

Université du Havre
Mémoire de Master deuxième année
Informatique

Réseaux viaires et évacuation

Auteur

**Thibaut
Démare**

Encadrants

**Damien
Olivier**

**Antoine
Dutot**

Date

**27 Juin
2012**



Ce travail est mis à disposition selon les termes de la [Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 3.0 non transposé](#).

Remerciements

Je tiens tout d'abord à remercier mes deux encadrants MM. Antoine DUTOT et Damien OLIVIER qui se sont montrés présents et à l'écoute. Ils ont été de bons conseillers dans mes choix et c'est sans nulle doute grâce à eux que nous traitons ici de fourmis.

Je souhaite également exprimer toute ma gratitude aux membres du laboratoire du Litis qui ont été très accueillants pendant ces quatre mois. Je tiens d'ailleurs à témoigner ma reconnaissance envers M. Cyrille BERTELLE qui a bien voulu accepter ma candidature pour une thèse l'année prochaine.

Bien sûr je ne saurais oublier mes camarades de stage avec qui l'ennui ne s'est jamais fait sentir. Que ce soit lors des discussions sur les jeux vidéo, sur les films, lors de celles plus sérieuses sur nos sujets respectifs, ou lors de cette mémorable randonnée à Étretat et son GR21, j'ai toujours passé des moments très appréciables. Dans l'ordre alphabétique de leur prénom je cite ainsi Alexis, Florent, Guillaume, Merwan et Mohamed Ali. Toutefois, désolé à ceux qui se reconnaîtront que nous n'ayons pas plus parler de moustaches.

Un grand merci également, et non des moindres, à Xavière et à mes parents pour partager ma passion de l'informatique alors que ce n'est pas forcément leur tasse de thé au départ.

Et pour finir, merci au lecteur qui passera par là et accordera un peu de son temps pour lire ces quelques pages.

Table des matières

Abstract	4
Introduction	6
1 Le problème	8
2 La centralité d'intermédiairité	9
2.1 L'approche classique	10
2.1.1 Les origines et les formules de calcul	10
2.1.2 L'algorithme de Brandes	11
2.2 Les variantes et les approximations	13
2.2.1 Les variantes	13
2.2.2 Les approximations	16
3 Les méthodes d'analyses	18
3.1 Les méthodes de comparaison	18
3.2 Les graphes utilisés	19
4 Les marches aléatoires	23
4.1 Les origines	23
4.2 Le modèle	25
4.3 Les résultats	26
5 La centralité d'intermédiairité avec pivots	28
5.1 La base du modèle	28
5.2 Le modèle avec des entités effectuant une marche aléatoire	30
5.2.1 Le fonctionnement des entités	30
5.2.2 Les résultats	30
5.3 Le modèle avec des entités utilisant A*	34
5.3.1 Le fonctionnement des entités	34
5.3.2 Les résultats	35
5.4 Le modèle avec des entités "fourmis"	37
5.4.1 Le fonctionnement des entités	37
5.4.2 Le résultat	39
6 Les perspectives	41
Conclusion	43

Abstract

The mesh of a road network can be very complex to understand, to analyze. Its particular topology is first a positive point because it allows good accessibility for users, but it can be a weakness if we route an important mass of vehicles on the same road. Tunnels, bridges, highways, interchanges are structures intended to facilitate transport, but they can become quite the bottlenecks when good or bad conditions are presents.

In order to identify these critical zones, the betweenness centrality can be used. It provides a measure on nodes of a graph by computing, for each node v , the number of shortest paths passing through v among all possible shortest paths of the graph.

This way, when a significant amount of paths traverse a set of nodes, we emphasize major road segments able to handle a large traffic.

If a part of the network becomes inaccessible because of, for example, an industrial accident, and if we wish do an evacuation, then it is useful to know the state of the betweenness centrality in order to provide adapted routes and avoid creating a bottleneck. However, the algorithm is very slow and very expensive to compute. Depending on the size of the graph taken into account, it can be unpractical to recompute all shortest paths at each change.

Thus, the purpose of this report is to present a computing method for the centrality, allowing real-time updates under acceptable delays. To answer to this problematic, we suggest robust methods for this dynamic.

The first model is derived from a hypothesis that the betweenness centrality and a random walk can be correlated. It was an observation from Michel Nabaa's thesis where the coloration of the road network of Le Havre according to these two methods seemed to emphasize the same areas.

The use of our comparatives measures, *e.g.* Pearson's and Spearman's correlation coefficients, have proved the contrary when we have repeated the experience.

So, we have designed a second model closer to the original definition. This one is inspired by Brandes' model using pivot nodes. His algorithm solves all shortest paths from these pivot sources to all the other nodes. For our part, we have chosen to select directly a subset of pairs of nodes rather than to prefer some sources in particular. With the intention of solving the path problem between the nodes of each pair, we have used some entities whose behavior is to find a solution.

The first entity executes a random walk and increments a counter when it crosses a node or an edge. It also deletes the cycle in the path. Unfortunately, the correlation between the centrality and this counter does not exist on some graphs but exists on some other types. In this particular case, the results are not positive on the road networks of two French cities, Le Havre and Rouen, even if we use a biased random walk.

Then, we wanted to prove the viability of our model with an entity using the A* algorithm. The latter allows to compute efficiently a shortest path between two nodes of a graph. Our comparative measures gave us positive results since there is a strong correlation between the betweenness centrality and our model.

However, the A* algorithm can not be applied in the general case because of the heuristics needed. Therefore, we have designed a final entity based on Dorigo's ants. It is a method inspired by the stigmergy where entities coordinate indirectly their actions through the environment and pheromones.

In our model, a master process holds a list of pairs of nodes, and we associate to each pair a different type of pheromone. During the computation, the master process orders the colony to make a path between the source and the destination of a given pair. The entities solve their problem and put down the pheromones on their computed path. The quantity depends on the length of the path.

Since the entities are sensible to the pheromone when they move, they will prefer to choose the shortest path a little more at each step.

Although we have just one result at this time for this model of ant entities, this method is promising and can provide some interesting results in the general case.

However, it is necessary to optimize again the mechanism of the ant entity because it still does not work perfectly on the road network. Furthermore, the method of selection of pairs can surely be improved by a most elaborated mechanism. We can imagine, for example, that we choose the nodes between different clusters determined by a clustering algorithm.

The number of pairs used is also an important parameter of our model but we have not defined his ideal value. Thus, it is an interesting prospect to implement, along with the distribution of our algorithm.

Introduction

Afin de rester compétitif et attractif, une ville doit se doter d'un réseau routier efficace où l'optimisation des temps de transport est recherché. Pour cela, il peut être décidé de créer des axes principaux sur lesquels les vitesses de déplacement et le volume du trafic sont importants.

Toutefois, en donnant une importance à certaines routes, cela risque de créer de véritables goulets d'étranglement. Effectivement, les risques industriels ou les accidents de la route peuvent bloquer ces axes et ainsi paralyser le trafic en très peu de temps. Il devient donc primordial d'étudier les réseaux viaires afin de mettre en valeur les zones sensibles pour comprendre comment améliorer ceux-ci et éviter des catastrophes en cas d'évacuation.

Grâce à la théorie des graphes, nous pouvons effectuer des mesures précises sur le réseau pour déterminer sa robustesse et sa résilience. La centralité d'intermédiairité est l'un de ces indices. Elle se base sur les plus courts chemins entre toutes les paires de nœuds d'un graphe et elle permet justement de répondre à nos attentes. Malheureusement, cette mesure possède un important défaut : son temps de calcul. Par conséquent, si le graphe traité est dynamique, alors il devient très délicat de recalculer tous les plus courts chemins à chaque changement.

Ainsi, au cours de ce rapport, nous nous proposons d'étudier la centralité d'intermédiairité puis de fournir une méthode de calcul robuste à la dynamique d'un graphe.

Dans une première partie, nous présenterons en détail notre problème en montrant l'intérêt de calculer la centralité d'intermédiairité et en particulier dans le contexte dynamique.

Par la suite nous détaillerons l'état de l'art concernant la centralité d'intermédiairité : l'approche classique mais également les méthodes dérivées dont certaines approximent le résultat réel, tandis que d'autres considèrent la recherche des chemins de façon complètement différente.

En deuxième partie, nous considérerons les méthodes d'analyses qui vont nous servir à comparer nos modèles à la centralité classique.

Dans la troisième partie, nous approfondirons notre première approche basée sur des marches aléatoires classiques. Nous verrons notamment que la méthode donne des résultats éloignés de la centralité d'intermédiarité.

Ainsi la quatrième partie sera consacrée à un nouveau modèle, plus prometteur, qui se fonde sur un concept de paires de nœuds dit "pivots".

Finalement, nous explorerons les perspectives d'améliorations avant de conclure.

Chapitre 1

Le problème

L’agglomération havraise est un système complexe dans lequel des entités hétérogènes cohabitent. Mais cela ne se fait pas toujours sans risques ni sans accidents. Le danger est bien présent et dans le but de le limiter ou même de le maîtriser, les collectivités locales ont financé des travaux de recherches pour trouver des solutions.

C’est dans cet objectif que Michel Nabaa a effectué sa thèse [1]. Au cours de son travail, il a utilisé différentes méthodes pour mettre en valeur la robustesse et la résilience du réseau viaire du Havre.

Pourtant la dynamique du système de cet espace géographique implique qu’une crise peut survenir très rapidement. Il est donc nécessaire de réagir promptement. Si un axe majeur se trouve bloqué car il mène ou il appartient à une zone de danger, alors il faut redéfinir les nouveaux axes majeurs. Cela permet de conseiller des itinéraires pour fluidifier le trafic au maximum et facilite une évacuation sans encombres.

La théorie des graphes nous offre des mesures sur ceux ceux-ci nous permettant d’évaluer l’importance de certains nœuds ou arcs, au sens de la connectivité avec par exemple les points d’articulations, au sens de la transitivité comme avec le coefficient de clustering, ou encore au sens des plus courts chemins comme avec la centralité d’intermédiarité. C’est donc sur cette dernière que nous allons orienter notre travail.

Malheureusement, c’est une mesure longue et coûteuse à calculer. Ainsi, nous souhaitons trouver une méthode palliant ces problèmes. Le sujet de ce stage consiste donc à trouver une méthode de calcul de la centralité qui soit rapide, ou permette de résister à la dynamique. De plus, pour minimiser les temps de calcul nous souhaitons favoriser les méthodes distribuables.

Chapitre 2

La centralité d'intermédiarité

Les mesures de centralité dans un graphe sont des indicateurs locaux de la position d'un nœud ou d'un arc dans le graphe. Les définitions de centralité sont toutefois nombreuses. Nous pouvons rencontrer par exemple la centralité de degré (le degré d'un nœud), la centralité de proximité [2] (l'inverse de la somme de la longueur des plus courts chemins entre chaque nœud du graphe), la centralité d'éloignement maximal (la distance maximale entre un nœud et l'ensemble des autres nœuds), et la centralité d'intermédiarité ou d'intermédiaire (voir figure 2.1).

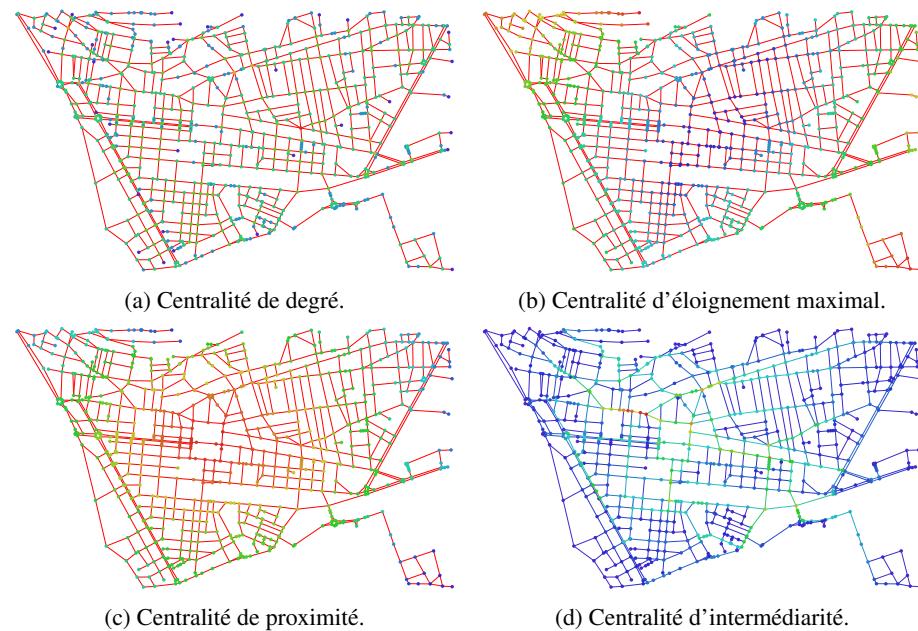


FIGURE 2.1 – Différentes centralités sur le réseau partiel et simplifié de la ville du Havre. Des centralités fortes en rouge aux centralités faibles en bleu.

Dans cette partie nous allons approfondir l'état de l'art de la centralité d'intermédiarité. Nous verrons donc au début de ce chapitre l'approche classique et les méthodes de

calcul, puis nous étudierons les variantes et les références concernant l'approximation de la mesure.

2.1 L'approche classique

2.1.1 Les origines et les formules de calcul

En 1948, Bavelas [3] introduit l'idée qu'un nœud possède une importance accrue lorsqu'il agit comme intermédiaire entre deux autres nœuds pour établir une communication. Cette idée est par la suite reprise par plusieurs personnes dont Anthonisse en 1971 [4]. Il faudra pourtant attendre 1977 et les travaux de Freeman [5, 6] pour que la centralité d'intermédiaire se précise et prenne un formalisme clair afin d'être popularisée.

Selon Freeman, la centralité d'intermédiaire revient à compter le nombre de plus courts chemins passant à travers un nœud parmi tous les plus courts chemins du graphe. Dans un graphe non valué, nous caractérisons la longueur d'un chemin par le nombre d'arêtes qui le constitue.

Soit :

- $G = (V, E)$ un graphe orienté (ou non) et non valué tel que V est l'ensemble de ses sommets, et E est l'ensemble de ses arcs. Nous posons également $n = |V|$ et $m = |E|$.
 - $C_B(k)$ est la centralité associée au nœud k .
 - g_{st} est le nombre de plus court chemin entre s et t .
 - $g_{st}(k)$ est le nombre de plus court chemin entre s et t passant par k .
 - Et nous posons $b_{st}(k) = \frac{g_{st}(k)}{g_{st}}$.
- S'il n'existe pas de chemin entre s et t , alors $b_{st}(k) = 0$.

Ainsi la centralité d'intermédiaire se mesure par :

$$C_B(k) = \sum_s^n \sum_t^n b_{st}(k) \quad s \neq t \neq k$$

Nous remarquons que les extrémités des plus courts chemins ne sont pas prises en compte car nous avons : $s \neq t \neq k$. Pourtant, dans la littérature nous pouvons rencontrer des cas où il n'est pas nécessaire de s'embarrasser de cette différence, comme le dit Newman dans [7]. Effectivement, dans un graphe connexe, le classement des nœuds selon leur centralité ne variera pas si nous considérons ou non les extrémités. Et si le graphe possède plusieurs composantes connexes, ce classement ne variera pas non plus au sein des composantes. Toutefois, il est possible que l'ordre global change en fonction de la taille des composantes.

Comme il est plus courant d'utiliser la forme classique avec $s \neq t \neq k$, nous étudierons ce cas de figure. Les modifications à apporter sont, de toute manière, minimes.

Normalisation Freeman propose une manière de comparer la centralité de deux nœuds n'étant pas dans un même graphe. Il normalise la centralité de cette manière :

$$C_{NB}(k) = \frac{2C_B(k)}{n^2 - 3n + 2}$$

Il considère ainsi que la centralité maximale est atteinte par le nœud central d'un graphe en étoile.

Nous pouvons également normaliser la centralité en la divisant par la centralité maximale trouvée dans le graphe. Nous avons donc :

$$C_{NB}(k) = \frac{C_B(k)}{\max_{u \leq n}(C_B(u))}$$

Le cas des graphes valués ne change pas énormément. Il ne s'agit pas de modifier la mesure en fonction des pondérations (bien que ce soit possible comme nous le verrons plus tard), mais simplement de caractériser les chemins différemment. Nous n'additionnons plus le nombre d'arêtes pour déterminer la longueur d'un chemin mais nous additionnons leur pondération.

2.1.2 L'algorithme de Brandes

La centralité d'intermédiairité est une mesure très coûteuse à calculer. Effectivement, si nous considérons un algorithme naïf reprenant la formule brute de la centralité d'intermédiairité, sa complexité en temps est de $O(n^3)$. C'est donc dans l'optique d'améliorer l'algorithme que Brandes en a proposé un autre en 2001, plus rapide, et dont la complexité est de $O(nm)$ pour les graphes non valués, et $O(nm + n^2 \log n)$ pour les graphes valués ([8]).

Son algorithme se base sur une formule de récursion. Si nous posons les termes suivants :

- $\sigma(s, t)$: le nombre de plus courts chemins entre s et t .
- $\sigma(s, t|v)$: le nombre de plus courts chemins entre s et t passant par v .
- et $\delta(s, t|v) = \sigma(s, t|v)/\sigma(s, t)$.

Nous obtenons ceci :

$$\delta(s|v) = \sum_{t \in V} \delta(s, t|v)$$

Ce qui nous permet d'exploiter la relation qui suit :

$$\delta(s|v) = \sum_{\substack{w \in V \\ w: (v,w) \in E \text{ et} \\ dist(s,w)=dist(s,v)+1}} \frac{\sigma(s, v)}{\sigma(s, w)} \cdot (1 + \delta(s, w))$$

Et comme $C_B(v)$ peut s'exprimer ainsi : $C_B(v) = \sum_{s \in V} \delta(s|v)$, il est possible de mesurer la centralité d'un nœud en accumulant les $\delta(s|v)$ progressivement calculés.

```

1 Entrée:
2   un graphe orienté  $G = (V, E)$ 
3 Données:
4   une queue  $Q$  initialisée à vide,
5   une pile  $S$  initialisée à vide,
6 Pour chaque  $v \in V$ 
7    $\text{dist}[v]$  : la distance depuis la source
8    $\text{Pred}[v]$  : la liste des prédecesseurs du plus court chemin depuis la source
9    $\sigma[v]$  : le nombre de plus court chemin depuis la source jusqu'à  $v$ 
10   $\delta[v]$  : la dépendance de la source sur  $v$ 
11 Sortie :
12  La centralité d'intermédiaire est  $C_B[v]$  pour chaque  $v \in V$  (initialisée à 0)
13 Début
14 Pour chaque  $s \in V$  faire
15 // Problème du plus court chemin
16 // Initialisation
17 Pour chaque  $w \in V$  faire  $\text{Pred}[w] \leftarrow$  liste vide
18 Pour chaque  $t \in V$  faire  $\text{dist}[t] \leftarrow \infty$  et  $\sigma[t] \leftarrow 0$ 
19  $\text{dist}[s] \leftarrow 0$ 
20  $\sigma[s] \leftarrow 1$ 
21 Ajouter  $s$  à  $Q$ 
22 // Construction du chemin
23 Tant que  $Q$  n'est pas vide faire
24   retirer  $v$  de  $Q$ 
25   empiler  $v$  à  $S$ 
26 Pour chaque  $w \in V | \exists(v, w) \in E$  faire
27   // Découverte du chemin
28   Si  $\text{dist}[w] = \infty$  alors
29      $\text{dist}[w] \leftarrow \text{dist}[v] + 1$ 
30     ajouter  $w$  à  $Q$ 
31 Fin si
32 // Comptage de chemin - l'arc  $(v, w)$  est-il dans un plus court chemin ?
33 Si  $\text{dist}[w] = \text{dist}[v] + 1$  alors
34    $\sigma[w] \leftarrow \sigma[w] + \sigma[v]$ 
35   ajouter  $v$  à  $\text{Pred}[w]$ 
36 Fin si
37 Fin pour chaque
38 Fin tant que
39 // Accumulation - rétro-propagation des dépendances
40 Pour chaque  $v \in V$  faire  $\delta[v] \leftarrow 0$ 
41 Tant que  $S$  n'est pas vide faire
42   dépiler  $w$  de  $S$ 
43   Pour chaque  $v \in \text{Pred}[w]$  faire  $\sigma[v] \leftarrow \sigma[v] + \frac{\sigma[w]}{\sigma[w]} \cdot (1 + \sigma[w])$ 
44   Si  $w \neq s$  alors  $C_B[w] \leftarrow C_B[W] + \delta[w]$ 
45 Fin tant que
46 Fin pour chaque
47 Fin

```

Cet algorithme est l'un des plus utilisés aujourd'hui pour effectuer le calcul exact de la centralité d'intermédiarité. Toutefois, sa complexité reste encore trop importante pour qu'il soit efficace dans de très grands graphes. C'est ainsi que plusieurs auteurs ont proposé des variantes de la mesure afin d'approximer l'ordre d'importance des nœuds ([9]), tandis que d'autres ont modifié l'approche classique lorsque celle-ci ne permettait pas de mettre en valeur certains nœuds importants ([10]). Nous allons donc approfondir dans la section qui suit ces différentes variantes.

2.2 Les variantes et les approximations

2.2.1 Les variantes

L'un des défauts de la centralité d'intermédiarité est que l'information transite au sein du graphe en empruntant les plus courts chemins. Pourtant, nous rendons rapidement compte que ce n'est pas toujours le cas. Effectivement, dans certains réseaux, notamment les réseaux sociaux, les entités associées aux nœuds ont une connaissance imparfaite de la topologie du graphe. Elles ne sont donc pas en mesure de "router" les messages efficacement en leur faisant emprunter exclusivement les plus courts chemins.

Ainsi, en 1991, Freeman *et al.* proposent un nouveau modèle basé sur les flots [11]. Il ne s'agit donc plus de calculer les plus courts chemins mais de considérer l'ensemble des nœuds appartenant au flot maximal. Ce dernier est défini selon l'algorithme de Ford et Fulkerson [12].

Ces idées ont ensuite été reprises, notamment par Borgatti [13]. Dans son article, il montre qu'il faut parfois considérer un processus particulier pour calculer le flot. Ceci implique que le modèle de Freeman n'est pas toujours approprié et qu'il est nécessaire de l'adapter.

La centralité basée sur les flots a l'avantage de fournir une liste de nœuds potentiellement plus importants. Et celle-ci peut contenir les nœuds appartenant au plus court chemin ou à de "bon" chemin. Cependant, cette propriété n'est pas toujours vérifiée. Le flot maximal entre deux sommets d'un graphe peut tout à fait emprunter un détour important, bien loin du plus court chemin ou des "bons" chemins.

De plus, l'algorithme de Ford et Fulkerson ne considère plus de nouveaux nœuds, ou chemins, lorsque nous avons atteint le flot maximal sur la coupe minimale. Par exemple sur la figure 2.2, la recherche de flot détermine qu'il est maximal entre les sommets de couleur en empruntant le chemin vert, et l'arête valuée à 1 appartient à la coupe minimale. L'algorithme stoppera donc son calcul pour le flot entre ces sommets alors que le chemin orange semble tout aussi intéressant.

Nous remarquons aussi que dans un réseau routier il n'est pas nécessaire d'atteindre la limite du volume d'une route avant que certains véhicules décident d'emprunter d'autres chemins. Nous pouvons ainsi voir que ce modèle n'est pas à l'image de la réalité (en tout cas pour notre objet d'étude). Ces remarques sont d'ailleurs mises en valeur dans [10] et [14].

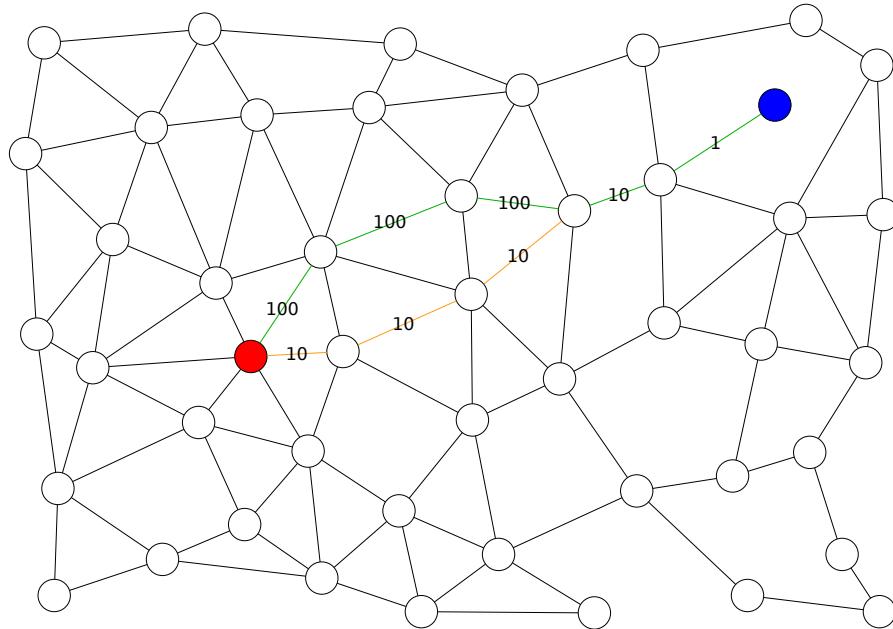


FIGURE 2.2 – Mise en valeur du flot maximal (en vert) entre deux sommets et d'une portion d'un chemin alternatif (en orange). Les valeurs correspondent à une capacité de flot.

D'autres méthodes ont également vu le jour afin de considérer plus de chemins, et pas nécessairement les plus courts. Elles se basent notamment sur la recherche de chemins d'une longueur maximale, comme la k-centralité d'intermédiairité (k-betweenness centrality) dont parle Borgatti dans [15]. Il s'agit de prendre en compte tous les chemins étant d'une longueur k au plus. Cette idée a été formalisée dans [16] qui en propose une implémentation parallélisée.

À partir de la même idée, il est également possible de considérer les chemins dont les arêtes (ou les nœuds) sont disjoints.

De même, il existe des formules prenant en compte les longueurs des chemins. Celles-ci se montrent souvent intéressantes et fournissent des données supplémentaires à la mesure. Certains auteurs, notamment ceux issus de la sociologie, préconisent de pondérer les plus courts chemins proportionnellement à leur longueur inverse ([15]). Ainsi, nous augmenterons la centralité d'un nœud jouant le rôle d'intermédiaire dans des chemins de petite taille. Soit :

$$C_B(k) = \sum_s^n \sum_{\substack{t \\ s \neq t \neq k}}^n \frac{1}{\text{dist}(s, t)} b_{st}(k)$$

tel que $\text{dist}(s, t)$ retourne la longueur du chemin entre s et t .

Dans [17] et [18], les auteurs montrent que nous pouvons également pondérer la contribution d'un chemin en fonction de la place du nœud dans ce chemin. Geisberger *et*

al. s'en servent pour effectuer une approximation de la centralité mais Brandes affirme qu'il peut s'agir d'une variante intéressante de celle-ci. Ainsi le nœud peut être mieux pondéré si :

- il se trouve au centre, tel que :

$$C_B(k) = \sum_s^n \sum_{\substack{t \\ s \neq t \neq k}}^n \frac{2 \times \min(\text{dist}(s, k), \text{dist}(k, t))}{\text{dist}(s, t)} b_{st}(k)$$

- il se trouve proche du nœud source du chemin, tel que :

$$C_B(k) = \sum_s^n \sum_{\substack{t \\ s \neq t \neq k}}^n \frac{\text{dist}(k, t)}{\text{dist}(s, t)} b_{st}(k)$$

- il se trouve proche du nœud cible du chemin, tel que :

$$C_B(k) = \sum_s^n \sum_{\substack{t \\ s \neq t \neq k}}^n \frac{\text{dist}(s, k)}{\text{dist}(s, t)} b_{st}(k)$$

Ces versions de la centralité peuvent notamment prendre de l'importance dans des graphes valués comme un réseau routier. Prenons l'exemple de la figure 2.3, si nous considérons une approche classique du calcul de la centralité, alors les nœuds rouge et orange devraient obtenir le même score. Pourtant, si nous considérons la pondération sur les arcs, il semble bien qu'il existe une différence entre ces deux nœuds. En traçant un cercle passant par les nœuds vert nous nous apercevons que le nœud rouge en est le centre. Si nous souhaitons faire apparaître ce type d'information dans la mesure finale, alors nous pouvons utiliser la formule ci-dessus favorisant les nœuds au centre des chemins.

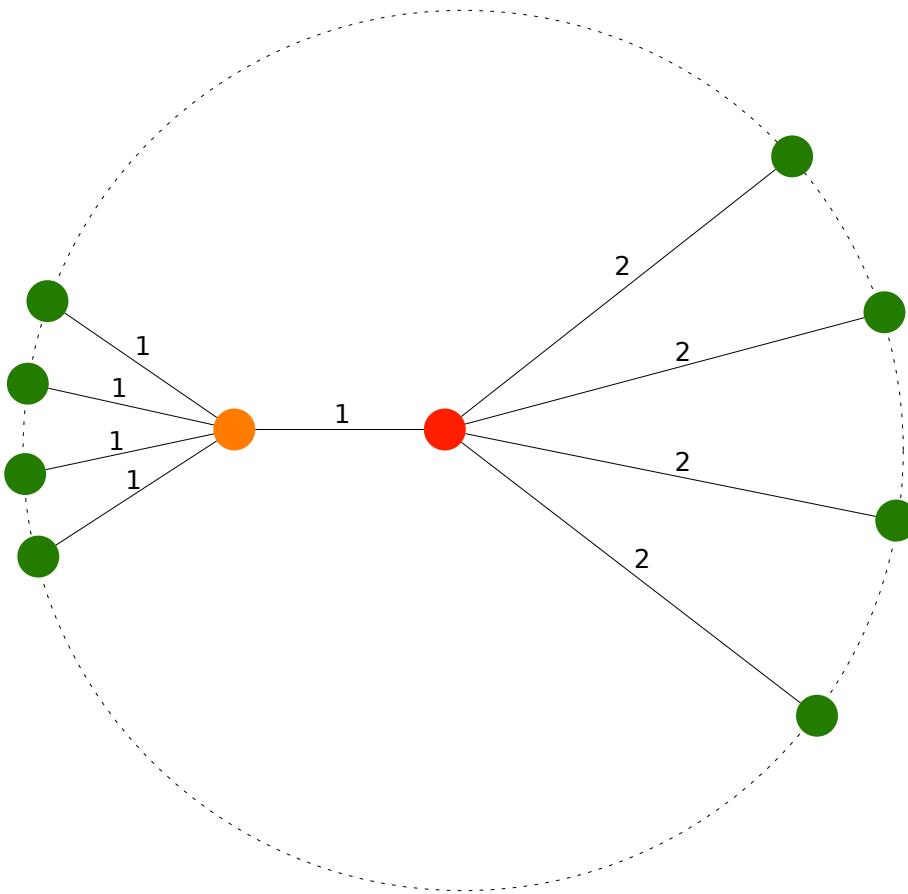


FIGURE 2.3 – Graphe valué dans lequel nous pouvons différencier le nœud rouge de celui orange avec une formule adaptée.

2.2.2 Les approximations

Un autre défaut, que nous avons déjà abordé, est le temps de calcul. Même s'il existe un algorithme résolvant le problème en $O(nm)$, lorsque nous traitons des graphes de plusieurs centaines de milliers ou de millions de nœuds, la complexité est encore bien trop grande. Ainsi, dans l'objectif de réduire le coût de calcul, Brandes a traité ce problème dans [9] en reprenant une idée issue de [19].

L'idée directrice est d'utiliser un sous ensemble de nœuds "pivots". Nous résolvons ensuite les problèmes de plus courts chemins partant de ces sommets vers tous les autres, et grâce à une formule de probabilité, il devient possible d'estimer la valeur réelle de la centralité de chaque nœud.

Brandes montre qu'il est préférable de sélectionner les pivots de façon aléatoire, et que leur nombre dépend de la taille du graphe.

Geisberger *et al.* proposent une extension de ce modèle dans [17]. Effectivement, la version de Brandes tend à favoriser les nœuds proches des pivots, ce qui biaise le

résultat final. Ils ont donc décidé de retirer ce biais à l'aide d'une méthode qui ajuste ces résultats.

Alahakoon *et al.* proposent également un modèle estimant la centralité dans [20]. Les auteurs définissent un algorithme dans lequel nous sélectionnons des nœuds pivots à partir desquels nous faisons transiter une information (ou une entité) selon une marche aléatoire, le chemin parcouru ne devant pas dépasser une certaine longueur comprise entre 1 et un paramètre K . Les auteurs de ce papier appuient une partie de leurs travaux sur ceux de Brandes. La principale contribution consiste à se servir d'une marche aléatoire et de limiter la longueur des chemins à une taille maximale.

Toutefois, il est nécessaire de noter que sa méthode s'applique très bien sur des réseaux sociaux mais il n'est pas garanti d'obtenir des résultats optimaux dans le cas général. Effectivement, les réseaux sociaux ont cette particularité : la contribution apportée à la centralité des nœuds par les chemins de grande taille est moins importante que la contribution apportée par les chemins de petite taille [21, 20]. Il est donc logique d'obtenir des résultats intéressants pour de tels réseaux en bornant la taille des chemins. Mais si nous utilisons un réseau viaire, les nœuds à forte centralité peuvent tout à fait faire partie d'axe de grande taille (comme les axes transversaux ou les points d'articulation).

L'utilisation d'une marche aléatoire pour calculer la centralité a quant à elle été proposée par Newman dans [10]. Toutefois, avant de présenter directement et plus en détail ce concept, nous allons expliquer les méthodes d'analyses que nous avons employées pour comparer les mesures développées et l'approche classique.

Chapitre 3

Les méthodes d'analyses

3.1 Les méthodes de comparaison

Nos hypothèses initiales se basaient sur nos impressions de similarités des colorations des graphes. Nous ne pouvions donc nous contenter de cette approche pour juger des résultats.

Dans un premier temps nous nous sommes intéressés aux coefficients de corrélation de Pearson et de Spearman. Il s'agit de mesures utilisées en statistiques. Elles fournissent une valeur entre -1 et 1 indiquant la dépendance qui existe ou non entre deux variables statistiques. Plus la valeur absolue de la mesure est proche de 1, plus les deux variables sont corrélées. Au contraire, plus la valeur est proche de zéro, moins les variables sont corrélées. La mesure de Pearson permet de détecter une corrélation linéaire, tandis que celle de Spearman met en valeur une corrélation non linéaire.

Si nous avons deux échantillons associés aux variables X et Y , alors le coefficient de Pearson se calcule ainsi :

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

tel que : x_i (resp. y_i) est la $i^{\text{ème}}$ valeur de X (resp. Y) et \bar{x}_i (resp. \bar{y}_i) est la moyenne des valeurs de l'échantillon de X (resp. Y).

Le coefficient de Spearman se calcule de la même manière mais plutôt que de considérer directement la valeur de chaque x_i et y_i , nous considérons le rang que possède ces variables dans une liste ordonnée.

Nous avons également décidé de compléter notre panel de méthodes avec une approche plus graphique. Nous construisons la liste ordonnée des nœuds selon l'une des deux mesures que nous souhaitons comparer. Puis nous traçons sur un graphique la courbe représentant ce classement avec en abscisses l'identifiant des nœuds et en ordonnées la mesure de chacun d'entre eux. Enfin, nous positionnons sur le même

graphique et pour chaque nœud, le score obtenu par la deuxième mesure. Si le deuxième tracé représente une courbe, alors nous pouvons affirmer que les deux mesures sont corrélées directement, et la finesse de cette courbe montrera le degré de corrélation.

Cette deuxième méthode permet de confirmer la corrélation indiquée par les coefficients, car parfois nous obtenons des coefficients à 0.6 ou 0.7 alors que la corrélation n'est pas du tout significative. L'interprétation des coefficients est donc délicate dans certains cas. Les graphiques permettent ainsi cette interprétation.

3.2 Les graphes utilisés

Afin de déterminer si nos modèles fournissaient des résultats corrects dans le cas général, nous les avons confrontés à plusieurs types de graphes différents. Nous avons donc utilisé d'un côté les générateurs de graphes présents dans la librairie GraphStream [22] et de l'autre des représentations des réseaux viaires de la ville du Havre et de celle de Rouen.

Le générateur de "small-world" Ce premier type de graphe est basé sur le modèle de Watts et Strogatz présent dans [23]. Il s'agit de créer une structure en anneau dont les noeuds ont k voisin(s). Puis nous modifions aléatoirement certaines arêtes afin qu'une de leurs extrémités soit connectée à un autre nœud choisi aléatoirement selon une probabilité β (voir figure 3.1).

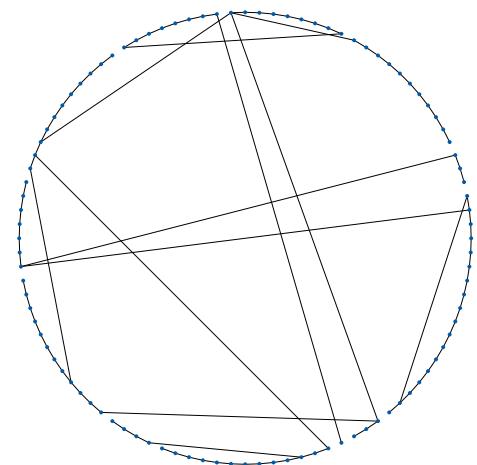


FIGURE 3.1 – Représentation d'un small-world avec 100 nœuds, $k = 2$ voisins et $\beta = 0.1$.

Le générateur de grille Nous avons également utilisé un générateur de grille car il s'agit d'une topologie très régulière et assez simple mais qui peut poser certains problèmes. Effectivement, entre deux nœuds quelconques d'une grille, il existe plusieurs plus courts chemins. Et la centralité classique prend en compte chacun de ces chemins. Pourtant, nous verrons par la suite que nos méthodes ne définissent qu'un seul chemin

entre les paires de nœuds. Il fallait donc vérifier si nous pouvions tout de même obtenir des résultats cohérents dans ce cas de figure.

Le générateur d'attachement préférentiel Celui-ci s'appuie sur le modèle de Barabàsi et d'Albert proposé dans [24]. À chaque étape nous générerons un nœud qui sera relié par une ou plusieurs arête(s) à un ou plusieurs nœuds choisi(s) selon une sélection aléatoire biaisée par son degré (voir figure 3.2).

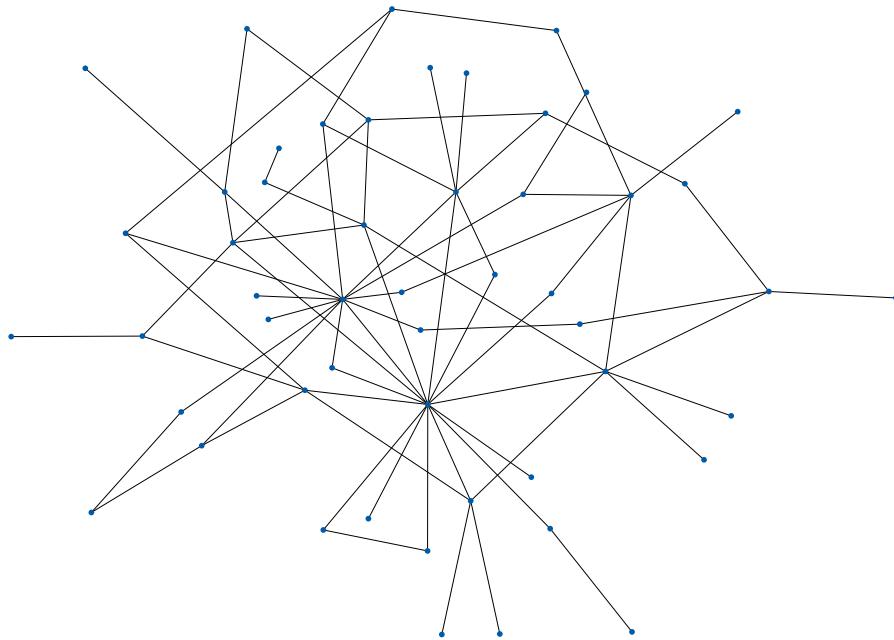


FIGURE 3.2 – Représentation d'un attachement préférentiel avec 50 nœuds, et 2 arêtes au maximum créées par tour.

Le générateur de graphe de Dorogovtsev-Mendes Il s'agit de l'algorithme proposé par Dorogovtsev et Mendes dans [25] qui crée des graphes planaires dont la distribution des degrés suit une loi de puissance. À l'initialisation, nous créons trois nœuds et trois arêtes disposés en triangle. Puis à chaque tour, nous ajoutons un nouveau nœud que nous connectons aux extrémités d'une arête choisie aléatoirement (voir figure 3.3).

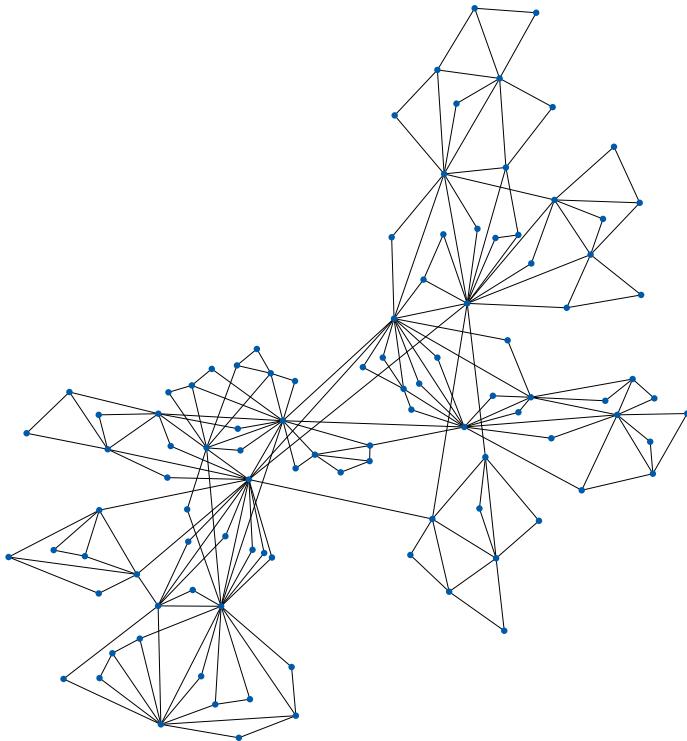
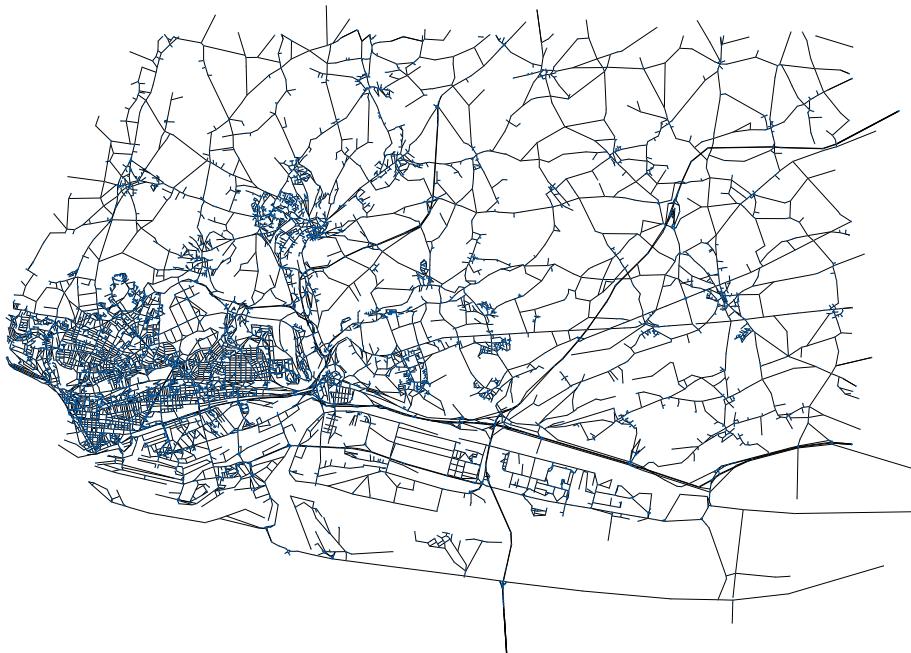
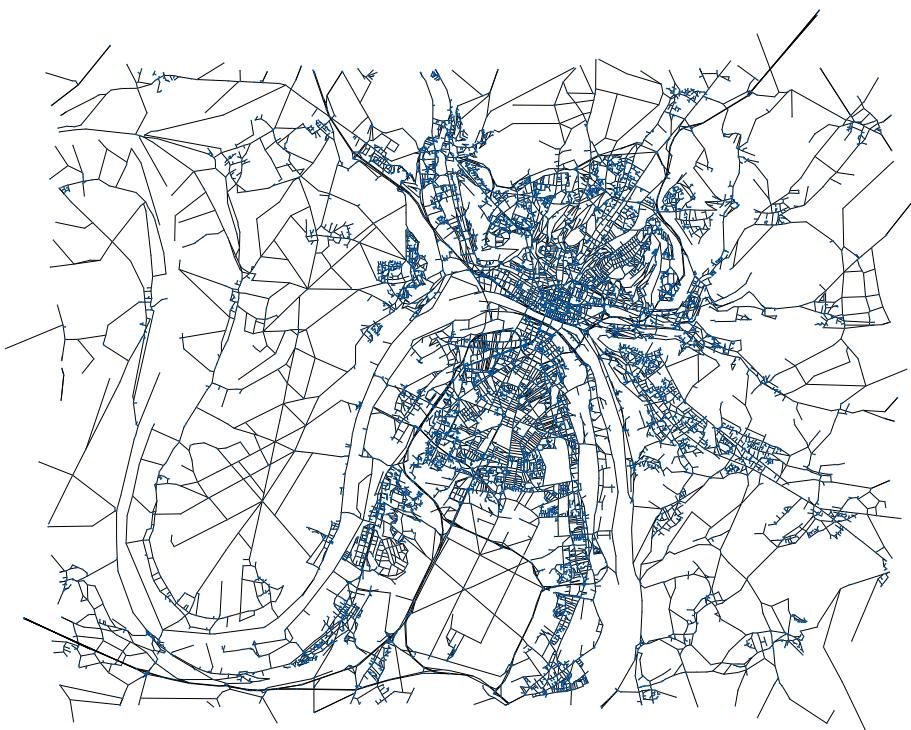


FIGURE 3.3 – Représentation d'un graphe généré par l'algorithme de Dorogovtsev et Mendes possèdant 100 nœuds.

Les réseaux viaires Enfin, nous avons également utilisé deux réseaux viaires : celui de l'agglomération du Havre et celui de l'agglomération de Rouen (voir figure 3.4). Il s'agit de graphe orienté et valué (soit par la longueur euclidienne des arcs, soit par leur temps de trajet pour les traverser). L'utilisation de deux réseaux s'est montrée intéressante car leurs structures sont très différentes l'une de l'autre. Alors que le Havre est très excentré puisqu'il est en bordure de mer, Rouen est plutôt au centre du graphe et possède un périphérique et des ponts traversant un fleuve (la Seine).



(a) Le réseau viaire du Havre - 11736 noeuds.



(b) Le réseau viaire de Rouen - 17442 noeuds.

FIGURE 3.4 – Réseau viaire de deux villes de la région.

Chapitre 4

Les marches aléatoires

4.1 Les origines

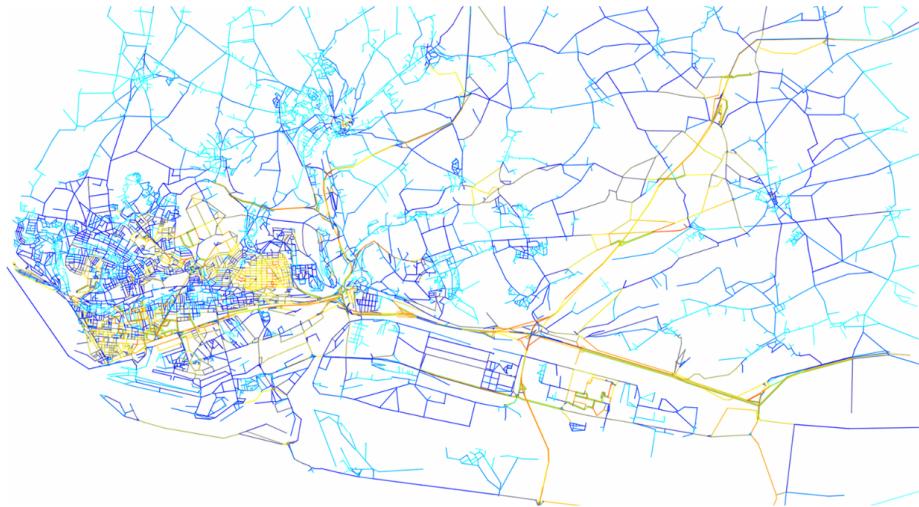
Nous avons pu voir précédemment que l'information ou l'entité qui se déplace dans un graphe n'emprunte pas nécessairement un plus court chemin. Freeman, conscient de cette limite avait alors proposé un modèle basé sur les flots [11]. Mais encore une fois, nous percevons une limite car, pour emprunter le flot maximal, l'entité doit avoir conscience du "bon" chemin à prendre.

Mais qu'en est-il lorsqu'elle ne connaît pas ce chemin ? Qu'en est-il lorsqu'elle possède une information incomplète, ou imparfaite du graphe ? Il s'agit là de questions primordiales puisque dans un système distribué et/ou dynamique, il n'est pas toujours possible de recalculer tous les plus courts chemins, soit par manque d'informations, soit par manque de temps.

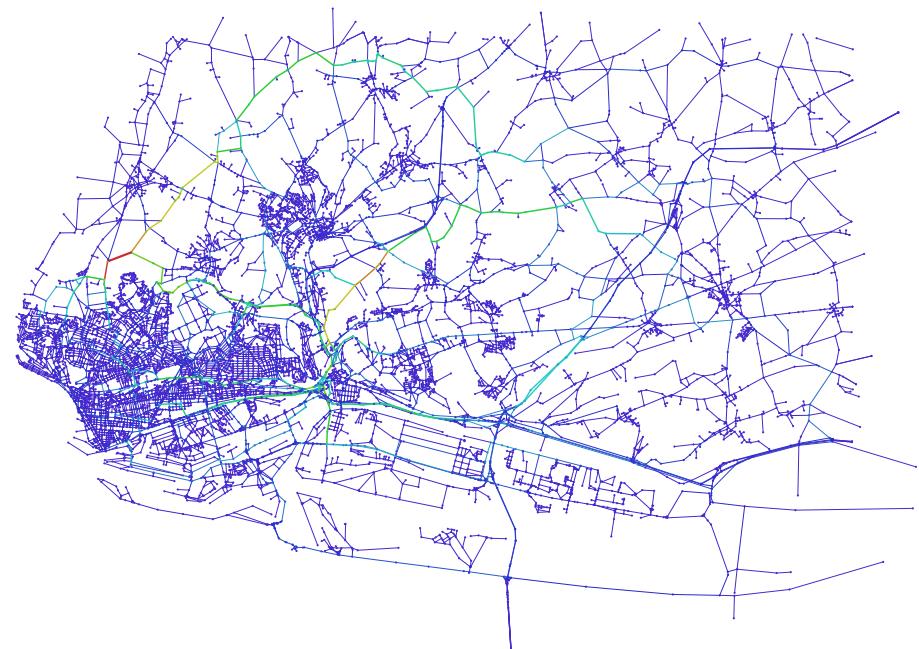
Dans l'idéal, il faudrait que nous soyons en mesure de simuler avec exactitude le comportement de cette entité dans le cas distribué et dynamique. Mais bien sûr plusieurs problèmes surviennent : tout d'abord, il est parfois impossible (ou seulement trop difficile) de modéliser un tel comportement tant la complexité du système peut être importante (par exemple le modèle de déplacement d'un véhicule dans une ville) ; d'autre part, nous ne pouvons créer un modèle général capable de prédire le comportement adéquate pour chaque type de réseau (le mode de transfert d'information dans un réseau social, type hiérarchie dans une entreprise, est différent dans un réseau peer-to-peer).

Afin de surmonter ces difficultés, Newman propose un modèle qui repose sur les mêmes principes que la centralité classique à la différence que les entités se déplacent selon une marche aléatoire [10] : il s'agit de la "random-walk betweenness" (que l'on peut traduire par "marche aléatoire d'intermédiairité").

D'autre part, Michel Nabaa avait mis en valeur un résultat intéressant dans sa thèse [1] (voir figure 4.1). Il avait implémenté une marche aléatoire classique sur le réseau viaire du Havre et avait également calculé sa centralité d'intermédiairité. Les "marcheurs" faisaient incrémenter un compteur sur chaque nœud traversé. Ainsi, par coloration du



(a) Résultat de la marche aléatoire sur le réseau viaire du Havre issu de la thèse de Michel Nabaa
- du non emprunté en bleu clair au congestionné en rouge.



(b) Résultat de la centralité d'intermédiairité sur le réseau viaire du Havre - de la centralité faible en bleu à la centralité forte en rouge.

FIGURE 4.1 – Comparaison visuelle des résultats d'une marche aléatoire et de la centralité d'intermédiairité. Corrélation sur les autoroutes, la Breque, l'entrée principale du Havre et la rocade.

réseau, les deux méthodes semblaient mettre en valeur les mêmes zones : notamment les autoroutes et les entrées principales du Havre.

Ainsi, dans cette partie, nous avons souhaité reproduire ce résultat et mesurer avec plus de précision la corrélation entre les deux méthodes.

4.2 Le modèle

Ce premier modèle s'appuie sur l'implémentation d'une marche aléatoire présente dans la librairie GraphStream. Nous allons donc exposer les caractéristiques de cette marche.

Nous utilisons une classe maître "RandomWalk" qui indique à chaque entité (et à chaque étape) de passer d'un nœud à l'autre. Cette classe se charge également de faire évaporer une partie du score obtenu sur chaque élément du graphe (les nœuds et les arêtes).

Ce mécanisme nous permet d'obtenir une certaine stabilité dans les scores au bout d'un certain nombre d'itérations. Ainsi, nous avons pu mettre en place un test de stabilité qui permet de stopper automatiquement le processus lorsque la stabilité apparaît. Notons que les taux d'évaporation sont définis par l'utilisateur. Au cours de nos tests, chaque élément perdait 3% de son score par tour.

Ce test compare la valeur moyenne des éléments du graphe avec les valeurs moyennes des x dernières itérations. Si la moyenne actuelle connaît un écart de plus de $y\%$ avec n'importe quelle autre moyenne alors nous considérons que la simulation n'est pas stable.

Les paramètres " x " et " y " sont quant à eux définis par l'utilisateur, mais dans le cas de nos tests nous avons choisi arbitrairement $x = 100$ et $y = 10$.

L'entité en elle-même s'inspire des mécanismes tabous. Elle ne peut retourner sur un nœud qu'elle a déjà visité. Toutefois, pour éviter tout blocage, sa mémoire est limitée par un paramètre et si l'entité ne pouvait vraiment pas se déplacer sur un voisin direct, elle est capable de se "téléporter" sur un autre nœud du graphe, choisi complètement au hasard.

4.3 Les résultats

Nous observons que le modèle ne semble fournir aucun résultat correct quelque soit le type de graphe. Le tableau 4.1 donne les coefficients de corrélation de Pearson et de Spearman pour ces différents graphes.

Type de graphe	Coef. de Pearson	Coef. de Spearman
Dorogovtsev et Mendes (1000 nœuds)	0.2405	0.7412
Grille (10000 nœuds)	0.1023	0.0073
Attachement préférentiel (1000 nœuds et 1 arête maximum par tour)	0.2396	0.6001
Attachement préférentiel (1000 nœuds et 2 arêtes maximum par tour)	0.3016	0.7217
Small-world (1000 nœuds, $k = 2$ et $\beta = 0.5$)	0.2783	0.6392
Small-world (1000 nœuds, $k = 4$ et $\beta = 0.01$)	-0.0559	-0.0158
Small-world (10000 nœuds, $k = 6$ et $\beta = 0.001$)	0.0008	0.0214
Le Havre (11736 nœuds)	0.3043	0.5540

TABLE 4.1 – Coefficient de corrélation entre la centralité d’intermédiarité et notre modèle de marche aléatoire pour les différents graphes.

Nous constatons que le coefficient de Spearman donne des résultats supérieurs à 0.6 dans certains cas. Pourtant, nous pouvons aisément affirmer grâce aux graphiques qui suivent en figure 4.2 que la corrélation n’est pas du tout significative dans ces cas là.

Ainsi, ce premier modèle ne permet pas de nous approcher de la centralité d’intermédiarité. Nous pouvons néanmoins imaginer que si les entités se déplaçaient depuis une source et vers une destination, le constat pourrait changer. Un mécanisme plus proche de la définition de la centralité serait potentiellement mieux corrélé. Nous allons donc développer dans la partie qui suit, un autre modèle dont les entités ont la particularité de naviguer entre deux nœuds.

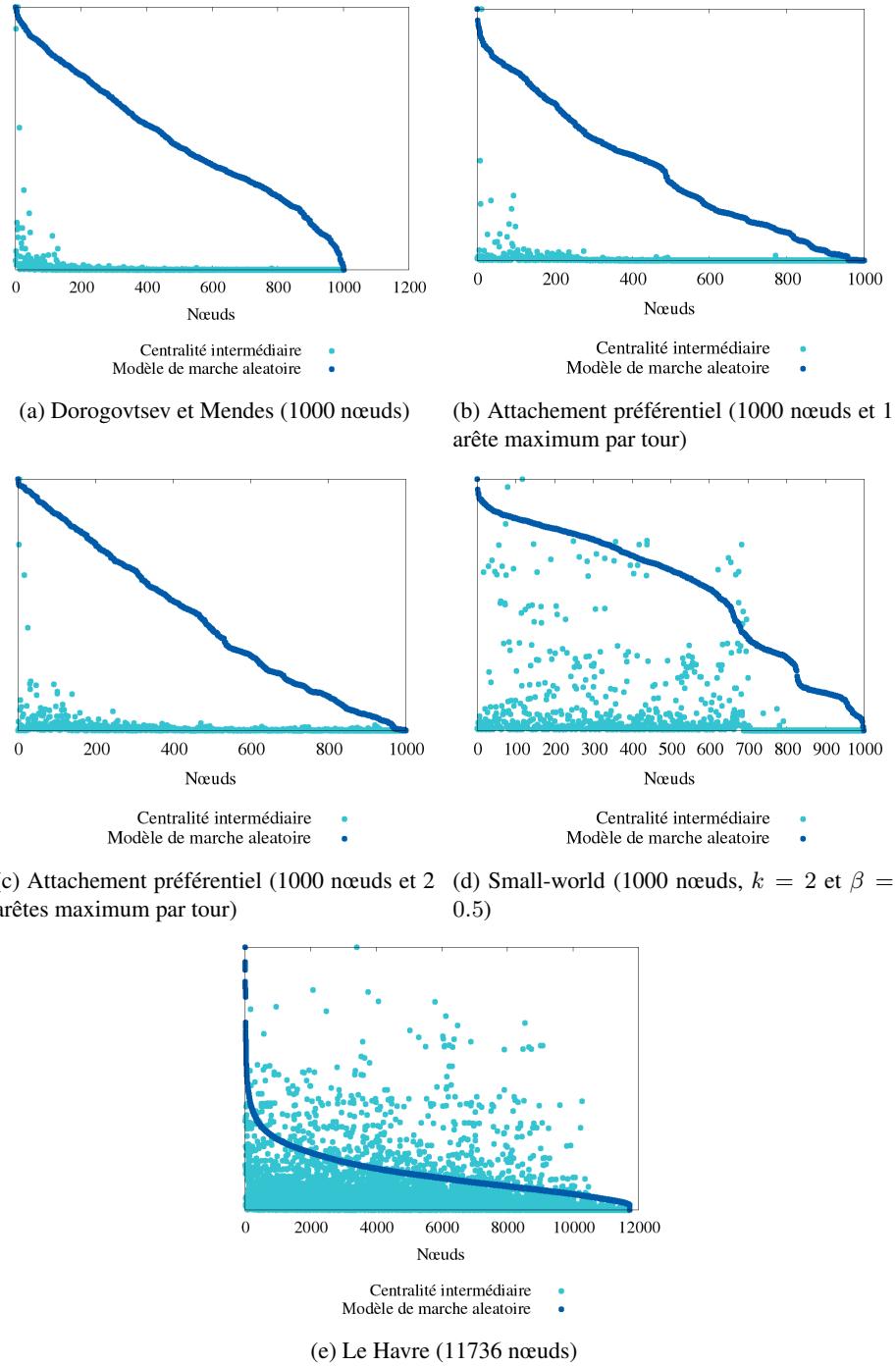


FIGURE 4.2 – Graphiques mettant en avant la non existence d'une corrélation de la centralité d'intermédiarité et d'une marche aléatoire. En abscisses, il s'agit de la liste ordonnée des nœuds selon la mesure de notre modèle ; en ordonnées, les points sont situés en fonction du score obtenu par l'une ou l'autre des mesures.

Chapitre 5

La centralité d'intermédiairité avec pivots

5.1 La base du modèle

Afin de pallier aux problèmes soulevés dans la partie précédente, nous allons développer dans celle-ci un autre modèle plus proche de la définition initiale.

Nous nous sommes inspirés de la démarche de Brandes dans [9], où l'auteur utilise un ensemble de nœuds "pivots" à partir desquels, et uniquement à partir de ceux-ci, il résout des problèmes de plus courts chemins vers tous les autres nœuds.

Brandes montre qu'avec un ensemble de pivots suffisamment grand, nous sommes en mesure d'obtenir des résultats fortement corrélés avec la centralité classique. Cela sous-entend donc l'hypothèse que des chemins contribuent plus efficacement que d'autres. Nous supposons alors que ces chemins sont constitués de sections ayant une forte centralité. Ainsi ces sections seraient mises en valeur plus ou moins facilement avant même de finir le calcul complet.

Partant de ce constat, il est possible d'émettre l'hypothèse qu'il n'est pas nécessaire de préférer une source en particulier. Effectivement, il est peu probable que toutes les paires calculées depuis un même nœud source soient pertinentes. Nous supposons donc qu'il n'existe qu'un nombre limité de chemin dont la contribution est vraiment plus importante que les autres et l'existence de ces chemins est potentiellement indépendante du nœud source.

A priori, il peut donc être préférable de considérer des paires de nœuds dont les extrémités seraient choisies de manière plus uniforme sur l'ensemble des nœuds du graphe.

Malheureusement, le calcul des plus courts chemins entre un nœud source et un nœud destination nécessite bien souvent de calculer les plus courts chemins partant de la source vers tous les autres nœuds afin de ne pas omettre un chemin moins intuitif. C'est par exemple ce que fait l'algorithme de Dijkstra. Il devient donc primordial de

trouver une manière de calculer les plus courts chemins sans parcourir le graphe tout entier.

Nous avons donc choisi d'utiliser des entités dont le comportement permet le calcul de ces chemins. Ces mêmes entités peuvent directement incrémenter le compteur de la centralité. Notre modèle se constitue donc d'un processus principal qui gère toutes les entités et leur fournit les paires de nœuds qu'elles doivent traiter. Voici l'algorithme de ce processus central :

```

1 Entrée:
2   - G : un graphe
3   - nbPaire : le nombre de paire de nœuds
4   - nbEntité : le nombre d'entité
5 Données:
6   - listeEntité() : la liste contenant l'ensemble des entités
7 Fonctions disponibles :
8   - initialiserCentralité() : initialise à zéro les compteurs correspondant à la centralité sur les nœuds
9     et sur les arêtes .
10  - nouvelleEntité() : crée et initialise une nouvelle entité
11  - étape() : s'applique sur une entité et lui indique de calculer une étape de son calcul interne
12 Sortie :
13  - G : le graphe dont les nœuds et les arêtes possèdent une pondération selon leur centralité
14 Début
15 //Initialisation
16 initialiserCentralité()
17 Pour i allant de 0 à nbEntité faire
18   listeEntité[i] ← nouvelleEntité()
19 Fin pour
20
21 //Calcul
22 Tant que nbPaire > 0 faire
23   Pour chaque e ∈ listeEntité faire
24     e.étape()
25   Fin pour chaque
26   Fin tant que
27 Fin

```

De leur côté, les entités sont capables de demander au processus maître quelle paire de nœuds doit être traitée. Ce moment mis à part, les entités sont complètement indépendantes. Elles tracent ainsi des chemins sans cycle puis incrémentent un compteur sur chaque élément composant leur chemin.

Nous allons donc expliquer dans la suite de ce chapitre trois comportements différents des entités susceptibles de résoudre notre problème.

5.2 Le modèle avec des entités effectuant une marche aléatoire

5.2.1 Le fonctionnement des entités

En développant ce premier type d'entité, nous souhaitions réunir le modèle de Brandes utilisant des pivots et le modèle de Newman utilisant des marches aléatoires.

Nous avons donc créé une entité basique qui commence par récupérer une paire de nœuds. Partant de la source, l'entité doit, à chaque étape du calcul, choisir uniformément l'un des voisins du nœud courant. Ce nœud est ensuite ajouté au chemin auquel nous supprimons les éventuels cycles. Le nœud courant devient alors le nœud choisi et nous passons à l'étape suivante.

Lorsque l'entité arrive à sa destination, elle refait le chemin à l'envers en incrémentant le compteur sur chaque nœud.

D'autre part, nous avons travaillé sur des réseaux réels : le réseau routier de la ville du Havre et celui de Rouen pour corroborer nos résultats. Nous pouvions donc nous servir des données contenues sur le graphe pour réaliser des heuristiques. Par conséquent, nous avons souhaité vérifier si des entités effectuant une marche aléatoire biaisée par la distance euclidienne pouvait fournir des résultats différents.

5.2.2 Les résultats

Nous allons commencer par présenter les coefficients de corrélation obtenus avec l'entité basique.

Type de graphe	Coef. de Pearson		Coef. de Spearman	
	100 paires	1000 paires	100 paires	1000 paires
Dorogovtsev et Mendes (1000 nœuds)	0.9662	0.9668	0.7436	0.7531
Grille (10000 nœuds)	0.9228	0.9368	0.9600	0.9807
Attachement préférentiel (1000 nœuds et 1 arête maximum par tour)	0.9997	0.9999	0.7517	0.7043
Attachement préférentiel (1000 nœuds et 2 arêtes maximum par tour)	0.9136	0.9133	0.8762	0.8743
Small-world (1000 nœuds, $k = 2$ et $\beta = 0.5$)	0.9702	0.9737	0.9720	0.9692
Small-world (1000 nœuds, $k = 4$ et $\beta = 0.01$)	0.7869	0.7964	0.5578	0.5748
Small-world (10000 nœuds, $k = 6$ et $\beta = 0.001$)	0.6255	0.6359	0.4985	0.5206
Le Havre (11736 nœuds)	0.6475	0.6481	0.8628	0.8653
Rouen (17442 nœuds)	0.5773	0.5805	0.8175	0.8219

TABLE 5.1 – Coefficient de corrélation entre la centralité d'intermédiairité et notre modèle avec pivots utilisant des pivots et des entités effectuant une marche aléatoire.

Le tableau 5.1 montre clairement une corrélation plus prononcée avec ce modèle utilisant les pivots. Les coefficients de Pearson sont souvent entre 0.9 et 1.0 pour la plupart des graphes. Les coefficients de Spearman sont néanmoins plus pessimistes puisqu'ils se situent plus souvent entre 0.7 et 0.9. Nous constatons sur les graphiques de la figure 5.1 que cette corrélation est significative pour les graphes dont la corrélation est forte.

Il faut toutefois modérer notre propos car les résultats sur les deux derniers small-world et les réseaux viaires du Havre et de Rouen sont plutôt mauvais par rapport aux autres comme le témoigne les graphiques relatifs à ces graphes (voir figure 5.2).

Nous allons maintenant voir si l'utilisation d'une entité biaisée permet d'obtenir de meilleurs résultats sur un réseau viaire.

Type de graphe	Coef. de Pearson		Coef. de Spearman	
	100 paires	1000 paires	100 paires	1000 paires
Le Havre (11736 noeuds)	0.6478	0.6517	0.8650	0.8659
Rouen (17442 noeuds)	0.5741	0.5791	0.8170	0.8219

TABLE 5.2 – Coefficient de corrélation entre la centralité d'intermédiarité et notre modèle avec pivots et des entités effectuant une marche aléatoire biaisée.

Les coefficients de corrélation du tableau 5.2 se situent autour de 0.6 pour celui de Pearson, et de 0.8 pour celui de Spearman. Il s'agit grossièrement des mêmes résultats obtenus avec les entités effectuant la marche aléatoire classique. La figure 5.3 est également une représentation similaire de la marche aléatoire classique.

Nous ne pouvons donc pas affirmer qu'une marche biaisée apporte un intérêt particulier. Bien sûr, le biais introduit n'est peut-être pas suffisant.

Pour conclure cette partie, cette méthode montre donc un certain intérêt puisqu'elle semble fournir des résultats corrects sur certains graphes. Toutefois, comme nous l'avons mis en évidence, ce n'est pas toujours le cas. Il est donc nécessaire d'approfondir le modèle. Cependant, une question se pose : est-ce que la non corrélation provient de la marche aléatoire utilisée par les entités, ou est-ce que le problème provient plutôt du mécanisme des paires qui ne serait pas pertinent ?

Afin de répondre à cette question, nous devons proposer une entité dont le modèle de déplacement suit davantage les plus courts chemins. C'est justement le point que nous allons aborder dans la partie qui suit.

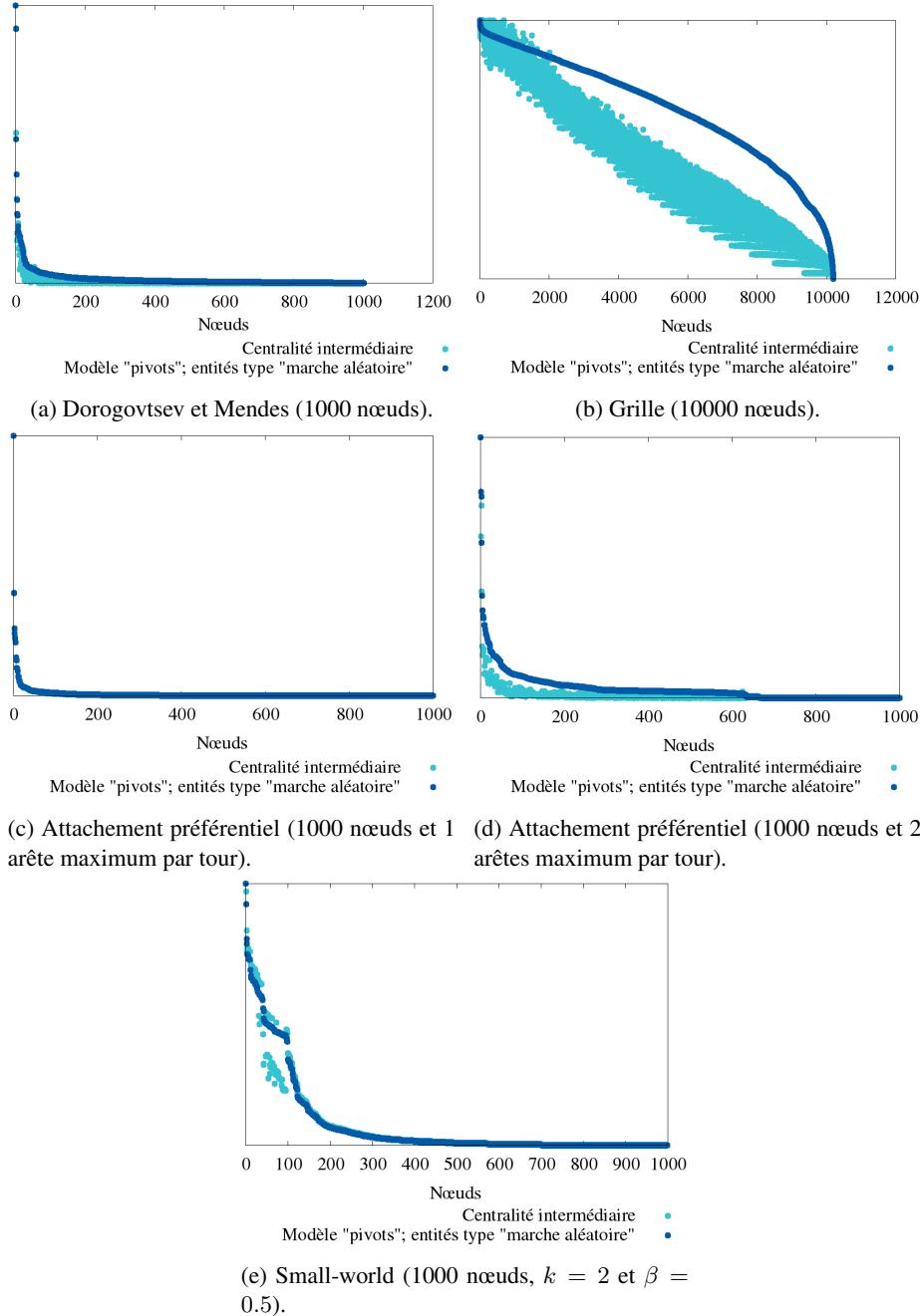


FIGURE 5.1 – Graphiques mettant en avant l’existence sur certains graphes d’une corrélation entre la centralité d’intermédiarité et notre modèle utilisant des pivots et des entités effectuant une marche aléatoire. En abscisses, il s’agit de la liste ordonnée des nœuds selon la mesure de notre modèle ; en ordonnées, les points sont situés en fonction du score obtenu par l’une ou l’autre des mesures.

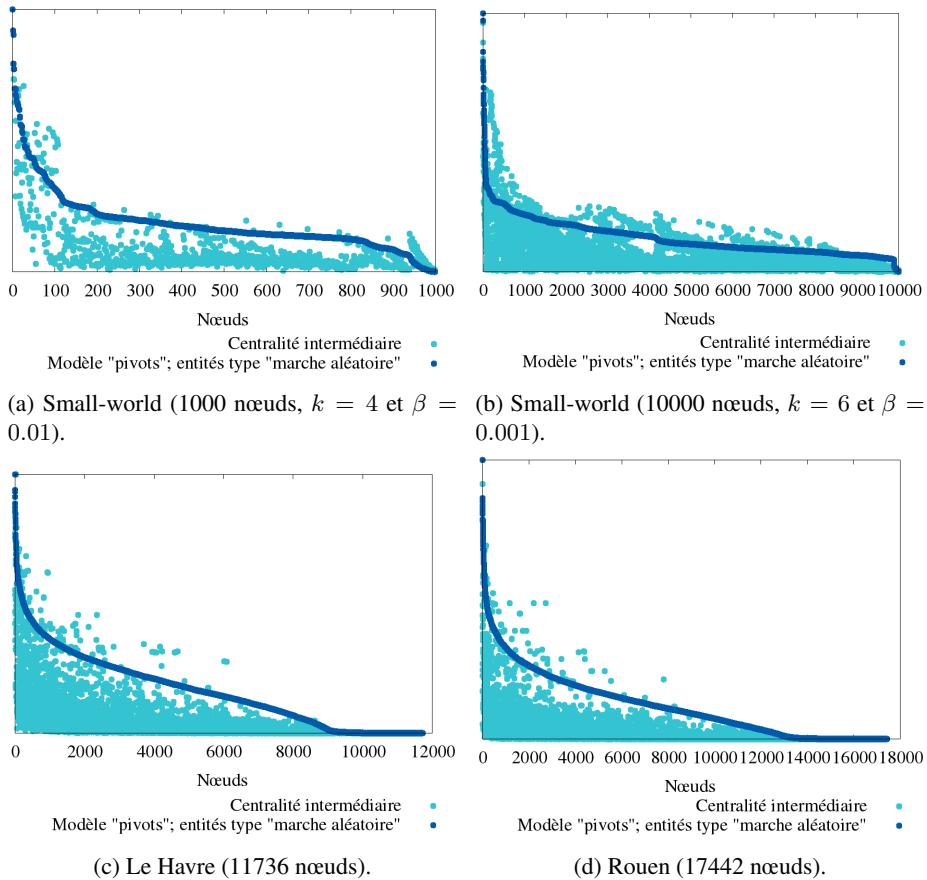


FIGURE 5.2 – Graphiques mettant en avant la non-existence sur certains graphes d'une corrélation suffisante entre la centralité d'intermédiarité et notre modèle utilisant des pivots et des entités effectuant une marche aléatoire. En abscisses, il s'agit de la liste ordonnée des nœuds selon la mesure de notre modèle ; en ordonnées, les points sont situés en fonction du score obtenu par l'une ou l'autre des mesures.

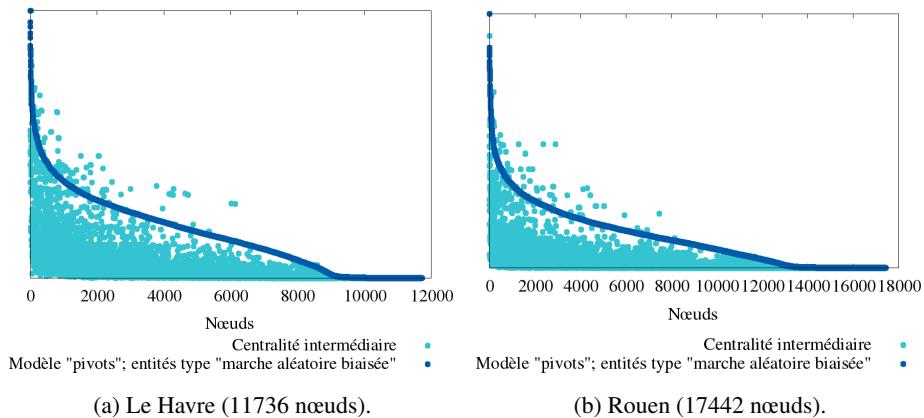


FIGURE 5.3 – Graphiques mettant en avant la non existence sur les réseaux viaires d'une corrélation suffisante entre la centralité d'intermédiarité et notre modèle utilisant des pivots et des entités effectuant une marche aléatoire biaisée. En abscisses, il s'agit de la liste ordonnée des noeuds selon la mesure de notre modèle ; en ordonnées, les points sont situés en fonction du score obtenu par l'une ou l'autre des mesures.

5.3 Le modèle avec des entités utilisant A^*

5.3.1 Le fonctionnement des entités

Puisque la première entité n'a pas fourni de résultats complètement concluants et notamment sur le Havre, nous avons décidé de rechercher des solutions plus proches des plus courts chemins. Tout naturellement nous nous sommes rapprochés de l'algorithme A^* .

Il s'agit d'un algorithme de recherche de chemin utilisant des heuristiques. Il fut présenté en premier dans [26] par Peter Hart, Nils Nilsson and Bertram Raphael en 1968.

Initialement, l'algorithme débute sur un nœud défini par l'utilisateur. À chaque itération, A^* estime un coût sur chaque voisin de son nœud courant à l'aide d'une heuristique. Il stocke ainsi les nœuds qu'il a rencontrés dans une liste d'attente prioritaire dont l'ordre est défini en fonction de l'heuristique. L'algorithme se déplace à la prochaine étape sur le premier nœud de cette file d'attente.

Il boucle ainsi jusqu'à ce que le nœud sélectionné soit la destination. L'algorithme est alors capable de reconstruire le chemin. De plus, si la file d'attente ne contient plus de nœud avant d'avoir rencontré la destination, cela signifie qu'il n'existe pas de chemin entre les deux nœuds.

Notre entité se contente donc de calculer le chemin à l'aide de cet algorithme et incrémenté le compteur de la centralité sur chaque nœud du chemin. Ci-dessous l'algorithme exécuté par l'entité au cours d'une étape.

```

1 Données:
2   — manager : l'objet associé au processus maître
3   — astar : un objet capable d'exécuter l' algorithme A*
4   — paire[] : liste des deux nœuds constituants la paire
5   — chemin[] : liste des nœuds et des arêtes constitutants le chemin
6 Fonctions disponibles :
7   — existeNouvellePaire() : appartient au processus maître; renvoie un booléen :
8     vrai s'il existe une nouvelle paire à calculer , faux dans le cas contraire .
9   — obtenirNouvellePaire() : renvoie la nouvelle paire traité par l'entité.
10  — calculer(nœudSource, nœudDest) : Calcul un chemin entre les nœuds en paramètres grâce à A*
11 Début
12  Si manager.existeNouvellePaire() alors
13    paire ← manager.obtenirNouvellePaire()
14    chemin ← astar.calculer(paire[1], paire[2])
15    Pour chaque e ∈ chemin faire
16      e.centralité ← e.centralité + 1
17    Fin pour chaque
18  Fin Si
19  Fin

```

Toutefois, cette entité ne peut en aucun cas être une solution définitive et acceptable pour le cas général. Nous souhaitions confirmer que le modèle avec les pivots pouvait fournir des résultats corrects ou si au contraire cette solution était inadaptée. Mais l'utilisation d'A* implique de posséder une heuristique admissible et monotone pour être utilisée puisque l'algorithme n'est capable de trouver le plus court chemin qu'à cette condition. Malheureusement, il est très rare de posséder une telle heuristique.

Sur un réseau routier nous pouvons nous servir d'une distance euclidienne comme heuristique si nous souhaitons un plus court chemin en terme de longueur euclidienne. Mais si nous souhaitons calculer les plus courts chemins en terme de temps de transport, alors il n'existe pas de telle heuristique. Et pour finir, sur un graphe quelconque, quelle heuristique pouvons-nous utiliser ? L'algorithme risque de perdre tout son intérêt.

5.3.2 Les résultats

Nous n'allons présenté ici que les résultats sur les réseaux viaires. Les graphes virtuels générés par GraphStream ne possèdent pas de sémantique particulière. Il est donc délicat de proposer une heuristique efficace.

Remarque La centralité d'intermédiairité dont nous nous servons dans cette partie prend en compte la pondération des arêtes (plus précisément leur longueur). Auparavant, nous n'utilisions pas la pondération afin de pouvoir comparer avec les graphes générés par GraphStream (qui eux, ne sont pas pondérés). De plus, sur le Havre et Rouen, les résultats étaient soit équivalents, soit moins bons.

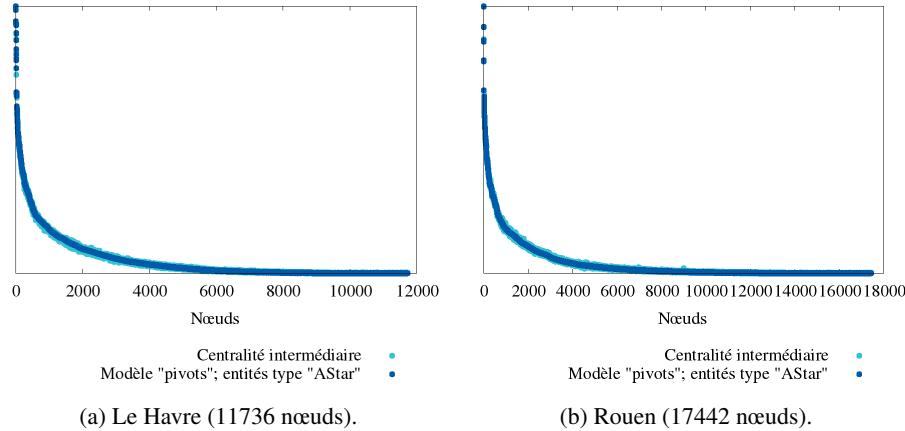


FIGURE 5.4 – Graphiques mettant en avant l’existence sur les réseaux viaires d’une corrélation entre la centralité d’intermédiarité et notre modèle utilisant des pivots et des entités se servant d’A*. En abscisses, il s’agit de la liste ordonnée des nœuds selon la mesure de notre modèle ; en ordonnées, les points sont situés en fonction du score obtenu par l’une ou l’autre des mesures.

Type de graphe	Coef. de Pearson			
	100 paires	1000 paires	10000 paires	20000 paires
Le Havre (11736 nœuds)	0.8482	0.9817	0.9978	0.9993
Rouen (17442 nœuds)	0.8686	0.9834	0.9986	0.9993

TABLE 5.3 – Coefficient de corrélation de Pearson entre la centralité d’intermédiarité et notre modèle avec pivots et des entités se servant d’A*

Type de graphe	Coef. de Spearman			
	100 paires	1000 paires	10000 paires	20000 paires
Le Havre (11736 nœuds)	0.5931	0.9100	0.9897	0.9951
Rouen (17442 nœuds)	0.5267	0.8678	0.9814	0.9912

TABLE 5.4 – Coefficient de corrélation de Spearman entre la centralité d’intermédiarité et notre modèle avec pivots et des entités se servant d’A*

Ainsi, nous constatons que les résultats de notre modèle utilisant A^* sont très corrélés avec la centralité d'intermédiarité pour ces deux réseaux viaires. Les courbes de la figures 5.4 se superposent quasiment, que ce soit pour le Havre ou pour Rouen. Les coefficients de corrélation des tableaux 5.3 et 5.4 indiquent également une bonne corrélation (en tout cas lorsqu'un nombre de paire suffisant est utilisé).

L'utilisation de paires de nœuds choisies au hasard sur l'ensemble des paires possibles est a priori un modèle viable. Toutefois, ce constat devra être étendu à d'autres types de graphes pour donner une affirmation définitive. De plus, comme nous l'avons exprimé dans la partie précédente, A^* nécessite une heuristique pour fonctionner. Or il n'est absolument pas garanti que nous puissions en développer une. Par conséquent, il reste toujours nécessaire de trouver une entité capable de satisfaire notre exigence de plus court chemin.

C'est donc pour cette raison que nous allons étudier un nouveau type d'entité dans la section suivante.

5.4 Le modèle avec des entités "fourmis"

5.4.1 Le fonctionnement des entités

Enfin, la dernière entité que nous développons actuellement se base sur les principes des colonies de fourmis. Ce concept a initialement été proposé par Marco Dorigo *et al.* dans [27]. Il s'agit d'une méthode d'optimisation, ou méta-heuristique, inspiré du vivant et plus précisément du comportement des fourmis.

Dans la nature, ces dernières sont capables de communiquer entre elles par le biais de leur environnement et de phéromones : nous appelons ce principe la stigmergie. Lorsqu'un ensemble de fourmis part à la découverte de son environnement pour, par exemple, rechercher de la nourriture, chaque fourmi se déplacera selon une marche plus ou moins aléatoire mais biaisée par les quantités de phéromones déjà déposées. Les fourmis laissent elles-même une certaine quantité de cette phéromone sur son passage.

Le marquage de phéromone s'évaporant s'il n'est pas entretenue, nous pouvons rapidement observer l'apparition d'un chemin qu'empruntera naturellement l'ensemble de la colonie. Mais le point remarquable ici est qu'il s'agit de plus courts chemins. Ce constat a été observé par une équipe de chercheurs, dont faisait parti Deneubourg, et qui a rendu public ces résultats dans [28] et [29].

Dorigo a ensuite adapté ce comportement à des fourmis virtuelles afin d'obtenir une technique décentralisée de recherche de chemin. Ce principe est aujourd'hui réputé pour s'adapter facilement à la dynamique du système. De plus, bien qu'il soit conseillé d'utiliser une heuristique, il est possible d'exécuter l'algorithme sans en avoir. Dans ce cas là, l'émergence de la solution se base sur les phéromones uniquement. Ainsi, cette solution devrait se montrer adaptée au cas général. Nous l'avons donc appliquée comme suit.

L'algorithme se déroule en différentes phases :

- La construction des solutions.

- Le dépôt des phéromones.
- L'évaporation des phéromones.

Selon son modèle, au cours de la construction des solutions, les fourmis parcourent un graphe de nœud en nœud. L'entité fourmi doit calculer une probabilité de choisir l'un des arcs adjacents à son nœud courant qui dépend d'une heuristique et de la quantité de phéromones présentes sur l'arc. Nous avons éventuellement la possibilité de nous passer de l'heuristique mais l'émergence d'une solution peut être plus longue à atteindre.

Une fois l'arc sélectionné, la fourmi se déplace dessus pour atteindre le prochain nœud.

La formule de Dorigo donnant la probabilité de sélectionner un arc est la suivante :

$$p_i = \frac{\tau_i^\alpha \cdot \eta_i^\beta}{\sum_{j=1}^k \tau_j^\alpha \cdot \eta_j^\beta}$$

tel que :

- p_i est la probabilité de choisir l'arc i .
- τ_i est la quantité de phéromones présentes sur l'arc i .
- η_i est une heuristique indiquant l'attractivité de l'arc i .
- j est un indice qui parcourt l'ensemble des arcs à disposition de la fourmi.
- Enfin, α et β sont des paramètres indiquant l'importance relative de la phéromone et de l'heuristique.

Lors de la seconde phase, la fourmi dépose une quantité de phéromones dépendant de la formule suivante :

$$\tau_s = \tau_s + \frac{Q}{l(C)}$$

tel que :

- s est l'un des arcs présents sur le chemin C déterminé dans la première phase par la fourmi.
- Q est un paramètre.
- $l(C)$ est la fonction l retournant la longueur du chemin C .

Par facilité, la dernière phase est exécutée par le processus maître. Il s'agit donc d'un processus centralisé. Nous calculons l'évaporation ainsi :

$$\tau_s = (1 - \rho)\tau_s$$

tel que ρ est un pourcentage d'évaporation paramétré par l'utilisateur.

Pour être efficace, la construction d'une seule solution doit être menée par un nombre suffisant de fourmis et cette quantité est déterminée par expérimentation. La colonie suit donc une même phéromone pour résoudre le problème. Toutefois dans notre problème de centralité, il faut résoudre plusieurs chemins. Ainsi, nous avons adapté l'algorithme pour répondre à nos besoins.

Le processus maître définit dès l'initialisation l'ensemble des paires de nœuds dont nous attribuons à chacune un type différent de phéromone. L'étape au niveau d'une fourmi correspond à la création complète d'un chemin entre les extrémités d'une paire.

L'ensemble des entités traite, au cours d'une même étape, la même paire de nœuds : d'abord la création du chemin, et ensuite le dépôt de phéromones. Puis le processus maître effectue l'évaporation sur les nœuds et les arêtes et fournit à la colonie la nouvelle paire.

De plus, nous conservons au sein du processus maître le chemin le plus court calculé par une fourmi pour chaque paire. Ainsi, il est plus facile de parcourir les chemins, pour cette fois, de donner un score de centralité aux éléments du graphe.

5.4.2 Le résultat

Nous ne pourrons présenter dans cette partie qu'un seul résultat car le modèle n'est pas encore complètement implémenté. Le fonctionnement des fourmis ne résiste pas encore au zone "impasse". Or les deux réseaux viaires en contiennent plusieurs ce qui bloque le traitement.

De plus, nous n'avons travaillé que sur une simple grille mais pas sur d'autres types de graphes. Cela tient au fait que pour le moment la fourmi a besoin d'une heuristique alors que nous ne pouvons pas en proposer une pour les autres graphes générés. Sur la grille, l'utilisation d'une distance euclidienne ou d'une distance de Manhattan est tout à fait envisageable.

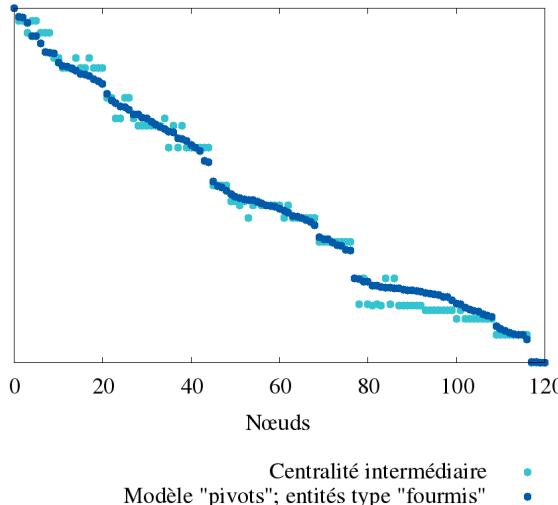


FIGURE 5.5 – Graphiques mettant en avant l'existence sur une grille d'une corrélation entre la centralité d'intermédiarité et notre modèle utilisant des pivots et des entités "fourmis". En abscisses, il s'agit de la liste ordonnée des nœuds selon la mesure de notre modèle ; en ordonnées, les points sont situés en fonction du score obtenu par l'une ou l'autre des mesures.

Nous constatons sur le graphique de la figure 5.5 que les deux mesures sont corrélées. Bien sûr il est trop tôt pour affirmer que le modèle est viable dans le cas général mais il s'agit d'un début prometteur.

Chapitre 6

Les perspectives

Maintenant que nous avons présenté notre modèle, nous allons expliquer les perspectives d'améliorations.

Le premier point nécessitant une remise en cause est certainement l'optimisation de notre entité fourmi en lui fournissant un comportement plus complexe mais plus performant. D'ailleurs, nous avons déjà commencé d'ajouter un comportement tabou pour qu'elle soit plus robuste aux zones "impasses" présentes dans certains graphes (notamment le Havre).

Le deuxième point que nous souhaiterions développer plus en détail est la méthode de sélection des paires. Nous avons préféré choisir les nœuds de manière complètement aléatoire car il semble difficile de considérer une stratégie en particulier sans la justifier. Ainsi, l'aléatoire est probablement la loi la plus impartiale qui soit. Toutefois, il serait très intéressant de proposer d'autres techniques de sélection.

Brandes et Pich avaient eux mêmes étudié la question dans [9] mais ils n'avaient pas nécessairement mis en valeur une méthode particulièrement plus efficace que la sélection aléatoire.

Nous sommes en tout cas en mesure d'affirmer que la centralité d'un nœud doit dépendre de la structure du graphe. Mais pouvons-nous supposer qu'un graphe est constitué de clusters, et que les arêtes qui les relient sont des zones critiques où la centralité est plus forte qu'ailleurs ? Si oui, il serait possible d'identifier ces clusters et de choisir des paires dont les extrémités font partie de communautés différentes. Cette méthode pourrait ainsi favoriser la mise en valeur de chemin contribuant plus efficacement au calcul de la centralité.

Le troisième point à mettre en perspective se rapporte au nombre de paires utilisées. Nous avons pu remarquer au cours des tests que le nombre de paires utilisées semblait améliorer la corrélation. Nous supposons que ce nombre doit varier en fonction de la taille du graphe traité, mais il serait intéressant et utile de découvrir le rapport précis qui lie la taille du graphe et notre paramètre.

Enfin, le dernier point serait de distribuer notre algorithme. Effectivement nous nous sommes efforcés de proposer un modèle auquel nous pourrions attribuer ce type de caractéristique. Mais l'implémentation peut prendre beaucoup de temps supplémentaire, donc nous avons préféré nous concentrer sur le problème principal.

Conclusion

Ce stage de recherche s'est établi dans la continuité d'un des axes d'études de la thèse de Michel Nabaa [1] portant sur la centralité d'intermédiarité dans un réseau viaire. Cette mesure avait été utilisée pour étudier la robustesse du réseau routier de la ville de Havre. Pourtant, en cas d'accident quelconque sur le réseau, le temps de calcul nécessaire rend impossible la mise à jour de l'information dans un délai acceptable.

Puisqu'il est nécessaire de réagir vite, le sujet a consisté à proposer un modèle robuste à la dynamique (et de préférence distribuable), capable de fournir une mesure de centralité d'intermédiairité.

En consultant les résultats de Nabaa sur le réseau du Havre, nous avons d'abord remarqué une éventuelle corrélation entre les marches aléatoires et la centralité d'intermédiarité. Après avoir implémenté cette marche sur différents graphes, nous avons analysé en détail la corrélation grâce aux coefficients de Pearson et de Spearman. Ces derniers ont clairement montré que les mesures ne sont pas corrélées.

Nous avons donc proposé une nouveau modèle inspiré des travaux de Brandes dans [9]. Un ensemble d'entité calcule des chemins entre les noeuds de différentes paires sélectionnées aléatoirement sur l'ensemble des noeuds du graphe traité.

Ce modèle a néanmoins nécessité la création de plusieurs types d'entités. Le premier correspond à une entité effectuant une simple marche aléatoire.

Nos méthodes d'analyse ont permis de mettre en avant la corrélation de notre méthode avec la centralité d'intermédiarité mais seulement avec quelques graphes. C'est pourquoi nous avons créé une entité utilisant l'algorithme A*. Nous voulions prouver la viabilité du modèle de paire lorsque les entités effectuent des plus courts chemins. Les coefficients de corrélation nous ont rassurés dans nos suppositions puisque la corrélation était cette fois, très nette. Toutefois, A* est difficilement utilisable dans le cas général à cause de l'heuristique. Il fallait donc créer une troisième entité.

Afin de garder une approche de plus court chemin, et de nous plier au besoin de dynamique, nous avons proposé une entité basée sur un comportement fourmis. Bien que cette entité ne soit pas encore pleinement fonctionnelle sur tous les types de graphes, il s'agit d'une conception prometteuse comme le témoigne le premier résultat issu du traitement sur une grille.

Comme le modèle fournit des premiers résultats correctement corrélés, nous avons proposé différentes perspectives : l'amélioration du comportement de la fourmi ; une méthode de sélection des paires de nœuds plus aboutie ; la détermination du rapport entre le nombre de paires nécessaires pour une corrélation correcte et la taille du graphe ; et enfin la distribution de l'algorithme.

La conclusion de ce rapport de stage se termine donc après le rappel succinct des éléments clés. S'il n'y avait qu'un seul point à retenir, ce serait que la centralité d'intermédiairité avec pivots est une approche qui, *a priori*, peut fournir des résultats corrélés avec la centralité d'intermédiairité classique mais qu'il est nécessaire d'approfondir ces résultats.

Bibliographie

- [1] M. Nabaa, *Morphodynamique des réseaux viaire, application au risque*. PhD thesis, Université du Havre, 2011.
- [2] G. Sabidussi, “The centrality index of a graph,” *Psychometrika*, vol. 31, pp. 581–603, 1966. 10.1007/BF02289527.
- [3] A. Bavelas, “A mathematical model for group structures,” *Human Organization*, vol. 7, pp. 16–30, 1948.
- [4] J. M. Anthonisse, “The rush in a directed graph,” Tech. Rep. BN9/71, Stichting Mahtematisch Centrum, Amsterdam, 1971.
- [5] L. C. Freeman, “A set of measures of centrality based on betweenness,” *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.
- [6] L. C. Freeman, “Centrality in social networks : conceptual clarification,” *Social networks*, vol. 1, no. 3, pp. 215–239, 1979.
- [7] M. E. J. Newman, *Networks : An Introduction*. Oxford University Press, 2010.
- [8] U. Brandes, “A faster algorithm for betweenness centrality,” *Journal of Mathematical Sociology*, vol. 25, pp. 163–177, 2001.
- [9] U. Brandes and C. Pich, “Centrality estimation in large networks,” *Internationnal Journal of Bifurcation and Chaos*, vol. 17, no. 7, pp. 2303–2318, 2007.
- [10] M. E. J. Newman, “A measure of betweenness centrality based on random walks,” 2005.
- [11] L. C. Freeman, S. P. Borgatti, and D. R. White, “Centrality in valued graphs : A measure of betweenness based on network flow,” *Social Networks*, vol. 13, no. 2, pp. 141–154, 1991.
- [12] L. R. Ford and D. R. Fulkerson, “Maximal Flow through a Network,” *Canadian Journal of Mathematics*, vol. 8, no. 3, pp. 399–404, 1956.
- [13] S. P. Borgatti, “Centrality and network flow,” *Social Networks*, vol. 27, pp. 55–71, Jan. 2005.
- [14] J.-F. Gleyze, *La vulnérabilité structurelle des réseaux de transport dans un contexte de risques*. PhD thesis, Université Paris VII, 2005.
- [15] S. P. Borgatti and R. Everett, “A graph-theoretic perspective on centrality,” *Social Networks*, 2006.
- [16] K. Jiang, D. Ediger, and D. A. Bader, “Generalizing k-betweenness centrality using short paths and a parallel multithreaded implementation,” *2009 International Conference on Parallel Processing*, pp. 542–549, 2009.
- [17] R. Geisberger, P. Sanders, and D. Schultes, “Better approximation of betweenness centrality,” in *ALNEX*, pp. 90–100, 2008.

- [18] U. Brandes, “On variants of shortest-path betweenness centrality and their generic computation,” 2007.
- [19] D. Eppstein and J. Wang, “Fast approximation of centrality,” *Journal of Graph Algorithms and Applications*, vol. 8, no. 1, pp. 39–45, 2004.
- [20] T. Alahakoon, R. Tripathi, N. Kourtellis, R. Simha, and A. Iamnitchi, “K-path centrality : a new centrality measure in social networks,” in *Proceedings of the 4th Workshop on Social Network Systems*, SNS ’11, (New York, NY, USA), pp. 1 :1–1 :6, ACM, 2011.
- [21] N. E. Friedkin, “Horizons of Observability and Limits of Informal Control in Organizations,” *Social Forces*, vol. 62, no. 1, pp. 57–77, 1983.
- [22] A. Dutot, F. Guinand, Y. Pigné, and D. Olivier, “Graphstream : A tool for bridging the gap between complex systems and dynamic graphs,” *CoRR*, vol. abs/0803.2093, 2008.
- [23] D. J. Watts and S. H. Strogatz, “Collective dynamics of small-world networks,” *Nature*, vol. 393, pp. 440–442, June 1998.
- [24] A. L. Barabasi and R. Albert, “Emergence of scaling in random networks,” *Science (New York, N.Y.)*, vol. 286, pp. 509–512, October 1999.
- [25] S. N. Dorogovtsev and J. F. F. Mendes, “Evolution of networks,” in *Adv. Phys.*, pp. 1079–1187, 2002.
- [26] P. Hart, N. Nilsson, and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, pp. 100–107, July 1968.
- [27] A. Colorni, M. Dorigo, and V. Maniezzo, “Distributed optimization by ant colonies,” *Compute*, vol. 142, no. 1, pp. 134–142, 1992.
- [28] S. Goss, S. Aron, J.-L. Deneubourg, and J.-M. Pasteels, “Self-organized shortcuts in the Argentine ant,” *Naturwissenschaften*, vol. 76, Dec. 1989.
- [29] J.-L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels, “The self-organizing exploratory pattern of the argentine ant,” *Journal of Insect Behavior*, vol. 3, pp. 159–168, Mar. 1990.