Extraction de co-localisations sous contrainte de la structure spatiale

R. Govan[1], N. Selmaoui-Folcher[1], A. Giannakos[2], P. Fournier-Viger[3]

[1] Université de la Nouvelle-Calédonie, ISEA
[2] Université de Picardie Jules Verne, EPROAD
[3] Shenzhen University, Big Data Institute

{rodrigue.govan, nazha.selmaoui}@unc.nc

Résumé

Une co-localisation est un sous-ensemble de caractéristiques géographiquement proches les unes des autres. La majorité des méthodes existantes utilise des mesures standards de proximité (par exemple la distance euclidienne). Cependant, ces mesures ne sont pas les plus adaptées selon la zone d'étude. La structure spatiale doit être prise en compte. Cet article propose CSS-Miner, une approche d'extraction de co-localisations sous la contrainte de la structure spatiale. Ici, nous utilisons comme contrainte le réseau routier d'une ville. CSS-Miner a été appliqué sur deux jeux de données des villes de Paris et Chicago en sélectionnant différents points d'intérêt.

Mots-clés

co-localisation, extraction de connaissances, données spatiales, structure spatiale.

Abstract

Spatial co-location pattern is a subset of object features that are geographically close to one another. The majority of existing methods employ standard proximity measures (e.g. Euclidean distance). However, depending on the study area, these standard measures do not work well. The spatial structure has to be considered. This article proposes CSS-Miner, a co-location pattern mining approach under the spatial structure constraint. In this case, we use the street network of a city as a constraint. CSS-Miner has been applied to two datasets from the cities of Paris and Chicago by selecting different POIs.

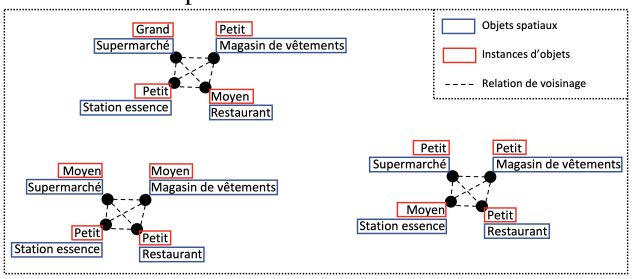
Keywords

co-location, knowledge mining, spatial data, spatial structure.

1 Introduction

Dans le domaine de la fouille des données, l'extraction de co-localisations est une des méthodes permettant d'extraire de l'information et des connaissances prenant en compte la dimension spatiale des données et pouvant aider les décideurs. Une co-localisation (ou motif spatial) est un sous-ensemble de caractéristiques spatiales qui sont fréquemment localisées dans une même zone géographique. Si l'on prend l'exemple des centres commerciaux dans une ville, ils contiennent fréquemment de grands supermarchés, des petits restaurants, des magasins de vêtements et une station essence. De ce fait, si l'on considère les centres commerciaux autour d'une ville comme une co-localisation, ses objets spatiaux sont décrits par les supermarchés, restaurants, magasins de vêtements et stations essence. Un supermarché peut être un objet spatial avec des attributs/instances tels que petit, moyen et grand. Un graphe représentant cet exemple est illustré dans la Fig. 1.

FIGURE 1 – Exemple d'une co-localisation avec ses objets spatiaux et leurs instances.



De nombreux travaux liés à l'extraction de co-localisations ont été menés [15, 19, 27]. Les méthodes d'extraction de co-localisations ont été appliquées dans divers domaines d'études, telles que l'analyse de concentration d'entreprises [6], l'explication de phénomènes anthropiques [1] et plus précisément l'analyse de l'érosion des sols [18]. Cependant, malgré un grand nombre de cas d'usage, la plupart de ces méthodes utilise des fonctions standards de distance (par exemple la distance euclidienne) pour mesurer la proximité des objets spatiaux. Pour certains cas d'usage, il est conseillé d'utiliser d'autres mesures de distance. Dans le cas d'une analyse de comportements de la population d'une ville à travers ses lieux d'intérêt, la distance euclidienne entre deux objets spatiaux n'a plus lieu d'être, car la longueur du chemin parcouru entre ces deux objets peut être significativement différente de sa distance euclidienne.
Selon la zone d'étude, il peut s'avérer essentiel de prendre en compte la structure spatiale, puisqu'elle impacte la dis-

tribution des objets spatiaux dans un espace donné. En employant la distance euclidienne dans l'analyse d'une zone urbaine, nous perdons totalement l'information sur la structure spatiale de cette zone. Afin de garder cette structure, il est nécessaire d'utiliser d'autres mesures de distance. Cependant, garder l'information de la structure spatiale de la zone d'étude peut augmenter la complexité de l'analyse, en termes de pré-traitement de données, mais aussi au niveau des paramètres à définir.

Dans cet article, nous proposons CSS-Miner (CSS pour **C**o-localisation sous contrainte de la **S**tructure **S**patiale), une méthode d'extraction de co-localisations sous contrainte de la structure spatiale de la zone d'étude. En premier lieu, la méthode construit un graphe sous cette contrainte en utilisant un algorithme de recherche du plus court chemin. Puis, CSS-Miner extrait les cliques maximales pour obtenir les motifs spatiaux. Pour les tests, la méthode proposée a été appliquée sur deux jeux de données des villes de Paris et Chicago, nous permettant d'extraire de nouveaux motifs pertinents, mais aussi de filtrer des motifs non pertinents.

L'article est organisé comme suit. La section 2 présente un bref état de l'art sur l'extraction de motifs spatiaux, en particulier avec l'approche par les événements. La section 3 décrit l'approche proposée qui prend en compte la contrainte de la structure spatiale. Puis, la section 4 présente les données utilisées dans cet article et les motifs extraits. Enfin, une conclusion est tirée et des perspectives sont discutées.

2 État de l'art

Dans leur article, Huang et al. [12] ont présenté deux approches d'extraction de motifs spatiaux : l'approche par les transactions et l'approche par les événements.

L'approche par les transactions consiste à transformer les objets spatiaux en données séquentielles dans le but d'appliquer les algorithmes standards d'extraction d'*itemsets* fréquents. Cette approche par les transactions a été initialement introduite par Koperski et al. [15].

L'approche par les événements se focalise sur la localisation des objets spatiaux et leurs proximités. Initialement proposé par Shekhar et al. [19], cette approche extrait tous les sous-ensembles d'objets qui sont géographiquement proches les uns des autres, aussi appelés co-localisations. De la même manière que l'approche par les transactions, des mesures d'intérêt ont été définies afin de ne garder que les co-localisations les plus pertinentes. Dans la littérature, nous pouvons observer que l'approche par les transactions est plus fréquemment utilisée que celle par les événements. Ces méthodes utilisent la distance euclidienne pour définir la relation de voisinage. Dans cet article, nous proposons une autre mesure de distance liée à une contrainte que nous avons nommé, la contrainte de la structure spatiale.

Pour cela, nous avons utilisé l'approche par les événements afin de tirer profit de la dimension spatiale de nos objets et leurs proximités. Pour appliquer l'approche par les événements sous notre contrainte de la structure spatiale, nous utilisons une méthode d'extraction de cliques maximales afin d'obtenir nos co-localisations. De ce fait, les sous-sections 2.1 et 2.2 suivantes donnent respectivement, un aperçu sur les approches d'extraction de cliques maximales et un aperçu des principales études menées sur l'extraction de co-localisations et leurs mesures d'intérêt.

2.1 Extraction de cliques maximales

(**Graphe complet**) Soit $G = (V, E)$ un graphe avec $V = \{v_1, v_2, \ldots, v_n\}$ l'ensemble des sommets et $E \subseteq \{(v_i, v_j) \in V^2 \mid \forall i, j \in \{1, \ldots, n\} \text{ et } i \neq j\}$ l'ensemble des arêtes. Si deux sommets v_i et v_j sont liés i.e., $(v_i, v_j) \in E$, alors v_i et v_j sont adjacents. Un graphe est dit complet si chaque paire de sommets du graphe est liée par une arête (adjacent).

(**Clique**) Soit $G = (V, E)$ un graphe et $g = (V_g, E_g)$ un sous-graphe tel que $V_g \subseteq V$ et $E_g \subseteq \{(v_{g,i}, v_{g,j}) \in E \mid v_{g,i} \in V_g \wedge v_{g,j} \in V_g\}$. Une clique de G est un sous-graphe $g \subseteq G$ tel que g est complet.

(**Clique maximale**) Pour $G = (V, E)$ un graphe donné et $g \subset G$ une clique, la clique g est dite maximale si et seulement s'il n'existe pas de clique g' telle que $g \subset g' \subseteq G$.

Avec l'approche par les événements, il est possible d'extraire nos motifs spatiaux par l'extraction de cliques maximales. Valiant [24] a démontré qu'énumérer toutes les cliques maximales est un problème #P-complet. De la même manière que les méthodes d'extraction d'itemsets fréquents, l'extraction des cliques maximales s'effectue en testant chaque combinaison de sommets d'un graphe afin d'obtenir les cliques maximales. En particulier, nous pouvons mentionner les algorithmes proposés par Bron et al. [4] et Tomita et al. [21] pour leur complexité de $O(3^{n/3})$ dans le pire scénario avec un graphe à n noeuds qui est optimal en fonction de n, mais aussi Moon et al. [17] et Cazals et al. [5] qui considèrent un appel récursif dans leur algorithme pour améliorer l'extraction des cliques maximales.

Dans la littérature, les méthodes d'extraction de cliques maximales sont communément utilisées afin d'extraire les co-localisations [2, 16, 22, 26]. En effet, en définissant un graphe où les sommets représentent des objets spatiaux et les arêtes représentant leurs voisinages puis en appliquant une méthode d'extraction de cliques maximales, nous pouvons obtenir les sous-ensembles d'objets qui sont tous voisins entre eux. Ainsi, dans cet article, nous utiliserons l'approche proposée dans [4] puis adaptée dans [21] pour sa rapidité étant donné la taille de nos jeux de données détaillés dans la section 4.1.

2.2 Extraction de co-localisations et leurs mesures d'intérêt

Le principe de l'approche par les événements est de projeter les données spatialisées par leurs coordonnées et de définir la proximité entre ces objets spatiaux dans le but d'extraire les motifs. Dans cette section, nous rappelons le formalisme de l'extraction de co-localisations proposé par Shekhar et Huang [19], Huang et al. [12] et Yoo et Shekhar [27]. Soit $\mathcal{F}$ un ensemble de caractéristiques et $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$ une base de données d'objets spatiaux. Chaque objet dans $\mathcal{O}$ se compose d'un triplet <object_id, localisation, c>, où c $\in \mathcal{F}$.

Par exemple, dans la Fig. 2.2, $\mathcal{F} = \{A, B, C\}$, $\mathcal{O} = \{A_1, B_2, \ldots, C_3\}$ avec $A_1 =< 1, (x_1, y_1), A >$, $B_2 =< 2, (x_2, y_2), B >$, etc. Une co-localisation $\mathcal{C}$ est un sous-ensemble de caractéristiques de $\mathcal{F}$ associées à des objets spatiaux appartenant à $\mathcal{O}$. Ces co-localisations représentent des caractéristiques apparaissant fréquemment dans des objets voisins. La relation de voisinage est définie par une relation binaire $\mathcal{R}(o, o')$ entre deux objets spatiaux o et o'. En fonction des besoins de l'utilisateur et des cas d'usage, $\mathcal{R}$ peut se baser sur un seuil de distance entre deux objets ou sur l'intersection de ces objets. Plusieurs travaux ont été menés, incluant Yoo et Shekhar [27], Wang et al. [25] et Kim et al. [14]. La plupart de ces travaux se basent généralement sur la distance euclidienne pour quantifier la proximité entre les objets spatiaux. Mais plus récemment, des travaux ont été menés sur l'extraction de co-localisations utilisant différentes relations de voisinage. Yu [28] a proposé dans son article la longueur du plus court chemin comme mesure de distance. Cependant, en proposant cette méthode, l'auteur ajoute un paramètre qui est le nombre maximum d'objets voisins. En définissant ce paramètre, cela assure un algorithme d'extraction rapide mais cela limite aussi la taille des co-localisations ce qui peut passer outre certains motifs qui peuvent s'avérer pertinents. Puis, Yu et al. [29] ont ajouté une fonction de décroissance de la distance afin de déterminer la dépendance spatiale entre les objets spatiaux. La fonction consiste à pondérer la contribution d'une co-localisation dans la mesure d'intérêt.

L'approche par les événements est basée sur la définition d'un seuil de voisinage. Pour déterminer si deux objets sont géographiquement proches, nous fixons un seuil de distance maximale d. Une fois que le voisinage est défini, le graphe est construit avec les objets spatiaux représentant les sommets. Deux sommets sont adjacents si leur distance associée respecte le seuil d (i.e., si la mesure de distance entre deux sommets est inférieure à d).

Pour les méthodes d'extraction de motifs spatiaux, des mesures d'intérêt ont été développées afin de quantifier la pertinence des motifs. Pour mesurer si une co-localisation est pertinente ou non, l'indice de participation, basé sur le ratio de participation est utilisé. L'indice de participation est aussi appelé la prévalence. Nous parlons ainsi de motif spatial prévalent.

(**Ratio de participation**) Soit $\mathcal{C}$ une co-localisation. Pour une instance $f_i \in \mathcal{C}$, le ratio de participation est défini par :

$$Pr(f_i, \mathcal{C}) = \frac{|\{ \text{ instances de } f_i \text{ participant à } \mathcal{C})\}|}{|\{ \text{ instances de } f_i \ \}|} \quad (1)$$

En prenant l'exemple de la Fig. 2, soit $\mathcal{C} = \{A, B\}$ une co-localisation et $I_{\mathcal{C}} = \{(A_1, B_1), (A_1, B_2), (A_3, B_4)\}$ l'ensemble des instances de $\mathcal{C}$. Avec A et B, deux caractéristiques ayant respectivement, 3 et 4 instances, nous avons :
$Pr(A, \{A, B\}) = \frac{|\{A_1, A_3\}|}{|\{A_1, A_2, A_3\}|} = \frac{2}{3}$ et
$Pr(B, \{A, B\}) = \frac{|\{B_1, B_2, B_4\}|}{|\{B_1, B_2, B_3, B_4\}|} = \frac{3}{4}$.

(**Indice de participation**) Soit $\mathcal{C}$ une co-localisation, $I_{\mathcal{C}} = \{I_1^{\mathcal{C}}, ..., I_k^{\mathcal{C}}\}$ l'ensemble des instances de $\mathcal{C}$ et $\mathcal{F} = \{f_1, \ldots, f_n\}$ l'ensemble des caractéristiques de la base de données $\mathcal{O}$. L'indice de participation est défini par :

$$Pi(\mathcal{C}) = \min_{f_i \in \mathcal{C}} Pr(f_i, \mathcal{C}) \quad (2)$$

En utilisant l'exemple précédent, nous avons pour indice de participation :

$$\begin{aligned} Pi(\{A, B\}) &= \min_{f_i \in \{A,B\}} Pr(f_i, \{A, B\}) \\ &= \min(\frac{2}{3}, \frac{3}{4}) = \frac{2}{3} \end{aligned}$$

Dans cet article, la mesure de prévalence sera utilisée afin de déterminer si les co-localisations dans la section 4 sont pertinentes ou non.

Cet indice de participation a été défini pour des données spatialisées de type point. Cependant, avec la croissance considérable de collecte de données, nous avons aujourd'hui différents types de données (lignes, polygones, ...). Dans leur contribution, Akbari et al. [1] ont proposé une variante de l'indice de participation pour chaque type de données. Pour prendre en compte tout type de données, les auteurs ont proposé de restreindre chaque région d'intérêt en appliquant le diagramme de Voronoï à partir des attributs cibles. Dans leur cas, la cible est une caractéristique spatiale qui est de type point. Une fois que le diagramme de Voronoï est appliqué, chaque cellule correspond à une instance de co-localisation. Puis, lors des calculs de prévalence, pour prendre en compte les données de type ligne/polygone, les auteurs pondèrent chaque objet spatial par la proportion de l'objet présente dans la cellule de Voronoï.

Puisque nous n'appliquons pas le diagramme de Voronoï dans cet article, nous n'allons pas utiliser la variante de la prévalence proposée dans [1]. De ce fait, nous allons devoir réduire nos données de type polygone à un seul point en prenant le centre de gravité du polygone (la moyenne de toutes les coordonnées du polygone).

Comme mentionné précédemment, les travaux autour de l'approche par les événements utilisent généralement la distance euclidienne comme relation de voisinage, ignorant la structure spatiale de la zone d'étude. Dans cet article, nous allons donc utiliser la longueur du plus court chemin existant comme mesure de proximité.

2.3 Recherche du plus court chemin

Au cours des dernières décennies, la recherche du plus court chemin a été un problème majeur dans la théorie des graphes. La vitesse de recherche dépend entièrement du nombre de sommets et d'arêtes dans un graphe. Une des premières solutions a été présentée par Dijkstra [7]. Pour un graphe de $|V|$ sommets et $|E|$ arêtes, l'algorithme de Dijkstra a une complexité polynomiale de $O((|V| + |E|) \log n)$. Puis, de nouvelles méthodes ont été développées afin d'accélérer la recherche du plus court chemin [13, 9, 20].

Plus récemment, Varia et Kurasova [23] ont proposé une version accélérée de l'algorithme de Dijkstra, en ajoutant deux composantes : le recherche bidirectionnelle et la parallélisation. Afin de chercher le plus court chemin entre

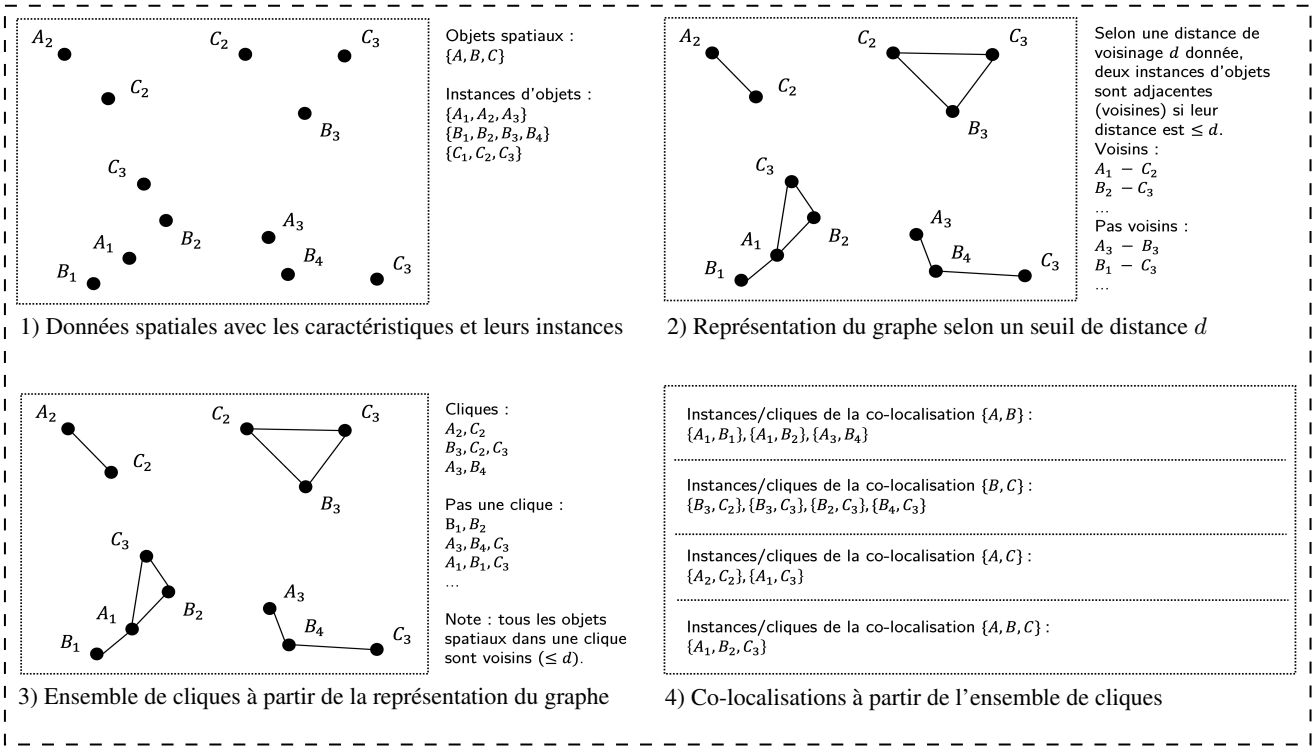


FIGURE 2 – Exemple de co-localisations basées sur un ensemble de cliques d'un jeu de données spatialisées.

deux sommets v_i et v_j, les auteurs appliquent l'algorithme de Dijkstra de v_i vers v_j et de v_j vers v_i. Puisque l'algorithme de Dijkstra est basé sur une file d'attente avec priorité, la composante bidirectionnelle utilise deux files d'attente. Cependant, lors des deux recherches, chaque recherche avance l'une après l'autre. L'avantage étant que les deux recherches seront plus courtes, mais elles avancent au tour par tour. Pour palier ce problème, les auteurs ont donc ajouté la parallélisation. Avec cette composante, les deux recherches avancent en même temps. Par l'ajout de ces deux composantes, selon les auteurs, le temps d'exécution de l'approche proposée est au minimum divisé par deux par rapport à la méthode initiale, selon le nombre de sommets dans un graphe.
Afin de prendre en compte la contrainte de la structure spatiale et d'accélérer notre processus, l'algorithme de Dijkstra bidirectionnel parallélisé sera donc utilisé.

3 Méthodes

Considérons un ensemble d'objets spatiaux $\mathcal{O}$ avec un ensemble de caractéristiques $\mathcal{F}$. Soit G_S un graphe représentant la structure spatiale telle que $G_S = (V_S, E_S)$ où V_S est l'ensemble des sommets et E_S est l'ensemble des arêtes.

3.1 Prise en compte de la contrainte de la structure spatiale

Pour analyser des points d'intérêt dans une structure spatiale (par exemple une zone urbaine), la longueur du plus court chemin parcouru entre deux localisations (x_i, y_i) et (x_j, y_j) associées respectivement, aux objets spatiaux o_i et o_j, semble la plus adaptée. Afin d'extraire les co-localisations, nous avons cherché à inclure la contrainte de la structure spatiale.
L'intégration de la contrainte est réalisée suivant plusieurs étapes :

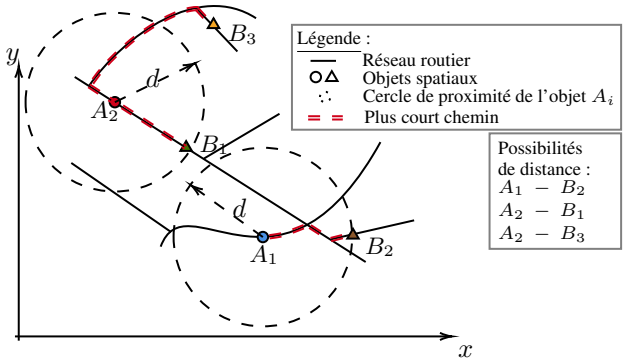
1. Pour chaque objet spatial $o_i \in \mathcal{O}$, nous l'associons dans la structure spatiale G_S au plus proche objet $o_S \in V_S$ (par la distance euclidienne) ;
2. Nous déterminons le plus court chemin pour chaque objet de V_S aux autres objets localisés dans un cercle de rayon d selon la distance euclidienne ;
3. Si la longueur du plus court chemin entre deux objets de V_S est inférieure ou égale à d, alors ces objets sont considérés comme voisins.

Afin de ne pas déterminer inutilement des plus courts chemins, nous n'appliquons pas l'algorithme de recherche du plus court chemin entre deux objets de V_S si ces deux objets ne sont pas respectivement associés à deux objets spatiaux de $\mathcal{O}$. Même si la distance euclidienne ne définit pas la relation de voisinage dans notre approche, nous l'utilisons tout de même dans le but d'élaguer le nombre de plus courts chemins calculés. Appliquer un seuil de distance par un cercle de rayon d va nous éviter de calculer des plus courts chemins qui nous le savons déjà, seront supérieurs à notre seuil. En effet, par inégalité triangulaire, un objet spatial localisé en dehors du cercle de rayon d d'un autre objet spatial a un plus court chemin qui sera supérieur à d.

3.2 Construction du graphe

Pour extraire nos motifs spatiaux (co-localisations) qui sont les cliques maximales, nous avons choisi de construire le

FIGURE 3 – Trois possibilités de distances que CSS-Miner peut rencontrer.



graphe $G = (\mathcal{O}, E_{\mathcal{O}})$ (sous la contrainte de la structure spatiale) où $E_{\mathcal{O}} = \{(o_i, o_j) \mid \exists (o_{S,i}, o_{S,j}) \in E_S, D_{sp}(o_{S,i}, o_{S,j}) \leq d, \forall (i,j) \in [\![1,n]\!]^2, i \neq j\}$ avec $o_{S,i}$ représentant l'objet de la structure spatiale associé à l'objet spatial $o_i \in \mathcal{O}$ et D_{sp} représentant la distance obtenue par l'algorithme du plus court chemin de Dijkstra.

La Fig. 3 illustre les trois possibilités que CSS-Miner peut rencontrer où les objets A_i et B_i sont des objets de V_S expliqué dans la section 3.1. Avec d en tant que rayon de distance et seuil de chemin le plus court, nous avons pour distance euclidienne, $d_2(A_2, B_3) > d$, donc CSS-Miner ne lancera pas l'algorithme de recherche du plus court chemin et ne considérera pas A_2 et B_3 comme voisins sous la contrainte de la structure spatiale. De l'autre côté, nous avons $d_2(A_1, B_2) \leq d$, donc notre algorithme lancera l'algorithme de Dijkstra. Cependant, nous avons $D_{sp}(A_1, B_2) > d$, donc nous ne considérerons pas A_1 et B_2 comme géographiquement proches (comme voisins) sous contrainte. Enfin, nous avons dans la Fig. 3, l'objet spatial B_1 localisé dans le cercle de rayon d et de centre A_2. CSS-Miner cherchera le plus court chemin, obtiendra $D_{sp}(A_2, B_1) < d$ et ainsi, considérera ces deux objets spatiaux comme voisins. Ainsi, leur valeur associée dans la matrice d'adjacence sera égale à 1.

Au final, dans CSS-Miner, nous manipulons deux graphes : Le premier représentant la structure spatiale et le second représentant le voisinage de nos jeux de données construit à partir du premier graphe.

4 Résultats

Dans cet article, nous appliquons notre approche sur deux jeux de données réelles. Le premier est créé en collectant des données de la plateforme *OpenData* de Paris [1] et sa périphérie [2] (voir la description des données dans le tableau 1). Le second jeu de données est aussi créé en collectant des données de la plateforme *OpenData* de Chicago [3] (voir la description des données dans le tableau 2).

Pour chaque jeu de données, le processus entier a été réalisé via le langage Python, sur un ordinateur avec un processeur AMD Ryzen 7 3700X 8-core, 64Go de RAM et une carte graphique NVIDIA GeForce RTX 2060 SUPER avec 8Go de RAM dédiée. Les temps d'exécution de tout le processus sur les données de Paris et Chicago ont été respectivement, d'environ 2 et 5 heures.

Cette étude de cas vise à analyser et comprendre le comportement de la jeunesse dans une grande ville. Cette approche reste tout de même générique, puisque nous pouvons l'appliquer dans une analyse de population plus large selon leurs catégories socio-professionnelles, par exemple : Quelles sont les habitudes quotidiennes d'un cadre face à un étudiant ? Une autre analyse de lieux d'intérêt peut aussi être pertinente pour le développement d'un outil d'aide à la décision afin de contribuer au développement du tourisme d'une ville. Finalement, l'analyse de lieux d'intérêt reste un sujet d'étude très varié.

TABLE 1 – Description des données de Paris.

Variable	Attributs	# Modalités	# Objets
Lycées	Type	7	239
Cinémas	# Sièges (**)	5	85
(*) Vélos	Capacité (**)	8	996
Parcs	Type	9	722
(*) Métros	Ligne	16	326

(*) : La variable concernent des stations.
(**) : Les données ont été discrétisées par quantile.

TABLE 2 – Description des données de Chicago.

Variable	Attributs	# Modalités	# Objets
Lycées	Type	13	142
(*) Bus	# Lignes	12	5 606
(*) Tramway	# Lignes	6	124
Fast Food		1	877
(*) Vélos	Capacité (*)	8	1 402
Parcs	Type	13	613

(*) : La variable concernent des stations.
(**) : Les données ont été discrétisées par quantile.

4.1 Pré-traitement des données

Pour intégrer notre contrainte de la structure spatiale, il est nécessaire d'avoir accès à cette information. Dans notre cas, nous avons utilisé le réseau routier comme structure spatiale. Ici, nous supposons que le parcours entre deux objets spatiaux s'effectue à pied. Ce choix est dû au fait que nous avons souhaité utiliser des données disponibles uniquement en libre accès, là où le trafic routier n'est pas toujours disponible.

Pour accéder aux réseaux routiers de Paris et Chicago, nous avons utilisé la méthode `OSMnx` [3]. Les auteurs ont rendu `OSMnx` simple d'utilisation. En effet, nous pouvons récupérer un réseau routier à partir du nom de la ville ou en fournissant les coordonnées de la zone d'étude via sa librairie Python. Une fois le réseau routier récupéré, il peut être représenté par un graphe avec les arêtes représentant les routes et les sommets représentant les intersections des

1. https ://opendata.paris.fr/
2. https ://data.iledefrance.fr/
3. https ://data.cityofchicago.org/

routes. Au final, le graphe associé au réseau routier de Paris contient 42 870 sommets et 241 016 arêtes tandis que le graphe associé au réseau routier de Chicago contient 184 476 sommets et 1 217 928 arêtes.

Pour les données de Paris, nous avons récupéré les données de Cinémas, Lycées, Stations de vélos, Parcs et Stations de métros. Pour restreindre le zone d'étude, nous n'avons gardé que les objets spatiaux de Paris intra-muros. Comme mentionné précédemment, nous avons choisi d'analyser que les données de type point. De ce fait, les Parcs qui sont initialement des polygones ont été réduits en un point, en l'occurrence le centre gravité. De plus, puisque les co-localisations ne sont compatibles qu'avec des données catégorielles, nous avons discrétisé deux variables (Cinémas et Stations de vélos) par quantile. Au final, nous avons 2 968 objets spatiaux décrits dans le tableau 1.

Pour les données de Chicago, nous avons récupéré les données de Tramway, Parcs, Lycées, Stations de vélos et Chaînes de *Fast Food* de la ville. Le même processus de pré-traitement des données de Paris a été appliqué aux données de Chicago. Ainsi, nous avons 8 764 objets spatiaux qui composent nos données de Chicago.

Pour les deux jeux de données et leur structure spatiale associée, nous avons projeté toutes les coordonnées dans le système de coordonnées WGS 84 / Pseudo-Mercator (EPSG : 3857). Ce système de coordonnées nous permet de pouvoir calculer les distances en mètres.

À l'étape d'élagage, nous avons fixé un seuil sur le rayon de $500m$ ($d = 500$). Chaque objet ne sera comparé qu'aux objets contenus dans ce cercle de rayon d. Lors de la construction du graphe et de sa matrice d'adjacence associée, pour déterminer si deux objets (deux sommets) sont contigus, nous avons fixé le seuil de la distance à pied au même seuil du rayon ($d = 500$). Ainsi, si le plus court chemin trouvé entre deux objets est inférieur à $500m$, alors leur valeur associée dans la matrice d'adjacence est égale à 1. Sinon, elle est égale à 0. De plus, pour éviter toute boucle dans le graphe (une arête reliant un sommet à lui-même), nous avons mis toute la diagonale de la matrice d'adjacence à 0.

4.2 Données de Paris

Suite au pré-traitement des données et à la construction du graphe, nous avons extrait les cliques maximales. Puisque l'article cherche à analyser le comportement de la population jeune de Paris, le tableau 3 ne montre que les co-localisations contenant la variable Lycées.

Le tableau 3 nous montre les possibles activités à proximité des lycées pour la population jeune de Paris, notamment par la présence des parcs et des cinémas. De plus, nous observons par ces co-localisations, l'omniprésence des variables Lycées et Vélos (stations de vélos en libre-service), ce qui montre aussi que la ville de Paris aide au mieux la jeunesse parisienne à se déplacer librement et en même temps pratiquer une activité sportive. Il serait intéressant d'appliquer CSS-Miner sur d'autres villes de France proposant ce service afin de pouvoir confirmer cette tendance.

Puisque CSS-Miner considère le réseau routier comme contrainte de structure spatiale, l'idée est de voir s'il y a une différence par rapport aux co-localisations sans cette contrainte, c'est-à-dire avec la distance euclidienne comme relation de voisinage. Les résultats nous montrent que les motifs extraits sous contrainte n'ont pas systématiquement une prévalence supérieure à la prévalence obtenue sans contrainte. Nous pouvons l'expliquer comme suit. Les motifs extraits sans contrainte ont été obtenus en utilisant le même seuil de distance (i.e., $500m$), tout comme CSS-Miner. Par inégalité triangulaire, un chemin parcouru à pied entre deux objets spatiaux est supérieur à sa distance euclidienne. De ce fait, sans la contrainte, les cliques maximales contiennent plus d'objets, augmentant la probabilité d'avoir un grand nombre d'instances par variable, ce qui peut réduire leur prévalence. Cela explique aussi pourquoi la co-localisation {Parcs, Lycées, Vélos} avec une prévalence de 0.89, voit sa prévalence diminuer à 0.56 si la variable Cinémas y est ajoutée. En effet, en ajoutant une variable dans une co-localisation, cela augmente le nombre d'objets spatiaux contenus dans cette co-localisation, ce qui peut diminuer sa prévalence.

TABLE 3 – Prévalences des co-localisations de Paris.

Co-localisation	Prévalence sous contrainte	Prévalence sans contrainte
{Parcs, Lycées, Vélos}	0.89	0.89
{Lycées, Vélos}	0.86	0.86
{Parcs, Lycées, Vélos, Métros}	0.78	0.89
{Lycées, Cinémas, Vélos}	0.71	0.71
{Lycées, Vélos, Métros}	0.71	0.71
{Lycées, Cinémas, Vélos, Métros}	0.71	0.71
{Parcs, Lycées, Cinémas, Vélos}	**0.56**	0.44

Enfin, sans prendre en compte la contrainte de la structure spatiale, l'algorithme a extrait des motifs que CSS-Miner n'a pas extraits. Ces motifs sont : {Lycées, Métros} et {Parcs, Lycées, Cinémas, Métros} avec des prévalences égales à 0.31 et 0.14 respectivement. Ces deux motifs ont une prévalence nulle si l'on prend en compte la contrainte. Cela montre que même si les objets spatiaux sont proches par la mesure euclidienne, la longueur de leur plus court chemin ne vérifie pas notre critère de voisinage. Ces objets ne peuvent donc pas être considérés comme voisins. Au final, en prenant en compte la structure spatiale de la zone d'étude, nous pouvons extraire les motifs pertinents et filtrer les motifs non pertinents, selon la zone d'étude.

4.3 Données de Chicago

De la même manière qu'avec les données de Paris, le tableau 4 ne montre que les co-localisations contenant la variable Lycées de Chicago.

Les prévalences du tableau 4 montrent que la majorité des Lycées de Chicago ont une chaîne de *Fast Food* à proximité, donc la population jeune de Chicago sera plus tentée

FIGURE 4 – Données de Paris avec les cliques maximales extraites.

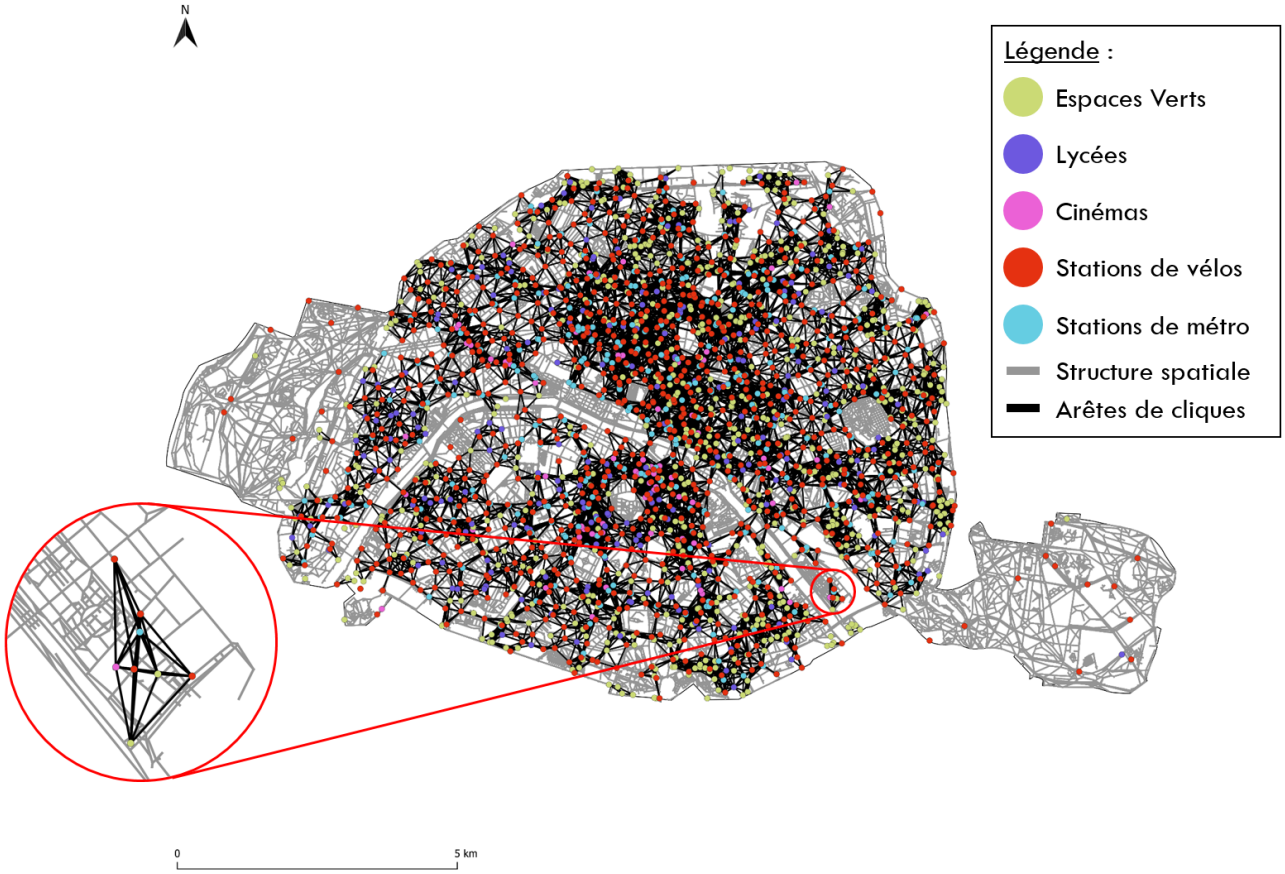


TABLE 4 – Prévalences des co-localisations de Chicago.

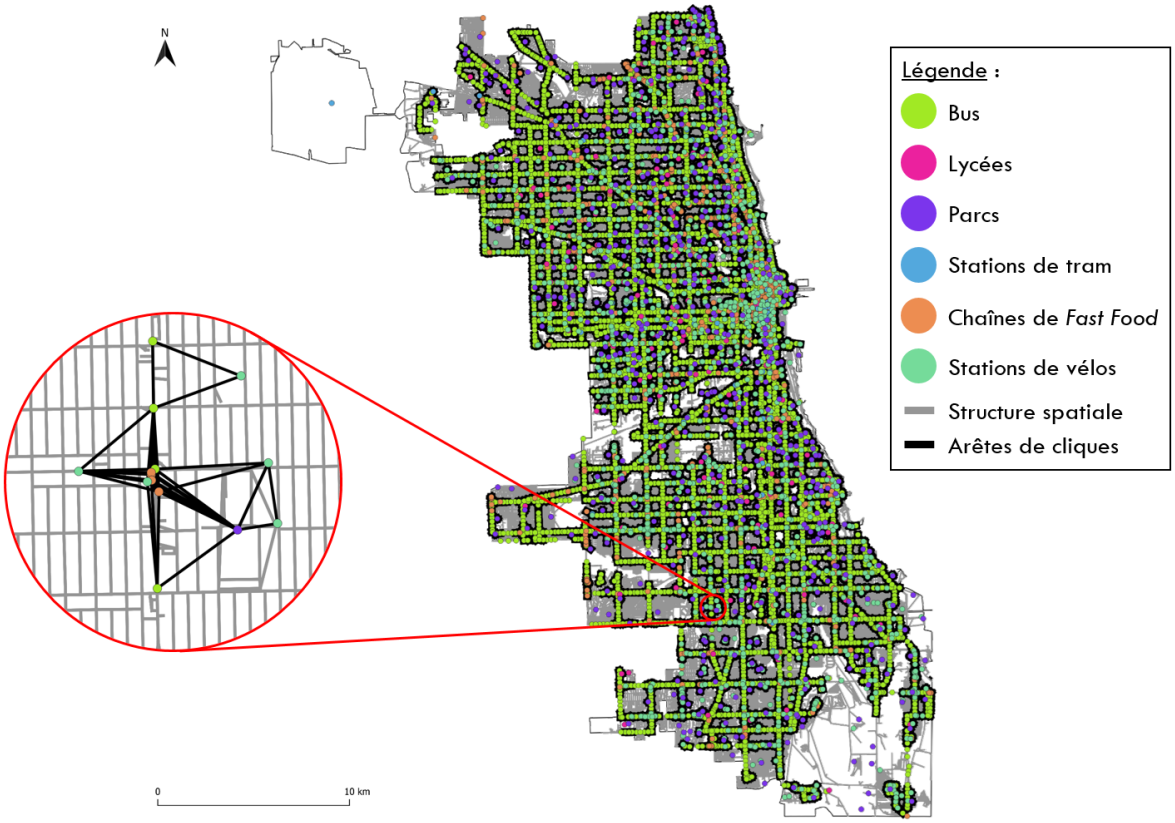
Co-localisation	Prévalence sous contrainte	Prévalence sans contrainte
{Bus, *Fast Food*, Lycées, Vélos}	**0.58**	0.5
{Bus, *Fast Food*, Lycées, Tramway, Vélos}	0.38	0.38
{Bus, *Fast Food*, Lycées}	**0.33**	0.17
{Bus, *Fast Food*, Lycées, Tramway}	0.3	0.3
{Bus, *Fast Food*, Lycées, Parcs}	0.17	0.17
{Bus, *Fast Food*, Lycées, Tramway, Vélos, Parcs}	0.15	0.15

de manger dans un *Fast Food* à midi ou en sortant d'école. L'omniprésence des variables Lycées et *Fast Food* dans nos co-localisations peut aussi alarmer la population sur la malnutrition des américains, ou du moins de la population jeune de Chicago. Pour confirmer cette affirmation, il serait intéressant d'appliquer cette approche sur les grandes villes des États-Unis et vérifier si nous pouvons extraire ces mêmes co-localisations. Il serait également intéressant d'obtenir un jeu de données recensant les *Fast Food* de Paris afin de démontrer si les chaînes de restauration rapide à Paris ciblent la jeune population de la même manière qu'à Chicago. Cependant, ce jeu de données n'est malheureusement pas disponible sur les plateformes *OpenData*.
Enfin, nous notons qu'à partir de ces co-localisations, l'omniprésence d'un moyen de transport en commun à proximité des Lycées de Chicago, ce qui peut supposer que la ville de Chicago est bien desservie. Tout comme les données de Paris, en se basant sur les résultats de prévalence, nous avons plus de motifs pertinents sous contrainte que sans la contrainte pour les mêmes raisons énoncées précédemment.

5 Conclusion et perspectives

Dans cet article, nous avons présenté CSS-Miner, une approche d'extraction de co-localisations sous contrainte de la structure spatiale de la zone d'étude. Nous avons décrit cette contrainte et comment nous l'avons prise en compte, en l'occurrence avec un réseau routier et une recherche du plus court chemin. Pour extraire nos motifs spatiaux, nous avons utilisé l'approche de l'extraction de cliques maximales avec une recherche de voisins restreinte par un cercle de rayon d modifiable par l'utilisateur, selon le cas d'étude.

FIGURE 5 – Données de Chicago avec les cliques maximales extraites.



Au final, grâce aux plateformes *OpenData* de Paris et Chicago, nous avons pu créer deux jeux de données réelles.

Cependant, durant l'étape de pré-traitement de données, nous avons choisi de réduire tous nos objets spatiaux en points. Nos futurs travaux seront garder le type initial de nos données (points, lignes, polygones, ...). De plus, à l'étape de la recherche du plus court chemin, un croisement entre la structure spatiale et nos objets spatiaux est effectué. Afin d'optimiser notre recherche du plus court chemin, une phase additionnelle d'élagage semble nécessaire. Plusieurs travaux ont été menés sur l'élagage de graphes, par exemple un élagage par filtre léger sur les réseaux de neurones convolutifs [11] ou encore un élagage de graphe appliqué sur une grille [10]. Donc une des prochaines étapes de nos travaux sera d'élaguer le graphe associé à notre structure spatiale afin d'accélérer notre étape de recherche du plus court chemin.

Par ailleurs, CSS-Miner reste une méthode d'analyse exploratoire, la prochaine étape de nos travaux sera donc d'intégrer la connaissance d'experts métier [8], tels que des urbanistes, des démographes et géographes, afin de vérifier la pertinence de nos co-localisations extraites.

Enfin, dans cet article, nous avons supposé que le chemin effectué entre deux objets spatiaux était à pied. En perspective, pour pouvoir considérer la distance en voiture (ou à vélo), nous prévoyons de considérer la structure spatiale par un graphe orienté étant donné que toutes les routes en voiture ne sont pas bidirectionnelles. Par la suite, pour que la distance en voiture soit pertinente, il serait donc nécessaire d'intégrer la dynamique temporelle avec les heures d'affluences impactant le trafic routier. Cependant, ces données intégrant la dimension temporelle ne sont pas forcément disponibles en libre accès. De ce fait, cette tâche nécessitera d'utiliser des *API* fournies par Google et d'autres entreprises de gestion du trafic routier.

Remerciements

Ces travaux ont été réalisés dans le cadre du projet SPIraL (ANR-19-CE35-0006-02) et financés par l'Agence Nationale de Recherche. Nous remercions Dr. Cyrille Goarant de l'Institut Pasteur de Nouvelle-Calédonie pour ses conseils durant la réalisation de cet article.

Références

[1] Mohammad Akbari, Farhad Samadzadegan, and Robert Weibel. A generic regional spatio-temporal co-occurrence pattern mining model : a case study for air pollution. *Journal of Geographical Systems*, 17(3) :249–274, 2015.

[2] Xuguang Bao and Lizhen Wang. A clique-based approach for co-location pattern mining. *Information Sciences*, 490 :244–264, 2019.

[3] Geoff Boeing. Osmnx : New methods for acquiring, constructing, analyzing, and visualizing complex

street networks. *Computers, Environment and Urban Systems*, 65 :126–139, 2017.

[4] Coen Bron and Joep Kerbosch. Algorithm 457 : finding all cliques of an undirected graph. *Communications of the ACM*, 16(9) :575–577, 1973.

[5] Frédéric Cazals and Chinmay Karande. A note on the problem of reporting maximal cliques. *Theoretical computer science*, 407(1-3) :564–568, 2008.

[6] Jeffrey Chiu, Amin Vahedian Khezerlou, and Xun Zhou. Understanding business location choice pattern : A co-location analysis on urban poi data. In *Proceedings of the 2nd INFORMS Workshop on Data Science, Phoenix, AZ, USA*, volume 3, 2018.

[7] Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1) :269–271, 1959.

[8] Frédéric Flouvat, Jean-François N'guyen Van Soc, Elise Desmier, and Nazha Selmaoui-Folcher. Domain-driven co-location mining : Extraction, visualization and integration in a gis. *Geoinformatica*, 19 :147–183, 2015.

[9] M.L. Fredman and R.E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. In *25th Annual Symposium onFoundations of Computer Science, 1984.*, pages 338–346, 1984.

[10] Daniel Harabor and Alban Grastien. Online graph pruning for pathfinding on grid maps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 1114–1119, 2011.

[11] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. Soft filter pruning for accelerating deep convolutional neural networks. *arXiv preprint arXiv :1808.06866*, 2018.

[12] Yan Huang, Shashi Shekhar, and Hui Xiong. Discovering colocation patterns from spatial data sets : a general approach. *IEEE Transactions on Knowledge and data engineering*, 16(12) :1472–1485, 2004.

[13] Donald B. Johnson. Efficient algorithms for shortest paths in sparse networks. *J. ACM*, 24(1) :1–13, jan 1977.

[14] Seung Kwan Kim, Jee Hyung Lee, Keun Ho Ryu, and Ungmo Kim. A framework of spatial co-location pattern mining for ubiquitous gis. *Multimedia tools and applications*, 71(1) :199–218, 2014.

[15] Krzysztof Koperski and Jiawei Han. Discovery of spatial association rules in geographic information databases. In *International Symposium on Spatial Databases*, pages 47–66. Springer, 1995.

[16] Seung Kwan Kim, Younghee Kim, and Ungmo Kim. Maximal cliques generating algorithm for spatial colocation pattern mining. In *Secure and Trust Computing, Data Management and Applications : 8th FIRA International Conference, STA 2011, Loutraki, Greece, June 28-30, 2011. Proceedings 8*, pages 241–250. Springer, 2011.

[17] J.W. Moon and L. Moser. On cliques in graphs. *Israel J. Math.*, 3 :23—-28, 1965.

[18] Nazha Selmaoui-Folcher, Frédéric Flouvat, Dominique Gay, and Isabelle Rouet. Spatial pattern mining for soil erosion characterization. In *New Technologies for Constructing Complex Agricultural and Environmental Systems*, pages 190–210. IGI Global, 2012.

[19] Shashi Shekhar and Yan Huang. Discovering spatial co-location patterns : A summary of results. In *International symposium on spatial and temporal databases*, pages 236–256. Springer, 2001.

[20] Mikkel Thorup. Undirected single-source shortest paths with positive integer weights in linear time. *J. ACM*, 46(3) :362–394, may 1999.

[21] Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363 :28–42, 2006.

[22] Vanha Tran, Lizhen Wang, Hongmei Chen, and Qing Xiao. Mcht : A maximal clique and hash table-based maximal prevalent co-location pattern mining algorithm. *Expert Systems with Applications*, 175 :114830, 2021.

[23] Gintaras Vaira and Olga Kurasova. Parallel bidirectional dijkstra's shortest path algorithm. *Databases and Information Systems VI, Frontiers in Artificial Intelligence and Applications*, 224 :422–435, 2011.

[24] L.G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3) :410—-421, 1979.

[25] Lizhen Wang, Yuzhen Bao, and Zhongyu Lu. Efficient discovery of spatial co-location patterns using the icpi-tree. *The Open Information Systems Journal*, 3(1), 2009.

[26] Xiaojing Yao, Ling Peng, Liang Yang, and Tianhe Chi. A fast space-saving algorithm for maximal colocation pattern mining. *Expert Systems with Applications*, 63 :310–323, 2016.

[27] Jin Soung Yoo and Shashi Shekhar. A joinless approach for mining spatial colocation patterns. *IEEE Transactions on Knowledge and Data Engineering*, 18(10) :1323–1337, 2006.

[28] Wenhao Yu. Spatial co-location pattern mining for location-based services in road networks. *Expert Systems with Applications*, 46 :324–335, 2016.

[29] Wenhao Yu, Tinghua Ai, Yakun He, and Shiwei Shao. Spatial co-location pattern mining of facility points-of-interest improved by network neighborhood and distance decay effects. *International Journal of Geographical Information Science*, 31(2) :280–296, 2017.