méthodes, la valeur d'une mesure de centralité de chaque nœud est représentée par le rayon d'un cercle sur lequel doit être placé ce nœud. Cette valeur est normalisée et les nœuds les moins centraux sont placés à la périphérie du graphe sur un cercle de rayon égal à 1 tandis que le nœud le plus central est associé à un cercle de rayon égal à 0 et se retrouve donc au centre du graphe. Ainsi, la distance au centre encode la mesure de centralité, qui est un résultat d'une étape d'analyse précédente. Rappelons que le rayon r_i du cercle associé à un nœud i est calculé comme suit :

$$r_i = \frac{\text{diam}(G)}{2} * \left(1 - \frac{C_i - \min_{u \in V} C_u}{\max_{u \in V} C_u - \min_{u \in V} C_u + C(G)}\right) \quad (4.1)$$

où

- $diam(G)$ est le diamètre d'un graphe G . Rappelons également que le diamètre d'un graphe est la plus grande distance entre deux nœuds (section 2.1.4). $\frac{diam(G)}{2}$ sert à ramener tous les rayons à la même échelle ;
- C_i est la valeur d'une mesure de centralité d'un nœud i ;
- $\min_{u \in V} C_u$ et $\max_{u \in V} C_u$ sont respectivement la valeur minimale et la valeur maximale d'une mesure de centralité des nœuds dans le graphe G ;
- $C(G)$ est un petit décalage supérieur à zéro s'il existe plusieurs nœuds avec une centralité maximale.

Brandes et Pich [16] proposent également de mettre l'emphase sur le centre du graphe ou bien sur sa périphérie afin de mieux visualiser les nœuds concernés en augmentant la distance entre eux. Par contre, les nœuds non concernés par la mise en emphase seront plus proches. Ainsi, l'emphase sur le centre consiste à augmenter la distance entre les nœuds qui sont proches du centre du graphe, tandis que l'emphase sur la périphérie augmente la distance entre les nœuds de la périphérie du graphe.

Raj et Whitaker [123] proposent quant à eux une visualisation radiale anisotrope (voir figure 3.7, p.68) permettant de mettre en évidence la centralité d'intermédiarité des nœuds. Au lieu des cercles concentriques, ils proposent d'utiliser plutôt des courbes d'iso-centralité qui offriraient plus de flexibilité pour préserver la structure d'un graphe. Cependant, il serait alors plus complexe d'identifier les nœuds qui ont une même valeur de centralité, contrairement à l'approche de Brandes et Pich où ces nœuds sont sur un même cercle. En effet, on serait obligé de suivre une courbe irrégulière pour pouvoir identifier par exemple deux nœuds qui ont une même valeur de centralité d'intermédiarité, contrairement à l'approche de Brandes et Pich [16] où ces nœuds peuvent être facilement identifiés sur un cercle. La méthode radiale de Brandes et Pich favorise donc une comparaison rapide de la centralité des nœuds.

Toutefois, pour un graphe qui contient par exemple plusieurs centaines de nœuds, les approches de Brandes et Pich et de Raj et Whitaker deviennent peu pratiques, car la résolution des dispositifs d'affichage rend les détails difficiles à discerner (voir figure 4.1). Ainsi, il est possible d'adapter certaines méthodes 2D à la 3D [19] afin de disposer de plus d'espace d'affichage pour pouvoir mieux distribuer les nœuds. Par conséquent, l'ajout d'une troisième dimension permet de réduire les occlusions et d'améliorer la lisibilité d'un graphe.

Par ailleurs, les approches de Brandes et Pich [16] et de Raj et Whitaker [123] n'ont pas traité d'autres propriétés des nœuds, ce qui est un problème lorsqu'un analyste de

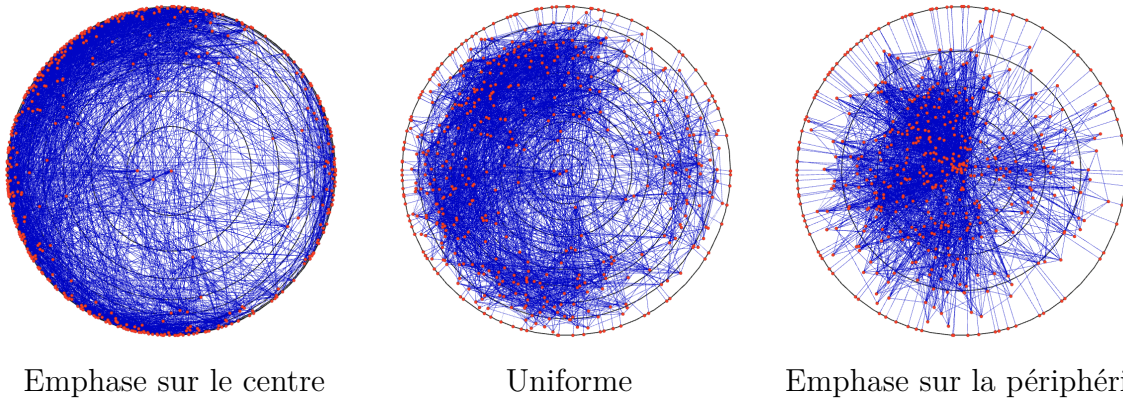


FIGURE 4.1 – Visualisation radiale 2D : mise en évidence de la centralité d’intermédiation des nœuds d’un graphe (500 nœuds et 3294 arêtes).

graphes veut par exemple observer simultanément la centralité des nœuds et autre une autre caractéristique (exemple : les nœuds connecteurs). De ce fait, nous proposons une visualisation radiale 3D qui permet de mettre en évidence la centralité des nœuds, mais aussi d’autres propriétés topologiques des nœuds, telles que les connecteurs et le coefficient de clustering.

4.2 Visualisation radiale 3D

4.2.1 Concept

Pour étendre la méthode radiale de Brandes et Pich [16] au domaine 3D, nous proposons de projeter les représentations 2D, selon l’axe vertical, sur différentes surfaces 3D. Ce type de projection garantit que si on regarde le résultat en vue de dessus, on retrouve une vue parfaitement équivalente à la représentation 2D initiale. Il est donc possible à tout moment de se ramener à l’équivalent de la vue 2D en prenant un point de vue adapté sur la représentation 3D.

La figure 4.2 permet d’illustrer notre concept de projection de la vue uniforme 2D sur une demi-sphère, un cône et une portion de tore, respectivement. La projection d’une vue 2D sur une demi-sphère étale moins les nœuds centraux que les nœuds de la périphérie. Ensuite, la projection sur une portion de tore étale plus les nœuds centraux que les nœuds de la périphérie. Enfin, la projection sur un cône est une version intermédiaire des projections sur une demi-sphère et sur une portion de tore, car elle distribue les nœuds de façon uniforme sur la surface.

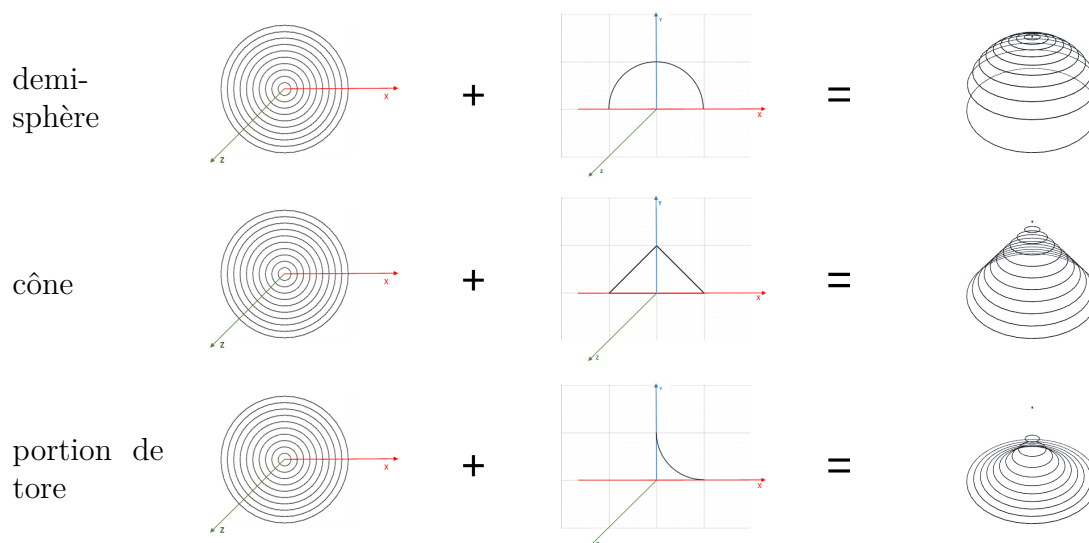


FIGURE 4.2 – Illustration de la projection uniforme sur une demi-sphère, un cône et une portion de tore, respectivement.

4.2.2 Projection de la vue uniforme

Nous allons analyser dans cette section les propriétés des projections de la vue 2D uniforme sur les 3 types de surfaces (voir figure 4.3)

Lorsqu'on change l'angle de vue de la projection sphérique de la représentation uniforme, on perçoit beaucoup mieux les écarts de positions des nœuds en périphérie. Ainsi, une élévation sphérique de la vue 2D uniforme peut fournir à la fois les avantages de la vue 2D uniforme et de l'emphase sur la périphérie. Par contre, elle n'améliore pas beaucoup la façon de dissocier visuellement les nœuds centraux les uns des autres (voir figure 4.4). La projection torique permet de mieux distinguer les nœuds centraux. De plus, on a une excellente visibilité des arêtes qui partent du centre, contrairement à la projection sphérique. La projection conique regroupe quant à elle à la fois les avantages de la vue 2D uniforme, de l'emphase sur le centre et de l'emphase sur la périphérie, car elle répartit les nœuds uniformément sur la surface.

Ainsi, une élévation de la vue 2D uniforme sur une demi-sphère, sur un cône et sur une portion de tore permet d'obtenir, en plus, les avantages des vues 2D mettant l'emphase sur le centre et sur la périphérie.

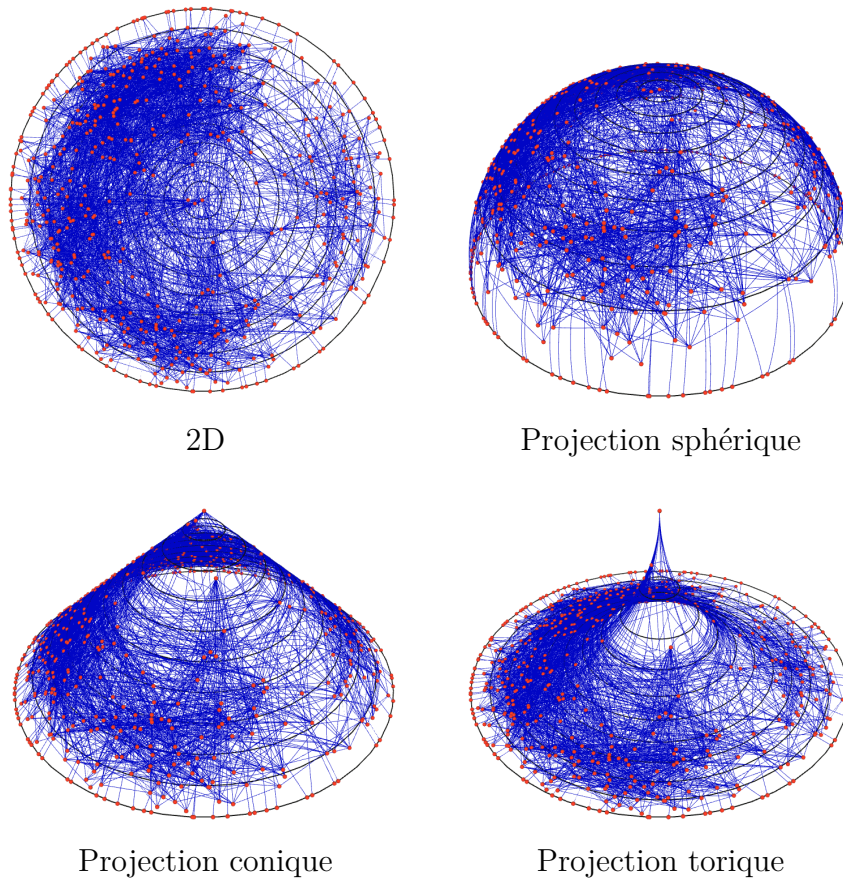


FIGURE 4.3 – Vue uniforme : mise en évidence de la centralité d’intermédiarité des nœuds d’un graphe (500 nœuds et 3294 arêtes).

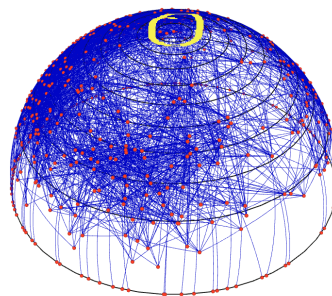


FIGURE 4.4 – La projection sphérique de la vue uniforme 2D améliore peu la façon de dissocier visuellement les nœuds centraux les uns des autres.

4.2.3 Projection de l'emphase sur le centre

La figure 4.5 montre le résultat des projections de la vue 2D qui met l'emphase sur le centre sur les 3 types de surfaces. L'élévation sphérique de l'emphase sur le centre répartit uniformément les nœuds intermédiaires sur la partie inférieure de la demi-sphère. Comparativement à la représentation 2D, la combinaison de l'élévation sphérique avec l'emphase centrale permet d'atténuer l'écrasement des nœuds intermédiaires et des nœuds en périphérie. Ainsi, grâce à la 3D, nous réduisons l'écrasement de la vue 2D qui met l'emphase sur le centre.

En combinant la projection conique avec l'emphase sur le centre, nous réduisons seulement un peu l'écrasement de la vue radiale 2D qui met l'emphase sur le centre, comparativement à l'approche sphérique où la réduction de l'écrasement est plus importante. La projection conique de l'emphase sur le centre fournit également une bonne visibilité des arêtes qui partent du centre, contrairement à la projection sphérique.

La combinaison de la projection torique avec l'emphase sur le centre fournit quant à elle un double étalement des nœuds centraux grâce, respectivement, aux étalements 2D et 3D. Par contre, elle écrase tous les nœuds intermédiaires et les nœuds en périphérie. Elle fournit par ailleurs une meilleure visibilité des arêtes qui partent du centre, comparativement aux projections conique et sphérique de l'emphase sur le centre.

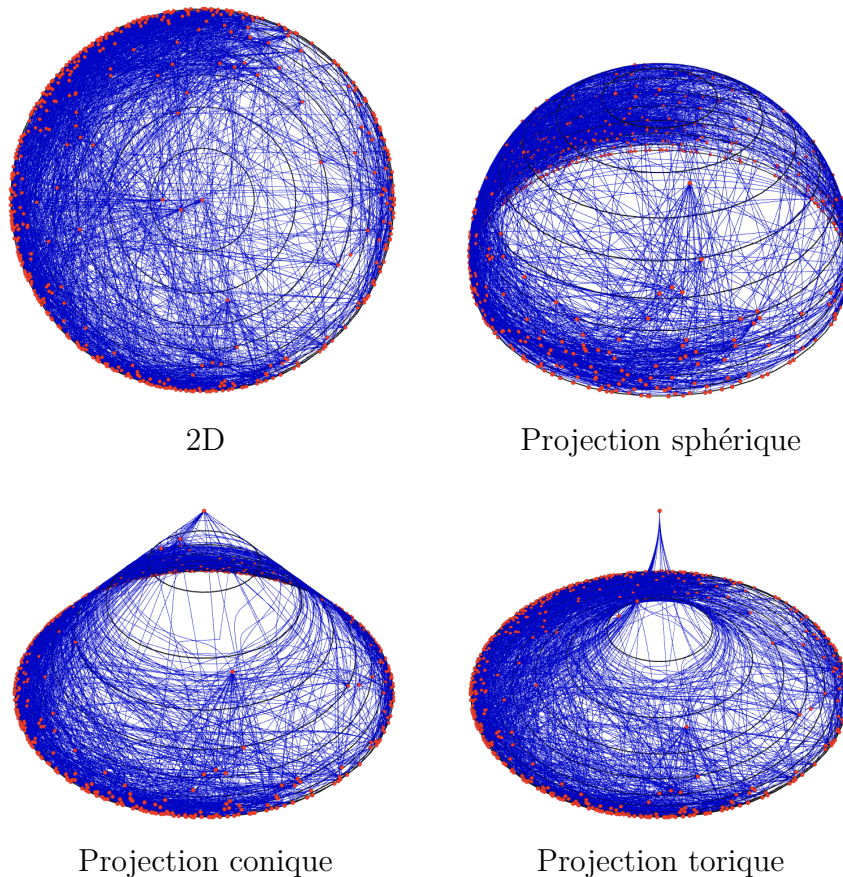


FIGURE 4.5 – Emphase sur le centre : mise en évidence de la centralité d’intermédierité des nœuds d’un graphe (500 nœuds et 3294 arêtes).

4.2.4 Projection de l’emphase sur la périphérie

La figure 4.6 montre que la combinaison d’une emphase sur la périphérie avec les élévations sphérique, conique et torique regroupent les nœuds centraux sur la partie supérieure de la surface et étalent les nœuds en périphérie.

La projection sphérique de l’emphase sur la périphérie fournit un double étalement des nœuds en périphérie grâce aux étalements respectifs 2D et 3D. Par contre, nous observons une perte de visibilité du centre à cause du type de surface (voir la figure 4.7). La combinaison de la projection torique avec l’emphase sur la périphérie met quant à elle en évidence la périphérie et le centre grâce respectivement à la 2D (emphase sur la périphérie) et à la 3D (la projection). La projection conique de l’emphase sur la périphérie met également en évidence la périphérie et le centre grâce respectivement à la 2D et à la 3D, mais de façon atténuée.

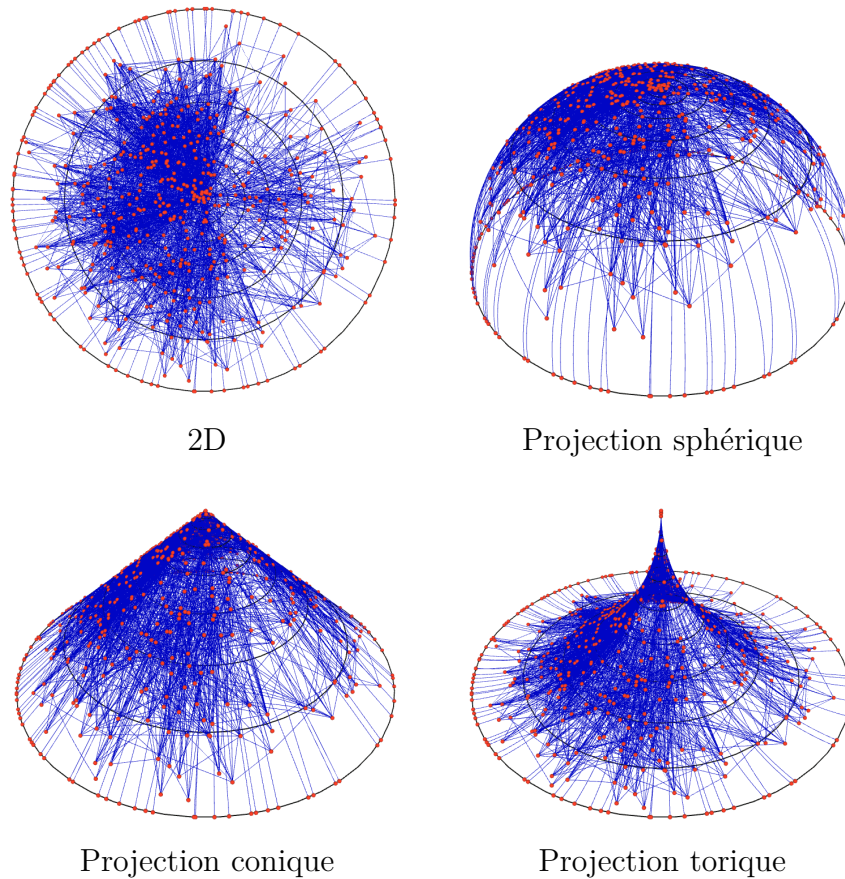


FIGURE 4.6 – Emphase sur la périphérie : mise en évidence de la centralité d’intermédiation des nœuds d’un graphe (500 nœuds et 3294 arêtes).

Ainsi, l’élévation sphérique de l’emphase sur la périphérie accentue la visibilité de la périphérie, alors que l’élévation torique fournit les mêmes avantages que l’emphase sur le centre et l’emphase sur la périphérie grâce respectivement à la 2D et à la 3D. L’élévation conique de l’emphase sur la périphérie fournit également les mêmes avantages que l’emphase sur la périphérie et l’emphase sur le centre, mais de façon modérée par rapport aux approches sphérique et torique.

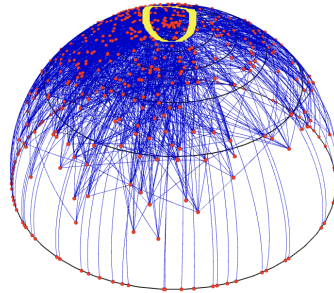


FIGURE 4.7 – La projection sphérique de l’emphase sur la périphérie entraîne une perte de visibilité sur le centre, contrairement aux projections conique et torique.

Le tableau 4.1 récapitule les propriétés mises en évidence de chacune des approches présentées et la figure 4.8 illustre la synthèse des différentes projections. La projection conique de la vue uniforme 2D met en évidence le centre et la périphérie. La projection sphérique de l’emphase sur la périphérie accentue la périphérie et la combinaison de la projection torique avec l’emphase sur le centre accentue quant à elle le centre.

	Visibilité du centre	Visibilité de la périphérie	Répartition uniforme des nœuds
Projection sphérique uniforme	Faible	Moyenne	Faible
Projection sphérique de l’emphase sur le centre	Normale	Normale	Faible
Projection sphérique de l’emphase sur la périphérie	Faible	Forte	Faible
Projection conique uniforme	Moyenne	Normale	Forte
Projection conique de l’emphase sur le centre	Moyenne	Normale	Forte
Projection conique de l’emphase sur la périphérie	Moyenne	Normale	Normale
Projection torique uniforme	Moyenne	Normale	Faible
Projection torique de l’emphase sur le centre	Forte	Faible	Faible
Projection torique de l’emphase sur la périphérie	Moyenne	Normale	Faible

TABLE 4.1 – Tableau récapitulatif des différentes approches et les propriétés mises en évidences.

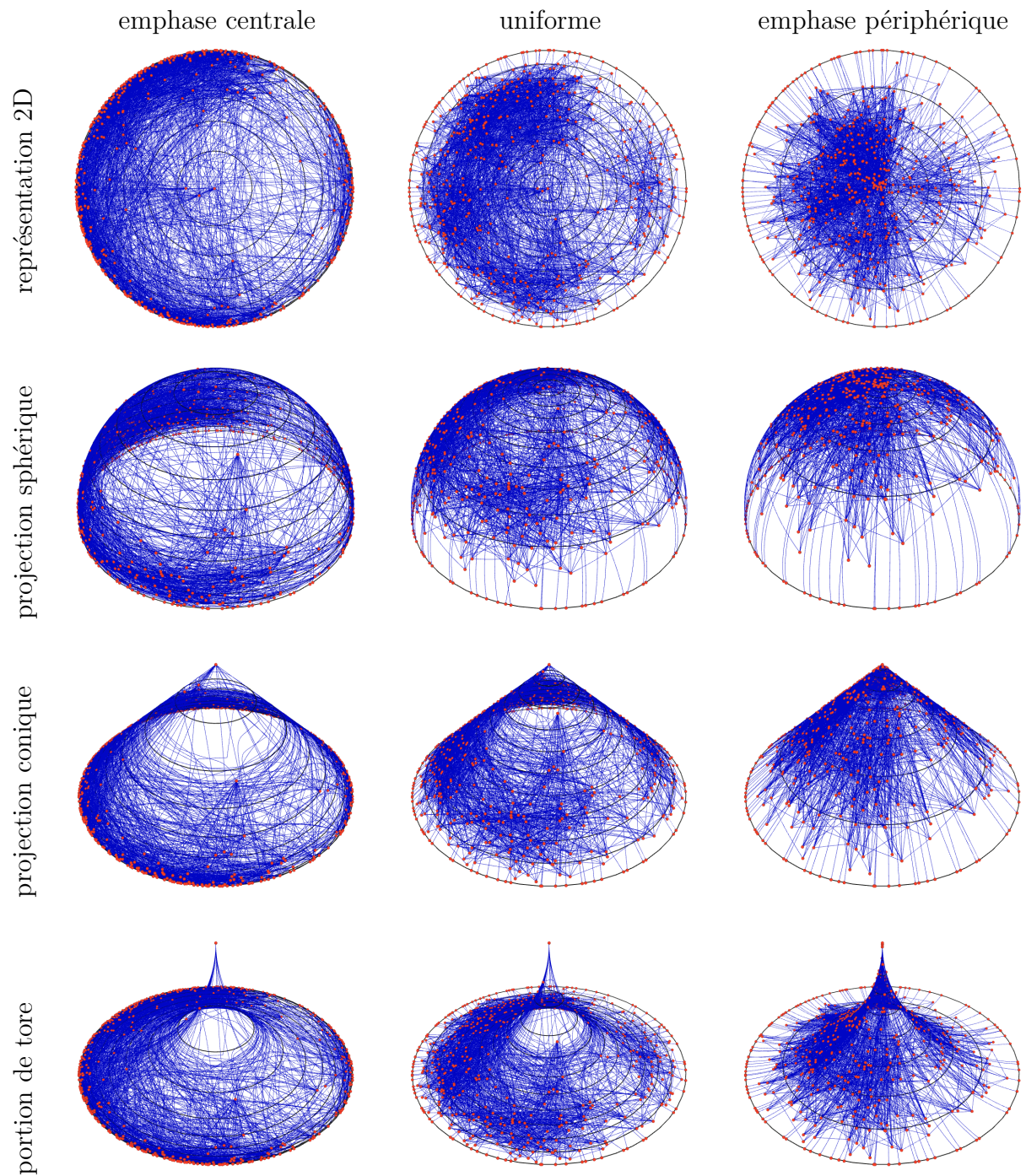


FIGURE 4.8 – Mise en évidence de la centralité d’intermédiarité des nœuds d’un graphe (500 nœuds et 3294 arêtes)

Par ailleurs, le croisement d’arêtes en 3D ne pose pas de problèmes, contrairement au croisement d’arêtes en 2D [141] car on peut changer de point de vue pour mieux observer

certaines parties d'un graphe. Par contre, l'encombrement visuel peut affecter l'analyse du graphe si ce dernier contient plusieurs milliers d'arêtes. Il est donc utile de regrouper les arêtes pour simplifier la lisibilité du graphe.

4.3 Regroupement d'arêtes

Comme évoqué précédemment dans la section 3.4, lorsqu'un graphe contient plusieurs milliers d'arêtes, la visualisation présente souvent des occlusions dues à la densité de ces arêtes (voir les images de gauche sur la figure 4.10). Par conséquent, le graphe peut être presque impossible à analyser.

Nous avons donc proposé d'utiliser l'algorithme de regroupement d'arêtes KDEEB (présenté dans la section 3.4) proposé par Hurter et al. [80] pour réduire le recouvrement d'arêtes (voir les images de droite sur la figure 4.10). Le regroupement d'arêtes permet ainsi de montrer la connectivité entre des groupes d'un graphe. Cependant, il fait perdre complètement la visibilité sur la connectivité d'un nœud.

En effet, un analyste pourrait s'intéresser à observer à la fois comment un nœud est connecté et la connectivité entre des groupes de nœuds. Pour cela, nous avons choisi de combiner le regroupement d'arêtes avec les arêtes détaillées en exploitant leur opacité. Nous disposons de deux curseurs dont le premier permet de gérer l'opacité des arêtes regroupées et le second celle des arêtes détaillées. La valeur de chaque curseur varie entre 0 (les arêtes sont totalement transparentes) et 1 (les arêtes sont totalement opaques). Ainsi, on peut faire varier cette valeur pour passer d'une vue détaillée des arêtes à une vue regroupée et inversement. La figure 4.9 illustre trois cas de variations de l'opacité des arêtes.

L'observation d'un groupe dans un graphe partitionné implique la compréhension des connectivités entre ce groupe et les autres, mais aussi le rôle que jouent certains nœuds dans ce groupe. Dans la section suivante, nous présentons une nouvelle approche permettant de mettre en évidence les connecteurs d'un groupe dans un graphe partitionné.

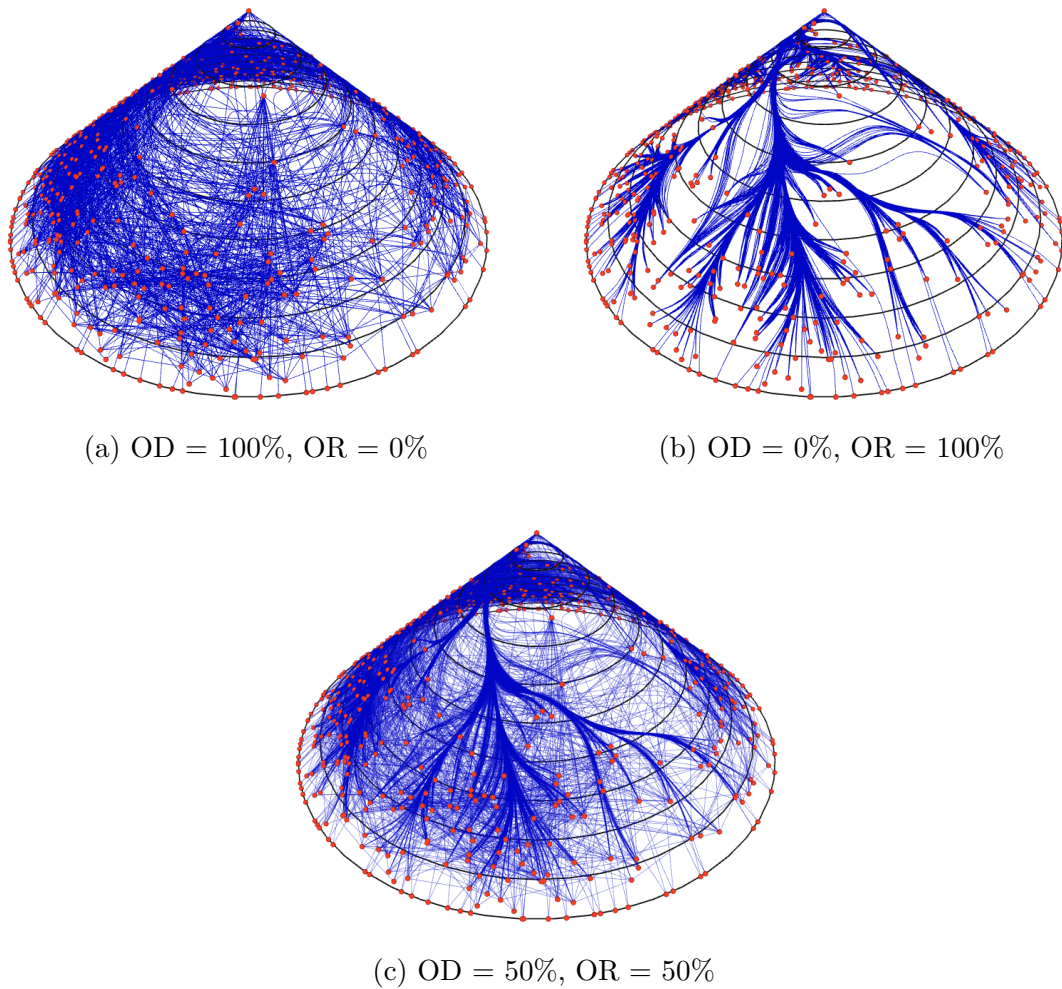


FIGURE 4.9 – Variation de l'opacité des arêtes à différents niveaux : (a) permet de voir la connectivité des nœuds, (b) permet de voir les connectivités entre groupes de nœuds et (c) est une configuration intermédiaire. **OD** = opacité des arêtes détaillées, **OR** = opacité des arêtes regroupées.

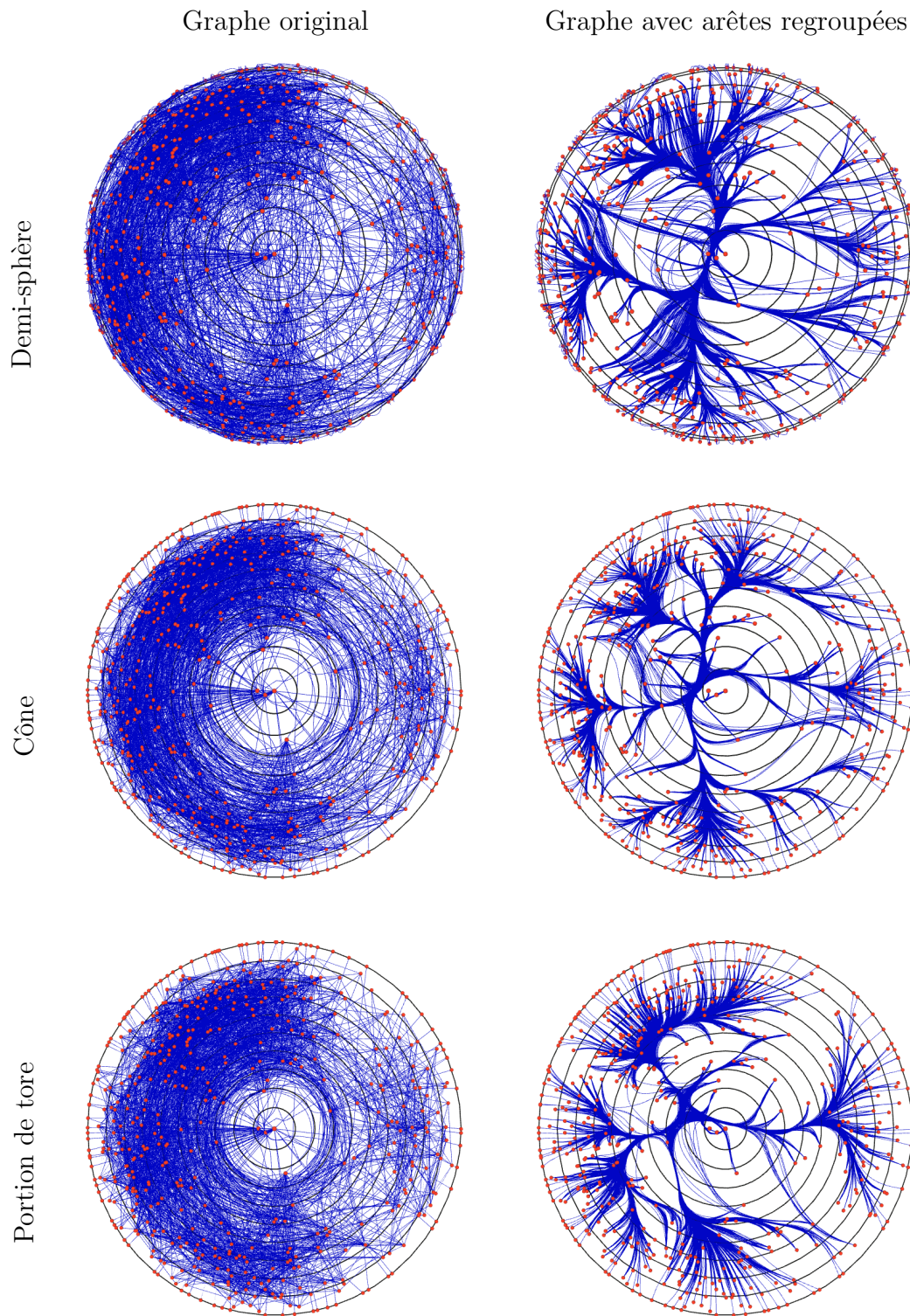


FIGURE 4.10 – Illustration synthétique d’une vue de dessus d’un graphe (500 nœuds et 3294 arêtes) montrant les arêtes en détail (images de gauche) et le regroupement des arêtes (images de droite) en utilisant KDEEB [80].

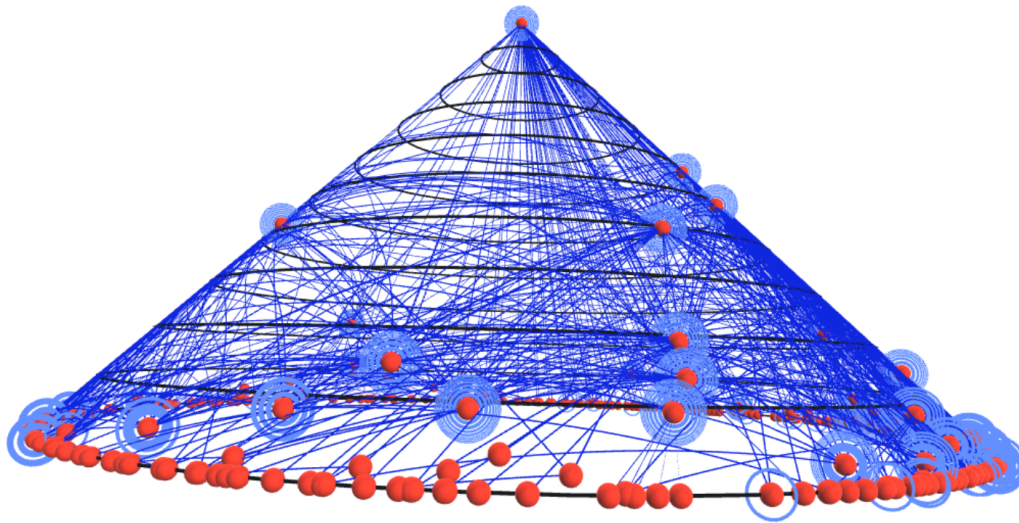
4.4 Mise en évidence des connecteurs

La structure de clusters ou communautés (dans le domaine de l'analyse des réseaux sociaux) a apporté des avancées significatives dans la compréhension des graphes représentant les données du monde réel [46]. Rappelons qu'un cluster est un groupe de nœuds qui partagent des propriétés communes dans un graphe. Cependant, le support des systèmes de visualisation de graphes existants pour des tâches liées aux communautés est assez limité. Ces tâches peuvent être par exemple l'exploration des relations intracommunautés et des relations intercommunautés.

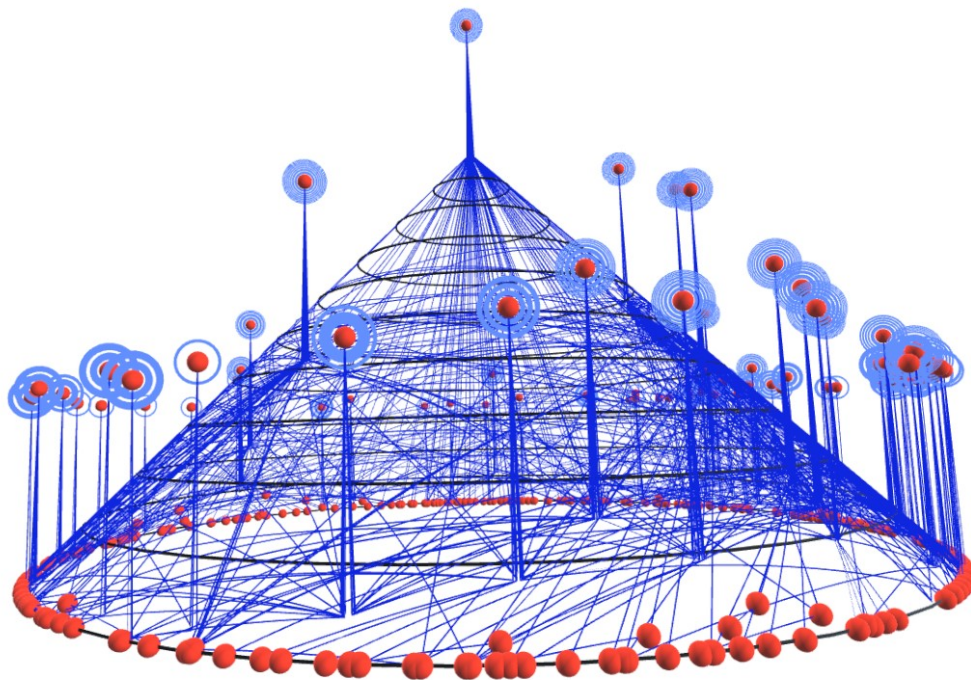
Avec une représentation visuelle d'un graphe partitionné, un analyste de graphes peut avoir besoin d'identifier et de manipuler certains nœuds avec des caractéristiques structurelles souhaitées. Par exemple, l'identification des nœuds connecteurs peut permettre à l'analyste de savoir quels nœuds jouent un rôle important de médiation et dirigent les échanges entre les différents groupes du graphe partitionné. Rappelons qu'un connecteur est un nœud qui connecte son groupe aux groupes voisins. Dans notre cas, un connecteur est nœud qui est connecté à des nœuds qui sont en dehors de la représentation visuelle du groupe, et donc pas forcément visualisés. La méthode de visualisation de clusters proposée par Cruz [30] (présentée dans la section 3.3) met en évidence les nœuds connecteurs dans un graphe partitionné. Cependant, cette méthode devient peu pratique si un analyste est intéressé à observer, à la fois, les connecteurs et la centralité des nœuds.

Pour cela, nous proposons une nouvelle méthode de mise en évidence des connecteurs (voir figure 4.11). La visualisation des connecteurs est indispensable si notre méthode est utilisée pour se focaliser sur un groupe sans afficher le reste du graphe afin de conserver l'information topologique. Elle permet à un analyste de connaître le nombre de groupes voisins par connecteur, mais aussi le nombre de nœuds voisins dans chacun des groupes. Nous proposons pour cela de :

- représenter chaque groupe voisin par un anneau (façon « anneau de Saturne ») autour du connecteur ;
- représenter l'épaisseur d'un anneau en fonction du nombre de nœuds voisins dans le groupe ;
- placer les anneaux les plus épais au centre ;
- mettre en place une possibilité de décoller le connecteur de la surface pour améliorer sa visibilité.



(a)



(b)

FIGURE 4.11 – Mise en évidence de la centralité d’intermédiarité et des connecteurs d’un groupe de 295 nœuds et 683 arêtes. (a) les connecteurs sont sur la surface de projection au même titre que les autres nœuds, (b) les connecteurs sont décollés de la surface afin d’améliorer leur visibilité par rapport aux autres nœuds du groupe.

La figure 4.12 illustre un zoom sur un nœud connecteur. Le nombre d'anneaux représente le nombre de clusters connectés par ce nœud. L'anneau le plus épais est au centre et représente le cluster où ce connecteur a plus de nœuds voisins.

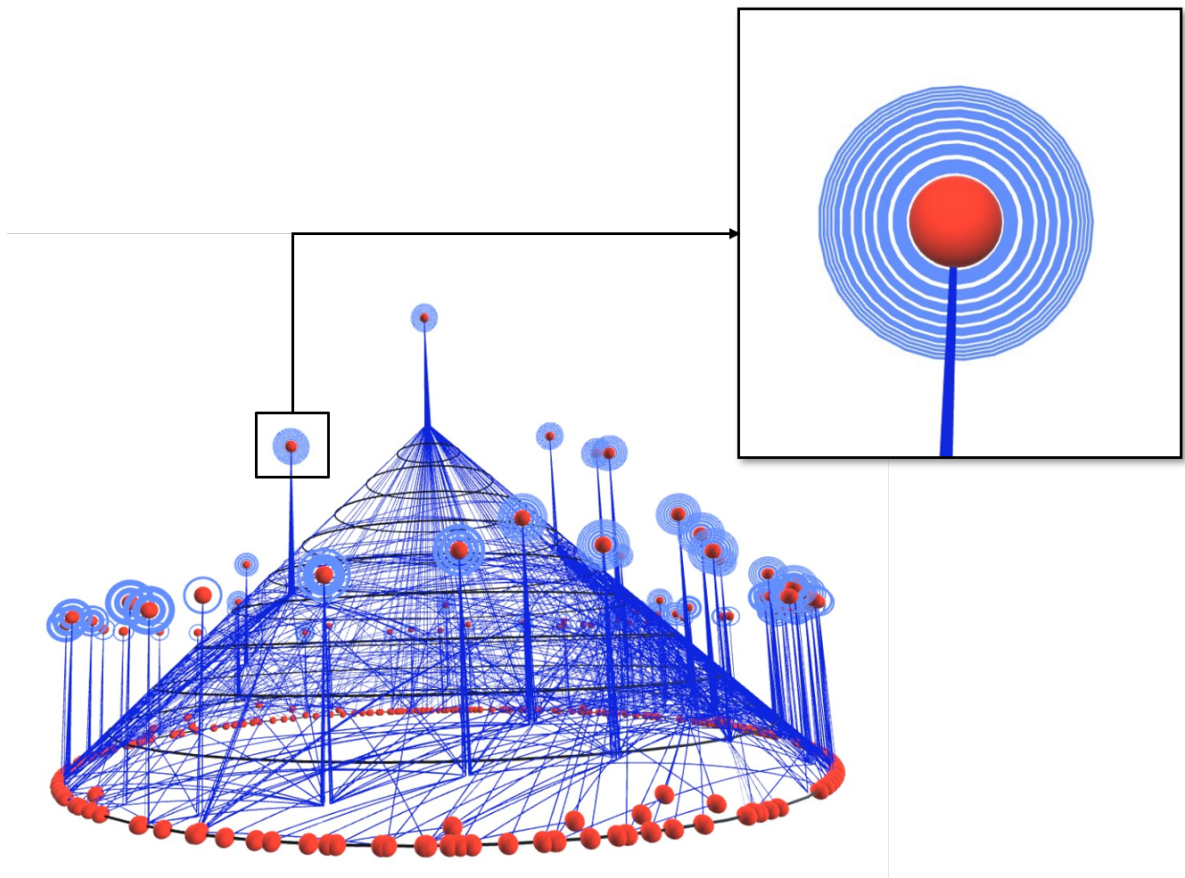


FIGURE 4.12 – Illustration d'un zoom sur un connecteur. Le nombre d'anneaux représente le nombre de clusters connectés. L'anneau le plus épais se trouve au centre et représente le cluster où le connecteur a plus de nœuds voisins.

Synthèse

Le passage à la 3D a permis de disposer de plus d'espace afin de mieux distribuer les nœuds que les méthodes 2D. Ainsi, notre approche radiale 3D a permis de réduire les occlusions tout en améliorant dans certains cas la perception de la connectivité des nœuds. De plus, elle permet d'observer à la fois la connectivité entre des groupes de nœuds et celle d'un nœud en faisant varier l'opacité des arêtes. Elle offre également une possibilité de visualiser simultanément la centralité des nœuds et les connecteurs d'un groupe dans le cas d'un graphe partitionné.

4.5 Évaluations

Nous avons effectué deux évaluations par le biais d'une série de tâches réalisées sur des graphes différents. La première a pour but de comparer l'efficacité de notre proposition 3D par rapport aux différentes représentations radiales 2D. La seconde vise à comparer l'efficacité de notre proposition en immersion par rapport à notre proposition 3D sur un grand écran de 75 pouces et à une représentation radiale 2D (avec une possibilité de faire une emphase sur le centre ou sur la périphérie) également sur un grand écran. Précisons que cet écran n'est pas stéréo.

Pour ces deux expériences, nous mesurons un score d'efficacité et un temps de réalisation d'une tâche pour chacune des visualisations. Nous utilisons ainsi ces deux métriques pour déterminer si un type de visualisation est plus ou moins performant que les autres pour une tâche donnée. Nous avons choisi d'utiliser la centralité d'intermédierité pour ces deux expériences. Toutefois, le résultat de nos évaluations n'est pas impacté par le type de mesure de centralité utilisé.

4.5.1 Évaluation 1

Cette première évaluation a été réalisée en ligne (WebGL) à cause de la Covid-19. Nous avons suggéré dans la section 4.2.2 qu'une projection 3D de la vue 2D uniforme permet d'obtenir les mêmes avantages que les vues 2D mettant l'emphase sur le centre ou sur la périphérie. Notre objectif est donc de montrer que les visualisations 3D pourraient

être mieux adaptées que les visualisations 2D pour comparer facilement la centralité des nœuds, mais aussi explorer les zones denses d'un graphe.

4.5.1.1 Tâches

Comme nous l'avons souligné plus haut, une projection 3D uniforme met en évidence soit le centre, soit la périphérie ou de façon modérée le centre et la périphérie selon le type de surface. Nous avons donc choisi trois tâches qui sont liées au centre, à la périphérie et aux zones denses d'un graphe :

- **Tâche 1 (liée aux nœuds centraux)** : les participants sont invités à trouver l'un des nœuds qui a le plus grand degré parmi les voisins du nœud central.
- **Tâche 2 (liée aux nœuds en périphérie)** : les participants sont invités à trouver l'un des nœuds les moins centraux qui a au moins deux voisins.
- **Tâche 3 (liée aux zones denses du graphe)** : les participants sont invités à trouver l'un des nœuds de degré au moins 3 et qui a le coefficient de clustering le plus élevé sauf 100%.

4.5.1.2 Données expérimentales

Pour cette étude, nous avons 6 visualisations à évaluer (3 représentations 2D et 3 représentations 3D). Nous avons donc généré 6 graphes différents avec l'algorithme SBM présenté dans la section 2.1.8.2. Les graphes générés ont la même taille (250 nœuds et 855 arêtes) et des caractéristiques topologiques équivalentes (densité = 0,027 et diamètre = 8), car il est difficile de trouver de tels graphes dans les bases de données en ligne. Rappelons que la densité d'un graphe représente le ratio entre le nombre d'arêtes existantes et le nombre maximal d'arêtes possibles. Le diamètre est quant à lui la plus grande distance (longueur du plus court chemin) entre deux nœuds du graphe.

Rappelons également que l'algorithme de blocs stochastiques est un algorithme génératif de graphes aléatoires qui produit généralement des graphes contenant une structure de clusters. Cela signifie que chaque nœud a une appartenance fixe à un cluster qui détermine la probabilité d'existence d'une arête vers d'autres nœuds [71, 136]. Par conséquent, les graphes générés ont des caractéristiques topologiques suffisamment équivalentes, tout en étant suffisamment différents pour éviter un effet d'apprentissage lors du passage d'un graphe à l'autre lors des expérimentations.

4.5.1.3 Interactions avec la visualisation de graphes

Nous avons mis en place des fonctionnalités qui permettent à un utilisateur d'interagir avec la visualisation d'un graphe.

Survol. Une infobulle affiche le degré et le coefficient de clustering lorsqu'un utilisateur place le pointeur de la souris au-dessus d'un nœud.

Sélection. La sélection d'un nœud se fait à l'aide d'un clic du bouton gauche de la souris. Elle permet de voir les voisins du nœud sélectionné, qui sont mis en évidence par une couleur différente de celle des autres nœuds. Dans cette expérience, un utilisateur ne peut sélectionner qu'un seul nœud à la fois.

Rotation. La rotation d'un graphe se fait à l'aide du bouton droit de la souris et permet à un utilisateur de changer de point de vue pour éviter, par exemple, certaines occlusions.

Contrôle de la caméra. Un utilisateur peut zoomer sur un graphe à l'aide de la roulette de la souris. Il peut également utiliser les touches directionnelles du clavier pour déplacer la caméra horizontalement et verticalement.

4.5.1.4 Hypothèses

Une meilleure lisibilité d'un graphe implique une meilleure efficacité dans une analyse, et par conséquent, une plus grande rapidité dans l'exploration. Ainsi, nous identifions des hypothèses ci-dessous par rapport à l'efficacité et à la rapidité.

Efficacité : nous nous attendions à ce que les visualisations 3D obtiennent un score d'efficacité nettement supérieur à celui des visualisations 2D. Ainsi, par rapport à la **tâche 1**, nous nous attendions à ce que les participants obtiennent de moins bons résultats sur la visualisation 2D qui met l'emphase sur la périphérie que sur les autres surfaces de visualisation. Par conséquent, nous avons posé les hypothèses suivantes :

H1 : la visualisation 2D qui met l'emphase sur la périphérie sera moins performante que les autres visualisations lorsqu'on s'intéresse aux nœuds centraux.

H2 : contrairement à la visualisation 2D avec emphase sur la périphérie, les projections 3D qui mettent naturellement l'emphase sur la périphérie (cône et sphère) ne seront pas moins performantes pour réaliser des tâches liées au centre.

Ensuite, concernant la **tâche 2**, nous nous attendions à ce que les participants aient un plus mauvais score d'efficacité sur la visualisation 2D qui met l'emphase sur le centre que sur les autres visualisations. Ainsi, nous avons posé les hypothèses suivantes :

H3 : la visualisation 2D qui met l'emphase sur le centre sera moins performante que les

autres visualisations lorsque les tâches sont liées à la périphérie.

H4 : contrairement à la visualisation 2D avec emphase sur le centre, les projections 3D qui mettent naturellement l'emphase sur le centre (cône et portion de tore) ne seront pas moins performantes pour réaliser des tâches liées à la périphérie.

En ce qui concerne les zones denses (**tâche 3**), nous nous attendions à ce que les participants obtiennent de bien meilleures performances sur les visualisations 3D que sur les visualisations 2D. Nous avons donc émis l'hypothèse suivante :

H5 : les visualisations 3D seront mieux adaptées que les visualisations 2D pour explorer les zones denses d'un graphe.

Rapidité : quelle que soit la tâche, nous nous attendions à ce que les participants soient plus rapides sur les visualisations 3D que sur les visualisations 2D. Par conséquent, nous avons émis l'hypothèse suivante :

H6 : le temps pour effectuer une tâche sera plus long avec les visualisations 2D qu'avec les visualisations 3D.

4.5.1.5 Dispositif expérimental

Une fois les graphes générés, nous avons ensuite construit 24 configurations de sorte que chaque combinaison (surface + graphe) soit placée autant de fois à chaque position sur l'ensemble des configurations. Pour limiter le nombre combinatoire, les configurations sont construites en utilisant quelque chose de similaire au concept du carré latin [126]. Un carré latin est un tableau de n lignes et n colonnes remplies de n éléments distincts de sorte que chaque élément apparaisse exactement une fois sur chaque ligne et exactement une fois dans chaque colonne.

Nous avons respecté un ordre de répartition dans chaque configuration entre les visualisations 2D et 3D. Ainsi, l'ordre d'apparition d'une visualisation 2D doit correspondre à celui de la visualisation 3D équivalente. Par exemple, si une configuration commence par les visualisations 2D, et si la première visualisation est celle qui met l'emphase sur le centre, alors la première surface 3D sera la portion de tore, puisqu'elle est la surface 3D qui met le plus en évidence les nœuds centraux (voir le tableau 4.1). Ainsi, nous nous assurons que chaque configuration est testée autant de fois avant qu'après chacune des autres configurations. La moitié des participants ont réalisé l'expérience en commençant avec les visualisations 2D pour finir avec les visualisations 3D et la seconde moitié a fait l'inverse (les visualisations 3D puis les visualisations 2D).

Ensuite, compte tenu des conditions de l'expérience, nous avons organisé une visiocon-

férence avec chaque participant afin de superviser le déroulement de l'expérience. Cette expérience est composée de quatre phases : une phase d'instruction, une phase d'entraînement, une phase d'évaluation et une phase de questionnaire.

La phase d'instruction consiste d'abord à décrire le déroulement de l'expérience et à présenter un formulaire de consentement au participant. Nous lui donnons ensuite les notions essentielles sur les graphes afin de nous assurer qu'il a les connaissances utiles pour l'expérience. Enfin, nous présentons l'environnement de l'expérience et les différentes interactions possibles (voir la section 4.5.1.3).

Ensuite, chaque participant s'entraîne à réaliser les tâches ci-dessus (section 4.5.1.1) avec le graphe « Karate Club » (34 nœuds et 78 arêtes) [160]. Le participant doit d'abord cliquer sur un bouton **Commencer** pour voir la première tâche à réaliser. La tâche suivante s'affiche automatiquement après validation de la réponse à la tâche précédente. Une fois familiarisé avec le système, le participant passe à l'expérience proprement dite, mais avec de grands graphes.

Lors de l'évaluation, les participants réalisent les mêmes tâches que celles de l'entraînement, mais avec nos graphes générés (250 nœuds et 855 arêtes). Pendant l'expérience, nous avons mesuré un score d'efficacité et le temps de réalisation pour chaque tâche et chaque visualisation. Les performances des participants sont automatiquement enregistrées lorsqu'ils valident leurs réponses. Notre procédure de calcul du score d'efficacité est présentée ci-dessous pour chacune des tâches :

Tâche 1 : trouver un des nœuds qui a le plus grand degré parmi les voisins du nœud central.

La formule du calcul du score pour la tâche 1 est :

$$score_i = \begin{cases} 100 * (deg_i/deg_{ideal}), & \text{si } d(centre, i) = 1 \\ 0, & \text{sinon} \end{cases} \quad (4.2)$$

où deg_i est le degré d'un nœud i sélectionné, deg_{ideal} est le plus grand degré parmi les voisins du nœud central et $d(centre, i)$ est la plus courte distance entre le nœud central et le nœud i . Ainsi, le nœud i doit être directement connecté au nœud central, c'est-à-dire $d(centre, i)$ doit être égale à 1.

Tâche 2 : trouver un des nœuds les moins centraux qui a au moins deux voisins.

La formule du calcul du score pour la tâche 2 est :

$$score_i = \begin{cases} 100 * (1 - c_i)/(1 - c_{ideal}), & \text{si } c_{ideal} \neq 1 \\ 0, & \text{sinon} \end{cases} \quad (4.3)$$

où c_i et c_{ideal} sont respectivement la valeur de centralité d'un nœud i et celle du nœud idéal. Par ailleurs, le score est égal à 0 si le degré du nœud sélectionné est inférieur à 2, car il est moins complexe de vérifier le degré du nœud sélectionné. Sinon, le score varie de 0 au centre à 1 pour un nœud de degré au moins 2 et le plus en périphérie.

Tâche 3 : trouver un des nœuds de degré au moins 3 et qui a le coefficient de clustering le plus élevé sauf 100%.

La formule du calcul du score pour la tâche 3 est :

$$score_i = \begin{cases} 100 * (ccf_i - ccf_{worst})/k, & \text{si } k > 0 \\ 0, & \text{sinon} \end{cases} \quad (4.4)$$

où $k = (ccf_{ideal} - ccf_{worst})$. ccf_i , ccf_{worst} et ccf_{ideal} sont respectivement le coefficient de clustering d'un nœud i , le pire coefficient de clustering et le coefficient de clustering le plus élevé sauf 100%. Ainsi, le score est égal à 0 si le degré du nœud sélectionné est inférieur à 3 ou si son coefficient de clustering est égal à 100%.

Enfin, la dernière phase consiste à remplir un questionnaire sur l'utilisabilité du système et sur l'expérience utilisateur (voir l'annexe 3). L'évaluation de l'utilisabilité d'un système offre un moyen rapide et efficace de juger de la convivialité de ce système. Elle se fait à travers le *System Usability Scale* (SUS) [21] et peut servir de support pour comparer plusieurs systèmes jugés utilisables. La procédure de calcul du score (compris entre 1 et 100) se déroule en trois étapes :

- Tout d'abord, on soustrait un point du score donné par un utilisateur pour les questions impaires (1, 3, 5, 7 et 9).
- Ensuite, pour les questions paires (2, 4, 6, 8 et 10), on soustrait le score de l'utilisateur de 5.
- On additionne enfin les 10 nouveaux scores et on multiplie le résultat par 2,5.

4.5.1.6 Participants

Nous avons besoin d'un nombre de participants qui serait un multiple de 24 afin de rencontrer le même nombre de ces 24 configurations ci-dessus évoquées. Nous avons donc eu 24 participants (9 femmes, 15 hommes) qui ont été recrutés parmi nos collègues du laboratoire et parmi des étudiants ayant un profil scientifique : 50% ont entre 18 et 25 ans, 37,5% ont entre 25 et 35 ans et 12,5% ont plus de 35 ans. La plupart des participants n'avaient aucune expérience en analyse de données et en visualisation, mais certains d'entre eux avaient une expérience en jeux vidéos.

4.5.1.7 Résultats

Suite au recueil de données fait lors de cette étude expérimentale, nous sommes en mesure d'évaluer l'efficacité et la rapidité des différentes visualisations 2D et 3D. Nous présentons d'abord les résultats des performances utilisateur grâce aux données mesurées objectivement lors de l'étude et ensuite les résultats de l'expérience utilisateur grâce aux données subjectives recueillies à l'aide du questionnaire SUS.

Performances des utilisateurs : nous présentons les résultats des performances des utilisateurs à travers des tests non paramétriques en utilisant la méthode de Kruskal [99] et des tests post-hoc en utilisant la méthode de Dunn [34, 131]. Nous avons utilisé des tests non paramétriques, car aucune de nos distributions ne suit une loi normale. Les tests de normalité ont été effectués à l'aide de la méthode de Shapiro-Wilk [134]. Rappelons que les variables analysées sont le score d'efficacité et le temps de réalisation d'une tâche.

- **Score d'efficacité**

Tâche 1 : le test non paramétrique montre qu'il y a une différence statistiquement significative entre les visualisations ($statistic = 31,45$ et $p = 10^{-5} < 0,05$). Le test post-hoc (voir le tableau 4.3) montre de plus que la représentation 2D qui met l'emphase sur la périphérie présente une différence de moyennes en faveur des autres visualisations, notamment le cône et la demi-sphère (voir le tableau 4.2). Nous validons donc l'hypothèse H1 que la visualisation 2D qui met l'emphase sur la périphérie est moins performante que les autres visualisations lorsqu'une tâche est liée aux nœuds centraux. Le résultat du test post-hoc montre qu'il y a une différence statistiquement significative entre les représentations 2D qui mettent l'emphase sur la périphérie et l'emphase sur le centre. Nous pouvons donc valider l'hypothèse H2 que nos projections 3D (sur un cône ou sur une demi-sphère) sont également performantes pour réaliser des tâches liées au centre d'un graphe.

	Moyenne	Écart-type
2D centrale	100	0
2D périphérique	63,43	46,27
2D uniforme	94,44	21,23
Cône	96,15	18,88
Demi-sphère	97,92	7,96
Portion de tore	94,72	19,94

TABLE 4.2 – Score d’efficacité (Tâche 1) : moyennes et écart-types du score d’efficacité par visualisation.

	2D centrale	2D périphérique	2D uniforme	Cône	Demi-sphère	Portion de tore
2D centrale	1	10^{-4***}	0,37	0,68	0,42	0,40
2D périphérique	10^{-4***}	1	10^{-3***}	10^{-4***}	10^{-3***}	10^{-3***}
2D uniforme	0,37	10^{-3***}	1	0,64	0,93	0,96
Cône	0,68	10^{-4***}	0,64	1	0,70	0,67
Demi-sphère	0,42	10^{-3***}	0,93	0,70	1	0,97
Portion de tore	0,40	10^{-3***}	0,96	0,67	0,97	1

TABLE 4.3 – Score d’efficacité (Tâche 1) : résultat des tests post-hoc en utilisant la méthode de Dunn [131] (p-value significative : $*p < 0,05$; $**p < 0,01$; $***p \leq 0,001$).

Tâche 2 : il y a une différence d’efficacité statistiquement significative entre la visualisation 2D qui met l’emphase sur le centre et les autres visualisations, notamment le cône et la portion de tore (voir les tableaux (voir les tableaux 4.4) et 4.5) car le test statistique est égal à 40,31 et la p-value correspondante est égale à $10^{-5} < 0,05$. Par conséquent, nous validons l’hypothèse H3 que la visualisation 2D qui met l’emphase sur le centre est moins performante que les autres visualisations lorsqu’une tâche est liée aux nœuds en périphérie. Grâce à ce résultat, nous validons de plus l’hypothèse H4 que nos projections 3D (sur un cône ou sur une portion de tore) sont également performantes lorsqu’on s’intéresse aux tâches liées à la périphérie d’un graphe.

	Moyenne	Écart-type
2D centrale	92,54	6,13
2D périphérique	95,08	20,38
2D uniforme	99,35	1,37
Cône	97,42	10,40
Demi-sphère	99,59	1,34
Portion de tore	98,59	4,99

TABLE 4.4 – Score d’efficacité (Tâche 2) : moyennes et écart-types du score d’efficacité par visualisation.

	2D centrale	2D péri- phérique	2D uniforme	Cône	Demi- sphère	Portion de tore
2D centrale	1	10^{-5***}	10^{-2**}	10^{-4***}	10^{-5***}	10^{-3***}
2D périphérique	10^{-5***}	1	0,16	0,69	0,71	0,22
2D uniforme	10^{-2**}	0,16	1	0,31	0,08	0,86
Cône	10^{-4***}	0,69	0,31	1	0,44	0,41
Demi-sphère	10^{-5***}	0,71	0,08	0,44	1	0,11
Portion de tore	10^{-3***}	0,22	0,86	0,41	0,11	1

TABLE 4.5 – Score d’efficacité (Tâche 2) : résultat des tests post-hoc en utilisant la méthode de Dunn [131] (p-value significative : * $p < 0,05$; ** $p < 0,01$; *** $p \leq 0,001$).

Tâche 3 : la différence de moyennes (voir le tableau 4.6 ou la figure 4.13) pourrait nous amener à croire que la visualisation 2D qui met l’emphase sur le centre est plus performante que les autres visualisations quand il faut regarder dans les zones denses d’un graphe. Cependant, le test montre que la différence d’efficacité entre les visualisations n’est pas statistiquement significative ($statistic = 6,0$ et $p = 0,31 > 0,05$). Par conséquent, nous rejetons l’hypothèse H5 que les visualisations 3D sont mieux adaptées que les visualisations 2D pour explorer les zones denses d’un graphe.

	Moyenne	Écart-type
2D centrale	76,93	25,01
2D périphérique	62,12	25,52
2D uniforme	68,15	21,68
Cône	70,07	20,59
Demi-sphère	65,09	20,82
Portion de tore	64,94	20,78

TABLE 4.6 – Score d’efficacité (Tâche 3) : moyennes et écart-types du score d’efficacité par visualisation.

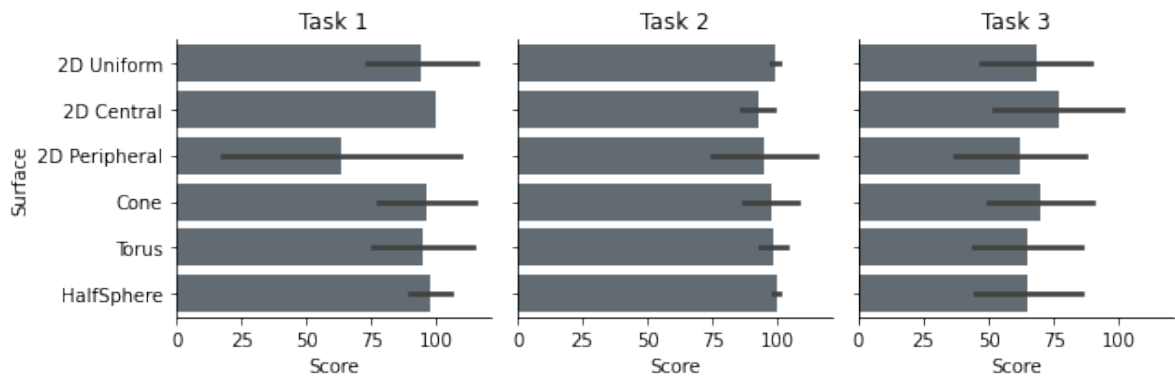


FIGURE 4.13 – Score d’efficacité : moyennes et écarts-types du score d’efficacité par tâche et par visualisation.

En conclusion, l’analyse du score d’efficacité montre que les surfaces 3D sont bien adaptées pour la réalisation des tâches liées aux nœuds centraux ou en périphérie, car nous avons validé les hypothèses H1, H2, H3 et H4. En revanche, nous avons rejeté l’hypothèse H5 que les visualisations 3D seront mieux adaptées que les visualisations 2D pour explorer les zones denses d’un graphe.

• Temps de réalisation

Tâche 1 : la figure 4.14 pourrait nous amener à dire que les participants ont passé plus de temps sur la visualisation 2D qui met l’emphase sur la périphérie, contrairement à toutes les autres visualisations. Cependant, le test statistique montre que la différence n’est pas statistiquement significative entre les visualisations ($statistic = 5,990$ et $p = 0,31 > 0,05$). Nous rejetons donc l’hypothèse H6 que le temps pour effectuer une tâche est plus long avec les visualisations 2D qu’avec les visualisations 3D.

Tâche 2 : la différence entre toutes les visualisations n’est pas statistiquement significative ($statistic = 1,65$ et $p = 0,90 > 0,05$). Par conséquent, nous rejetons encore l’hypothèse H6.

Tâche 3 : nous rejetons l’hypothèse H6, car la différence n’est pas statistiquement significative entre toutes les visualisations ($statistic = 1,04$ et $p = 0,96 > 0,05$)

En définitive, l’analyse du temps de réalisation des tâches nous amène à rejeter l’hypothèse H6, car les tests ont montré que la différence entre les visualisations 2D et 3D n’est pas statistiquement significative.

Expérience des utilisateurs : rappelons qu’à la fin de l’expérience, les participants ont été invités à remplir un questionnaire lié à l’utilisabilité du système et à leur expé-

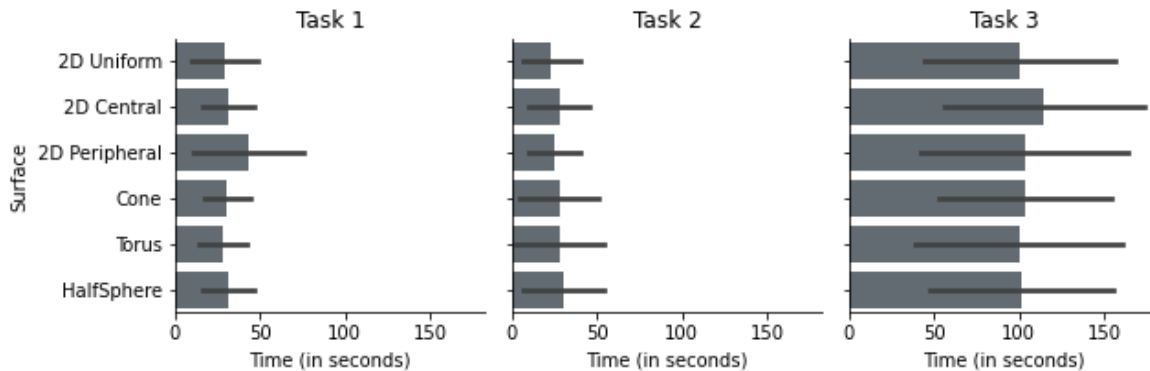


FIGURE 4.14 – Temps de réalisation : moyennes et écarts-types du temps de réalisation par tâche et par visualisation.

rience. L'évaluation de l'utilisabilité permet de mesurer la facilité d'utilisation des différentes visualisations. Ainsi, les systèmes 2D et 3D ont respectivement obtenu des scores d'utilisabilité SUS de 75,21 et 81,46 sur 100, ce qui signifie qu'ils sont utilisables (le seuil d'utilisabilité d'un système étant fixé à 70/100 pour le SUS). Cependant, la visualisation 3D a obtenu un meilleur score d'utilisabilité que la visualisation 2D.

En ce qui concerne l'expérience utilisateur, nous avons demandé aux participants s'ils avaient compris les tâches demandées, s'ils avaient eu des difficultés à interagir avec le système et s'ils avaient ressenti une fatigue visuelle. Les résultats sont les suivants : 23 participants sur 24 ont déclaré avoir compris les tâches demandées ; 7 participants sur 24 ont déclaré avoir eu des difficultés à interagir avec le système et 7 participants sur 24 ont déclaré avoir ressenti une fatigue visuelle.

Nous avons également invité les participants à préciser les visualisations qui leur ont permis de bien réaliser les tâches demandées, mais aussi à identifier les visualisations avec lesquelles ils ont eu des difficultés. D'après leurs réponses, les visualisations 3D facilitent la réalisation des différentes tâches, contrairement aux visualisations 2D. La figure 4.15 permet d'illustrer les préférences des participants. Elle permet également de montrer que la visualisation 2D qui met l'accent sur le centre et la visualisation 2D qui met l'accent sur la périphérie totalisent à elles seules 80% des votes des visualisations que les participants n'aiment pas, tandis que le cône fait 0%.

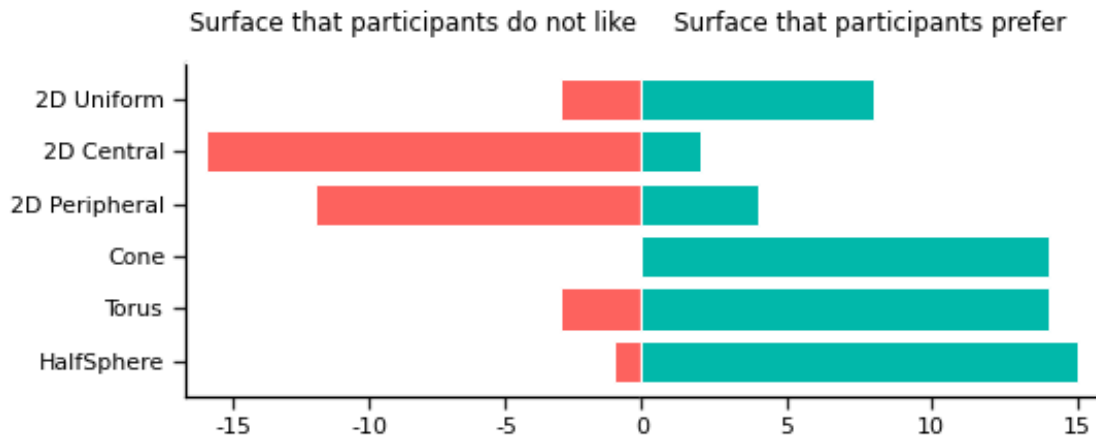


FIGURE 4.15 – Les participants préfèrent nettement les visualisations 3D pour la réalisation des différentes tâches.

4.5.1.8 Discussion

Les différentes projections 3D des représentations radiales 2D ont permis d'améliorer la lisibilité d'un graphe. Ainsi, nous pensions que chaque visualisation 3D pouvait être la meilleure pour une des tâches, d'où l'intérêt de passer d'un type de projection à un autre en fonction de la tâche à réaliser. Nous nous attendions également à ce que les visualisations 3D soient plus performantes que les visualisations 2D quand on s'intéresse aux zones denses d'un graphe ou à certains nœuds particuliers (forte centralité, par exemple).

Les résultats de l'évaluation ont montré que la différence d'efficacité entre les visualisations 2D et 3D est statistiquement significative pour les tâches liées au centre et à la périphérie. Ils nous ont ainsi permis de valider quatre hypothèses qui sont liées à ces tâches (voir le tableau 4.7). En revanche, nous avons rejeté l'hypothèse H5, car nous ne sommes pas en mesure de prouver que les visualisations 3D sont plus adaptées que les visualisations 2D pour explorer les zones denses d'un graphe. Nous n'avons pu prouver également l'hypothèse H6 que les visualisations 3D sont plus rapides que les visualisations 2D, car la différence en temps de réalisation n'est pas statistiquement significative entre toutes les visualisations.

Les retours des participants ont quant à eux montré que les visualisations 3D pourraient être bien adaptées pour mener à bien les différentes tâches qui pourraient être liées au centre, à la périphérie ou aux zones denses d'un graphe, contrairement aux visualisations 2D. De plus, la visualisation 3D a obtenu un meilleur score d'utilisabilité que la visualisation 2D (81,46 contre 75,21).

Métriques	Efficacité					Rapidité
	H1	H2	H3	H4	H5	H6
Tâche 1	✓	✓				✗
Tâche 2			✓	✓		✗
Tâche 3					✗	✗

TABLE 4.7 – Tableau récapitulatif des hypothèses d'évaluation. Une coche (✓) indique que l'hypothèse est validée, tandis qu'une croix (✗) indique que l'hypothèse est rejetée. Une cellule vide indique que l'hypothèse n'est pas pertinente pour la tâche correspondante.

Synthèse : visualisation 2D vs visualisation 3D

Les résultats de certains tests statistiques nous ont permis de montrer que les visualisations 3D étaient significativement plus performantes que les visualisations 2D lorsqu'une tâche est liée aux nœuds centraux ou aux nœuds en périphérie. De plus, les participants préfèrent les visualisations 3D, car elles fournissent une bonne lisibilité d'un graphe et permettent de comparer rapidement la centralité des nœuds. Cependant, la différence en temps de réalisation des tâches n'est pas statistiquement significative entre les visualisations 2D et 3D.

4.5.2 Évaluation 2

L'objectif de cette évaluation est de comparer l'efficacité de la visualisation 3D immersive par rapport à la visualisation 3D desktop et à la visualisation 2D desktop en mesurant un score d'efficacité. Nous voulons également vérifier si les tendances observées lors de la première évaluation sont confirmées pour un graphe de taille un peu plus importante que celle des graphes précédemment utilisés.

Nous avons en outre choisi d'utiliser la représentation conique (pour la visualisation 3D desktop et la visualisation 3D immersive) du fait qu'elle répartit les nœuds de manière uniforme, tout en améliorant la visibilité sur le centre et sur la périphérie. De plus, elle est la seule représentation que personne n'ait rejetée lors de notre expérience précédente (voir la figure 4.15).

4.5.2.1 Tâches

Nous avons choisi de conserver les deux dernières tâches de notre expérimentation précédente et de substituer la première par une nouvelle tâche, car nous voulions une tâche qui nécessiterait une bonne visibilité à la fois du centre et des zones denses d'un graphe.

- **Tâche 1 (liée aux nœuds centraux et aux zones denses)** : *les participants sont invités à trouver l'un des nœuds de degré au moins 3 dont le coefficient de clustering est le plus élevé et dont un voisin est connecté au nœud central.* Notons que pour cette tâche et pour nos graphes utilisés, il n'y a aucun nœud ayant un coefficient de clustering égal à 100% parmi les nœuds potentiels, ce qui nous permet de forcer les utilisateurs à explorer les graphes.
- **Tâche 2 (liée aux nœuds en périphérie)** : *les participants sont invités à trouver l'un des nœuds les moins centraux qui a au moins deux voisins.*
- **Tâche 3 (liée aux zones denses du graphe)** : *les participants sont invités à trouver l'un des nœuds de degré au moins 3 et qui a le coefficient de clustering le plus élevé sauf 100%.*

4.5.2.2 Données expérimentales

Rappelons que l'un des objectifs de cette étude est de comparer les résultats de deux graphes de tailles différentes d'une visualisation 3D immersive avec des visualisations 3D et 2D desktop. Pour cela, nous avons utilisé une série de graphes de 250 nœuds et 855 arêtes de notre expérience précédente et une autre série de plus grands graphes (500 nœuds et 3294 arêtes) également générés avec l'algorithme SBM (section 2.1.8.2).

4.5.2.3 Interactions avec la visualisation de graphes

Mode desktop

En plus des interactions mises en place lors de notre expérience précédente (le survol, la sélection, la rotation et le zoom), un utilisateur a la possibilité de mémoriser un nœud lors de l'exploration d'un graphe. Cette action consiste d'abord à placer le pointeur de la souris au-dessus d'un nœud puis à appuyer ensuite la touche **Espace**. Le nœud ainsi mémorisé devient un peu plus gros que les autres. Il prend également une couleur différente des autres, ce qui permet à l'utilisateur de le retrouver facilement plus tard. Tout comme pour la sélection, l'utilisateur ne peut mémoriser qu'un seul nœud à la fois. Il dispose

également d'un curseur lui permettant de faire l'emphase sur le centre ou sur la périphérie du graphe.

Mode immersif

En immersion, les interactions sur un graphe se font à l'aide des manettes HTC Vive (voir la figure 4.16). Cependant, l'utilisateur peut se déplacer dans l'environnement à l'aide des mouvements de son corps. Il peut ainsi tourner autour d'un graphe en marchant. Il peut également changer de point de vue à l'aide des mouvements de la tête ou saisir le graphe comme un objet réel à l'aide de la gâchette (*Trigger*) de sa main non dominante.

Survol : un laser au bout de la manette de la main dominante de l'utilisateur permet de pointer un nœud et d'afficher son degré et son coefficient de clustering.

Sélection : la sélection se fait en appuyant sur la gâchette de la manette de la main dominante de l'utilisateur. La sélection d'un nœud permet d'observer ses voisins.

Mémorisation : la mémorisation consiste à marquer un nœud en le pointant et en appuyant sur le « Touchpad » de la manette de la main dominante de l'utilisateur.

Rotation : la rotation d'un graphe se fait à l'aide des boutons de l'axe des abscisses du « Touchpad » (voir la figure 4.16) de la main non dominante de l'utilisateur. Rappelons qu'un utilisateur peut également tourner autour d'un graphe en marchant.

Zoom : le zoom se fait à l'aide des boutons de l'axe des ordonnées du « Touchpad » de la main secondaire (voir la figure 4.16). Un utilisateur peut augmenter la taille globale d'un graphe pour se retrouver immergé dans le graphe afin de l'observer de l'intérieur.

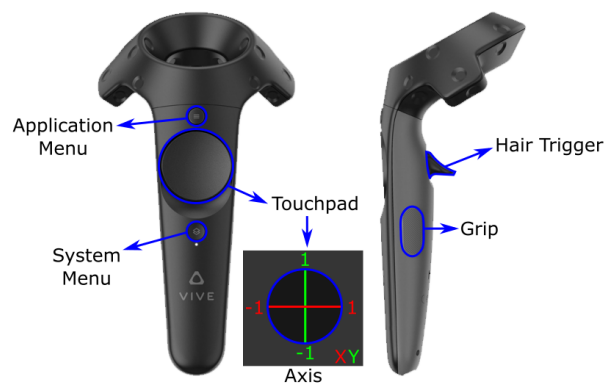


FIGURE 4.16 – Manette HTC Vive avec illustration des différents boutons d'interaction. (Image extraite du site <https://www.raywenderlich.com/149239/htc-vive-tutorial-unity>).

4.5.2.4 Hypothèses

Certains résultats de notre expérience précédente nous ont permis de montrer que la visualisation 3D était plus performante que la visualisation 2D pour nos tâches qui sont liées au centre ou à la périphérie d'un graphe. Par conséquent, nous nous attendons à ce que la visualisation 3D immersive soit plus performante que la visualisation 2D et la visualisation 3D desktop, car nous avons amélioré les interactions utilisateur. De plus, nous disposons de plus d'espace en immersion pour visualiser un graphe et améliorer la perception dans les zones denses. Nous identifions ainsi nos hypothèses par rapport à l'efficacité d'une visualisation, mais nous espérons que les participants ne perdront pas trop de temps sur la visualisation 3D immersive.

- **Efficacité** : les participants ont la possibilité de faire une emphase sur le centre ou sur la périphérie avec la visualisation 2D. Ainsi, nous nous attendons à ce que la visualisation 3D immersive soit encore plus performante que la visualisation 2D et la visualisation 3D desktop avec un graphe de 500 nœuds et 3294 arêtes. Nous formulons donc l'hypothèse ci-dessous relative à toutes nos tâches :

H7 : l'écart de performance entre la visualisation 3D immersive et la visualisation 2D augmente lorsqu'on augmente la taille des graphes.

4.5.2.5 Dispositif expérimental

Pour cette étude, nous avons construit 6 configurations par graphe, de sorte que chaque combinaison (surface + graphe) soit placée autant de fois à chaque position sur l'ensemble des configurations. De plus, chaque configuration est testée aussi souvent avant qu'après chacune des autres. L'expérimentation dure une heure environ et se déroule en cinq phases.

La première consiste d'abord à décrire le déroulement de l'expérience aux participants. Nous leur présentons ensuite un formulaire de consentement à signer après lecture. Ensuite, nous leur donnons les notions essentielles sur les graphes. Nous leur présentons enfin les différents environnements de l'expérience et les interactions possibles.

Ensuite, nous alternons entre un entraînement (avec le graphe « Karate Club »), une première évaluation (avec un graphe de 250 nœuds et 855 arêtes) et un questionnaire SUS pour chacune des visualisations. La seconde évaluation se déroule avec un graphe de 500 nœuds et 3294 arêtes. Pendant l'expérimentation, chaque participant est placé à environ 2 mètres du grand écran (75 pouces) pour les visualisations desktop. Rappelons que l'écran utilisé n'est pas stéréo. Nous mesurons un score d'efficacité et un temps de

réalisation d'une tâche, tout comme lors de notre expérience précédente. La procédure de calcul du score d'efficacité des deux dernières tâches reste inchangée (voir la section 4.5.1.5). Ci-dessous la procédure de calcul du score d'efficacité pour la tâche 1 :

Tâche 1 : *trouver l'un des nœuds de degré au moins 3 dont le coefficient de clustering est le plus élevé et dont un voisin est connecté au nœud central.*

La formule du calcul du score pour la tâche 1 est :

$$score_i = \begin{cases} 100 * (ccf_i - ccf_{worst})/k, & \text{si } k > 0 \text{ et } d(\text{centre}, i) \in [1, 2] \\ 0, & \text{sinon} \end{cases} \quad (4.5)$$

où

- $k = (ccf_{ideal} - ccf_{worst})$
- ccf_i est le coefficient de clustering d'un nœud i sélectionné
- ccf_{worst} est le pire coefficient de clustering
- ccf_{ideal} est le plus élevé parmi les nœuds potentiels.
- $d(\text{centre}, i)$ est la plus courte distance entre le nœud central et le nœud i .

À la fin de l'expérience, chaque participant remplit un questionnaire par rapport à son expérience avec les différentes visualisations (voir l'annexe 4).

4.5.2.6 Participants

Nous avons besoin d'un nombre de participants qui serait un multiple de 6 afin de rencontrer le même nombre de nos configurations. Nous avons eu au total 24 participants (3 femmes, 21 hommes) qui ont été recrutés parmi nos collègues du laboratoire et parmi des étudiants d'IMT Atlantique, de l'Énib et de l'UBO : 50% ont entre 18 et 25 ans, 29,2% ont entre 25 et 35 ans et 20,8% ont plus de 35 ans.

Aucun participant n'a déclaré avoir des problèmes de perception de couleurs. La plupart des participants n'avaient aucune expérience en analyse de données et en visualisation. Ils n'étaient pas non plus familiers avec les environnements immersifs.

4.5.2.7 Résultats

Nous présentons d'abord les résultats liés aux performances des participants et ensuite ceux de leur expérience avec nos différents systèmes de visualisation.

Performances des utilisateurs : comme lors de l'analyse statistique des résultats de l'expérimentation précédente, aucune de nos distributions ne suit une loi normale.

Par conséquent, nous présentons des résultats à travers des tests non paramétriques en utilisant la méthode de Kruskal [99] et des tests post-hoc en utilisant la méthode de Dunn [131].

- **Score d'efficacité**

Tâche 1 : les tests montrent que la différence entre les visualisations n'est pas statistiquement significative pour le graphe de 250 nœuds et 855 arêtes ($statistic = 2,78$ et $p = 0,25 > 0,05$), même si le tableau 4.8 montre une différence de moyennes en faveur des visualisations 3D. En revanche, elle l'est pour le graphe de 500 nœuds et 3294 arêtes ($statistic = 20,42$ et $p = 10^{-4} < 0,05$). De plus, le test post-hoc (tableau 4.9) montre qu'il y a une différence entre la visualisation 2D et toutes les autres visualisations 3D. Par conséquent, les visualisations 3D sont plus performantes que la visualisation 2D. Nous validons ainsi l'hypothèse H7 que nous avons un gain de performance en immersion lorsqu'on augmente la taille des graphes (voir les figures 4.17 et 4.18).

	Petit graphe (250 nœuds)		Grand graphe (500 nœuds)	
	Moyenne	Écart-type	Moyenne	Écart-type
2D	69,82	32,56	54,16	24,79
3D desktop	85,87	15,35	82,49	21,78
3D VR	82,62	23,69	81,87	35,30

TABLE 4.8 – Score d'efficacité (Tâche 1) : moyennes et écart-types du score d'efficacité par visualisation.

	2D	3D desktop	3D VR
2D	1	10^{-2**}	10^{-4***}
3D desktop	10^{-2**}	1	0,23
3D VR	10^{-4***}	0,23	1

TABLE 4.9 – Score d'efficacité avec le graphe de 500 nœuds et 3294 arêtes (Tâche 1) : résultat des tests post-hoc en utilisant la méthode de Dunn [131] (p-value significative : $*p < 0,05$; $**p < 0,01$; $***p \leq 0,001$).

Tâche 2 : le tableau 4.10 montre une différence de moyennes vraiment très faible en faveur de la visualisation 2D pour le petit graphe et en faveur des visualisations 3D pour le grand graphe. Le test de Kruskal-Wallis montre quant à lui que cette différence n'est pas statistiquement significative entre les visualisations tant avec le graphe de 250 nœuds et 855 arêtes ($statistic = 0,94$ et $p = 0,63 > 0,05$) qu'avec le graphe de 500 nœuds et 3294 arêtes ($statistic = 2,25$ et $p = 0,32 > 0,05$). Par conséquent, nous rejetons l'hypothèse H7

que l'écart de performance augmente en faveur de la visualisation 3D immersive lorsqu'on augmente la taille des graphes.

	Petit graphe (250 nœuds)		Grand graphe (500 nœuds)	
	Moyenne	Écart-type	Moyenne	Écart-type
2D	99,70	0,39	95,06	20,27
3D desktop	99,65	0,35	99,46	0,76
3D VR	95,57	20,36	99,45	0,82

TABLE 4.10 – Score d'efficacité (Tâche 2) : moyennes et écart-types du score d'efficacité par visualisation.

Tâche 3 : le tableau 4.11 montre que le score d'efficacité de la visualisation 2D est inférieur aux autres scores. Le test de Kruskal-Wallis montre quant à lui que la différence est statistiquement significative entre les visualisations du graphe de 250 nœuds et 855 arêtes ($statistic = 8,40$ et $p = 10^{-2} < 0,05$) et du graphe de 500 nœuds et 3294 arêtes ($statistic = 19,49$ et $p = 10^{-4} < 0,05$). De plus, le tableau 4.12 illustre un gain de performance pour la visualisation 3D immersive, car la différence entre la visualisation 2D et la visualisation 3D desktop n'est pas statistiquement significative. Par ailleurs, en comparant les p-values de la visualisation 3D immersive des tableaux 4.12 et 4.13, nous pouvons valider l'hypothèse que nous avons un gain de performance en immersion lorsqu'on augmente la taille des graphes, car les p-values de nos tests diminuent notablement et la qualité des tests statistiques passe de deux étoiles (**) à trois étoiles (***)).

	Petit graphe (250 nœuds)		Grand graphe (500 nœuds)	
	Moyenne	Écart-type	Moyenne	Écart-type
2D	73,45	18,08	73,75	15,66
3D desktop	86,69	16,55	92,12	10,10
3D VR	89,09	15,67	91,41	12,70

TABLE 4.11 – Score d'efficacité (Tâche 3) : moyennes et écart-types du score d'efficacité par visualisation.

- **Temps de réalisation**

Tâche 1 : la différence de temps de réalisation entre les différentes visualisations n'est pas statistiquement significative. Le test statistique de la visualisation du graphe de 250 nœuds et 855 arêtes est égal 0,67 avec une p-value égale à $0,72 > 0,05$ et celui de la visualisation du graphe de 500 nœuds et 3294 arêtes est égal à 1,89 avec une p-value égale à $0,39 > 0,05$.

	2D	3D desktop	3D VR
2D	1	0,09	10^{-2**}
3D desktop	0,09	1	0,40
3D VR	10^{-2**}	0,40	1

TABLE 4.12 – Score d’efficacité avec le graphe de 250 nœuds et 855 arêtes (Tâche 3) : résultat des tests post-hoc en utilisant la méthode de Dunn [131] (p-value significative : $*p < 0,05$; $**p < 0,01$; $***p \leq 0,001$).

	2D	3D desktop	3D VR
2D	1	10^{-3***}	10^{-3***}
3D desktop	10^{-3***}	1	0,93
3D VR	10^{-3***}	0,93	1

TABLE 4.13 – Score d’efficacité avec le graphe de 500 nœuds et 3294 arêtes (Tâche 3) : résultat des tests post-hoc en utilisant la méthode de Dunn [131] (p-value significative : $*p < 0,05$; $**p < 0,01$; $***p \leq 0,001$).

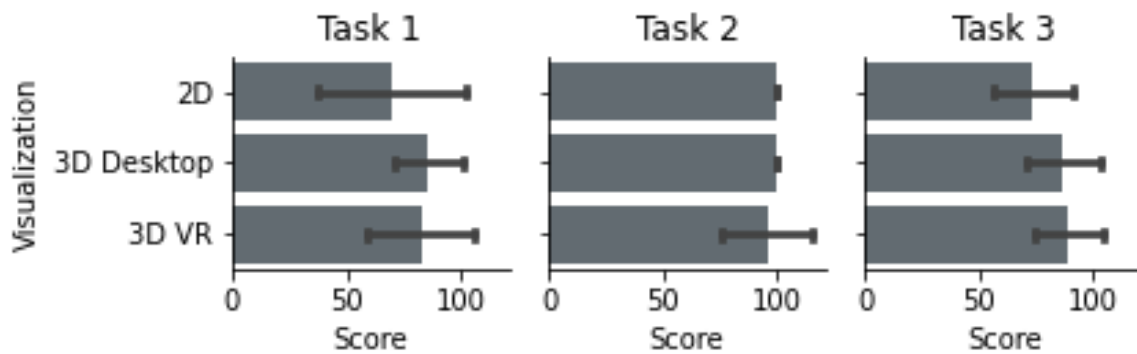


FIGURE 4.17 – Score d’efficacité (Graphe de 250 nœuds et 855 arêtes) : moyennes et écarts-types du score d’efficacité par tâche et par visualisation.

Tâche 2 : la différence de temps de réalisation entre les visualisations du graphe de 250 nœuds et 855 arêtes n’est pas statistiquement significative ($statistic = 0,82$ et $p = 0,66 > 0,05$). La différence de temps de réalisation entre les visualisations du graphe de 500 nœuds et 3294 arêtes n’est pas non plus statistiquement significative ($statistic = 0,96$ et $p = 0,62 > 0,05$).

Tâche 3 : la différence de temps de réalisation entre les visualisations du graphe de 250 nœuds et 855 arêtes n’est pas statistiquement significative ($statistic = 0,07$ et $p = 0,97 > 0,05$). La différence de temps de réalisation entre les visualisations du graphe de 500 nœuds et 3294 arêtes n’est pas non plus statistiquement significative ($statistic = 0,49$

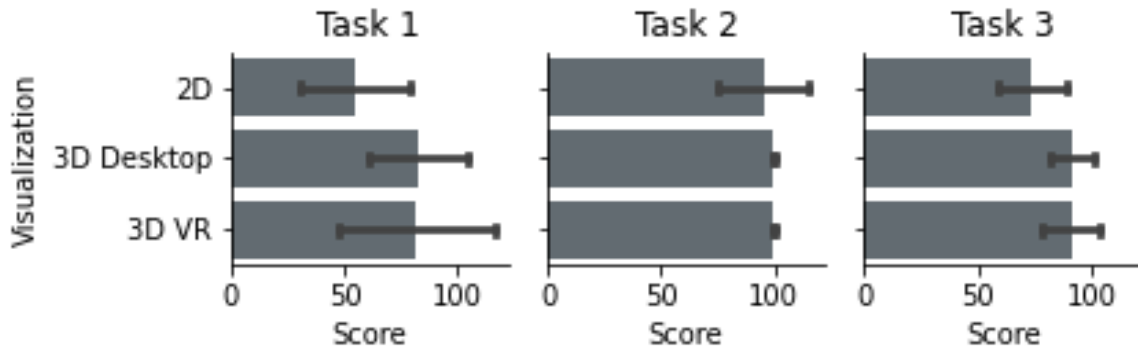


FIGURE 4.18 – Score d’efficacité (Graphe de 500 nœuds et 3294 arêtes) : moyennes et écarts-types du score d’efficacité par tâche et par visualisation.

et $p = 0,78 > 0,05$).

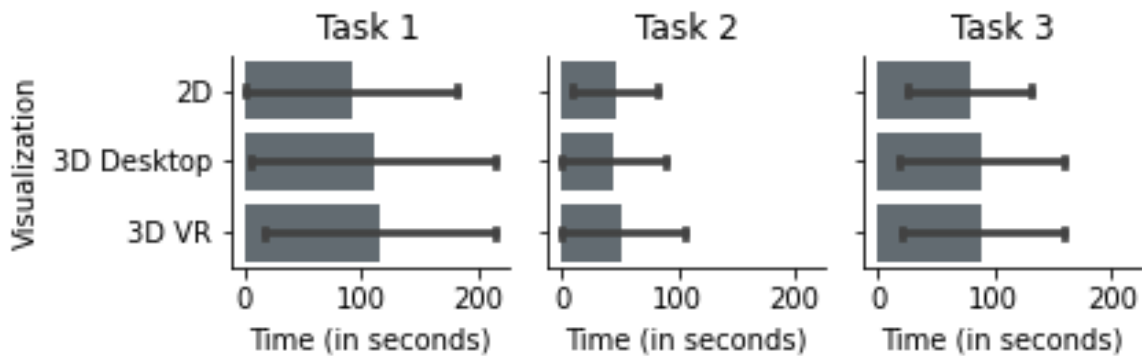


FIGURE 4.19 – Temps de réalisation (Graphe de 250 nœuds et 855 arêtes) : moyennes et écarts-types du temps de réalisation par tâche et par visualisation.

L’analyse du temps de réalisation a montré que la différence de temps de réalisation d’une tâche entre les différentes visualisations n’est pas statistiquement significative. Les figures 4.19 et 4.20 montrent quant à elles que les visualisations 3D ont le plus mauvais temps de réalisation par rapport à la visualisation 2D.

Expérience utilisateur : les résultats d’évaluation de l’utilisabilité des systèmes montrent que la visualisation 2D, la visualisation 3D desktop et la visualisation 3D immersive sont toutes utilisables (voir le tableau 4.14), car le score de chaque système est supérieur au seuil (70/100). Cependant, la visualisation 3D a obtenu un meilleur score d’utilisabilité par rapport à la visualisation 2D. Le test de Kruskal-Wallis montre quant à lui qu’il y a un effet du mode de visualisation sur l’utilisabilité, car le résultat du test

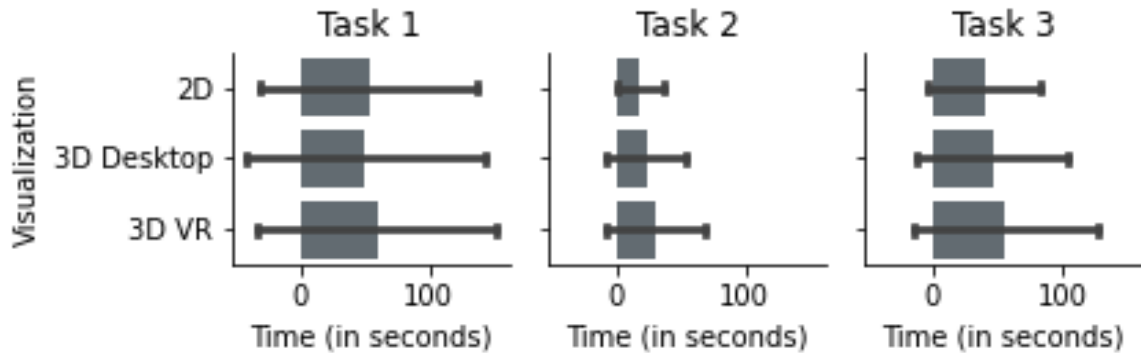


FIGURE 4.20 – Temps de réalisation (Graphe de 500 nœuds et 3294 arêtes) : moyennes et écarts-types du temps de réalisation par tâche et par visualisation.

statistique est égal à 14,32 et la p-value correspondante est égale à $10^{-3} < 0,05$. Le résultat des tests post-hoc (voir le tableau 4.15) montre que c'est la visualisation 2D qui est différente des autres visualisations. Par conséquent, elle est moins utilisable que les visualisations 3D et 3D immersive.

Score SUS	2D	3D desktop	3D VR
Moyenne	78,33	81,67	85,42
Écart-type	6,20	11,05	10,60

TABLE 4.14 – Score d'utilisabilité : moyennes et écart-types du score d'utilisabilité de chaque système de visualisation.

	2D	3D desktop	3D VR
2D	1	10^{-2**}	10^{-3***}
3D desktop	10^{-2**}	1	0,15
3D VR	10^{-3***}	0,15	1

TABLE 4.15 – Score d'utilisabilité : résultat des tests post-hoc en utilisant la méthode de Dunn [131] (p-value significative : * $p < 0,05$; ** $p < 0,01$; *** $p \leq 0,001$)

En ce qui concerne l'expérience utilisateur, tous les participants ont déclaré avoir compris ce que nous attendions d'eux. Ils ont également tous déclaré avoir compris les tâches demandées. Le tableau 4.16 récapitule d'autres critères d'évaluation de l'expérience des utilisateurs. Personne n'a déclaré avoir eu des difficultés à interagir avec l'environnement virtuel ; 5 participants sur 24 ont déclaré avoir eu une fatigue visuelle avec la visualisation

2D et avec la visualisation immersive. 5 participants 24 ont également déclaré avoir eu des maux de tête avec l’environnement virtuel.

	Difficulté d’interaction	Fatigue visuelle	Maux de tête
2D	1/24	5/24	0/24
3D desktop	2/24	3/24	1/24
3D VR	0/24	5/24	5/24

TABLE 4.16 – Récapitulatif d’autres critères d’évaluation de l’expérience utilisateur.

Nous avons également invité les participants à identifier la(les) visualisation(s) qui semble(nt) la(les) plus adaptée(s) pour visualiser des graphes de plusieurs centaines de nœuds, et à préciser les visualisations qu’ils préfèrent. Les participants pensent donc que la visualisation 3D immersive serait la plus adaptée par rapport à la visualisation 3D et à la visualisation 2D desktop (voir figure 4.21). Ils ont préféré les visualisations 3D, en l’occurrence la visualisation 3D immersive (voir la figure 4.22). La plupart des participants préfèrent la visualisation 3D immersive pour deux raisons. Premièrement, l’aspect immersif qui donne l’impression d’être dans un environnement réel. Deuxièmement, le fait qu’on peut interagir avec un graphe à l’aide des gestes naturels tels que la marche et les mouvements de la tête.

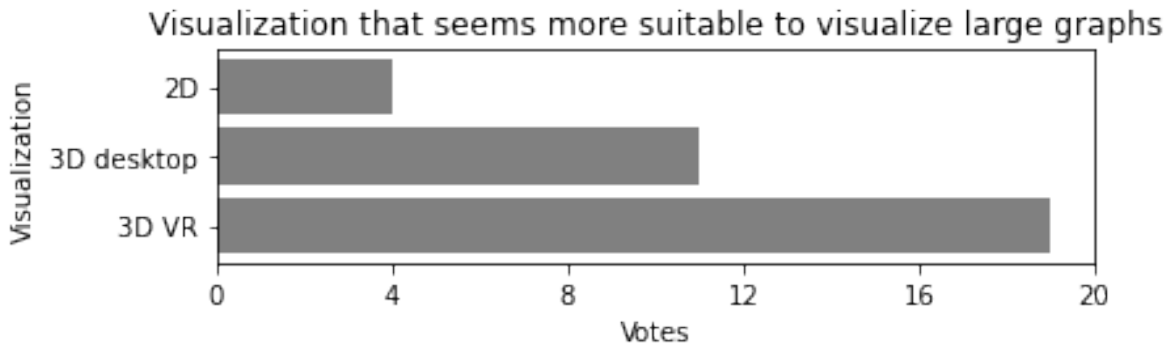


FIGURE 4.21 – Les participants pensent que la visualisation 3D immersive serait la mieux adaptée pour visualiser les graphes comportant plusieurs centaines nœuds.

4.5.2.8 Discussion

La première évaluation visait à comparer l’efficacité de notre proposition 3D par rapport aux représentations radiales 2D, tandis que la seconde avait pour but de comparer

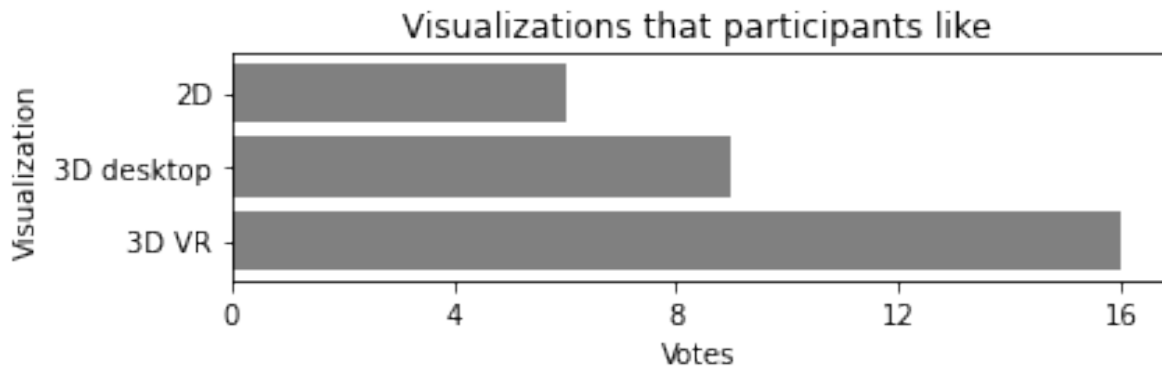


FIGURE 4.22 – Les participants préfèrent la visualisation 3D immersive.

l'efficacité de notre proposition 3D en immersion par rapport à la visualisation 3D et à la visualisation 2D sur un grand écran de 75 pouces. Nous voulions déterminer à travers ces évaluations si un type de visualisation est plus ou moins performant que toutes les autres pour une tâche donnée.

Les résultats de notre première évaluation (section 4.5.1.7) nous ont permis de montrer que la visualisation 3D est plus performante que la visualisation 2D pour les tâches liées au centre et à la périphérie d'un graphe. Par conséquent, nous nous attendions à ce que la visualisation 3D immersive soit plus performante que la visualisation 2D avec un graphe de 500 nœuds, car nous avons amélioré les interactions utilisateur.

Les tests statistiques du score d'efficacité des tâches 1 et 3 nous ont permis de montrer que nous avons de meilleures performances avec la visualisation 3D immersive en passant d'un graphe de 250 nœuds à un graphe de 500 nœuds, comparativement à la visualisation 2D (voir tableau 4.17). En effet, certains tests ont montré que la différence d'efficacité entre la visualisation 2D et la visualisation 3D immersive est bien plus significative que celle entre la visualisation 2D et la visualisation 3D desktop (voir le tableau 4.9). D'autres ont montré que passer d'un graphe de 250 nœuds à un graphe de 500 nœuds augmente également l'écart de performance en faveur de la visualisation 3D immersive (voir les tableaux 4.12 et 4.13).

Cependant, les tests statistiques ont montré que la différence de temps de réalisation entre la visualisation 2D et la visualisation 3D n'est pas statistiquement significative. Les résultats exploratoires (figures 4.19 et 4.20) montrent quant à eux que le temps de réalisation d'une tâche a été plus long avec la visualisation 3D qu'avec la visualisation 2D.

Les résultats d'évaluation de l'expérience utilisateur ont quant à eux montré que les

	Tâche 1	Tâche 2	Tâche 3
H7	✓	✗	✓

TABLE 4.17 – Validation de H7 (l'écart de performance entre la visualisation 3D immersive et la visualisation 2D augmente lorsqu'on augmente la taille des graphes) par rapport au score d'efficacité. Une coche (✓) indique une validation, tandis qu'une croix (✗) indique un rejet.

participants préfèrent la visualisation 3D (en l'occurrence la visualisation 3D immersive) et pensent qu'elle permet de mieux visualiser des graphes comportant plusieurs centaines de nœuds, contrairement à la visualisation 2D. De plus, l'évaluation de l'utilisabilité du système a permis d'obtenir un meilleur score pour la visualisation 3D immersive, comparativement à la visualisation 3D desktop et à la visualisation 2D (voir le tableau 4.14). Rappelons que l'évaluation de l'utilisabilité d'un système permet de mesurer la facilité d'utilisation de ce système.

Visualisation 2D vs Visualisation 3D vs Visualisation 3D immersive

Les résultats de la première évaluation nous ont permis de montrer que la visualisation 3D est plus performante pour les tâches où l'on s'intéresse au centre ou à la périphérie d'un graphe, comparativement à la visualisation 2D. Les résultats de notre seconde évaluation nous ont permis de montrer que la visualisation 3D immersive accentue ce gain de performance en passant d'un graphe de 250 nœuds et 855 arêtes à un graphe de 500 nœuds et 3294 arêtes. Cependant, nous ne sommes pas parvenus à démontrer qu'on a un gain de temps entre les différentes visualisations. Mais du point de vue d'un analyste cette absence de gain de temps n'a aucune importance et ce qui compte c'est l'amélioration de l'utilisabilité de la visualisation que nous obtenons en passant d'une visualisation 2D à une visualisation 3D. Par ailleurs, les participants préfèrent la visualisation 3D immersive et pensent qu'elle est la mieux adaptée pour visualiser les graphes comportant plusieurs centaines de nœuds.

4.6 Conclusion

Pour permettre à un analyste de pouvoir observer les zones denses ou comparer la centralité des nœuds dans un graphe, nous avons proposé une visualisation 3D. Elle consiste à projeter les représentations 2D radiales sur des surfaces 3D tout en gardant la vue 2D en vue de dessus. La projection sphérique accentue la visibilité des nœuds en périphérie d'un graphe alors que la projection torique donne une excellente visibilité des nœuds au centre du graphe. La projection conique répartit quant à lui les nœuds de manière uniforme.

Les projections 3D des représentations 2D mettant déjà l'emphase sur le centre ou sur la périphérie présentent également des avantages considérables. En effet, la projection torique de l'emphase sur le centre accentue la visibilité sur le centre du graphe. La projection sphérique de l'emphase sur la périphérie accentue quant à elle la visibilité sur la périphérie du graphe. Ces contributions ont été présentées dans les ateliers de visualisation **Atelier EGC-VIF, Newcastle University Visualization Workshop** [89] et dans une conférence internationale **International Conference on Augmented Reality, Virtual Reality and Computer Graphics (AVR)** [87].

En plus de la centralité des nœuds, notre proposition 3D permet également de mettre en évidence les connecteurs dans un graphe partitionné. Elle intègre également un algorithme de regroupement d'arêtes afin d'améliorer la visibilité de la connectivité entre les groupes de nœuds du graphe.

Les résultats de la première évaluation, qui a consisté à comparer l'efficacité de notre proposition par rapport à la visualisation 2D, ont montré que la visualisation 3D est plus performante que la visualisation 2D pour des tâches liées au centre et à la périphérie d'un graphe. Ces résultats ont été présentés dans une conférence internationale **International Conference on Computer Graphics, Visualization and Computer Vision** [88]. Ceux de la seconde évaluation, qui a consisté à comparer l'efficacité de la visualisation 3D immersive par rapport à la visualisation 3D desktop et à la visualisation 2D, ont montré que nous avons un gain de performance en faveur de la visualisation 3D immersive lorsqu'on augmente la taille des graphes. De plus, la visualisation 3D immersive a obtenu un meilleur score d'utilisabilité que la visualisation 2D. L'analyse statistique du temps de réalisation des tâches a quant à elle montré que la différence n'est pas significative entre les différentes visualisations.

VISUALISATION ÉGOCENTRIQUE DE GRAPHES

Une tâche essentielle dans la représentation de graphes consiste à déterminer un agencement géométrique approprié des nœuds et des arêtes. Plusieurs critères [32], tels qu'une minimisation de croisements d'arêtes, une homogénéité des longueurs des arêtes et une distribution rationnelle des nœuds, permettent d'évaluer la qualité d'une représentation de graphe. Les méthodes les plus couramment utilisées pour représenter un graphe sont les algorithmes de force (présentés dans la section 3.1.2). Cependant, ces algorithmes ne sont pas toujours adaptés aux graphes contenant plusieurs milliers de nœuds et d'arêtes [14].

D'autres algorithmes ont été proposés dans le but d'améliorer la représentation des graphes en termes de qualité et d'efficacité algorithmique, notamment pour les grands graphes. Certains de ces algorithmes sont basés sur une variante des méthodes de réduction de dimension MDS (présentée dans la section 3.1.3.2) [15, 14] et ont pour but de minimiser le « stress ». Dans ces approches, le but de la minimisation du stress est de déterminer la position des nœuds de sorte que la distance euclidienne entre deux nœuds se rapproche de leur distance théorique, c'est-à-dire la distance du plus court chemin entre ces nœuds. Toutefois, de telles approches basées sur les distances ont tendance à regrouper les nœuds pour des graphes ayant un faible diamètre (la plus grande distance entre deux nœuds d'un graphe, section 2.1.4), notamment les graphes qui représentent les données du monde réel.

Cependant, toutes ces méthodes peuvent être peu adaptées pour les grands graphes à cause du manque d'espace d'affichage. On peut noter une présence de beaucoup d'occlusions de nœuds et d'arêtes. Certaines méthodes (présentées dans la section 3.4) ont été proposées pour réduire l'occlusion en regroupant les arêtes, tandis que d'autres ont proposé d'exploiter la 3D pour bien distribuer les nœuds et pour améliorer la perception de la connectivité entre les nœuds [151, 107]. Les méthodes exploitant la 3D sont limitées lorsqu'elles sont associées à un environnement classique (affichage 2D), car les indices de profondeur en 3D peuvent rendre la compréhension d'un grand graphe plus complexe [161].

Des travaux ont suggéré qu’une immersion dans les données peut atténuer cette complexité et faire de la profondeur un canal viable pour encoder les données [60, 29]. Ainsi, plusieurs méthodes de visualisation de graphes en environnement immersif ont été proposées [101, 100, 19]. La plupart des méthodes utilisaient une approche exocentrique dans laquelle un utilisateur se trouvait à l’extérieur de la représentation d’un graphe [111] pour donner une vision d’ensemble du graphe, mais certains travaux suggèrent qu’une approche égocentrique peut être utile pour la visualisation des graphes [101]. Une visualisation égocentrique d’un graphe consiste par exemple à placer un analyste de graphes à l’intérieur d’un graphe. D’autres travaux soutiennent qu’une telle visualisation peut également augmenter l’engagement des utilisateurs et réduire les symptômes de cybermalaise qu’on peut observer avec une approche exocentrique [111, 139]. Toutefois, une étude de Yang et al. [158] a montré que l’exploration des cartes dans une perspective égocentrique entraînait des performances inférieures à celles de toute autre représentation en environnement immersif.

Afin d’exploiter le potentiel de la 3D immersive et améliorer l’exploration d’un graphe, nous proposons une nouvelle métaphore de visualisation égocentrique avec un espace rétractable où un utilisateur peut passer de manière flexible d’une vue détaillée égocentrique à une vue d’ensemble du graphe tout en restant dans un mode égocentrique. Cependant, nous n’avons pas encore pu mener d’expérimentation afin de valider notre approche par une évaluation de performance et une étude de cas avec des utilisateurs. Dans ce chapitre, nous présentons des travaux relatifs à la visualisation égocentrique de graphes, notre approche égocentrique et ses avantages et ses limites.

5.1 Travaux relatifs à la visualisation égocentrique

Adapter la visualisation à de nouvelles modalités, telles que les environnements immersifs, peut nécessiter de nouvelles façons de représenter et d’interagir avec les données [130]. Plusieurs travaux ont donc exploré des moyens de manipuler, naviguer et représenter de manière interactive des graphes en réalité virtuelle. Kwon et al. [101, 100] ont introduit une approche égocentrique dans laquelle un graphe est positionné sur une sphère de sorte que tous les nœuds du graphe soient également visibles pour l’utilisateur (voir les figures 5.1 et 5.2). Les arêtes sont regroupées afin de réduire l’occlusion des nœuds. Ils ont ensuite comparé les performances de leur méthode avec la visualisation classique 2D à l’aide d’une étude utilisateur (figure 5.3) [100]. Les interactions sont faites à l’aide de la souris

et les tâches de l'évaluation ont porté sur la recherche de voisins communs à deux nœuds donnés, la recherche d'un nœud qui a le plus de nœuds voisins, la recherche d'un chemin entre deux nœuds et le rappel de la position de certains nœuds. Les résultats de leur évaluation ont montré que les participants ont obtenu de meilleurs résultats avec moins d'interactions en utilisant la visualisation sphérique qu'en utilisant la visualisation 2D, notamment pour les tâches plus difficiles et les grands graphes.

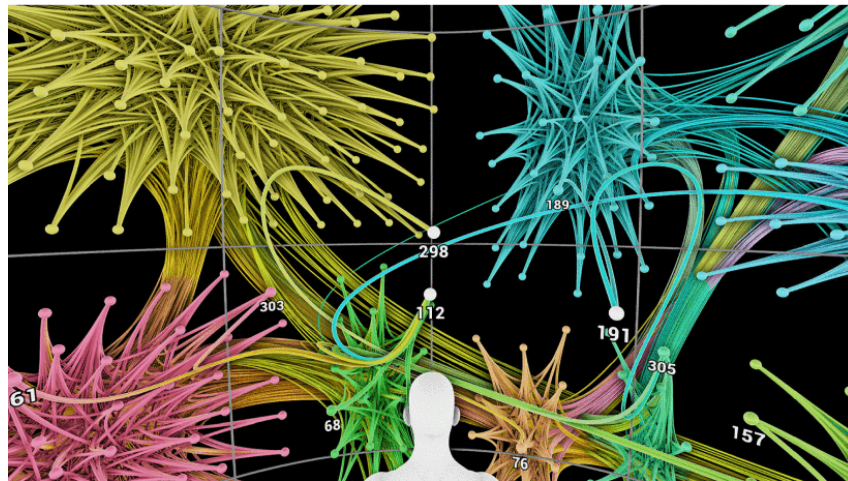


FIGURE 5.1 – Visualisations sphériques de graphes proposées par Kwon et al. [100]. L'utilisateur est placé au centre de la sphère sur laquelle le graphe est positionné (© 2016, IEEE).

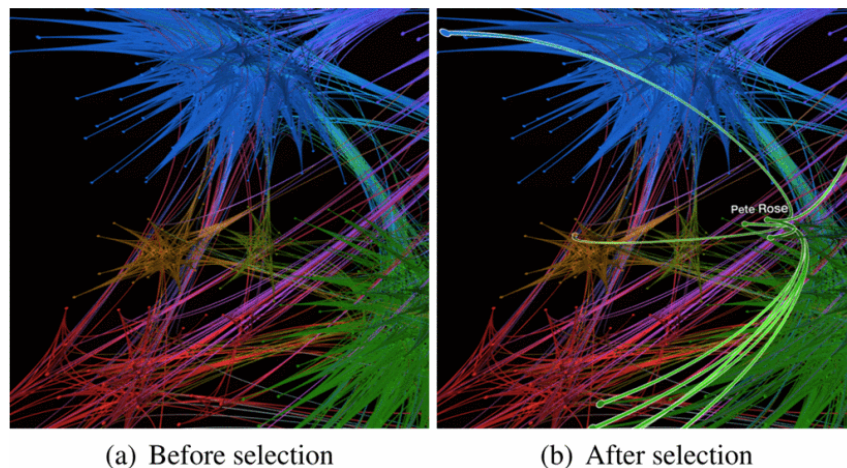


FIGURE 5.2 – Visualisations sphériques de graphes proposées par Kwon et al. [101]. Lorsqu'un utilisateur sélectionne un nœud, celui-ci et ses nœuds voisins sont rapprochés de l'utilisateur (© 2015, IEEE).

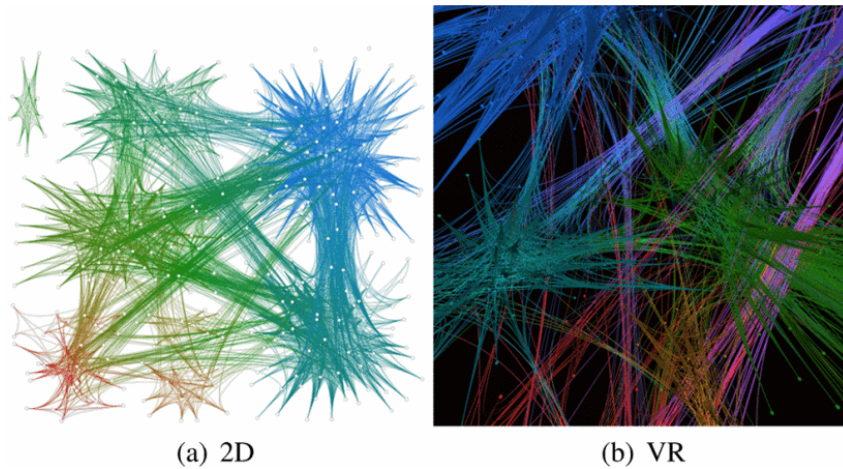


FIGURE 5.3 – Dans la visualisation 2D le graphe peut être entièrement visible, mais il est difficile de discerner les connectivités de certains nœuds. Dans l’environnement immersif, seule une petite section du graphe est visible à un moment donné, mais la structure est plus tangible (© 2015, IEEE).

Pour améliorer l’exploration d’un graphe dans un environnement immersif, Sorger et al. [139] ont proposé deux visualisations égocentriques (*Ego-highlight* et *Ego-bubble*) dans lesquelles un utilisateur peut prendre la place d’un nœud afin d’analyser sa connectivité (voir la figure 5.4). Dans l’approche *Ego-highlight* l’utilisateur peut prendre la place d’un nœud et identifier les nœuds voisins qui sont automatiquement mis en évidence. Les arêtes connectant ce nœud à ses voisins sont supprimées de la vue de l’utilisateur afin de réduire l’encombrement visuel. Mais on peut noter l’existence d’occlusions et certains nœuds peuvent être très éloignés de l’utilisateur à cause de l’algorithme de force utilisé. L’approche *Ego-bubble* est quant à elle une extension de l’approche *Ego-highlight*. Afin d’atténuer davantage l’occlusion dans la vue locale de l’utilisateur, les nœuds voisins directs sont déplacés pour être uniformément répartis autour de l’utilisateur. Les arêtes passant à proximité de l’utilisateur sont coupées pour éviter les obstructions de vue. Sorger et al. [139] espéraient que l’exploration d’un graphe à l’aide d’une visualisation égocentrique puisse conduire à un compromis qui affecte plusieurs facteurs tels que l’efficacité de la recherche visuelle, la performance de navigation, l’orientation spatiale et le cybermalaise. Ainsi, ils ont ensuite effectué une étude expérimentale pour quantifier ces compromis, mais aussi pour évaluer les avantages et les limites de l’égocentrisme pour l’exploration de graphes en environnement immersif. Leur évaluation visait à comparer l’efficacité et l’efficience de certaines tâches d’analyse de graphes entre une visualisation classique en environnement immersif et leurs méthodes. La visualisation classique est le

résultat d'un algorithme de force et un utilisateur peut se rapprocher d'un nœud pour analyser sa connectivité. Il peut cependant être gêné par des arêtes et on peut noter l'existence d'occlusions. L'évaluation reposait sur les tâches d'analyse de graphes telles que la recherche d'un nœud voisin, la recherche de nœuds voisins communs, l'estimation du degré d'un nœud donné (comptage du nombre d'arêtes) et la recherche de chemin entre deux nœuds.

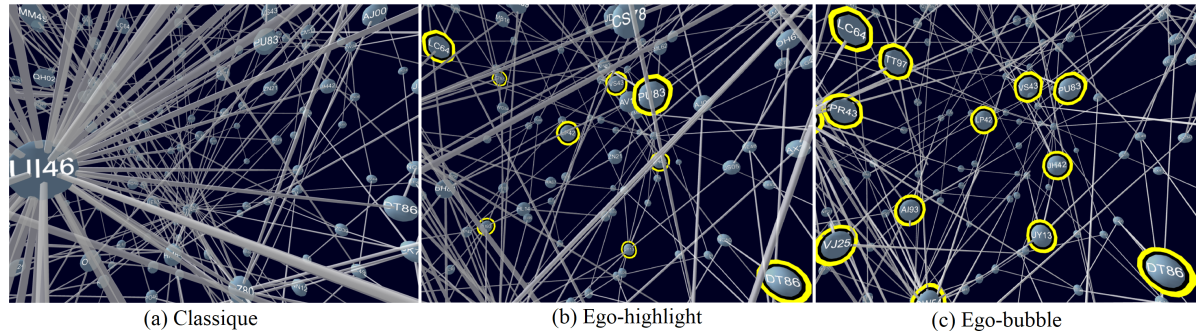


FIGURE 5.4 – Exploration immersive des voisins du nœud *UI46*. (a) Classique : l'utilisateur inspecte le nœud *UI46* de près. (b) Ego-highlight : l'utilisateur prend la position du nœud *UI46*, les voisins directs sont mis en évidence en jaune. (c) Ego-bubble : les voisins directs sont positionnés de manière équidistante par rapport à l'utilisateur. (Images extraites de Sorger et al. [139]).

Les résultats de l'évaluation confirment leur hypothèse de base que les visualisations égocentriques de graphes sont plus efficace qu'une visualisation immersive classique. Sorger et al. [139] soutiennent que la recherche visuelle est clairement facilitée par une visualisation égocentrique désencombrée. En ce qui concerne le cybermalaise la visualisation immersive classique a obtenu les notes moyennes les plus élevées. Ainsi, les résultats de leur évaluation ont montré qu'une visualisation égocentrique peut augmenter considérablement l'efficacité de la recherche visuelle. Ces résultats suggèrent également que l'égocentrisme peut réduire les symptômes de cybermalaise, comparativement aux visualisations immersives traditionnelles.

Cependant, pour avoir une vision globale d'un graphe, il faudra quitter le mode égocentrique proposé par Sorger et al. Certains nœuds apparaîtront très éloignés, et par conséquent, il sera difficile d'observer leur connectivité. Ainsi, l'approche égocentrique proposée par Kwon et al. [101] semble plus pratique pour une vision d'ensemble d'un graphe. Dans les approches Ego-highlight et Ego-bubble la mise en évidence des voisins d'un nœud n'est possible que si un utilisateur prend la place de celui-ci, alors qu'un analyste de graphes pourrait s'intéresser à identifier les voisins d'un nœud sans forcément

se rapprocher de celui-ci. Dans l’approche Ego-bubble les arêtes passant à proximité de l’utilisateur sont coupées pour éviter les obstructions de vue. Toutefois, cela peut perturber un analyste de graphes dans l’identification de certaines connectivités si celles-ci sont coupées. Les voisins directs d’un nœud sont positionnés de manière équidistante par rapport à l’utilisateur dans l’approche Ego-bubble. Cependant, cela pourrait occasionner une perte d’orientation de l’utilisateur par le déplacement des positions des nœuds même si Sorger et al. [139] ont proposé d’animer ce déplacement lorsque l’utilisateur passe à un autre nœud.

En conclusion, les approches égocentriques de Sorger et al. [139] permettent à un analyste de graphe d’étudier la connectivité d’un nœud en prenant sa place dans le graphe. Cependant, ces approches sont peu pratiques pour avoir une vision d’ensemble du graphe. De plus, elles n’améliorent pas le résultat de l’algorithme de spatialisation des nœuds. Par exemple, si l’algorithme de force utilisé fournit un résultat où certains nœuds apparaissent groupés, ce résultat serait utilisé tel quel. Par conséquent, un analyste pourrait avoir des difficultés à identifier certaines connectivités. Ainsi, améliorer la spatialisation améliorerait la perception de la connectivité entre les nœuds. Dans une approche immersive traditionnelle, le fait qu’un utilisateur se retrouve empêtré dans les nœuds et les arêtes peut occasionner un inconfort pendant la visualisation. Cela pourrait également causer ou augmenter le cybermalaise. Le fait de couper les arêtes à proximité de l’utilisateur peut affecter la compréhension de certaines connectivités. Il peut donc être utile de représenter différemment ses arêtes au lieu de les couper.

5.2 Métaphore d’un espace de visualisation rétractable

5.2.1 Concept

Selon notre analyse de l’état de l’art, notre métaphore égocentrique vise à améliorer le résultat d’un algorithme de spatialisation des nœuds et la représentation des arêtes. Elle vise également à fournir une vision d’ensemble du graphe.

Les résultats d’évaluation des approches égocentriques proposées par Sorger et al. [139] ont montré qu’une visualisation égocentrique et une conception d’interaction adaptée peuvent apporter des avantages significatifs dans l’exploration immersive de graphes. Ainsi, afin d’améliorer la spatialisation et d’éviter qu’un utilisateur se retrouve empê-

tré dans les nœuds et les arêtes, nous créons une zone vide dans son entourage proche. Cette zone vide est une sphère d'un rayon r_{int} minimum de 2,5m de sorte que la sphère enveloppe entièrement l'utilisateur en immersion. Notons que l'utilisateur a la possibilité de sortir de cette zone (voir figure 5.5b). Nous positionnons ensuite les nœuds (résultat d'un algorithme de spatialisation en 3D) entre la zone de l'utilisateur et une autre sphère de rayon $r_{ext} > r_{int}$. La valeur minimale de r_{ext} est par défaut égale à $2 \times r_{int}$ et sa valeur maximale est fonction du nombre de nœuds dans le graphe. La valeur maximale de r_{int} est définie en fonction de la valeur courante de r_{ext} . L'équation 5.1 illustre le calcul de la position d'un nœud i entre les sphères :

$$P'_i = d'_i \cdot \frac{P_i}{\|P_i\|} \quad (5.1)$$

où :

- P_i est la position du nœud i à l'aide d'un algorithme de spatialisation en 3D ;
- P'_i la position du nœud i entre les sphères ;
- $d'_i = r_{int} + \frac{\|P_i\|}{d_{max}} \cdot (r_{ext} - r_{int})$ est la nouvelle distance entre un nœud i et le centre, et d_{max} est la distance entre un nœud le plus éloigné et le centre.

Pour améliorer la représentation des arêtes, nous ne laissons aucune arête traverser l'espace vide de l'utilisateur (la sphère interne). Pour cela, nous repoussons les arêtes dans la zone délimitée par nos deux sphères concentriques (figure 5.5a). Ainsi, l'utilisateur peut se déplacer librement dans son espace sans être encombré par des arêtes. Les figures 5.6a et 5.6b présentent une visualisation 3D du graphe « Karate Club » (34 nœuds et 78 arêtes) avec une spatialisation des nœuds effectuée, respectivement, par l'algorithme de Fruchterman et Reingold [50] et par notre approche basée sur une métaphore égocentrique. Le résultat de la figure 5.6b montre une bonne expansion des nœuds avec notre méthode, comparativement au résultat de l'algorithme de Fruchterman et Reingold (figure 5.6a).

Avec les approches égocentriques de Sorger et al. [139] la vision d'ensemble implique que l'utilisateur s'éloigne du graphe. Il peut cependant perdre la perception de certains nœuds et de leur connectivité. Il en est de même pour une méthode immersive classique. Avoir une vue détaillée seule ne peut pas permettre d'identifier des zones intéressantes dans un graphe. Ainsi, il peut être utile d'associer une vision d'ensemble d'un graphe à une vision égocentrique pour permettre à un analyste de passer d'une vue à l'autre. Cela introduit cependant plusieurs nouveaux défis. Comment afficher les nœuds qui sont :

- derrière un utilisateur ?

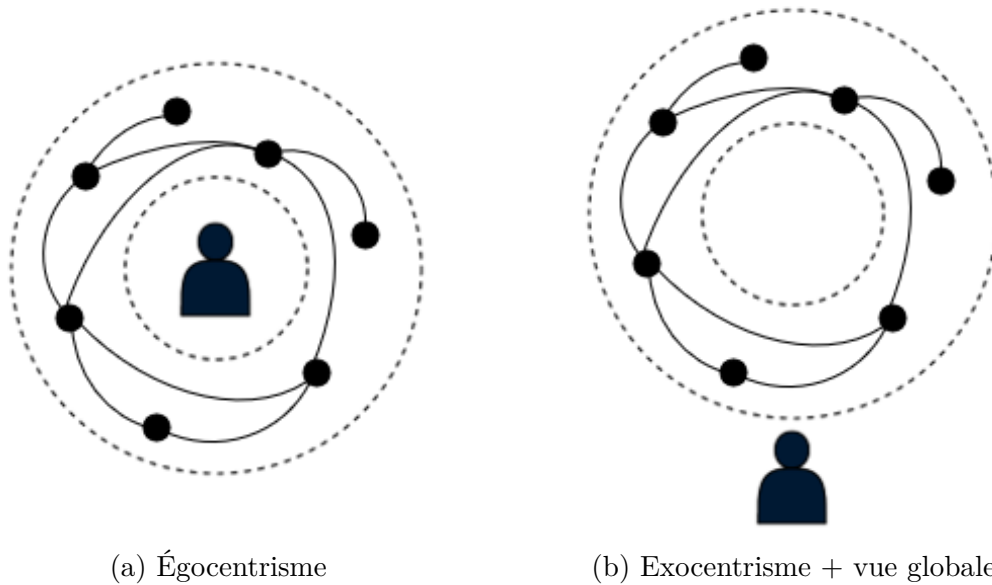


FIGURE 5.5 – Illustration de notre métaphore de visualisation égocentrique (une vue de coupe) : (a) l'utilisateur est placé au centre de la sphère interne et certaines arêtes sont projetées en dehors ; (b) vision d'ensemble du graphe en sortant des sphères, mais certains nœuds apparaîtront éloignés de l'utilisateur.

- en dehors du champ de vision d'un utilisateur ?

Pour cela, nous proposons une possibilité de rétracter l'espace d'affichage du graphe selon les axes vertical et horizontal (voir la figure 5.7) tout en restant égocentré. À l'aide d'un curseur permettant de contrôler l'axe horizontal et d'un curseur permettant de contrôler l'axe vertical, un analyste peut rétracter horizontalement et/ou verticalement l'espace d'affichage d'un graphe pour pouvoir obtenir une vue d'ensemble du graphe sans modifier sa structure. Ce type de visualisation égocentrique du graphe pourrait offrir des avantages perceptifs qui permettent à un analyste de comprendre la structure du graphe plus rapidement qu'avec une visualisation exocentrique. La figure 5.8 illustre quelques aspects de la rétractation de l'espace d'affichage du graphe « Karate Club » (34 nœuds et 78 arêtes) [160].

5.2.2 Synthèse

Nous avons proposé une nouvelle métaphore égocentrique de visualisation de graphes qui consiste à créer une zone vide autour d'un utilisateur afin d'éviter qu'il soit entravé

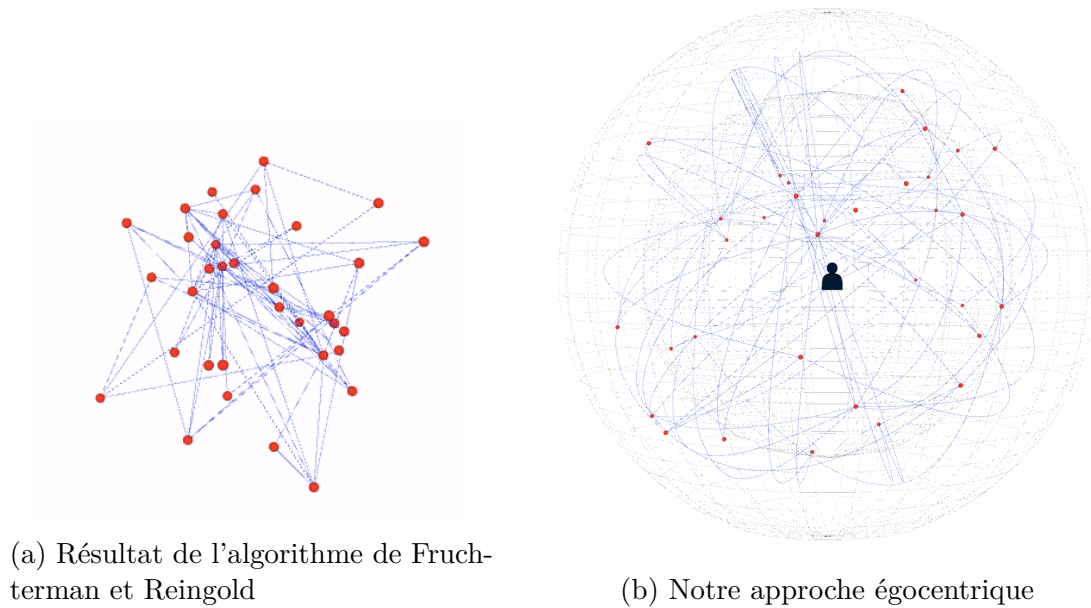


FIGURE 5.6 – Visualisation du graphe « Karate Club » (34 nœuds et 78 arêtes) [160]. (a) visualisation à l'aide de l'algorithme de Fruchterman et Reingold [50] ; (b) notre métaphore égocentrique où le graphe est positionné entre deux sphères concentriques.

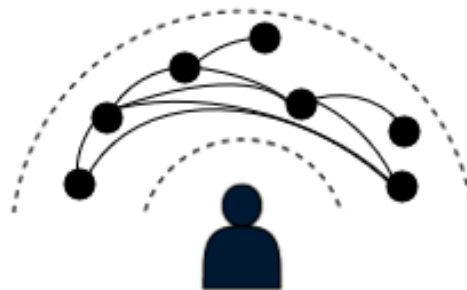


FIGURE 5.7 – Vision d'ensemble : un utilisateur peut rétracter l'espace d'affichage du graphe afin d'avoir une vue globale tout en restant égocentré.

par les arêtes et à positionner un graphe entre cette zone et une sphère. Cette approche permet d'améliorer l'expansion des nœuds dans l'espace, contrairement aux approches égocentriques proposées par Sorger et al. [139]. Nous avons ensuite repoussé les arêtes dans la zone délimitée par nos deux sphères concentriques pour désencombrer l'espace utilisateur. Nous avons également proposé une possibilité de rétracter l'espace d'affichage

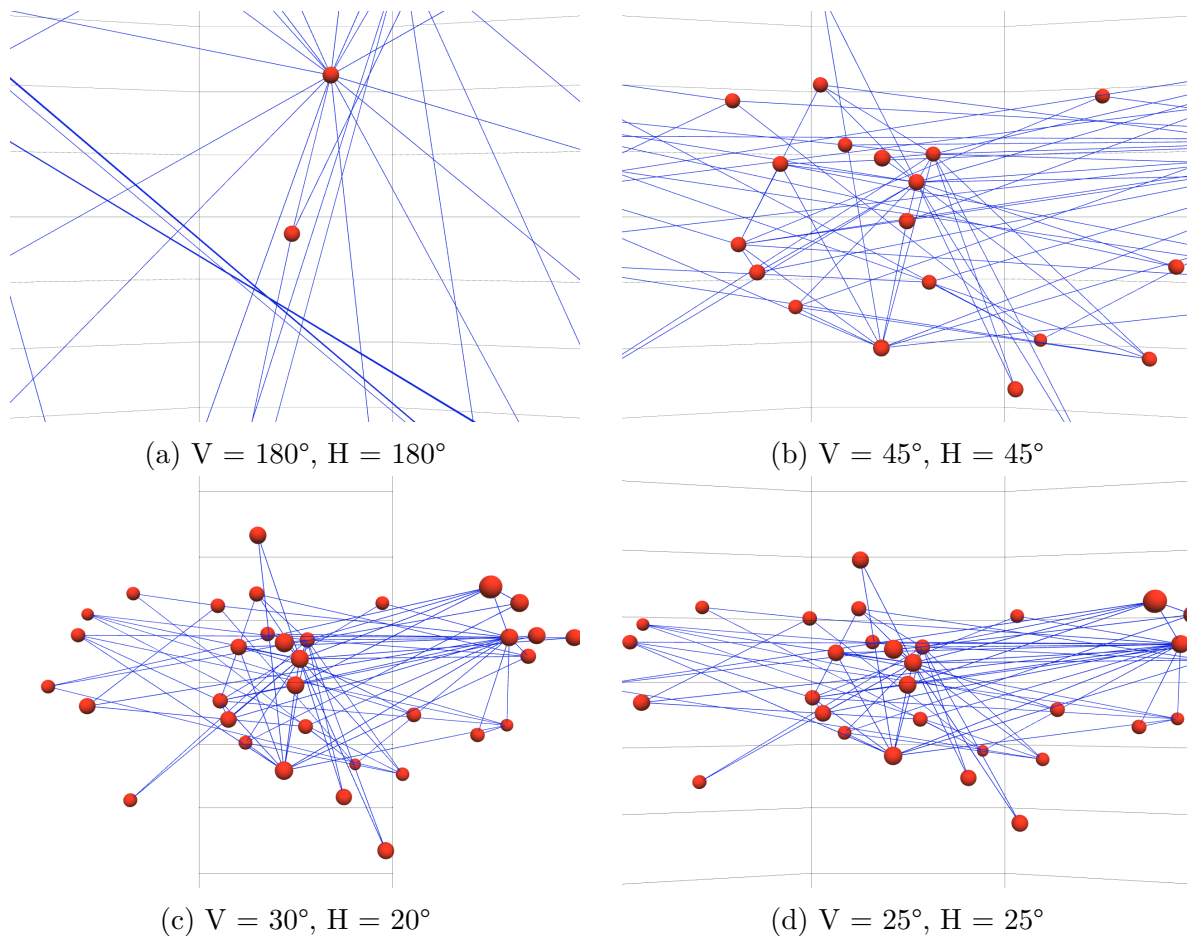


FIGURE 5.8 – Visualisation égocentrique (vue de l’intérieur) du graphe « Karate Club » (34 nœuds et 78 arêtes) [160] en rétractant l’espace d’affichage selon les axes vertical et horizontal : \mathbf{V} = angle selon l’axe vertical et \mathbf{H} = angle selon l’axe horizontal.

afin d’obtenir une vision globale du graphe sans quitter l’égocentrisme. Cependant, nous n’avons pas encore pu mener d’expérimentation afin de valider notre proposition à l’aide d’une évaluation de performance.

5.2.3 Discussion

L’encombrement visuel est toujours un problème crucial des techniques de visualisation de certains graphes, et par conséquent, de nombreuses améliorations sur la visualisation de graphes continuent d’être présentées [133], en l’occurrence des techniques immersives.

Cependant, certains travaux ont montré que les environnements virtuels présentent des inconvénients tels que les cybermalaises [137, 138]. Mais les résultats d’évaluation de

la métaphore égocentrique proposée par Sorger et al. [139] suggèrent que l'égocentrisme a le potentiel de surmonter certains des inconvénients inhérents aux environnements virtuels (par exemple, l'encombrement visuel et le cybermalaise).

Notre approche égocentrique pourrait donc contribuer à améliorer l'exploration d'un graphe puisque l'utilisateur dispose d'un espace libre autour de lui. Repousser les arêtes entre la zone délimitée par nos deux sphères concentriques pourrait également améliorer la visibilité de la connectivité entre les nœuds du graphe. Notre approche d'espace rétractable peut permettre à un analyste de passer d'une vue détaillée à une vue d'ensemble du graphe et inversement. Cette approche pourrait être utile pour identifier des zones denses du graphe en mode globale et ensuite pour comprendre la connectivité entre les nœuds d'une zone donnée.

Cependant, le passage d'une vue détaillée à une vue d'ensemble en rétractant l'espace d'affichage du graphe peut occasionner la présence d'occlusions d'arêtes. Un algorithme de regroupement d'arêtes pourrait alors être une alternative pour réduire cet encombrement visuel et améliorer la lisibilité du graphe.

APPLICATION DE NOS VISUALISATIONS À L'ANALYSE D'UN RÉSEAU SOCIAL

Dans ce chapitre, nous allons illustrer l'instanciation de nos propositions à l'analyse des réseaux sociaux, notamment comment nos propositions de visualisation pourraient aider à l'analyse du réseau des contributeurs de la plateforme de financement participatif que nous avons présenté dans la section 2.2, page 44.

Rappelons que nous avons identifié deux types d'experts qui ont des objectifs différents : un économiste et un data scientist. Les objectifs de l'économiste étaient de déterminer s'il existe une structure communautaire sous-jacente à la plateforme de financement participatif et de caractériser l'influence de ces communautés et de certains individus sur la performance des campagnes de levée de fonds. Le data scientist devait quant à lui aider à modéliser le graphe permettant de représenter les interactions sur la plateforme de financement participatif, à choisir les propriétés topologiques à considérer lors de l'étude et à conseiller sur le choix de l'algorithme de détection de communautés à utiliser.

Rappelons également que la visualisation d'un graphe permet d'obtenir une intuition pour ensuite identifier des zones denses, mais aussi des individus particuliers (les individus centraux, par exemple) puis émettre des hypothèses. Elle permet également de présenter les résultats d'analyses sur le graphe.

Dans cette étude de cas appliquée à un réseau social, les clusters identifiés sont des communautés. Une vision d'ensemble d'un graphe de communautés peut permettre une compréhension facile des connectivités entre communautés et une analyse rapide de l'ensemble du graphe en identifiant par exemple les communautés les plus denses. Elle peut ensuite permettre de se focaliser sur une communauté particulière : par exemple une communauté dont les individus sont densément connectés entre eux ou une communauté connectée à beaucoup d'autres communautés.

6.1 Analyse des relations intercommunautés

Pour vérifier l'hypothèse de l'existence d'une structure de communauté sous-jacente à la plateforme de financement participatif, le graphe de co-contributions a été découpé en 23 communautés (figure 6.1) à l'aide de l'algorithme de Louvain [8]. Cependant, il est un peu complexe d'identifier toutes les communautés détectées dans la figure 6.1. Nous avons donc choisi de construire un méta graphe où un nœud représente une communauté et où une arête représente l'ensemble des arêtes entre deux communautés connectées. Les figures 6.3 et 6.2b montrent la visualisation de ce méta graphe avec respectivement notre visualisation égoцентриque et notre visualisation radiale 3D.

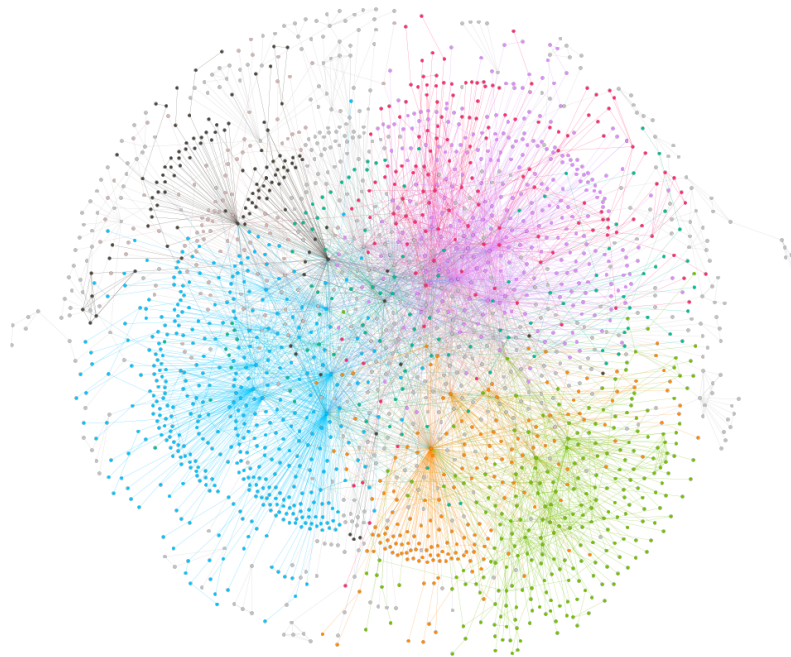
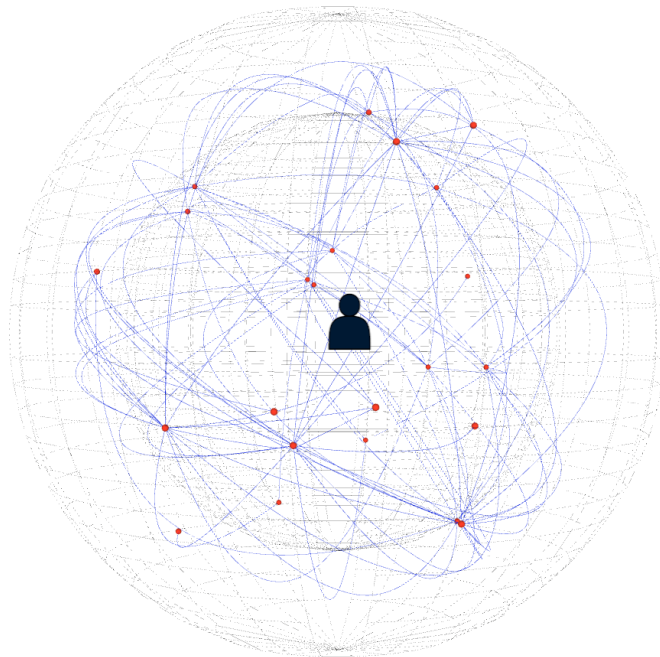
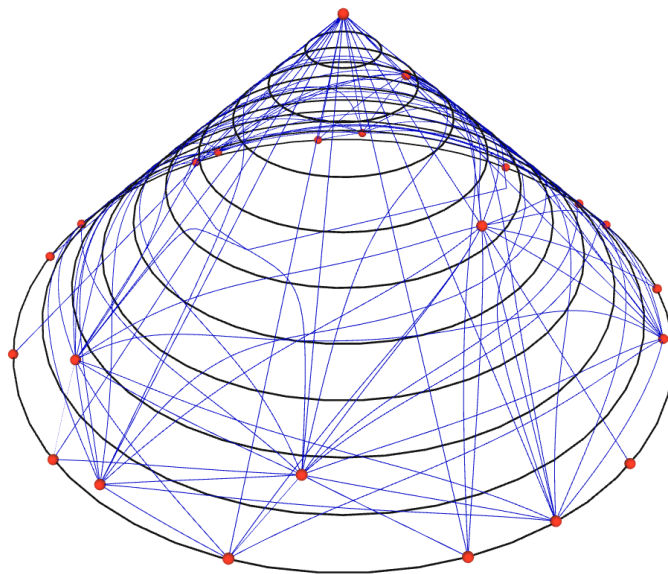


FIGURE 6.1 – Visualisation des 23 communautés identifiées à l'aide de l'algorithme de Louvain : chaque communauté est représentée d'une couleur différente des autres.



(a) Visualisation égocentrique (vue de l'extérieur)



(b) Visualisation radiale 3D : mise en évidence de la centralité d'intermédierité.

FIGURE 6.2 – Visualisation du méta graphe de co-contributions (23 nœuds et 98 arêtes). Un nœud représente une communauté et une arête représente l'ensemble des connexions entre deux communautés connectées.

Nos visualisations pourraient aider à mieux comprendre les connectivités entre les communautés, comparativement à l'approche 2D utilisée. La possibilité de rétracter l'espace d'affichage (figure 6.3 qu'offre notre visualisation égocentrique pourrait aider à obtenir une vision d'ensemble du méta graphe afin de mieux identifier par exemple les communautés les plus connectées : dans notre cas il s'agirait d'identifier les nœuds qui ont beaucoup de voisins. Cependant, notre visualisation égocentrique ne pourrait pas permettre de compa-

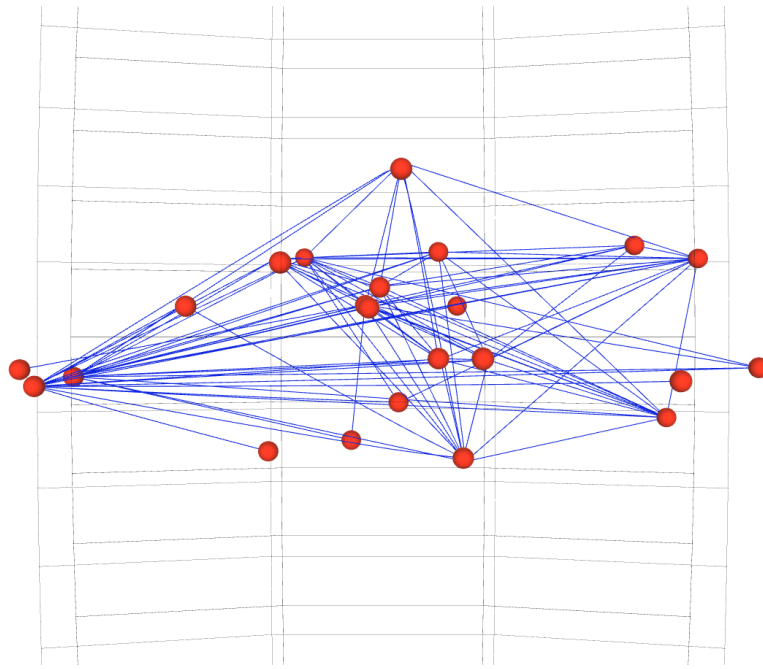


FIGURE 6.3 – Visualisation égocentrique (vue de l'intérieur) du méta graphe avec rétraction de l'espace (angle selon l'axe horizontal = 25° , angle selon l'axe vertical = 25°).

rer la centralité des communautés, contrairement à notre visualisation radiale 3D qui offre cette possibilité. En effet, la figure 6.2b fournit une vision d'ensemble du graphe et pourrait permettre de comparer visuellement la position topologique des communautés pour ensuite se focaliser sur une communauté donnée. Rappelons que dans notre cas chaque communauté est représentée par un nœud. Ainsi, notre visualisation radiale 3D pourrait aider à identifier rapidement les communautés les plus centrales ou les plus influentes dans le réseau : dans notre cas les communautés influentes sont proches du centre, tandis que les communautés les moins influentes sont vers la périphérie du méta graphe.

En définitive, la vision d'ensemble du graphe de co-contributions à l'aide de nos visualisations égocentrique et radiale 3D pourrait aider à identifier les communautés les plus connectées, comparativement à la visualisation 2D. Notre visualisation radiale 3D pour-

rait en plus aider à comparer visuellement la centralité des communautés pour ensuite se focaliser sur une communauté particulière, notamment la plus influente (forte centralité d'intermédiarité). Nous présentons dans la section suivante une analyse des relations entre les individus de la communauté la plus centrale.

6.2 Analyse des relations intracommunauté

À partir de la visualisation radiale 3D du méta graphe (figure 6.2b), nos experts (l'économiste et le data scientist) peuvent choisir d'approfondir leur compréhension des relations entre les contributeurs d'une communauté. Pour cela, il suffirait de faire un clic droit sur un nœud représentant une communauté pour afficher tous les individus de cette communauté. Nous avons choisi d'explorer les individus de la communauté la plus centrale par rapport à la centralité d'intermédiarité. Cette communauté est également la plus grande et est représentée par un graphe contenant 419 nœuds et 695 arêtes (voir la figure 6.4). Notre visualisation radiale 3D pourrait aider à comparer facilement la position topologique des individus de cette communauté.

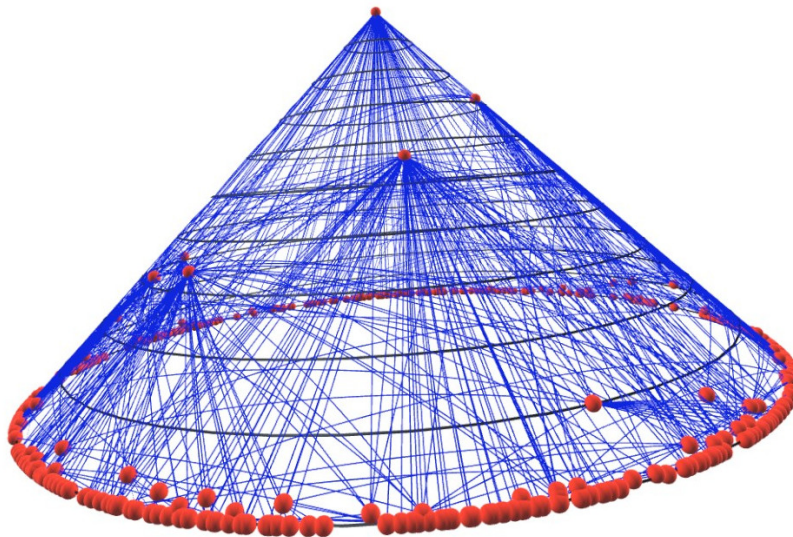


FIGURE 6.4 – Centralité d'intermédiarité : visualisation des individus de la communauté la plus centrale. Les individus les plus influents de cette communauté sont proches du centre, tandis que les moins influents sont vers la périphérie de la communauté.

En plus de la comparaison de la centralité d'intermédiarité, notre visualisation radiale

3D pourrait également aider nos experts à visualiser les connecteurs de cette communauté (voir les figures 6.5 et 6.6). Rappelons qu'un connecteur dans ce cas est un individu qui est connecté à des individus qui sont en dehors de la représentation visuelle de la communauté, et donc pas forcément visualisés. La visualisation des connecteurs pourrait être indispensable ici, car nous nous focalisons sur une communauté sans afficher le reste du graphe afin de conserver l'information topologique. Elle permettrait à nos experts de connaître le nombre de communautés voisines par connecteur, mais aussi le nombre d'individus voisins dans chacune des communautés. Ainsi, notre visualisation radiale 3D pourrait aider à visualiser à la fois la centralité des individus et les connecteurs d'une communauté.

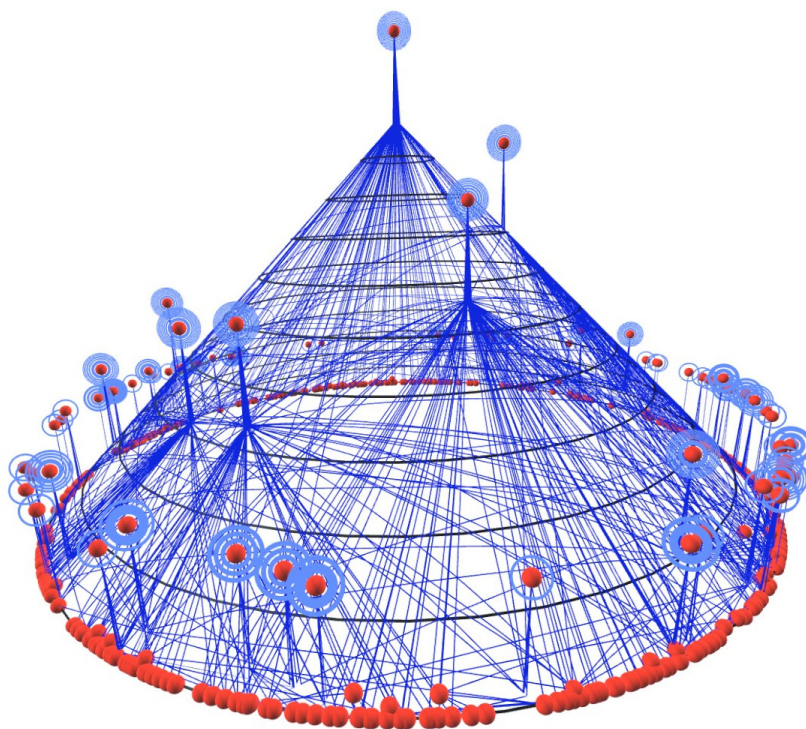


FIGURE 6.5 – Mise en évidence de la centralité d'intermédiation et des connecteurs dans la communauté la plus centrale. Une communauté voisine est représentée par un anneau autour d'un connecteur. L'épaisseur d'un anneau représente le nombre d'individus voisins du connecteur dans la communauté voisine. Les anneaux les plus épais sont au centre.

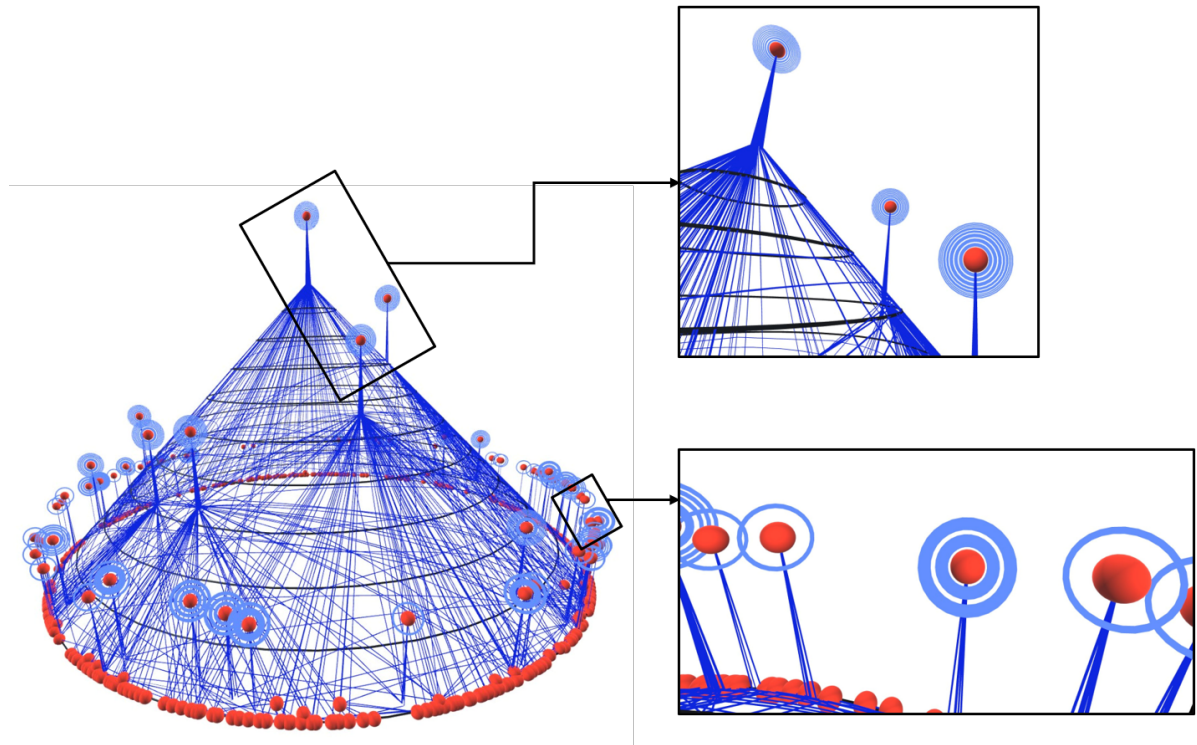


FIGURE 6.6 – Zoom sur certains connecteurs de la communauté la plus centrale. Une communauté voisine est représentée par un anneau autour d'un connecteur. L'épaisseur d'un anneau représente le nombre d'individus voisins du connecteur dans la communauté voisine. Les anneaux les plus épais sont au centre.

En conclusion, notre visualisation radiale 3D pourrait aider nos experts à comparer visuellement la centralité d'intermédiation et à identifier les connecteurs d'une communauté afin d'approfondir leur compréhension des interactions entre les individus d'une communauté sans perdre de vue certaines interactions avec les communautés voisines.

CONCLUSION ET PERSPECTIVES

Conclusion

L'objectif de nos travaux était d'étudier la contribution de la 3D immersive comparativement à la visualisation 2D traditionnelle afin de simplifier l'exploration et l'analyse des relations intracommunautés et intercommunautés représentées sous forme de graphes. Pour cela, nous avons proposé d'utiliser certaines propriétés et mesures sur les graphes et d'exploiter la 3D pour proposer de nouvelles approches de visualisation de graphes. Les applications de ces travaux sont tout aussi civiles que militaires.

Les graphes peuvent aider à répondre à de nombreuses questions, telles que l'identification des nœuds ayant une caractéristique particulière (forte centralité, connecteur entre groupes, etc.). La visualisation de tels graphes est pertinente pour susciter des questions chez un analyste afin d'émettre des hypothèses. Elle est ainsi considérée comme un outil d'accompagnement de la recherche, notamment pour un data scientist. Nous avons fait des contributions pour donner une vision globale d'un graphe tout en mettant en valeur la centralité des nœuds. Notre approche de solution permet également de s'intéresser à la fois à la visualisation des connexions entre des nœuds et à la visualisation des connexions entre communautés. Pour cela, nous avons d'abord étudié deux classes d'algorithmes de représentation des graphes les plus couramment utilisés : les algorithmes basés sur un paradigme de force et les algorithmes utilisant une approche de réduction de dimension. Certains algorithmes de force s'adaptent mieux aux graphes de taille relativement réduite, tandis que d'autres sont plus adaptés aux grands graphes. Les algorithmes basés sur la réduction de dimension semblent quant à eux être plus adaptés pour mettre en évidence les structures locales des graphes.

Nous avons ensuite étudié des méthodes permettant de mettre en évidence, d'une part la centralité des nœuds dans un graphe et d'autre part les connecteurs dans un graphe partitionné. Cependant, les méthodes classiques 2D mettant en évidence la centralité ne sont pas bien adaptées aux grands graphes à cause du manque d'espace d'affichage. On

note par conséquent une présence d’occlusions et une difficulté à observer certaines arêtes. Les méthodes mettant en évidence les connecteurs ne sont pas pratiques pour visualiser plusieurs caractéristiques simultanément (la centralité et les connecteurs).

Afin de pouvoir répondre aux questions de recherche (posées dans la section 1.2), telles que l’optimisation de la distribution des nœuds et l’exploration d’un graphe, nous avons proposé de nouvelles méthodes de visualisation de graphes en exploitant la 3D et un environnement immersif. La 3D nous a permis de disposer de plus d’espace d’affichage afin d’optimiser la spatialisation des nœuds, contrairement aux approches classiques. L’immersion nous a quant à elle permis d’améliorer l’exploration d’un graphe. Ainsi, nous avons proposé deux approches de visualisation de graphes. La première est une visualisation exocentrique, qui permet à un utilisateur de manipuler un graphe comme un objet, pour visualiser intuitivement la centralité des nœuds. La seconde est une visualisation égocentrique, particulièrement adaptée à l’exploration d’un graphe dans un environnement immersif, où notre technique de rétractation de l’espace permet à un utilisateur d’avoir une vision d’ensemble de l’intérieur du graphe.

La visualisation exocentrique a consisté à projeter des représentations radiales 2D, selon l’axe vertical, sur différentes surfaces 3D (une demi-sphère, un cône et une portion de tore) tout en gardant la vue radiale 2D en vue de dessus. La projection sphérique distribue le plus les nœuds de la périphérie d’un graphe, alors que la projection torique distribue le plus les nœuds centraux. La projection conique distribue quant à elle uniformément les tous nœuds sur la surface. Ces différentes visualisations sont donc complémentaires pour se focaliser sur un aspect du graphe. Nous avons ensuite comparé l’efficacité de cette visualisation par rapport aux visualisations radiales 2D au moyen des expérimentations utilisateur. Les résultats de ces expérimentations ont permis de montrer que notre visualisation exocentrique est significativement plus performante que les visualisations radiales 2D pour des tâches liées au centre ou à la périphérie d’un graphe. Ces résultats ont également permis de montrer que le passage en immersion peut accentuer le gain de performance en passant d’un graphe de 250 nœuds et 855 arêtes à un graphe de 500 nœuds et 3294 arêtes. Cependant, nous n’avons pas pu démontrer un gain du temps de réalisation des tâches en passant en immersion. Ceci n’est pas pénalisant, car du point de vue d’un analyste, ce qui compte c’est l’amélioration de l’utilisabilité que l’on obtient en passant d’une visualisation 2D à une visualisation 3D. Les visualisations 3D permettent de mieux percevoir les propriétés et la connectivité des nœuds d’un graphe que les visualisations 2D. Ainsi, il s’agit d’une réelle amélioration pour l’analyste même s’il devait y

passer davantage de temps.

Nous avons également proposé de combiner un mode de visualisation d'arêtes regroupées avec un mode de visualisation directe (arêtes détaillées). Cette combinaison permet à un utilisateur de trouver un compromis entre la connectivité des nœuds et des groupes créés par le regroupement d'arêtes en faisant varier l'opacité des arêtes. Aucune des méthodes de visualisation de graphes étudiées n'offre cette possibilité. Notre approche de visualisation pourrait ainsi améliorer l'exploration d'un graphe et la compréhension de la connectivité des nœuds.

La visualisation exocentrique peut également être utilisée pour visualiser à la fois la centralité des nœuds et les connecteurs dans un groupe de nœuds à la suite d'un découpage du graphe en groupes. Elle permet de connaître la position topologique d'un connecteur dans son groupe. Notre visualisation exocentrique pourrait être mieux adaptée que les approches de visualisation étudiées pour l'exploration des connectivités entre groupes de nœuds dans un graphe partitionné.

Enfin, notre visualisation égocentrique crée une zone sphérique vide autour d'un utilisateur immergé et positionne un graphe entre cette zone et une autre sphère concentrique. Ainsi, l'utilisateur n'est pas gêné par des arêtes qui traverseraient son champ de vision. Cette approche a permis d'améliorer la spatialisation des nœuds, contrairement aux méthodes étudiées dans l'état de l'art. Elle permettrait de réduire l'encombrement visuel et le cybermalaise grâce à l'espace libre autour de l'utilisateur et à la projection des arêtes entre deux sphères. Nous avons également proposé une possibilité de rétracter l'espace d'affichage du graphe pour pouvoir obtenir une vue d'ensemble du graphe. Cependant, nous n'améliorons pas le positionnement de certains nœuds qui peuvent être éloignés de leurs voisins à cause des forces d'attraction et de répulsion. Nous n'avons pas pu évaluer les performances de cette approche égocentrique.

Pour résumer, nos contributions dans le cadre des travaux de cette thèse sont les suivantes :

- proposition d'une visualisation radiale 3D qui peut permettre de mieux distribuer les nœuds dans l'espace par rapport aux méthodes 2D étudiées et de mettre en évidence des propriétés topologiques des nœuds dans un graphe [89, 87] ;
- proposition d'une métaphore d'un espace de visualisation rétractable pour la visualisation des graphes qui peut permettre de passer d'une vue détaillée à une vue d'ensemble d'un graphe ;
- comparaison de l'efficacité de la visualisation 3D de graphes et pour certaines tâches

-
- spécifiques par rapport à la visualisation 2D [88] ;
 - évaluation de l'utilisabilité de la visualisation 3D de graphes pour des tâches liées au centre, à la périphérie et aux zones denses d'un graphe.

Perspectives

Les perspectives à court terme consisteraient à comparer la qualité visuelle de notre visualisation exocentrique par rapport aux visualisations radiales 2D à l'aide des métriques de qualité [66], telles que la dispersion des nœuds et l'occlusion des nœuds, et à valider les performances de notre visualisation égocentrique au moyen d'expérimentations utilisateur. L'évaluation de la qualité visuelle de la visualisation exocentrique permettrait de soutenir que la visualisation 3D a permis de mieux distribuer les nœuds que la visualisation 2D. La validation des performances de la visualisation égocentrique consisterait à comparer l'efficacité de la visualisation égocentrique par rapport au résultat d'un algorithme de spatialisation 3D en mesurant un score d'efficacité. L'évaluation pourrait porter sur des tâches telles que la recherche de chemin entre deux nœuds et la recherche de voisins communs à un certain nombre de nœuds donnés. Nous pourrions également évaluer l'utilisabilité de la visualisation égocentrique pour l'exploration de graphes par rapport à une méthode exocentrique traditionnelle.

À moyen terme, nous pourrions proposer une extension 3D de la visualisation anisotrope [123], une méthode qui permet de mettre en évidence la centralité des nœuds (présentée dans la section 3.2). Cette extension pourrait représenter les courbes d'isocentralité sous forme d'un relief où les nœuds ayant une même valeur de centralité sont à la même altitude. Nous pensons que cette approche pourrait, elle aussi, améliorer la comparaison visuelle de la centralité des nœuds, tout en proposant une approche plus immersive que la visualisation exocentrique que nous avons proposée.

À long terme, nous pourrions intégrer une collaboration entre plusieurs utilisateurs à nos méthodes de visualisation. Cela pourrait améliorer les tâches d'exploration des graphes et les échanges entre les utilisateurs. Les utilisateurs peuvent avoir des domaines d'expertise différents. Un défi à relever à ce niveau pourrait porter sur les techniques de communication entre ces utilisateurs. Un autre défi pourrait porter sur une manière de faire collaborer par exemple deux utilisateurs immergés avec la visualisation égocentrique.

RÉSUMÉ DES CONTRIBUTIONS

Publications Scientifiques

- P. Kobina, T. Duval et L. Brisson. « 3D Radial Layout for Centrality Visualization in Graphs » in : *Augmented Reality, Virtual Reality, and Computer Graphics*, Springer International Publishing, 2020, p. 452-460
- P. Kobina, T. Duval et L. Brisson. « Human-centered Evaluation of 3D Radial Layouts for Centrality Visualization », *30th International Conference on Computer Graphics, Visualization and Computer Vision* (**accepted paper**) (2022).

Communications

- P. Kobina, T. Duval et L. Brisson. « Layout radial 3D pour la visualisation de la centralité dans les graphes » in : *Atelier EGC-VIF 2020*, Bruxelles, Belgium, 2020, url : <https://hal.archives-ouvertes.fr/hal-02422512>
- P. Kobina, T. Duval et L. Brisson. « 3D Radial Layout for Centrality Visualization in Graphs », *4th Newcastle University Visualization Workshop*, Apr 2020 (Online)

BIBLIOGRAPHIE

- [1] Michael H. ALBERT, « Measures on the Random Graph », in : *Journal of the London Mathematical Society* 50.3 (déc. 1994), p. 417-429, ISSN : 0024-6107, DOI : 10.1112/jlms/50.3.417, URL : <https://doi.org/10.1112/jlms/50.3.417>.
- [2] Albert-László BARABÁSI et Réka ALBERT, « Emergence of Scaling in Random Networks », in : *Science* 286.5439 (1999), p. 509-512, DOI : 10.1126/science.286.5439.509.
- [3] A. BARRAT et M. WEIGT, « On the properties of small-world network models », in : *The European Physical Journal B* 13.3 (2000), p. 547-560, DOI : 10.1007/s100510050067, URL : <https://doi.org/10.1007/s100510050067>.
- [4] Mathieu BASTIAN, Sebastien HEYMANN et Mathieu JACOMY, « Gephi : An Open Source Software for Exploring and Manipulating Networks », in : *Proceedings of the International AAAI Conference on Web and Social Media* 3.1 (mars 2009), p. 361-362, URL : <https://ojs.aaai.org/index.php/ICWSM/article/view/13937>.
- [5] Giuseppe Di BATTISTA et al., *Graph Drawing : Algorithms for the Visualization of Graphs*, Prentice Hall PTR, 1998, ISBN : 0133016153.
- [6] Chris BENNETT et al., « The Aesthetics of Graph Visualization », in : *Computational Aesthetics in Graphics, Visualization, and Imaging*, sous la dir. de Douglas W. CUNNINGHAM, Gary MEYER et Laszlo NEUMANN, The Eurographics Association, 2007, ISBN : 978-3-905673-43-2, DOI : 10.2312/COMPAESTH/COMPAESTH07/057-064.
- [7] Norman L. BIGGS, Robin J. WILSON et E. Keith LLOYD, *Graph Theory, 1736-1936*, Clarendon Press Oxford, 1976, ISBN : 0198539010.
- [8] Vincent D BLONDEL et al., « Fast unfolding of communities in large networks », in : *Journal of Statistical Mechanics : Theory and Experiment* 2008.10 (oct. 2008), p. 10008, ISSN : 1742-5468, DOI : 10.1088/1742-5468/2008/10/p10008, URL : <http://dx.doi.org/10.1088/1742-5468/2008/10/P10008>.

-
- [9] Phillip BONACICH, « Some unique properties of eigenvector centrality », in : *Social Networks* 29.4 (2007), p. 555-564, ISSN : 0378-8733, DOI : <https://doi.org/10.1016/j.socnet.2007.04.002>.
- [10] Denny BORSBOOM, Gideon J. MELLENBERGH et Jaap Van HEERDEN, « The Theoretical Status of Latent Variables », in : *Psychological Review* 110 (2003), p. 203-219.
- [11] P. Oscar BOYKIN et Vwani P. ROYCHOWDHURY, « Leveraging social networks to fight spam », in : *Computer* 38.4 (2005), p. 61-68, DOI : 10.1109/MC.2005.132.
- [12] U. BRANDES et al., « Explorations into the Visualization of Policy Networks », English, in : *Journal of Theoretical Politics* 11.1 (1999), p. 75-106, ISSN : 0951-6298.
- [13] Ulrik BRANDES, Patrick KENIS et Dorothea WAGNER, « Communicating centrality in policy network drawings », in : *IEEE Transactions on Visualization and Computer Graphics* 9 (2003), p. 241-253.
- [14] Ulrik BRANDES et Christian PICH, « An experimental study on distance-based graph drawing », in : *Graph Drawing*, sous la dir. d'Ioannis G. TOLLIS et Maurizio PATRIGNANI, Lecture Notes in Computer Science 5417, Berlin, Heidelberg : Springer Berlin Heidelberg, 2009, p. 218-229, ISBN : 978-3-642-00218-2, DOI : 10.1007/978-3-642-00219-9_21.
- [15] Ulrik BRANDES et Christian PICH, « Eigensolver Methods for Progressive Multidimensional Scaling of Large Data », in : *Graph Drawing, 14th International Symposium, GD 2006, Karlsruhe, Germany, September 18-20, 2006. Revised Papers*, sous la dir. de Michael KAUFMANN et Dorothea WAGNER, t. 4372, Lecture Notes in Computer Science, Springer, 2006, p. 42-53, DOI : 10.1007/978-3-540-70904-6_6, URL : https://doi.org/10.1007/978-3-540-70904-6%5C_6.
- [16] Ulrik BRANDES et Christian PICH, « More flexible radial layout », in : *Journal of Graph Algorithms and Applications* 15 (2011), p. 151-173.
- [17] Sergey BRIN et Lawrence PAGE, « The Anatomy of a Large-Scale Hypertextual Web Search Engine », in : *Computer Networks and ISDN Systems* 30 (1998), p. 107-117.

-
- [18] Govert G. BRINKMANN, Kristian F.D. RIETVELD et Frank W. TAKES, « Exploiting GPUs for Fast Force-Directed Visualization of Large-Scale Networks », in : *2017 46th International Conference on Parallel Processing (ICPP)*, 2017, p. 382-391, DOI : 10.1109/ICPP.2017.47.
- [19] Laurent BRISSON, Thierry DUVAL et Rémi SAHL, « Visualisation Immersive de Graphes en 3D pour explorer des graphes de communautés », in : *EGC-VIF 2018*, Paris, France, jan. 2018, URL : <https://hal.archives-ouvertes.fr/hal-01759070>.
- [20] Michael BRONSTEIN et Emanuele ROSSI, *Deep learning on dynamic graphs*, URL : https://blog.twitter.com/engineering/en_us/topics/insights/2021/temporal-graph-networks.
- [21] John BROOKE, « SUS : A quick and dirty usability scale », in : *Usability Evaluation in Industry* 189 (1996).
- [22] Chris BRYAN, Kwan-Liu MA et Yang-Chih FU, « An interactive visualization interface for studying egocentric, categorical, contact diary datasets », in : *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, 2013, p. 771-778, DOI : 10.1145/2492517.2492636.
- [23] Ed BULLMORE et Olaf SPORNS, « The economy of brain network organization », in : *Nature Reviews Neuroscience* 13.5 (mai 2012), p. 336-349, ISSN : 1471-0048, DOI : 10.1038/nrn3214, URL : <https://doi.org/10.1038/nrn3214>.
- [24] Abhijit CHAKRABORTY et al., « Characterization of the community structure in a large-scale production network in Japan », in : *Physica A : Statistical Mechanics and its Applications* 513 (2019), p. 210-221, ISSN : 0378-4371, DOI : <https://doi.org/10.1016/j.physa.2018.08.175>.
- [25] Chaomei CHEN, *Information visualization : Beyond the horizon*, English, United Kingdom : Springer London, 2006, ISBN : 184628340X, DOI : 10.1007/1-84628-579-8.
- [26] Jilin CHEN et al., « Make New Friends, but Keep the Old : Recommending People on Social Networking Sites », in : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, Association for Computing Machinery, 2009, p. 201-210, ISBN : 9781605582467, DOI : 10.1145/1518701.1518735.

-
- [27] Wei CHEN, Yajun WANG et Siyu YANG, « Efficient Influence Maximization in Social Networks », in : *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, Association for Computing Machinery, 2009, p. 199-208, ISBN : 9781605584959, DOI : 10.1145/1557019.1557047.
- [28] Se-Hang CHEONG et Yain-Whar SI, « Force-directed algorithms for schematic drawings and placement : A survey », in : *Information Visualization 19.1* (2020), p. 65-91, DOI : 10.1177/1473871618821740, URL : <https://doi.org/10.1177/1473871618821740>.
- [29] Grégoire CLIQUET et al., « Towards HMD-based Immersive Analytics », in : *Immersive analytics Workshop, IEEE VIS 2017*, Phoenix, United States, oct. 2017, URL : <https://hal.archives-ouvertes.fr/hal-01631306>.
- [30] J.D. Gómez CRUZ, « Socio-semantic Networks Algorithm for a Point of View Based Visualization of On-line Communities », thèse de doct., Télécom Bretagne, Université de Rennes 1, déc. 2012.
- [31] Vinh-Loc DAO, « Characterizing community detection algorithms and detected modules in large-scale complex networks », thèse de doct., Brest, France : IMT Atlantique Bretagne-Pays de la Loire, déc. 2018.
- [32] Giuseppe DI BATTISTA et al., « Algorithms for Drawing Graphs : An Annotated Bibliography », in : *Comput. Geom. Theory Appl.* 4.5 (oct. 1994), p. 235-282, ISSN : 0925-7721, DOI : 10.1016/0925-7721(94)00014-X, URL : [https://doi.org/10.1016/0925-7721\(94\)00014-X](https://doi.org/10.1016/0925-7721(94)00014-X).
- [33] E. W. DIJKSTRA, « A note on two problems in connexion with graphs », in : *Numerische Mathematik* 1.1 (déc. 1959), p. 269-271, ISSN : 0945-3245, DOI : 10.1007/BF01386390, URL : <https://doi.org/10.1007/BF01386390>.
- [34] Olive J. DUNN, « Multiple Comparisons Using Rank Sums », in : *Technometrics* 6.3 (1964), p. 241-252, DOI : 10.1080/00401706.1964.10490181.
- [35] C. DUNNE et Ben SHNEIDERMAN, « Improving graph drawing readability by incorporating readability metrics : A software tool for network analysts », in : *University of Maryland, HCIL Tech Report HCIL-2009-13* (2009).

-
- [36] Tim DWYER et al., « Visual analysis of network centralities », in : *Asia-Pacific Symposium on Information Visualisation, APVIS 2006, Tokyo, Japan, February 1-3, 2006*, sous la dir. de Kazuo MISUE, Kozo SUGIYAMA et Jiro TANAKA, t. 60, Australian Computer Society, 2006, p. 189-197, URL : <https://dl.acm.org/citation.cfm?id=1151931>.
- [37] P. EADES, « A heuristic for graph drawing », in : *Congressus Numerantium* vol.42 (1984), p. 149-160, URL : <https://ci.nii.ac.jp/naid/10000023432/en/>.
- [38] Peter EADES et Xuemin LIN, « Spring algorithms and symmetry », in : *Theoretical Computer Science* 240.2 (2000), p. 379-405, ISSN : 0304-3975, DOI : [https://doi.org/10.1016/S0304-3975\(99\)00239-X](https://doi.org/10.1016/S0304-3975(99)00239-X), URL : <https://www.sciencedirect.com/science/article/pii/S030439759900239X>.
- [39] Zakariyaa Ait EL MOUDEN et al., « Towards Using Graph Analytics for Tracking Covid-19 », in : *Procedia Computer Science* 177 (2020), p. 204-211, ISSN : 1877-0509, DOI : <https://doi.org/10.1016/j.procs.2020.10.029>, URL : <https://www.sciencedirect.com/science/article/pii/S1877050920322961>.
- [40] P. ERDÖS et A. RÉNYI, « On Random Graphs I », in : *Publicationes Mathematicae Debrecen* 6 (1959), p. 290.
- [41] Ozan ERSOY et al., « Skeleton-Based Edge Bundling for Graph Visualization », in : *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), p. 2364-2373, DOI : 10.1109/TVCG.2011.233.
- [42] Leonhard EULER, « Solutio problematis ad geometriam situs pertinentis », in : *Commentarii Academiae Scientiarum Imperialis Petropolitanae* 8 (1741), p. 128-140.
- [43] Chong FENG et al., « Community Detection by Node Betweenness and Similarity in Complex Network », in : *Complexity* 2021 (2021), ISSN : 1076-2787, DOI : 10.1155/2021/9986895, URL : <https://doi.org/10.1155/2021/9986895>.
- [44] François FERLAND et al., « Egocentric and exocentric teleoperation interface using real-time, 3D video projection », in : *2009 4th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2009, p. 37-44, DOI : 10.1145/1514095.1514105.

-
- [45] Paola FLOCCHINI, Bernard MANS et Nicola SANTORO, « On the exploration of time-varying networks », in : *Theoretical Computer Science* 469 (2013), p. 53-68, ISSN : 0304-3975, DOI : <https://doi.org/10.1016/j.tcs.2012.10.029>, URL : <https://www.sciencedirect.com/science/article/pii/S0304397512009504>.
- [46] Santo FORTUNATO, « Community detection in graphs », in : *Physics Reports* 486.3 (2010), p. 75-174, ISSN : 0370-1573, DOI : <https://doi.org/10.1016/j.physrep.2009.11.002>.
- [47] Linton FREEMAN, « The development of social network analysis », in : *A Study in the Sociology of Science* 1.687 (2004), p. 159-167.
- [48] Linton C. FREEMAN, « Centrality in social networks conceptual clarification », in : *Social networks* 1.3 (1978), p. 215-239, ISSN : 0378-8733, DOI : [https://doi.org/10.1016/0378-8733\(78\)90021-7](https://doi.org/10.1016/0378-8733(78)90021-7).
- [49] Y. FRISHMAN et Ayellet TAL, « Dynamic Drawing of Clustered Graphs », in : *IEEE Symposium on Information Visualization*, 2004, p. 191-198, DOI : 10.1109/INFVIS.2004.18.
- [50] Thomas M. J. FRUCHTERMAN et Edward M. REINGOLD, « Graph drawing by force-directed placement », in : *Software : Practice and Experience* 21.11 (1991), p. 1129-1164, DOI : <https://doi.org/10.1002/spe.4380211102>.
- [51] Pawel GAJER, Michael T. GOODRICH et Stephen G. KOBOUROV, « A Multi-dimensional Approach to Force-Directed Layouts of Large Graphs », in : *Graph Drawing*, sous la dir. de Joe MARKS, Berlin, Heidelberg : Springer Berlin Heidelberg, 2001, p. 211-221, ISBN : 978-3-540-44541-8.
- [52] E.R. GANSNER et al., « A technique for drawing directed graphs », in : *IEEE Transactions on Software Engineering* 19.3 (1993), p. 214-230, DOI : 10.1109/32.221135.
- [53] Emden R. GANSNER, Yehuda KOREN et Stephen NORTH, « Graph Drawing by Stress Majorization », in : *Graph Drawing*, sous la dir. de János PACH, Berlin, Heidelberg : Springer Berlin Heidelberg, 2005, p. 239-250, ISBN : 978-3-540-31843-9.
- [54] Emden R. GANSNER et al., « Multilevel agglomerative edge bundling for visualizing large graphs », in : *2011 IEEE Pacific Visualization Symposium*, 2011, p. 187-194, DOI : 10.1109/PACIFICVIS.2011.5742389.

-
- [55] Helen GIBSON, Joe FAITH et Paul VICKERS, « A survey of two-dimensional graph layout techniques for information visualisation », in : *Information Visualization* 12.3-4 (2013), p. 324-357, DOI : 10.1177/1473871612455749, URL : <https://doi.org/10.1177/1473871612455749>.
- [56] E. N. GILBERT, « Random Graphs », in : *The Annals of Mathematical Statistics* 30.4 (1959), p. 1141-1144, DOI : 10.1214/aoms/1177706098, URL : <https://doi.org/10.1214/aoms/1177706098>.
- [57] M. GIRVAN et M. E. J. NEWMAN, « Community structure in social and biological networks », in : *Proceedings of the National Academy of Sciences* 99.12 (2002), p. 7821-7826, ISSN : 0027-8424, DOI : 10.1073/pnas.122653799, URL : <https://www.pnas.org/content/99/12/7821>.
- [58] Simone GÖTTLICH et al., « Data-driven graph drawing techniques with applications for conveyor systems », in : *Journal of Mathematics in Industry* 10.1 (2020), p. 24, ISSN : 2190-5983, DOI : 10.1186/s13362-020-00092-2.
- [59] Martin GRANDJEAN, « La connaissance est un réseau : perspective sur l'organisation archivistique et encyclopédique », in : *Les Cahiers du Numérique* 10 (oct. 2014), p. 37-54, DOI : 10.3166/LCN.10.3.37-54.
- [60] Nicolas GREFFARD, Fabien PICAROUGNE et Pascale KUNTZ, « Beyond the classical monoscopic 3D in graph analytics : An experimental study of the impact of stereoscopy », in : *2014 IEEE VIS International Workshop on 3DVis (3DVis)*, 2014, p. 19-24, DOI : 10.1109/3DVis.2014.7160095.
- [61] Aditya GROVER, Aaron ZWEIG et Stefano ERMON, « Graphite : Iterative Generative Modeling of Graphs », in : *Proceedings of the 36th International Conference on Machine Learning*, sous la dir. de Kamalika CHAUDHURI et Ruslan SALAKHUTDINOV, t. 97, Proceedings of Machine Learning Research, PMLR, sept. 2019, p. 2434-2444, URL : <https://proceedings.mlr.press/v97/grover19a.html>.
- [62] Roger GUIMERÀ et Luís A Nunes AMARAL, « Cartography of complex networks : modules and universal roles », in : *Journal of Statistical Mechanics : Theory and Experiment* 2005.02 (2005), P02001, DOI : 10.1088/1742-5468/2005/02/p02001, URL : <https://doi.org/10.1088/1742-5468/2005/02/p02001>.

-
- [63] Stefan HACHUL et Michael JÜNGER, « Drawing Large Graphs with a Potential-Field-Based Multilevel Algorithm », in : *Graph Drawing*, sous la dir. de János PACH, Berlin, Heidelberg : Springer Berlin Heidelberg, 2005, p. 285-295, ISBN : 978-3-540-31843-9.
- [64] Stefan HACHUL et Michael JÜNGER, « Large-Graph Layout Algorithms at Work : An Experimental Study », in : *Journal of Graph Algorithms and Applications* 11.2 (2007), p. 345-369, DOI : [10.7155/jgaa.00150](https://doi.org/10.7155/jgaa.00150), URL : <https://doi.org/10.7155/jgaa.00150>.
- [65] Ronny HADANY et David HAREL, « A multi-scale algorithm for drawing graphs nicely », in : *Discrete Applied Mathematics* 113.1 (2001), p. 3-21, ISSN : 0166-218X, DOI : [https://doi.org/10.1016/S0166-218X\(00\)00389-9](https://doi.org/10.1016/S0166-218X(00)00389-9).
- [66] H. HALEEM et al., « Evaluating the Readability of Force Directed Graph Layouts : A Deep Learning Approach », in : *IEEE Computer Graphics and Applications* 39.4 (2019), p. 40-53.
- [67] William L. HAMILTON, « Graph Representation Learning », in : *Synthesis Lectures on Artificial Intelligence and Machine Learning* 14.3 (2020), p. 1-159.
- [68] Derek L. HANSEN et al., « Chapter 3 - Social network analysis : Measuring, mapping, and modeling collections of connections », in : *Analyzing Social Media Networks with NodeXL (Second Edition)*, sous la dir. de Derek L. HANSEN et al., Second Edition, Morgan Kaufmann, 2020, p. 31-51, ISBN : 978-0-12-817756-3, DOI : <https://doi.org/10.1016/B978-0-12-817756-3.00003-0>.
- [69] David HAREL et Yehuda KOREN, « Graph Drawing by High-Dimensional Embedding », in : *Graph Drawing*, sous la dir. de Michael T. GOODRICH et Stephen G. KOBouROV, Berlin, Heidelberg : Springer Berlin Heidelberg, 2002, p. 207-219, ISBN : 978-3-540-36151-0.
- [70] Pascal HELD, Julia HEMPEL et Rudolf KRUSE, « Cluster-based Visualization of Dynamic Graphs », in : *Proceedings. 23. Workshop Computational Intelligence*, sous la dir. de Frank HOFFMANN et E. HÜLLERMEIER, Dortmund : KIT Scientific Publishing, déc. 2013, p. 21-37, ISBN : 978-3-7315-0126-8.
- [71] Paul W. HOLLAND, Kathryn Blackmond LASKEY et Samuel LEINHARDT, « Stochastic blockmodels : First steps », in : *Social Networks* 5.2 (1983), p. 109-137, ISSN : 0378-8733, DOI : [https://doi.org/10.1016/0378-8733\(83\)90021-7](https://doi.org/10.1016/0378-8733(83)90021-7).

-
- [72] Paul W. HOLLAND et Samuel LEINHARDT, « Transitivity in Structural Models of Small Groups », in : *Comparative Group Studies* 2.2 (1971), p. 107-124, DOI : 10.1177/104649647100200201, URL : <https://doi.org/10.1177/104649647100200201>.
- [73] Danny HOLTEN, « Hierarchical Edge Bundles : Visualization of Adjacency Relations in Hierarchical Data », in : *IEEE Transactions on Visualization and Computer Graphics* 12.5 (2006), p. 741-748, DOI : 10.1109/TVCG.2006.147.
- [74] Danny HOLTEN et Jarke J. VAN WIJK, « Force-Directed Edge Bundling for Graph Visualization », in : *Computer Graphics Forum* 28.3 (2009), p. 983-990, DOI : <https://doi.org/10.1111/j.1467-8659.2009.01450.x>.
- [75] Yifan HU, « Efficient, High-Quality Force-Directed Graph Drawing », in : *The Mathematica journal* 10 (2005), p. 37-71.
- [76] Yifan HU et Lei SHI, « Visualizing large graphs », in : *WIREs Computational Statistics* 7.2 (2015), p. 115-136, DOI : <https://doi.org/10.1002/wics.1343>, URL : <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.1343>.
- [77] Jie HUA et al., « Applying Graph Centrality Metrics in Visual Analytics of Scientific Standard Datasets », in : *Symmetry* 11.1 (2019), ISSN : 2073-8994, DOI : 10.3390/sym11010030, URL : <https://www.mdpi.com/2073-8994/11/1/30>.
- [78] Weidong HUANG et Maolin HUANG, « Exploring the Relative Importance of Number of Edge Crossings and Size of Crossing Angles : A Quantitative Perspective », in : *International Journal of Advanced Intelligence Paradigms* 3 (jan. 2010), p. 10, DOI : 10.1145/1865841.1865854.
- [79] Weidong HUANG et al., « Making graph visualization a user-centered process », in : *Journal of Visual Languages and Computing* 48 (2018), p. 1-8, ISSN : 1045-926X, DOI : <https://doi.org/10.1016/j.jvlc.2018.07.001>, URL : <https://www.sciencedirect.com/science/article/pii/S1045926X1830051X>.
- [80] Christophe HURTER, Ozan ERSOY et Alexandru TELEA, « Graph Bundling by Kernel Density Estimation », in : *Computer Graphics Forum* 31 (juin 2012), p. 865-874, DOI : 10.1111/j.1467-8659.2012.03079.x.
- [81] Stephen INGRAM, Tamara MUNZNER et Marc OLANO, « Glimmer : Multilevel MDS on the GPU », in : *IEEE Transactions on Visualization and Computer Graphics* 15.2 (2009), p. 249-261, DOI : 10.1109/TVCG.2008.85.

-
- [82] Ian T. JOLLIFFE et Jorge CADIMA, « Principal component analysis : a review and recent developments », in : *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences* 374.2065 (2016), DOI : 10.1098/rsta.2015.0202.
- [83] Tomihisa KAMADA et Satoru KAWAI, « An algorithm for drawing general undirected graphs », in : *Information Processing Letters* 31.1 (1989), p. 7-15, ISSN : 0020-0190, DOI : [https://doi.org/10.1016/0020-0190\(89\)90102-6](https://doi.org/10.1016/0020-0190(89)90102-6).
- [84] Leo KATZ, « A new status index derived from sociometric analysis », in : *Psychometrika* 18.1 (1953), p. 39-43, ISSN : 1860-0980, DOI : 10.1007/BF02289026, URL : <https://doi.org/10.1007/BF02289026>.
- [85] Beom Jun KIM, « Performance of networks of artificial neurons : The role of clustering », in : *Physical review. E, Statistical, nonlinear, and soft matter physics* 69 (4 2004), p. 045101, DOI : 10.1103/PhysRevE.69.045101.
- [86] Maksim KITSACK et al., « Identification of influential spreaders in complex networks », in : *Nature Physics* 6.11 (2010), p. 888-893, DOI : 10.1038/nphys1746, URL : <https://doi.org/10.1038/nphys1746>.
- [87] Pirizivè KOBINA, Thierry DUVAL et Laurent BRISSON, « 3D Radial Layout for Centrality Visualization in Graphs », in : *Augmented Reality, Virtual Reality, and Computer Graphics*, sous la dir. de Lucio Tommaso DE PAOLIS et Patrick BOURDOT, Springer International Publishing, 2020, p. 452-460, ISBN : 978-3-030-58465-8.
- [88] Pirizivè KOBINA, Thierry DUVAL et Laurent BRISSON, « Human-centered Evaluation of 3D Radial Layouts for Centrality Visualization », in : *30th International Conference on Computer Graphics, Visualization and Computer Vision (accepted paper)* (2022).
- [89] Pirizivè KOBINA, Thierry DUVAL et Laurent BRISSON, « Layout radial 3D pour la visualisation de la centralité dans les graphes », in : *Atelier EGC-VIF 2020*, Bruxelles, Belgium, 2020, URL : <https://hal.archives-ouvertes.fr/hal-02422512>.
- [90] Pirizivè KOBINA, Thierry DUVAL et Laurent BRISSON, « The Spherical Retractable Bubble Space : an Egocentric Graph Visualization throughout a Retractable

-
- Visualization Space », in : *19th EuroXR International Conference (under review)* (2022).
- [91] Stephen G. KOBOUROV, « Spring Embedders and Force Directed Graph Drawing Algorithms », in : *CoRR* abs/1201.3011 (2012), URL : <http://arxiv.org/abs/1201.3011>.
- [92] Jacob KÖHLER et al., « Graph-based analysis and visualization of experimental results with ONDEX », in : *Bioinformatics* 22.11 (mars 2006), p. 1383-1390, ISSN : 1367-4803, DOI : 10.1093/bioinformatics/btl081, URL : <https://doi.org/10.1093/bioinformatics/btl081>.
- [93] Eric KOLACZYK, *Statistical Analysis of Network Data : Methods and Models*, 2009, ISBN : 978-0-387-88145-4, DOI : 10.1007/978-0-387-88146-1.
- [94] Yehuda KOREN, « Graph Drawing by Subspace Optimization », in : *Proceedings of the Sixth Joint Eurographics - IEEE TCVG Conference on Visualization, VIS-SYM'04*, Eurographics Association, 2004, p. 65-74, ISBN : 390567307X.
- [95] Aleksandra KOROLOVA et al., « Link Privacy in Social Networks », in : *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08*, Association for Computing Machinery, 2008, p. 289-298, ISBN : 9781595939913, DOI : 10.1145/1458082.1458123.
- [96] J. F. KRUIGER et al., « Graph Layouts by t-SNE », in : *Comput. Graph. Forum* 36.3 (2017), p. 283-294, ISSN : 0167-7055, DOI : 10.1111/cgf.13187.
- [97] J. B. KRUSKAL, « Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis », in : *Psychometrika* 29.1 (1964), p. 1-27, ISSN : 1860-0980, DOI : 10.1007/BF02289565.
- [98] Joseph B. KRUSKAL et Judith B. SEERY, « Designing network diagrams », in : *Proceedings of the First General Conference on Social Graphics*, 1980, p. 22-50.
- [99] William H. KRUSKAL et W. Allen WALLIS, « Use of Ranks in One-Criterion Variance Analysis », in : *Journal of the American Statistical Association* 47.260 (1952), p. 583-621, ISSN : 01621459.
- [100] Oh-Hyun KWON et al., « A Study of Layout, Rendering, and Interaction Methods for Immersive Graph Visualization », in : *IEEE Transactions on Visualization and Computer Graphics* 22.7 (2016), p. 1802-1815, DOI : 10.1109/TVCG.2016.2520921.

-
- [101] Oh-Hyun KWON et al., « Spherical layout and rendering methods for immersive graph visualization », in : *2015 IEEE Pacific Visualization Symposium (Pacific-Vis)*, 2015, p. 63-67, DOI : 10.1109/PACIFICVIS.2015.7156357.
- [102] A. LAMBERT, R. BOURQUI et D. AUBER, « 3D Edge Bundling for Geographical Data Visualization », in : *2010 14th International Conference Information Visualisation (IV 2010)*, Los Alamitos, CA, USA : IEEE Computer Society, juill. 2010, p. 329-335, DOI : 10.1109/IV.2010.53, URL : <https://doi.ieeecomputersociety.org/10.1109/IV.2010.53>.
- [103] Tatiana von LANDESBERGER et al., « Visual analysis of large graphs : State-of-the-art and future research challenges », in : *Computer Graphics Forum* 30.6 (2011), p. 1719-1749, ISSN : 0167-7055, DOI : 10.1111/j.1467-8659.2011.01898.x.
- [104] Clement LEE et Darren J. WILKINSON, « A review of stochastic block models and extensions for graph clustering », in : *Applied Network Science* 4.1 (déc. 2019), ISSN : 2364-8228, DOI : 10.1007/s41109-019-0232-2, URL : <http://dx.doi.org/10.1007/s41109-019-0232-2>.
- [105] Jongshin LEE et al., « Betweenness centrality of teams in social networks », in : *Chaos : An Interdisciplinary Journal of Nonlinear Science* 31.6 (2021), p. 061108, DOI : 10.1063/5.0056683.
- [106] Antoine LHUILLIER, Christophe HURTER et Alexandru C TELEA, « State of the Art in Edge and Trail Bundling Techniques », in : *Computer Graphics Forum* 36.3 (juin 2017), p. 619-645, DOI : 10.1111/cgf.13213, URL : <https://hal-enac.archives-ouvertes.fr/hal-01559020>.
- [107] Jiawei LU et Yain-Whar SI, « Clustering-based force-directed algorithms for 3D graph visualization », in : *The Journal of Supercomputing* 76 (2020), p. 1-62, DOI : 10.1007/s11227-020-03226-w.
- [108] Douglas LUKE, *A User's Guide to Network Analysis in R*, jan. 2015, ISBN : 978-3-319-23882-1, DOI : 10.1007/978-3-319-23883-8.
- [109] Inna LYUBAREVA et al., « Une plateforme de crowdfunding et son réseau social : L'exemple Ulule », in : *Revue Française de Gestion* 46 (juill. 2020), p. 135-151, DOI : <https://doi.org/10.3166/rfg.2019.00402>.
- [110] Laurens van der MAATEN et Geoffrey HINTON, « Visualizing Data using t-SNE », in : *Journal of Machine Learning Research* 9.86 (2008), p. 2579-2605.

-
- [111] Kim MARRIOTT et al., « Immersive Analytics : Time to Reconsider the Value of 3D for Information Visualisation », in : *Immersive Analytics*, sous la dir. de Kim MARRIOTT et al., Springer International Publishing, 2018, p. 25-55, ISBN : 978-3-030-01388-2, DOI : 10.1007/978-3-030-01388-2_2.
- [112] Francesco MARTINO et Andrea SPOTO, « Social Network Analysis : A brief theoretical review and further perspectives in the study of Information Technology », in : *PsychNology Journal* 4 (jan. 2006), p. 53-86.
- [113] Tamara MUNZNER, « Exploring large graphs in 3D hyperbolic space », in : *IEEE Computer Graphics and Applications* 18.4 (1998), p. 18-23, DOI : 10.1109/38.689657.
- [114] M. E. J. NEWMAN, « Modularity and community structure in networks », in : *Proceedings of the National Academy of Sciences* 103.23 (2006), p. 8577-8582, DOI : 10.1073/pnas.0601602103.
- [115] M. E. J. NEWMAN et M. GIRVAN, « Finding and evaluating community structure in networks », in : *Physical Review E* 69 (2 fév. 2004), p. 026113, DOI : 10.1103/PhysRevE.69.026113.
- [116] Mark E.J. NEWMAN, « Power laws, Pareto distributions and Zipf's law », in : *Contemporary Physics* 46.5 (2005), p. 323-351, DOI : 10.1080/00107510500052444.
- [117] Arlind NOCAJ, Mark ORTMANN et Ulrik BRANDES, « Untangling the Hairballs of Multi-Centered, Small-World Online Social Media Networks », in : *Journal of Graph Algorithms and Applications : JGAA* 19.2 (2015), p. 595-618, DOI : 10.7155/jgaa.00370.
- [118] Reza Yousefi NOORAIE et al., « Social Network Analysis : An Example of Fusion Between Quantitative and Qualitative Methods », in : *Journal of Mixed Methods Research* 14.1 (2020), p. 110-124, DOI : 10.1177/1558689818804060, eprint : <https://doi.org/10.1177/1558689818804060>, URL : <https://doi.org/10.1177/1558689818804060>.
- [119] Zhang PANPAN, Wang TIANDONG et Yan JUN, « PageRank centrality and algorithms for weighted, directed networks », in : *Physica A : Statistical Mechanics and its Applications* 586 (2022), p. 126438, ISSN : 0378-4371, DOI : <https://doi.org/10.1016/j.physa.2021.126438>.

-
- [120] Nicola PEZZOTTI et al., « Approximated and User Steerable tSNE for Progressive Visual Analytics », in : *IEEE Transactions on Visualization and Computer Graphics* 23.7 (2017), p. 1739-1752, DOI : 10.1109/TVCG.2016.2570755.
- [121] Pascal PONS et Matthieu LATAPY, « Computing Communities in Large Networks Using Random Walks », in : *Computer and Information Sciences - ISCIS 2005*, sous la dir. de pInar YOLUM et al., Berlin, Heidelberg : Springer Berlin Heidelberg, 2005, p. 284-293, ISBN : 978-3-540-32085-2.
- [122] Michalis POTAMIAS et al., « Fast Shortest Path Distance Estimation in Large Networks », in : *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09*, Association for Computing Machinery, 2009, p. 867-876, ISBN : 9781605585123, DOI : 10.1145/1645953.1646063.
- [123] Mukund RAJ et Ross T. WHITAKER, « Anisotropic Radial Layout for Visualizing Centrality and Structure in Graphs », in : *Graph Drawing and Network Visualization*, sous la dir. de Fabrizio FRATI et Kwan-Liu MA, Cham : Springer International Publishing, 2018, p. 351-364, ISBN : 978-3-319-73915-1.
- [124] John RANOLA et al., « A Poisson Model for Random Multigraphs », in : *Bioinformatics (Oxford, England)* 26 (août 2010), p. 2004-11, DOI : 10.1093/bioinformatics/btq309.
- [125] Jörg REICHARDT, Roberto ALAMINO et David SAAD, « The Interplay between Microscopic and Mesoscopic Structures in Complex Networks », in : *PLOS ONE* 6.8 (2011), p. 1-8, DOI : 10.1371/journal.pone.0021282.
- [126] John T.E. RICHARDSON, « The use of Latin-square designs in educational and psychological research », in : *Educational Research Review* 24 (2018), p. 84-97, ISSN : 1747-938X, DOI : <https://doi.org/10.1016/j.edurev.2018.03.003>, URL : <https://www.sciencedirect.com/science/article/pii/S1747938X18301623>.
- [127] B. RICKS, C.W. NIELSEN et M.A. GOODRICH, « Ecological displays for robot interaction : a new perspective », in : *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, t. 3, 2004, 2855-2860 vol.3, DOI : 10.1109/IROS.2004.1389842.
- [128] Garry ROBINS et al., « Recent developments in exponential random graph (p*) models for social networks », in : *Social Networks* 29.2 (2007), p. 192-215, ISSN :

-
- 0378-8733, DOI : <https://doi.org/10.1016/j.socnet.2006.08.003>, URL : <https://www.sciencedirect.com/science/article/pii/S0378873306000384>.
- [129] Emanuele ROSSI et al., « Temporal Graph Networks for Deep Learning on Dynamic Graphs », in : *CoRR* abs/2006.10637 (2020), URL : <https://arxiv.org/abs/2006.10637>.
- [130] Ramik SADANA, Vidya SETLUR et John STASKO, « Redefining a Contribution for Immersive Visualization Research », in : *Proceedings of the 2016 ACM Companion on Interactive Surfaces and Spaces*, ISS '16 Companion, Association for Computing Machinery, 2016, p. 41-45, ISBN : 9781450345309, DOI : 10.1145/3009939.3009946.
- [131] Lee SANGSEOK et Lee Dong KYU, « What is the proper way to apply the multiple comparison test? », in : *Korean J Anesthesiol* 71.5 (2018), p. 353-360, DOI : 10.4097/kja.d.18.00242.
- [132] Mohammed SAQR, Uno FORS et Jalal NOURI, « Using social network analysis to understand online Problem-Based Learning and predict performance », in : *PLOS ONE* 13.9 (sept. 2018), p. 1-20, DOI : 10.1371/journal.pone.0203590, URL : <https://doi.org/10.1371/journal.pone.0203590>.
- [133] Hinako SASSA, Takayuki ITOH et Mitsuo YOSHIDA, « 3D Visualization of Network Including Nodes with Labels », in : *2019 23rd International Conference Information Visualisation (IV)*, 2019, p. 19-24, DOI : 10.1109/IV.2019.00013.
- [134] Samuel S. SHAPIRO et Martin B. WILK, « An Analysis of Variance Test for Normality (Complete Samples) », in : *Biometrika* 52.3/4 (1965), p. 591-611, ISSN : 00063444.
- [135] Nitai B. SILVA et al., « A graph-based friend recommendation system using Genetic Algorithm », in : *IEEE Congress on Evolutionary Computation*, 2010, p. 1-7, DOI : 10.1109/CEC.2010.5586144.
- [136] Tom A.B. SNIJDERS et Krzysztof NOWICKI, « Estimation and Prediction for Stochastic Blockmodels for Graphs with Latent Block Structure », in : *Journal of Classification* 14.1 (jan. 1997), p. 75-100, ISSN : 1432-1343, DOI : 10.1007/s003579900004, URL : <https://doi.org/10.1007/s003579900004>.

-
- [137] Richard H. Y. SO, Andy HO et W. T. LO, « A Metric to Quantify Virtual Scene Movement for the Study of Cybersickness : Definition, Implementation, and Verification », in : *Presence : Teleoperators and Virtual Environments* 10.2 (avr. 2001), p. 193-215, DOI : 10.1162/105474601750216803.
- [138] Richard H. Y. SO, W. T. LO et Andy T. K. HO, « Effects of Navigation Speed on Motion Sickness Caused by an Immersive Virtual Environment », in : *Human Factors* 43.3 (2001), p. 452-461, DOI : 10.1518/001872001775898223.
- [139] J. SORGER et al., « Egocentric Network Exploration for Immersive Analytics », in : *Computer Graphics Forum* 40.7 (2021), p. 241-252, DOI : <https://doi.org/10.1111/cgf.14417>, URL : <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14417>.
- [140] Simon STOLZ et Christian SCHLERETH, « Predicting Tie Strength with Ego Network Structures », in : *Journal of Interactive Marketing* 54 (2021), p. 40-52, ISSN : 1094-9968, DOI : <https://doi.org/10.1016/j.intmar.2020.10.001>.
- [141] Alfredo R. TEYSEYRE et Marcelo R. CAMPO, « An Overview of 3D Software Visualization », in : *IEEE Transactions on Visualization and Computer Graphics* 15.1 (2009), p. 87-105, DOI : 10.1109/TVCG.2008.86.
- [142] Warren S. TORGERSON, « Multidimensional scaling : I. Theory and method », in : *Psychometrika* 17.4 (1952), p. 401-419, ISSN : 1860-0980, DOI : 10.1007/BF02288916, URL : <https://doi.org/10.1007/BF02288916>.
- [143] Melanie TORY et al., « Visualization task performance with 2D, 3D, and combination displays », in : *IEEE Transactions on Visualization and Computer Graphics* 12.1 (2006), p. 2-13, DOI : 10.1109/TVCG.2006.17.
- [144] V. A. TRAAG, L. WALTMAN et N. J. van ECK, « From Louvain to Leiden : guaranteeing well-connected communities », in : *Scientific Reports* 9.1 (mars 2019), p. 5233, ISSN : 2045-2322, DOI : 10.1038/s41598-019-41695-z, URL : <https://doi.org/10.1038/s41598-019-41695-z>.
- [145] Cong TRAN et al., « DeepNC : Deep Generative Network Completion », in : *CoRR* abs/1907.07381 (2019), URL : <http://arxiv.org/abs/1907.07381>.

-
- [146] Nitesh TRIVEDI et Anurag SINGH, « Efficient Influence Maximization in Social-Networks Under Independent Cascade Model », in : *Procedia Computer Science* 173 (2020), p. 315-324, ISSN : 1877-0509, DOI : <https://doi.org/10.1016/j.procs.2020.06.037>.
- [147] Chris WALSHAW, « A Multilevel Algorithm for Force-Directed Graph Drawing », in : *Journal of Graph Algorithms and Applications* 7 (jan. 2003), p. 253-285, DOI : [10.1007/3-540-44541-2_17](https://doi.org/10.1007/3-540-44541-2_17).
- [148] Junyi WANG et al., « A novel weight neighborhood centrality algorithm for identifying influential spreaders in complex networks », in : *Physica A : Statistical Mechanics and its Applications* 475 (2017), p. 88-105, ISSN : 0378-4371, DOI : <https://doi.org/10.1016/j.physa.2017.02.007>, URL : <https://www.sciencedirect.com/science/article/pii/S0378437117301218>.
- [149] Xiaobo WANG et Isao MIYAMOTO, « Generating Customized Layouts », in : *Proceedings of the Symposium on Graph Drawing, GD '95*, Berlin, Heidelberg : Springer-Verlag, 1995, p. 504-515, ISBN : 3540607234.
- [150] Colin WARE et Glenn FRANCK, « Evaluating Stereo and Motion Cues for Visualizing Information Nets in Three Dimensions », in : *ACM Trans. Graph.* 15.2 (1996), p. 121-140, ISSN : 0730-0301, DOI : [10.1145/234972.234975](https://doi.org/10.1145/234972.234975).
- [151] Colin WARE et Peter MITCHELL, « Visualizing Graphs in Three Dimensions », in : *ACM Transactions on Applied Perception* 5.1 (2008), ISSN : 1544-3558, DOI : [10.1145/1279640.1279642](https://doi.org/10.1145/1279640.1279642).
- [152] Duncan J. WATTS et Steven H. STROGATZ, « Collective dynamics of 'small-world' networks », in : *Nature* 393.6684 (juin 1998), p. 440-442, ISSN : 1476-4687, DOI : [10.1038/30918](https://doi.org/10.1038/30918), URL : <https://doi.org/10.1038/30918>.
- [153] Eric W. WEISSTEIN, "*Bipartite Graph.*" *From MathWorld—A Wolfram Web Resource*, URL : <https://mathworld.wolfram.com/BipartiteGraph.html>.
- [154] Eric W. WEISSTEIN, "*Complete Graph.*" *From MathWorld—A Wolfram Web Resource*, URL : <https://mathworld.wolfram.com/CompleteGraph.html>.
- [155] Christo WILSON et al., « User Interactions in Social Networks and Their Implications », in : *Proceedings of the 4th ACM European Conference on Computer Systems, EuroSys '09*, Association for Computing Machinery, 2009, p. 205-218, ISBN : 9781605584829, DOI : [10.1145/1519065.1519089](https://doi.org/10.1145/1519065.1519089).

-
- [156] Jieting WU, Lina YU et Hongfeng YU, « Texture-based edge bundling : A web-based approach for interactively visualizing large graphs », in : *2015 IEEE International Conference on Big Data (Big Data)*, 2015, p. 2501-2508, DOI : 10.1109/BigData.2015.7364046.
- [157] Erjia YAN et Ying DING, « Discovering author impact : A PageRank perspective », in : *Information Processing and Management* 47.1 (2011), p. 125-134, ISSN : 0306-4573, DOI : <https://doi.org/10.1016/j.ipm.2010.05.002>.
- [158] Yalong YANG et al., « Maps and Globes in Virtual Reality », in : *Computer Graphics Forum* 37.3 (2018), p. 427-438, DOI : 10.1111/cgf.13431.
- [159] Zhirong YANG, Jaakko PELTONEN et Samuel KASKI, « Optimization Equivalence of Divergences Improves Neighbor Embedding », in : *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, Beijing, China : Journal of Machine Learning Research, 2014, II-460-II-468.
- [160] Wayne W. ZACHARY, « An Information Flow Model for Conflict and Fission in Small Groups », in : *Journal of anthropological research* 33 (1977), p. 452-473.
- [161] Jeff ZACKS et al., « Reading Bar Graphs : Effects of Extraneous Depth Cues and Graphical Context », in : *Journal of Experimental Psychology : Applied* 4 (juin 1998), p. 119-138, DOI : 10.1037/1076-898X.4.2.119.
- [162] Lukáš ZAORÁLEK, Tomáš BURIÁNEK et Václav SNÁŠEL, « Dimension reduction methods in graph drawing problem », in : *2014 14th International Conference on Intelligent Systems Design and Applications*, 2014, p. 13-18, DOI : 10.1109/ISDA.2014.7066252.
- [163] Hao ZHANG et al., « Efficient Local Clustering Coefficient Estimation in Massive Graphs », in : *Database Systems for Advanced Applications*, sous la dir. de Selçuk CANDAN et al., Cham : Springer International Publishing, 2017, p. 371-386, ISBN : 978-3-319-55699-4.
- [164] Delu ZHU et al., « Parallelized Force-Directed Edge Bundling on the GPU », in : *2012 11th International Symposium on Distributed Computing and Applications to Business, Engineering Science*, 2012, p. 52-56, DOI : 10.1109/DCABES.2012.16.

Annexes

ANNEXE 1

Projection des nœuds

Le tableau voir tableau 7.1 récapitule les formules de projection des nœuds sur la demi-sphère, sur le cône et sur la portion de tore.

2D	Projection sur la demi-sphère	Projection sur le cône	Projection sur la portion de tore
$y = 0$	$y = \sqrt{1 - r^2}$	$y = 1 - r$	$y = 1 - \sqrt{1 - (r - 1)^2}$

TABLE 7.1 – Illustration des différentes projections selon l’axe vertical. x , y et z représentent les coordonnées d’un point et $r = \sqrt{x^2 + z^2}$.

Projection des arêtes

Soient $\{j, k\}$, une arête entre les nœuds j et k , et $P_i(x_i, y_i, z_i)$ chaque point de contrôle appartenant à $\{j, k\}$.

$P_i = P_j + (P_k - P_j)t$ où P_j et P_k sont respectivement la position des nœuds j et k , et $t = i/(n - 1)$ où n le nombre de points de contrôle de $\{j, k\}$.

Projection des arêtes sur la demi-sphère

La projection des arêtes sur la demi-sphère se déroule en deux étapes :

- Calcul du point projeté $P_p = \frac{P_i}{\|P_i\|}$
- Calcul de l’altitude $y_{P_p} = \sqrt{1 - (x_{P_p}^2 + z_{P_p}^2)}$

Projection des arêtes sur le cône

Ci-après les quatre étapes qui permettent de projeter les arêtes sur le cône :

- Calcul de l’angle θ entre les axes X et Z du point à projeter : $\theta = \frac{180}{\pi} \text{atan2}(z_i, x_i)$

-
- Rotation de θ autour de l'axe Y. Soit R le résultat de cette rotation :

$$R = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- Calcul du point projeté : $P_p = \frac{x_i \cdot x_R + y_i \cdot y_R + z_i \cdot z_R}{\|R\|} \cdot R$
- Calcul de l'altitude : $y_{P_p} = 1 - \sqrt{x_{P_p}^2 + z_{P_p}^2}$

Projection des arêtes sur la portion de tore

La projection des arêtes sur la portion de tore se déroule également en quatre étapes :

- Calcul de l'angle θ entre les axes X et Z du point P_i : $\theta = \frac{180}{\pi} \text{atan2}(z_i, x_i)$
- Rotation de θ autour de l'axe Y. Soit R le résultat de cette rotation :

$$R = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- Calcul du point projeté : $P_p = \frac{P_i}{\|P_i\|} + R$
- Calcul de l'altitude du point P_p :

$$y_{P_p} = 1 - \sqrt{1 - ((r - 1)(r - 1))}, \text{ avec } r = \sqrt{x_{P_p}^2 + z_{P_p}^2}.$$

ANNEXE 2

Fiche de consentement

La fiche de consentement suivante a été utilisée lors de nos évaluations présentées dans la section 4.5.



Fiche de consentement éclairé pour la participation à une expérimentation en visualisation de graphes

Nom de la recherche: Visualisation radiale 3D

Etudiant-chercheur :

- Piriziwè Kobina, Doctorant, IMT Atlantique / Lab-STICC

Responsables :

- Thierry Duval, Professeur, IMT Atlantique / Lab-STICC
- Laurent Brisson, Maître de conférences, IMT Atlantique / Lab-STICC

Organismes subventionnaires : Région Bretagne et Pôle d'Excellence Cyber

Préambule

Nous vous proposons de participer à un projet de recherche en Sciences de l'Information et de la Communication. Votre participation permettra d'améliorer d'une manière ou d'une autre les systèmes de visualisation d'informations.

Veillez lire attentivement ce formulaire ou demandez qu'on vous le lise avant d'accepter de participer à cette expérimentation. Vous rencontrerez éventuellement des expressions ou des termes que vous ne comprenez pas. Pour cela, n'hésitez pas à poser des questions à l'étudiant-chercheur et/ou aux autres membres affectés à ce projet. Si vous êtes d'accord pour y participer, veuillez accepter et valider ce formulaire. Nous vous en remettrons un exemplaire.

Nature et but du projet de recherche

Les graphes sont le plus souvent utilisés pour représenter les systèmes complexes tels que les réseaux sociaux et les réseaux biologiques. Ainsi, la visualisation d'un graphe peut aider à mieux comprendre la structure des données.

Pour cela, il existe de nombreuses méthodes de visualisation de graphes. L'utilisation de différentes méthodes pour visualiser un graphe peut fournir des apparences visuelles très différentes.

Ainsi, notre étude consiste, d'une part, à analyser l'utilisabilité de la 3D pour explorer et analyser les données représentées sous forme de graphes, et d'autre part, à identifier la meilleure représentation pour visualiser les graphes.

Nature de la participation et déroulement du projet de recherche

Après la validation du présent formulaire, vous commencerez par une série de tutoriels qui vous permet de vous familiariser avec notre outil. L'expérimentation consiste à réaliser une série de tâches liées aux graphes et sur différentes surfaces.

À la fin de l'expérience, vous aurez à répondre à un questionnaire sur l'utilisabilité du système et l'expérience utilisateur.

Risques et avantages

La participation à ce projet pourrait créer chez certains utilisateurs, pour une courte durée, des effets peu agréables assimilables au mal de transports. Ainsi, votre participation pourrait occasionner une fatigue et/ou un inconfort visuel. Dans ce cas, vous pouvez arrêter l'expérience à tout moment. L'étudiant-chercheur vous offrira l'occasion de poursuivre cette expérience à tout moment si vous le désirez.

Par contre, les résultats obtenus lors de ce projet de recherche pourraient contribuer à l'amélioration des systèmes de visualisation de données.

Participation volontaire et possibilité de retrait

Votre participation à ce projet de recherche est entièrement volontaire. Vous êtes donc libre de refuser d'y participer. Vous pouvez également cesser de participer à ce projet à n'importe quel moment, sans avoir à donner de raisons, en faisant connaître votre décision à l'étudiant-chercheur du projet.

De même, l'étudiant-chercheur du projet peut mettre fin à votre participation, sans consentement, notamment si vous ne respectez pas les consignes du projet de recherche ou s'il existe des raisons administratives d'abandonner le projet.

Si vous vous retirez ou êtes retiré.e.s du projet, les informations déjà obtenues dans le cadre de ce projet seront conservées aussi longtemps que nécessaire pour assurer l'intégrité de l'étude et respecter les exigences réglementaires.

Toute nouvelle connaissance acquise durant le déroulement du projet qui pourrait affecter votre décision de continuer d'y participer vous sera communiquée, sans délai, verbalement et par écrit.

Confidentialité

Durant votre participation à ce projet, l'étudiant-chercheur recueillera et consignera dans un dossier de recherche les renseignements vous concernant, mais sans jamais que votre nom soit mentionné. Seuls les renseignements nécessaires pour répondre aux objectifs scientifiques de ce projet seront recueillis.

Tous les renseignements recueillis demeureront strictement confidentiels dans les limites prévues par la Loi 78-17 du 6 janvier 1978 modifiée relative à l'informatique, aux fichiers et aux libertés. Toutes les mesures seront prises pour s'assurer de la confidentialité de votre dossier s'il est consulté et toute information collectée demeurera confidentielle. Afin de préserver votre identité et la confidentialité des renseignements, vous ne serez identifiés que par un numéro de code. La clé du code reliant votre nom à votre dossier de recherche sera conservée par l'étudiant-chercheur. Il utilisera les données à des fins de recherche dans le but de répondre aux objectifs scientifiques du projet décrit dans la présente lettre d'information et de consentement.

Les résultats pourront être publiés dans les revues spécialisées ou faire l'objet de discussions scientifiques, mais il ne sera pas possible de vous identifier.

À des fins de protection, notamment afin de pouvoir communiquer avec vous rapidement, vos noms et prénoms, vos coordonnées et les dates de début et de fin de votre participation au projet seront conservés et maintenus dans un répertoire par l'étudiant-chercheur pendant un an après la fin du projet.

Vous avez le droit de consulter votre dossier de recherche pour vérifier les renseignements recueillis et les faire modifier au besoin, et ce, aussi longtemps que l'étudiant-chercheur du projet ou l'établissement détienne cette information. Cependant, afin de préserver l'intégrité scientifique du projet, vous pourriez n'avoir accès qu'à certains de ces renseignements, une fois votre participation terminée.

Droit du sujet de recherche

En acceptant de participer à ce projet de recherche, vous ne renoncez à aucun de vos droits ni ne libérez les chercheurs ou l'établissement (IMT Atlantique) de leurs responsabilités civile et professionnelle.

Communication des résultats généraux

Les résultats globaux de cette étude pourront être à votre disposition sur demande et après la fin du projet. Cette demande devra être faite par écrit auprès de l'étudiant-chercheur.

Personnes ressources

Si vous avez des questions en rapport avec le projet de recherche ou si vous éprouvez un problème concernant votre participation au projet, vous pouvez contacter l'étudiant-chercheur:

Piriziwè Kobina, doctorant

IMT Atlantique

655 Avenue du Technopôle, 29280 Plouzané

piriziwe.kobina@imt-atlantique.fr

Si vous avez des questions concernant vos droits en tant que participant à une étude de recherche ou si vous avez des plaintes à formuler, vous pouvez communiquer avec l'étudiant-chercheur à travers l'adresse ci-dessus mentionnée.

Fait à, le

Signature du participant (précédée de la mention « lu et approuvé »):

ANNEXE 3

Questionnaire et configurations de la première évaluation

Le questionnaire et les configurations suivants ont été utilisés lors de notre première évaluation présentée dans la section 4.5.1.

Questionnaire expérience utilisateur

1. Je pense avoir compris ce qu'on attendait de moi

1 2 3 4 5

Pas du tout d'accord Tout à fait d'accord

2. Je pense avoir compris les tâches demandées

1 2 3 4 5

Pas du tout d'accord Tout à fait d'accord

3. Je pense avoir réalisé de manière pertinente les tâches demandées

1 2 3 4 5

Pas du tout d'accord Tout à fait d'accord

4. Je pense que cet outil est pertinent pour une utilisation par des experts en analyse des graphes

1 2 3 4 5

Pas du tout d'accord Tout à fait d'accord

5. J'ai éprouvé des difficultés à interagir avec l'outil

1 2 3 4 5

Pas du tout d'accord Tout à fait d'accord

6. Remarques complémentaires

Conséquences de l'expérience

7. Je me suis senti.e fatigué.e durant mon interaction avec l'environnement web

1 2 3 4 5 6 7 8 9 10

Pas du tout d'accord Tout à fait d'accord

8. J'ai ressenti les maux de tête durant mon interaction avec l'environnement web

1 2 3 4 5 6 7 8 9 10

Pas du tout d'accord Tout à fait d'accord

9. J'ai ressenti une fatigue visuelle durant mon interaction avec l'environnement web

1 2 3 4 5 6 7 8 9 10

Pas du tout d'accord Tout à fait d'accord

Questions ouvertes

10. À votre avis, quels ont été les points positifs de votre expérience ?

11. À votre avis, quels ont été les points négatifs de votre expérience ?

12. Quelle(s) surface(s) vous a (ont) permis de bien réaliser les tâches demandées ?

- 2D uniforme
- 2D avec emphase sur le centre
- 2D avec emphase sur la périphérie
- Demi-sphère
- Cône
- Portion de tore

13. Avec quelle(s) surface(s) avez-vous éprouvé des difficultés à réaliser les tâches demandées ?

- 2D uniforme
- 2D avec emphase sur le centre
- 2D avec emphase sur la périphérie
- Demi-sphère
- Cône
- Portion de tore

14. Préférences

	2D uniforme	2D avec emphase sur le centre	2D avec emphase sur la périphérie	Demi-sphère	Cône	Portion de tore
Quelle(s) surface(s) préférez-vous ?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Quelle(s) surface(s) n'aimez-vous pas ?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

15. Avez-vous des suggestions pour améliorer cet outil de visualisation ?

		2D			3D				2DU	Uniform 2D
Bloc 1	1	2DU+G1	2DC+G2	2DP+G3	C+G4	T+G5	S+G6		2DC	2D with Central emphasis
	2	2DU+G2	2DP+G3	2DC+G4	C+G5	S+G6	T+G1		2DP	2D with Peripheral emphasis
	3	2DC+G3	2DU+G4	2DP+G5	T+G6	C+G1	S+G2		C	Cone
	4	2DC+G4	2DP+G5	2DU+G6	T+G1	S+G2	C+G3		T	Torus
	5	2DP+G5	2DU+G6	2DC+G1	S+G2	C+G3	T+G4		H	Half sphere
	6	2DP+G6	2DC+G1	2DU+G2	S+G3	T+G4	C+G5		Gi	Graph number i
		3D			2D					
Bloc 2	7	C+G6	T+G5	S+G4	2DU+G3	2DC+G2	2DP+G1			
	8	C+G5	S+G4	T+G3	2DU+G2	2DP+G1	2DC+G6			
	9	T+G4	C+G3	S+G2	2DC+G1	2DU+G6	2DP+G5			
	10	T+G3	S+G2	C+G1	2DC+G6	2DP+G5	2DU+G4			
	11	S+G2	C+G1	T+G6	2DP+G5	2DU+G4	2DC+G3			
	12	S+G1	T+G6	C+G5	2DP+G4	2DC+G3	2DU+G2			
		2D			3D					
Bloc 3	13	2DU+G6	2DC+G5	2DP+G4	C+G3	T+G2	S+G1			
	14	2DU+G5	2DP+G4	2DC+G3	C+G2	S+G1	T+G6			
	15	2DC+G4	2DU+G3	2DP+G2	T+G1	C+G6	S+G5			
	16	2DC+G3	2DP+G2	2DU+G1	T+G6	S+G5	C+G4			
	17	2DP+G2	2DU+G1	2DC+G6	S+G5	C+G4	T+G3			
	18	2DP+G1	2DC+G6	2DU+G5	S+G4	T+G3	C+G2			
		3D			2D					
Bloc 4	19	C+G1	T+G2	S+G3	2DU+G4	2DC+G5	2DP+G6			
	20	C+G2	S+G3	T+G4	2DU+G5	2DP+G6	2DC+G1			
	21	T+G3	C+G4	S+G5	2DC+G6	2DU+G1	2DP+G2			
	22	T+G4	S+G5	C+G6	2DC+G1	2DP+G2	2DU+G3			
	23	S+G5	C+G6	T+G1	2DP+G2	2DU+G3	2DC+G4			
	24	S+G6	T+G1	C+G2	2DP+G3	2DC+G4	2DU+G5			

ANNEXE 4

Questionnaire et configurations de la deuxième évaluation

Le questionnaire et les configurations suivants ont été utilisés lors de notre deuxième évaluation présentée dans la section 4.5.2.

Expérience utilisateur

Utilisation du système

* Required

1. Utilisateur *

2. Je pense avoir compris ce qu'on attendait de moi *

1 2 3 4 5

Pas du tout d'accord Tout à fait d'accord

3. Je pense avoir compris les tâches demandées *

1 2 3 4 5

Pas du tout d'accord Tout à fait d'accord

4. Je pense avoir réalisé de manière pertinente les tâches demandées *

1 2 3 4 5

Pas du tout d'accord Tout à fait d'accord

5. Je pense que la 2D est pertinente pour une utilisation par des experts en analyse des graphes *

1 2 3 4 5

Pas du tout d'accord Tout à fait d'accord

6. Je pense que la 3D desktop est pertinente pour une utilisation par des experts en analyse des graphes *

1 2 3 4 5

Pas du tout d'accord Tout à fait d'accord

21. Si j'utilise de nouveau l'environnement virtuel, mon interaction avec l'environnement sera claire et compréhensible pour moi *

1 2 3 4 5 6 7 8 9 10

Pas du tout d'accord Tout à fait d'accord

22. J'arriverai aisément à acquérir les compétences requises pour utiliser l'environnement desktop *

1 2 3 4 5 6 7 8 9 10

Pas du tout d'accord Tout à fait d'accord

23. J'arriverai aisément à acquérir les compétences requises pour utiliser l'environnement virtuel *

1 2 3 4 5 6 7 8 9 10

Pas du tout d'accord Tout à fait d'accord

24. Utiliser ces périphériques d'interaction (clavier et souris) en desktop est une mauvaise idée *

1 2 3 4 5 6 7 8 9 10

Pas du tout d'accord Tout à fait d'accord

25. Utiliser ces périphériques d'interaction (manettes) en environnement virtuel est une mauvaise idée *

1 2 3 4 5 6 7 8 9 10

Pas du tout d'accord Tout à fait d'accord

Questions ouvertes

26. À votre avis, quels ont été les points positifs de votre expérience avec la 2D ?

27. À votre avis, quels ont été les points positifs de votre expérience avec la 3D desktop ?

28. À votre avis, quels ont été les points positifs de votre expérience avec la 3D immersive ?

29. À votre avis, quels ont été les points négatifs de votre expérience avec la 2D ?

30. À votre avis, quels ont été les points négatifs de votre expérience avec la 3D desktop ?

31. À votre avis, quels ont été les points négatifs de votre expérience avec la 3D immersive ?

32. Quel (s) environnement (s) vous semble (nt) le (s) plus adapté (s) pour visualiser les graphes de grandes tailles (beaucoup de nœuds et d'arêtes) ? *

- 2D
- 3D desktop
- 3D immersive

33. Quel (s) environnement (s) préférez-vous ? *

- 2D
- 3D desktop
- 3D immersive

34. Avez-vous des suggestions pour améliorer cet outil de visualisation ?

Configurations				2D	2D visualization on classique display
1	2D+G1	3D+G1	3D_VR+G1	3D	3D visualization on classique display
2	2D+G1	3D_VR+G1	3D+G1	3D_VR	Immersive 3D visualization
3	3D+G1	3D_VR+G1	2D+G1	G_i	Graph number i
4	3D+G1	2D+G1	3D_VR+G1		
5	3D_VR+G1	2D+G1	3D+G1		
6	3D_VR+G1	3D+G1	2D+G1		
7	2D+G2	3D+G2	3D_VR+G2		
8	2D+G2	3D_VR+G2	3D+G2		
9	3D+G2	3D_VR+G2	2D+G2		
10	3D+G2	2D+G2	3D_VR+G2		
11	3D_VR+G2	2D+G2	3D+G2		
12	3D_VR+G2	3D+G2	2D+G2		

Titre : Nouvelles visualisations 3D de graphes pour l'exploration des relations intra et inter communautés

Mot clés : Visualisation 3D, Réalité virtuelle, Graphes de communautés, Analyse de graphes

Résumé : L'objectif de cette thèse était d'étudier les contributions de la visualisation 3D immersive par rapport à la visualisation classique 2D afin d'améliorer l'exploration et l'analyse des relations intracommunautés et intercommunautés représentées sous forme de graphes. Après une analyse de l'état de l'art sur la théorie des graphes et ses applications à l'analyse des réseaux sociaux et sur des méthodes de visualisation de graphes, nous avons proposé deux approches de visualisation de graphes : une visualisation exocentrique, qui permet à un utilisateur de manipuler un graphe comme un objet, pour permettre de visualiser intuitivement la centralité des nœuds et une visualisation égocentrique, spécialement adaptée à l'exploration d'un graphe en environnement immersif, où notre mécanisme de rétractation de l'espace d'affichage du graphe permet à un utilisateur d'avoir une vision d'ensemble du graphe tout en restant en mode égocentrique. Les applications de ces travaux sont tout aussi civiles que militaires.

Title: New 3D graph visualizations for intra and inter communities' relationships exploration

Keywords: 3D visualization, Virtual reality, Community graphs, Graph analysis

Abstract: The aim of this thesis was to study the contributions of immersive 3D visualization compared to classical 2D visualization in order to improve the exploration and analysis of intra-community and inter-community relationships represented as graphs. After an analysis of the state of the art on graph theory and its applications to social network analysis and on graph visualization methods, we proposed two graph visualization approaches: an exocentric visualization, which allows a user to manipulate a graph as an object, to intuitively visualize the centrality of the nodes and an egocentric visualization, especially adapted to the exploration of a graph in an immersive environment, where our mechanism of retraction of the graph display space allows a user to have an overall view of the graph while remaining in egocentric mode. The applications of this work are both civilian and military.