
Une approche basée sur l'Analyse Formelle de Concepts pour la Recherche en Continue des K Plus Proches Voisins dans les Réseaux Routiers

Hafedh FERCHICHI ¹, Jalel AKAICHI ²

1. ISET de Jendouba, Laboratoire BESTMOD,
Campus Universitaire 8189 Jendouba Nord, Tunisie
ferchichi.hafedh@gmail.com
2. ISG de Tunis, Laboratoire BESTMOD,
41, Avenue de la Liberté, Cité Bouchoucha, Le Bardo 2000 - Tunis
j.akaichi@gmail.com

RESUME. Ce papier présente une nouvelle approche de recherche en continue des K plus proches voisins (C-KNN). Notre approche, qui s'applique sur des réseaux routiers, adopte l'Analyse Formelle de Concepts (FCA : Formel Concept Analysis) qui est un paradigme à base de fondement mathématique. FCA permettra de présenter une abstraction du réseau qui se base sur les voisinages. L'approche proposée consiste à construire le treillis de Galois à partir des relations binaires entre les points d'intérêts candidats et leurs propriétés. Ces dernières sont extraites à partir d'un ensemble de capteurs associés au réseau routier en question. Une phase d'indexation permet d'accélérer le processus de recherche afin de réduire le temps de traitement. Une étude de cas est aussi proposée afin de montrer la faisabilité de notre approche de recherche en continue des KNN à base de FCA.

ABSTRACT. This paper presents a new approach to the continuous K nearest neighbors search (C-KNN) problem, in the context of road networks. Our approach is based on Formal Concepts Analysis (FCA) which has a mathematical foundation. FCA offers an abstraction of the network based on the neighborhoods. We build the concept lattice based on the binary relations between the target points as well as theirs properties. The latter are collected from various sensors on the road network. An indexing phase is also defined to speed up the search process and to reduce the processing time. Finally, a case study is presented to show the effectiveness of our FCA-based solution.

MOTS-CLES : Analyse formelle de concepts, requêtes de plus proche voisin, CKNN, indexation, réseau spatial.

KEYWORDS: Formel Concept Analysis, K-Nearest Neighbors queries, CKNN, indexation, spatial network.

1. Introduction

Avec l'intégration des communications sans fils et les technologies de positionnement (GPS-Global Positioning System), les applications et les services basés sur la localisation (LBS Location-based services) émergent et gagnent rapidement du progrès. Une classe importante de problème dans les LBSs consiste en la recherche du plus proche voisin (NN). Jusqu'à présent, la recherche des k-NN constitue un problème majeur dans les entrepôts de données notamment dans les entrepôts de données des objets mobiles qui décrivent des environnements dynamiques. Il existe plusieurs techniques de traitement efficaces des requêtes de recherches des k-NN dans un espace de données statique. Récemment, la recherche s'est focalisée sur les requêtes de recherche en continu des K-NN (C-KNN : Continuous K Nearest Neighbor) qui s'intéresse aux objets mobiles sur un réseau routier. Une requête continue est une requête qui, au lieu d'être évaluée une seule fois à l'instant où elle est soumise au système, s'évalue de manière continue pour un intervalle de temps donné (Terry. D et al, 1992). Les approches actuelles de traitement de ce genre de requêtes dans des environnements dynamiques ont montré des insuffisances et ont été, dans la majorité des cas, incapables de donner satisfaction aux usagers. Cependant, une pertinence et une efficacité des résultats attendus dépendent énormément de la façon avec laquelle est indexé l'espace de recherche ainsi que les méthodes de recherche utilisées dans ces structures d'indexés. L'émergence de l'analyse formelle de concepts (FCA) dans divers domaines de l'informatique, que ce soit en extraction et représentation des connaissances (Lakhel et Stumme, 2005), dans les domaines des ontologies (Bendaoud et al., 2010) ou encore des bases de données, a ainsi mis en avant les structures de treillis des concepts. La recherche en continu des K-NN est un domaine en plein essor. Avec l'absence d'un standard de traitement de ce genre de requête dans un environnement dynamique, plusieurs approches ont été proposées. Le dilemme est de fournir aux usagers des réponses valides au moment de réception. Pour aboutir à des tels résultats, l'FCA semble être un bon candidat, permettant de regrouper les points d'intérêts en une hiérarchie de niveaux. Chaque niveau correspond à un regroupement d'objets mobiles qui partagent un ensemble de propriétés communes (Vitesse, Position, Direction ...). Nous proposons, ainsi, une nouvelle approche de traitement en continu des k-NN qui s'applique sur des réseaux routiers. Notre contribution profite d'une technique mathématique, l'analyse formelle de concepts, et ce afin de présenter une abstraction du réseau qui se base sur les voisinages. Notre approche vise à répondre aux requêtes des usagers en tenant compte de l'état de la route et du contexte de l'utilisateur.

Le reste du papier est organisé comme suit : La section 2 synthétise l'état de l'art des méthodes de recherche des k-plus proches voisins dans un réseau routier. La section 3 donne une idée sur la solution adoptée. Dans la section 4 nous présentons notre approche à base de FCA pour la recherche des k-plus proches voisins dans un réseau routier. La section 5 présente une synthèse de notre approche. La section 6 est consacrée à la conclusion et aux travaux futurs.

2. Etat de l'art

La problématique de recherche des K plus proches voisins présente plusieurs axes où les chercheurs essaient de trouver des solutions. Le classement le plus fréquent de cette problématique se base sur la façon avec laquelle seront calculées les distances entre les points requêtes et les points d'intérêts, dont on distingue la distance euclidienne et la distance de plus court chemin. Pour chercher les K plus proches voisins on pourrait détecter plusieurs points de différences entre l'espace euclidien et le réseau routier. Ces différences touchent essentiellement la métrique utilisée (distance euclidienne entre deux points ou distance du plus court chemin dans le réseau routier), le type de recherche à appliquer (Par point (kNN) ou Par intervalle (kNN+CkNN)) et les techniques de recherches utilisées (Structure d'index ou Techniques géométriques+Structures d'index).

Les techniques d'indexation multidimensionnelles ont été largement étudiées. En 1984, Guttman propose le R-Tree (Guttman, 1984), une technique référence dans ce domaine. Plusieurs variantes du R-Tree sont apparues, parmi elles le R*Tree (Beckmann et al., 1990) ou encore le X-Tree (Berchtold et al., 1997). Ces structures d'indexation ont montré leurs limites lors du passage à des dimensions élevées. La plupart des travaux existants essaient d'offrir une réponse valide pour différents types de requêtes instantanées. Ce qui a été proposé par (Song et Roussopoulos, 2001) dépend du nombre d'exemples fournis en entrée, si le nombre d'exemples est faible le résultat sera erroné. (Tao et Papadias, 2002 ; 2003) et (Feng et Watanabe, 2004) ont proposé une technique valable uniquement pour la recherche du premier plus proche voisin (1NN). (Khayati et Akaichi, 2008) Utilisent la triangulation de Delaunay pour modéliser un réseau routier constitué de routes directes joignant des points de l'espace. La technique ne peut être appliquée, que sur un réseau routier avec des routes directes (sans virage) car leur proposition ne prend en considération que les points inclus dans la triangulation. Les auteurs proposent comme perspectives d'appliquer ce modèle de partitionnement de l'espace de recherche sur des réseaux routiers avec des routes sinueuses en ajoutant des facteurs de pondérations comme le trafic urbain, le temps écoulé, la vitesse...

Plusieurs modèles et algorithmes ont été proposés pour faire face aux changements constamment des positions des objets en mouvement. Toutefois, ces travaux présentent encore énormément de limites :

- la majorité des travaux existants se basent sur la métrique de la distance euclidienne. Toutefois, dans un monde réel, les requêtes se déplacent sur un réseau routier. Etant donné un chemin d'un point P_r à un point P_i , on appelle longueur du chemin, la somme des longueurs des arcs qui le constituent qui s'oppose au calcul de la distance euclidienne entre P_r et P_i qui n'est que la longueur de la ligne directe qui sépare ces deux points.
- la plupart des techniques existantes pour la recherche des K-NN ignorent l'état du réseau routier et ne prennent pas en charge la requête en temps réel ;

- la plupart des algorithmes existants génèrent un surcoût de stockage considérable ;
- les approches précédentes ne peuvent pas bien fonctionner s'il ya des mises à jour fréquentes du déplacement d'objets, qui se produisent généralement dans des applications réelles ;
- les Travaux existant ne sont pas en mesure de faire face à la représentation dynamique des attributs des objets en mouvement ;

Même les tentatives récentes, telles que (Zhong et al., 2013) qui propose un indice G-arbre pour trouver les K-NN à un emplacement donné, ont montré des limites par rapport à la taille du réseau étudié. G-arbre nécessite un prétraitement de plusieurs heures pour les réseaux routiers (16,8 heures de prétraitement pour le réseau complet Etats-Unis) par conséquent ça ne peut pas être applicable en cas des objets mobiles sur un réseau routier car G-arbre est construit avec une vue statique de l'information sur le trafic du réseau routier. Dans le même contexte Notre travail (Ferchichi et Akaichi, 2013) s'inspire de la validité des approches statiques pour proposer une technique efficace qui détermine le premier plus proche voisin (1NN) dans un environnement dynamique.

La majorité des approches étudiées présentent des insuffisances surtout lors du passage à des dimensions élevées ou dans le cas où il s'agit d'un contexte dynamique. Pour en résumer le problème, actuellement pour connaître les plus proches voisins d'un point requête dans l'espace, on est obligé de parcourir tous les points déjà existants, ce qui implique beaucoup de comparaisons.

3. Définition du problème et exemple de motivation

3.1 Analyse Formelle de Concepts

L'Analyse Formelle de Concepts (AFC) est un formalisme mathématique qui permet d'obtenir des concepts structurés hiérarchiquement regroupant des objets possédant les mêmes attributs. La hiérarchie résultant de l'AFC est appelée treillis de Galois (Barbut et Monjardet, 1970) ou treillis de concepts (Ganter et Wille, 1999). Le fondement mathématique de l'AFC et les structures conceptuelles qu'elle permet de dériver (Godin et al., 1995) ont été exploités dans plusieurs domaines d'analyse et d'exploitation de données tels que la classification, la recherche d'information (Carpinetto et Romano, 2005), la sélection de services Web (Azmeah et al., 2008), la construction d'ontologies (Bendaoud et al., 2008), l'extraction de connaissances (Lakhal et Stumme, 2005), l'ingénierie des logiciels (Tilley et al., 2005 ; Godin et Valtchev, 2005), la linguistique (Priss, 2005), etc.

L'AFC consiste à construire un treillis de concepts à partir d'un tableau binaire Objets \times Attributs. Formellement, un contexte K est la donnée d'un triplet (O, A, I) où O est un ensemble d'objets, A est un ensemble d'attributs, et $I \subset O \times A$ une relation entre O et A.

3.2 Définitions et symboles préliminaires

Définition 1 (Contexte Formel) : Un Contexte Formel est un triplet $K = (G, M, I)$ où G est un ensemble d'objets, M est un ensemble d'attributs et I une relation binaire entre G et M vérifiant: $I \subseteq G \times M$; $(g, m) \in I$ avec $g \in G$ et $m \in M$ signifie que l'objet g possède l'attribut m ou que l'attribut m est possédé par l'objet g .

Définition 2 (Concept Formel) : Un Concept Formel d'un contexte $K = (G, M, I)$ est une paire (A, B) avec : $A \subseteq G$, $B \subseteq M$, $A' = B$ et $B' = A$, où A' est l'ensemble de tous les attributs de B possédés par les objets de A et de façon duale B' est l'ensemble de tous les objets possédant les attributs de B . Les ensembles A et B sont appelés respectivement *extension* et *intension* du concept formel C . C (G, M, I) dénote l'ensemble de tous les concepts du contexte $K = (G, M, I)$.

Définition 3 (Objet Mobile) : Un Objet Mobile « O_i », avec $0 < i \leq n$, est caractérisé par une Position (P), une Vitesse (V) et une Direction (D). « O_i » pourrait être soit un point requête (Pr) soit un point d'intérêt (Pi). O_i (P_i, V_i, D_i) avec O_i , Objet Mobile i , P_i position de l'objet mobile i , V_i Vitesse de l'objet mobile i , D_i direction de l'objet mobile i .

Définition 4 (Point requête- Pr) : un Point requête (Pr) désigne chaque Objet mobile « O_i » qui lance une requête (r). « Pr » est caractérisé par une Position, une Vitesse et une Direction. Pr (P, V, D), avec Pr point requête, P position, V vitesse et D direction (East, West, North, South).

Définition 5 {Point d'intérêt (Pi) avec $0 < i \leq n$ } : un Point d'intérêt (Pi) désigne chaque objet mobile sur un réseau routier qui satisfait les critères d'une requête donnée. Pi est Caractérisé par une Position, une Vitesse et une Direction. Pi (P_i, V_i, D_i), avec Pi point d'intérêt i , V_i vitesse de l'objet mobile i et D_i direction de l'objet mobile.

Définition 6 Statut (O_1, O_2) : Chaque objet mobile (O_1) admet un Statut par rapport à un autre objet mobile (O_2). Généralement, O_1 est un point requête alors qu' O_2 est un point d'intérêt. Statut (O_1, O_2) est une fonction qui retourne soit « loin » soit « près ». Statut (Pr, Pi) = Loin. Statut (Pr, Pi) = Prés.

Illustration 1 Statut (Pr, Pi) = {Loin || Prés}. Déterminer le statut d'un objet mobile revient à étudier les caractéristiques suivantes : la direction de l'objet mobile « Pr » ainsi que celle de l'objet mobile « Pi » (même direction ou direction inverse) ; les vitesses de l'objet mobile « Pr » ainsi que celle de l'objet mobile « Pi » (V_r est-elle supérieure ou inférieure à V_i ?)

Algorithm 1. Statut (in ($Pr, Pi, Vr, Vi, D1, D2$), out ($Near, Far$))

Input: $Pr (X_{pr}, Y_{pr}), Pi (X_{pi}, Y_{pi}), Vr, Vi, Dr, Di$

Output: "Near", "Far"

```

1:   si (( $V_i < V_r$ ) et ( $D_r = D_i$ )) ou (( $V_i \geq V_r$ ) et ( $D_r \neq D_i$ )) alors
2:       return "Near"
3:   fin condition

```

Copyright © by the paper's authors. Copying permitted for private and academic purposes. Proceedings of the Spatial Analysis and GEOmatics conference, SAGEO 2015.

```

4:      si ((Vi > Vr) et (Dr= Di)) Ou ((Vi < Vr) et (Dr != Di)) alors
5:          return "Far"
6:      fin condition
7:      fin

```

Statut (Pr, Pi) = {Prés} dans le cas où les deux objets mobiles Pr et Pi admettent une même direction alors que Statut (Pr, Pi) = {Loin} dans tous les autres cas.

Soit l'ensemble d'objet mobile : $O_i = \{O_1, O_2, O_3, \dots, O_n\}$ tel que $0 < i \leq n$. À un instant « T » donné, chaque Objet mobile « O_i » est caractérisé par :

- **Position** : On désigne par « P » la position d'un point d'intérêt et par « P_i » l'ensemble des positions des points d'intérêts $P_i = \{P_1, P_2, P_3, \dots, P_n\}$ tel que $0 < i \leq n$. L'objet 1 ($i=1$) est situé alors à la position P_1 , l'objet 2 ($i=2$) est situé à la position P_2 , l'objet n ($i=n$) est situé à la position P_n . Alors que le point requête « Pr » est situé à la position P-Pr. Position P_i est étroitement liée à l'arc sur lequel est situé l'Objet en question.
- **Vitesse** : les vitesses des objets mobiles étudiés vont être classées selon trois intervalles [0 km-50 km], [51 km-90 km] et [91 km-110 km]. On désigne par V_i la vitesse de l'objet i .
- **Direction** : On désigne par « W » la direction West, par « E » la direction East, par « N » la direction North, et par « S » la direction South. Au moment d'interrogation le système doit indiquer si l'objet en question suit la même direction (Same Direction =SD) ou une direction inverse (Reverse Direction =RD) par rapport au point requête.
- **Facteur de perturbation** : On définit un facteur de perturbation par, tout genre d'obstacle qui pourrait ralentir ou empêcher l'avancement des objets mobiles sur un réseau routier (un embouteillage, des travaux routiers, un mauvais temps, un accident routier). Un facteur de perturbation est une caractéristique propre à un arc.

Exemple. L'objet O_2 se trouve à une position P_2 . La position P_2 appartient à un arc porteur du caractère facteur de perturbation. On admettra que l'objet O_2 possèdera cette caractéristique. Autrement dit si O_2 sera l'un des NN cherchés par le point requête il sera automatiquement éliminé à cause de cette caractéristique qui pourrait empêcher le point requête de l'atteindre.

- Statut (X, Y) : Un point d'intérêt admet un statut par rapport à un point requête. Le résultat attendu est déterminé en étudiant la direction et la vitesse de l'objet mobile dont on veut déterminer son statut.

3.3 Exemple de motivation

Citons par exemple le cas d'un usager (dans sa voiture) qui recherche les centres d'urgence mobile les plus proches de lui, tout au long de son trajet. La réponse à ce type de requête doit être valide au moment de sa réception par le client. Un point requête commencera son trajet à partir du nœud « N_1 » pour atteindre le nœud « N_{12} » (Figure 1). On suppose que tous les objets mobiles présents sont des réponses candidates à la requête en question. Figure 1 présente un graphe de réseau

routier formé de 12 nœuds, et considéré comme notre espace de recherche. Les propriétés des objets mobiles changent avec le temps.

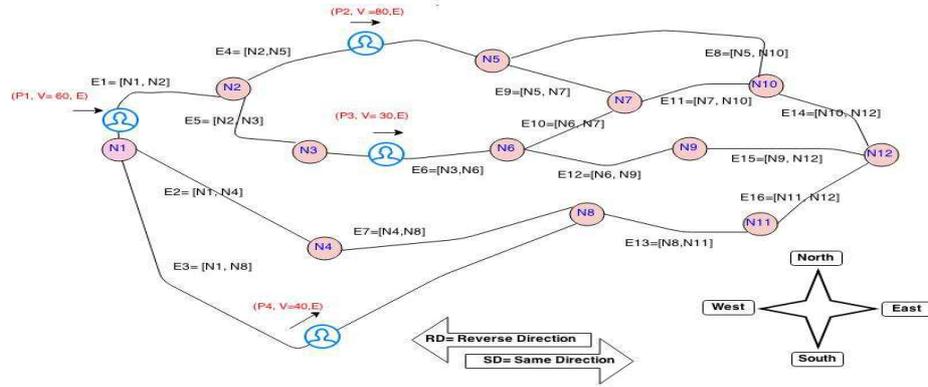


Figure 1 Espace de recherche à l'instant $t = 0$,

4. Approche de recherche des K-NN à base de FCA : FCA-Cknn

4.1. Fondement algorithmique de l'approche proposée FCA-Cknn

Etape 1 : Modélisation de Réseau routier : Graphe {nœuds « N » + arcs « E »} / données statiques (les nœuds et les arcs) + données dynamiques (ensemble d'objets mobiles = points requêtes et points d'intérêts).

Etape 2 : À partir d'un ensemble de capteurs, associés au réseau routier étudié, les caractéristiques des objets mobiles seront livrées en temps réel à notre système. A partir du graphe et des capteurs dédiés on vise à générer les concepts formels associés, binarisation des caractéristiques des objets mobiles et la génération de la matrice des concepts formels. Soient $O_i \subseteq O$ et $A_i \subseteq A$, on définit « f » et « g » comme suit: Intension. $f: P(O) \rightarrow P(A)$ $f(O_i) = \{ a \in A / (o,a) \in I, \forall o \in O_i \}$, Extension. $g: P(A) \rightarrow P(O)$ $g(A_i) = \{ o \in O / (o,a) \in I, \forall a \in A_i \}$

Etape 3 : Extraction des concepts formels : Les points d'intérêt candidats qui vont satisfaire la requête de l'utilisateur. Soient $O_i \subseteq O$ et $A_i \subseteq A$, (O_i, A_i) est un concept si et seulement si O_i est l'extension de A_i et A_i est l'intension de O_i . Soit l'ensemble des concepts $L = \{(O_i, A_i) \in P(O) \times P(A) / O_i = g(A_i) \text{ et } A_i = f(O_i)\}$. Relation d'ordre (\leq) sur L : Sous-concept / Sur-concept (spécialisation / généralisation). $(O_1, A_1) \leq (O_2, A_2)$ si et seulement si $O_1 \subseteq O_2$ (ou $A_1 \supseteq A_2$).

Algorithme 2. FCA-CKNN (in $(T = (L, \leq), i)$, out $(TAB-Res [i])$)

Input: $T = (L, \leq)$, int $i = 0$

Output: $TAB-Res [i]$

```

1:   pour tout Concept  $C_i \in T = (L, \leq)$  faire
2:     pour tout Extension  $\in C_i$  faire
3:       si  $(Statut()) = \text{"Near"}$  alors
4:         Add Concept  $C_i$  to  $TAB-Res [i]$ 

```

```

5:            $i = i+1$ 
6:           fin condition
7:       fin boucle
8:       fin boucle
9:       Return TAB-Res []
10:      fin

```

Etape 4 : Génération de treillis (technique de regroupement). Treillis de Galois : Chaque couple d'éléments possède son « supremum » et son « infimum ». $T = (L, \leq)$, ensemble des concepts muni de la relation d'ordre.

Etape 5 : L'ajout des niveaux aux concepts générés.

Algorithme 3. Level (in $(T = (L, \leq))$, out $(T\text{-Level} [])$)

```

Input:  $T = (L, \leq)$ 
Output:  $T\text{-Level} []$ 
1:   pour tout Concept  $C_i \in T = (L, \leq)$  faire
2:        $N \leftarrow \text{size of } (C_i.\text{Intension})$ 
3:       Add  $C_i$  to  $T\text{-Level} [N]$ 
4:   fin boucle
5:   Return  $T\text{-Level} []$ 
6:   fin

```

Etape 6 : Indexation des résultats avec R-Tree → Tableau d'index (Level + concepts).

4.2. Génération des concepts formels (étape 2 et étape 3)

Au moment du lancement de la requête, on suppose que tous les arcs du graphe ne présentent aucun facteur de perturbation, qui pourrait ralentir l'avancement des objets mobiles. Les valeurs de cette caractéristique sont négligeables ($F_p = 0$) à l'instant « $T=0$ ».

Input du système (voir Figure 1)

- $O1 = Pr (PPr, Vr=60, E)$
- A déterminer si les objets en question sont aux portés du point requête Pr ou non. Résultats de la fonction Statut (\cdot): Statut (Pr, $O2$) = « Far », Statut (Pr, $O3$) = « Near », Statut (Pr, $O4$) = « Near ».
- Les points d'intérêt $O2$, $O3$ et $O4$: $O2 (P2, V1=80, SD, Far)$, $O3 (P3, V2=40, SD, Near)$, $O4 (P4, V3=30, SD, Near)$

Définition 7 (Relation Binaire) : Soit E un ensemble. Une relation binaire R sur E est un sous-ensemble de $E \times E$. On note xRy pour signifier que $(x, y) \in R$

Soit le contexte formel $K=(S, P, I)$. « S » désigne l'ensemble des objets mobiles, « P » désigne l'ensemble des propriétés de ces objets (Position, Vitesse, Direction, Statut et Facteur de perturbation) et « I » est une relation binaire entre « S » et « P » vérifiant : $I \subseteq S \times P$ et $(s, p) \in I$ avec $s \in S$ et $p \in P$. La relation $(s, p) \in I$ signifie que l'objet s possède l'attribut ou la propriété p . Le contexte formel « K » sera présenté par un tableau de sources « Objets Mobile X Propriétés » où toute case de la matrice contiendra une relation binaire : « 1 » si l'évènement est présent « 0 » sinon. Selon que le couple (S, P) de $S \times P$ appartient à I ou non. L'ensemble « S »

contient tous les objets mobiles figurant dans le graphe étudié et on retiendra dans l'ensemble « P » les propriétés pouvant servir comme critères pour livrer des réponses valides aux requêtes des utilisateurs. Le Tableau1 présente la matrice des concepts formels « Objets Mobile X Propriétés ». Soient les vitesses $V = \{V1= 60, V2= 80, V3=30, V4 = 40\}$, Position $P = \{P1= [N1, N2], P2= [N2, N5], P3= [N3, N6], P4= [N1, N8]\}$, les directions N: North, S: South, E: East, W: West. Au moment de l'envoi d'une requête, le système détermine les résultants des différentes fonctions par rapport au point requête (Pr).

Tableau 1 Contexte formel au moment de l'envoi de requête

	Caractéristiques OM												Propriétés Calculables				Edge
	Vitesse				Position				Direction				N-Direction		Statut		Fp[P(OM)]
	V1	V2	V3	V4	P1	P2	P3	P4	N	S	E	W	SD	RD	Far	Near	Fp
O1(Pr)	X	0	0	0	X	0	0	0	0	0	X	0	X	0	0	X	0
O2	0	X	0	0	0	X	0	0	0	0	X	0	X	0	X	0	0
O3	0	0	X	0	0	0	X	0	0	0	X	0	X	0	0	X	0
O4	0	0	0	X	0	0	0	X	0	0	X	0	X	0	0	X	0

Définition 8 N-Direction (Pr, Pi) : Une fonction qui permet de comparer les directions entre le point d'intérêt (Pi) et le point requête (Pr). N-Direction retourne « SD » si les deux objets mobiles Pi et Pr ont les mêmes directions, et retourne « RD » dans le cas contraire.

Algorithme 4. N-Direction (in (Pi, Pr, D (Pi), D (Pr)), out (“SD”, “RD”))

Input: Pi, Pr, D (Pi), D (Pr)

Output: “SD”, “RD”

```

1:   pour tout Concept Ci ∈ T = (L, ≤) faire
2:       si (D (Pi) == D (Pr)) alors
3:           return “SD”
4:       si non
5:           return “RD”
6:       fin condition
7:   fin boucle
8:   fin

```

A ce stade, notre système doit assurer la génération du Treillis après l'intégration des propriétés calculables à savoir N-Direction () et Statut (). Ensuite il faut étudier pour chaque point d'intérêt suivant la position de l'objet mobile en question s'il est disposé à un facteur de perturbation ou non.

La dernière colonne du contexte formel généré Fp[P(OM)] détermine tout d'abord P(OM) qui présente la position de l'objet mobile en question. OM est situé éventuellement sur un arc bien déterminé. Si l'arc P(OM) présente au moment de l'interrogation l'un des facteurs qui peuvent freiner l'avancement de l'objet, Fp[P(OM)] est évaluée à « vraie » (« faux » dans le cas échéant). Dans cette version

Copyright © by the paper's authors. Copying permitted for private and academic purposes. Proceedings of the Spatial Analysis and GEOmatics conference, SAGEO 2015.

du papier, nous ne tenons pas compte des facteurs de perturbation et nous laissons ce critère à des travaux futurs.

On pourrait bien remarquer que les objets qui ont plus de chance à satisfaire les besoins de l'utilisateur sont ceux qui ont le Statut « Prés » par rapport au point requête. Les deux objets O3 et O4 peuvent garantir cette condition avec les deux caractéristiques « SD » et « Near » pour dire même direction que le point requête et près du point requête.

Dans notre contexte, on cherche à maximiser le gain par rapport à un point requête c'est à dire que les points d'intérêt doivent garantir certaines caractéristiques pour satisfaire la requête de l'utilisateur.

A l'instant « T=0 », une réponse valide = {O3, O4}, ce qui est bien exprimé par la matrice des concepts formels :

- L'objet O2 admet un Statut () = « Far » car sa vitesse (V2) est plus grande que la vitesse du point requête (V1) (la propriété SD = 1 pourrait être interprétée dans ce cas comme suit : le point Pr ne pourrait jamais atteindre le point Pi =O2).
- L'objet O3 admet un Statut () = « Near » car sa vitesse est inférieure à la vitesse du point requête (la propriété SD = 1 pourrait être interprétée dans ce cas comme suit : le point Pr pourrait atteindre le point Pi =O3).
- L'objet O4 admet un Statut () = « Near » sa vitesse est inférieure à la vitesse du point requête (la propriété SD = 1 pourrait être interprétée dans ce cas comme suit : le point Pr pourrait atteindre le point Pi =O4).

4.3. Treillis de Galois (étape 4)

Les nœuds du treillis sont structurés sous forme de concepts. Un concept peut être vu comme un ensemble d'objets (l'extension) caractérisé par un ensemble de propriétés (l'intension). Pour la recherche des K plus proches voisins, les concepts du treillis fournissent un regroupement des objets mobiles dans des nœuds homogènes qui ont chacun un ensemble de propriétés en commun. Connaissant les critères que doit satisfaire les objets mobiles cherchés, on pourrait facilement identifier le ou les nœuds qui vérifient ces critères et avoir donc des réponses valides. Dans un treillis de Galois, les concepts sont ordonnés selon des critères liés au nombre des objets mobiles figurant dans l'extension de chaque concept et au nombre des propriétés communes entre ces objets présents dans les intensions des concepts. Les concepts les plus généraux sont situés en haut du treillis alors que les concepts les plus spécifiques sont situés en bas du treillis. Les liens entre les différents niveaux du treillis peuvent être interprétés comme des généralisations ou des spécialisations entre les niveaux représentés par les concepts. En effet, un parcours ascendant des niveaux d'un treillis se traduit à chaque niveau par la diminution progressive du nombre des propriétés dans les intensions des concepts et l'augmentation progressive du nombre d'objets mobiles dans leurs extensions. Cela correspond au passage d'un niveau plus spécifique, qui contient peu d'objets mobiles qui vérifient plusieurs critères, à un niveau plus général, qui contient plus d'objets mobiles qui ne vérifient qu'une partie des propriétés en commun d'un niveau spécifique. Un parcours descendant des niveaux d'un treillis correspond au

passage d'un niveau général à un niveau plus spécifique. Le diagramme de Hasse (Figure 2), facilite la compréhension et l'interprétation de la relation entre les objets mobiles et leurs propriétés. L'avantage de cette représentation est qu'à partir d'un treillis de concepts il est toujours possible de retrouver le contexte formel correspondant et inversement.

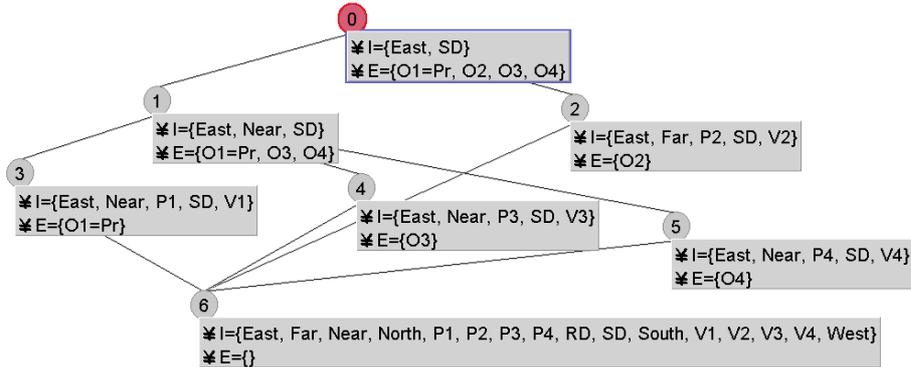


Figure 2 Diagramme de Hasse (Grphe du Treillis de Gallois)

Le parcours du treillis permet de livrer les K plus proches voisins qui représentent l'ensemble des K réponse valide à un instant « T = 0 ».

Définition 9 (Niveau d'un nœud) : Le niveau d'un nœud correspond au nombre des propriétés qui figurent à l'intension « I ». Un nœud « n » admet le niveau « m » si et seulement si l'intension « I » du nœud « n » contient « m » caractéristiques. Soit Nœud « n » avec l'intension $I = \{C1, C2, \dots, Cm\}$ tel que « C1 » caractéristique 1, « C2 » caractéristique 2 et « Cm » Caractéristique m. « I » contient « m » caractéristiques alors Niveau du nœud n = « m ».

Illustration 3 Regroupement par niveau. Le treillis généré contient 4 niveaux (Figure 3) . Chaque niveau contient au moins un nœud à l'exception du premier et du dernier nœud qui peuvent ne pas contenir des nœuds : Niveau 2 = {Nœud 0}, Niveau 3 = {Nœud 1}, Niveau 5 = {Nœud 2, Nœud 3, Nœud 4, Nœud 5}, Niveau 16 = {}.

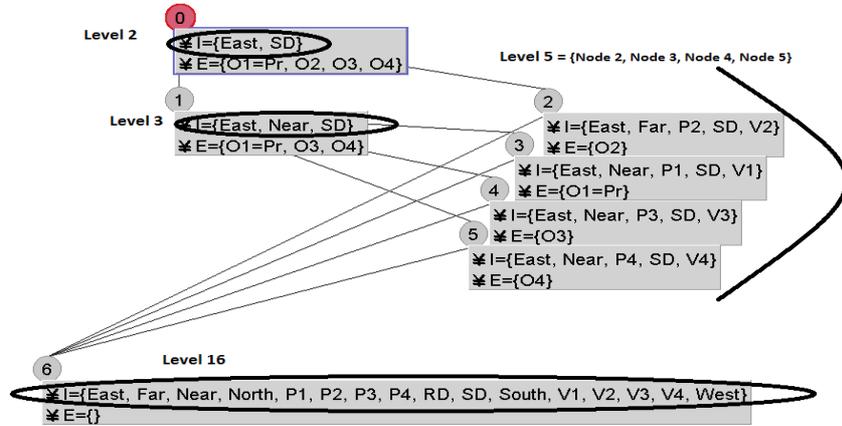


Figure 3 : FCA et niveaux

4.4. Raffinement des résultats (étape 5 et étape 6)

Le treillis de Galois correspondant aux points d'intérêts candidats, présente 9 nœuds classifié selon 4 niveaux (Figure 3). On cherche à indexer les niveaux qui peuvent contenir éventuellement des réponses valides, en fonction des nœuds de chaque niveau.

Tableau 2 Indexe Niveau / nœuds

Niveau n	Niveau 2	Niveau 3	Niveau 5	Niveau 16
{ } Nœuds	Nœud 0	Nœud 1	Nœud 2, Nœud 3, Nœud 4, Nœud 5	{ }

Synthèse : « Un usager dans sa voiture cherche les centres d'urgence mobile les plus proches de sa position tout au long de son trajet ».

Usager Pr (P, V, D) (Pr= O1). Centres d'urgence mobile Pi (Pi, Vi, Di, Si) (Pi = {O2, O3, O4}). Les Pi doivent tenir compte du contexte de Pr et doivent satisfaire deux conditions primordiales.

- Les Vitesses de Pr et des Pi doivent être plus au moins proches l'une de l'autre :
 - Si les objets ont les mêmes directions, pour que l'objet mobile Pr puisse atteindre les objets mobiles Pi, il faut que la vitesse de Pr soit supérieure aux vitesses des Pi.
 - Si la direction de l'objet mobile Pr est opposée aux directions des objets mobiles Pi, il y aura éventuellement un point d'intersection entre ces derniers s'ils vont passer par un même chemin. Pas de contrainte dans ce cas sur les vitesses de Pr et des Pi.

- Si Statut (Pr, Pi) = Prés, alors ajouter Pi à la liste des points d'intérêts valides.

Les K plus proches centres d'urgence mobiles recherchés admettent au moins trois caractéristiques en communs : Statut (Pr, Pi) = « Prés » (1) + Direction (Pr, Pi) = SD (2) + V-Pr Supérieure à V-Pi (3) Ou bien Statut (Pr, Pi) = Prés (1) + Direction (Pr, Pi) = RD (2) + V-Pr Inférieure à V-Pi (3).

On s'intéresse aux différents nœuds du Treillis de Galois qui présentent dans leurs intensions « I » la propriété «Near». Le tableau d'indexe accélère la recherche et donne satisfaction à l'utilisateur en un temps optimale. Dans notre cas les réponses valides appartiennent au Niveau 5 et au Niveau 3.

Nœud 1(Niveau 3 les trois objets O1=Pr, O3 et O4 ont les caractéristiques Near, SD et East en communs) : $I = \{\text{East, SD, Near}\}$, $E = \{\text{O4, O1=Pr, O3}\}$ // Réponses valides. Nœud 3 (Niveau 5, l'objet O1 admet tout seul les cinq caractéristiques mentionnées) : $I = \{\text{East, Near, P1, SD, V1}\}$, $E = \{\text{O1=Pr}\}$ // Réponses valides. Nœud 4 (Niveau 5, l'objet O3 admet tout seul les cinq caractéristiques mentionnées) : $I = \{\text{East, Near, P3, SD, V3}\}$, $E = \{\text{O3}\}$ // Réponses valides. Nœud 5 (Niveau 5, l'objet O4 admet tout seul les caractéristiques mentionnées) : $I = \{\text{East, Near, P4, SD, V4}\}$, $E = \{\text{O4}\}$ // Réponses valides. Finalement, les réponses valides après élimination des doublant sont : $\{\text{O1=Pr, O3, O4}\}$.

5. Performance et avantage de FCA-Cknn

Préserver la pertinence des résultats revient en premier lieu à réduire le temps de réponse puisque les résultats doivent être valides au moment de leur réception. Afin d'atteindre nos objectifs, l'approche proposée vise à réduire l'espace de recherche. L'utilisation de l'FCA pour la recherche des k plus proches voisins est motivée par deux principales caractéristiques : la structuration conceptuelle des données du treillis et l'ordre hiérarchique entre les concepts. En plus du temps de calcul pris par la génération de treillis, la complexité de construction du treillis de points d'intérêts dépend du nombre d'objets mobiles ainsi que de leurs propriétés. La construction des concepts est d'ordre $O(k.m)$, ou k est le nombre de propriétés que peut avoir un point d'intérêt, m est le nombre de points d'intérêts. Le parcours du treillis généré prend un temps d'ordre $O(L)$ ou L est le nombre de concepts d'objets mobiles.

6. Conclusion et travaux futurs

Dans ce papier, nous avons proposé une approche de recherche des k-plus proches voisin en continu en appliquant l'analyse formelle de concepts pour le regroupement des objets mobiles. Cette méthode consiste à créer, à partir d'un graphe, un contexte formel qui servira à la construction du Treillis de Galois représentant la hiérarchie des caractéristiques des objets mobiles. Une fois le treillis construit, l'étape de recherche de points d'intérêt peut être effectuée grâce au mode de classification offert par les treillis de Galois. La méthode permet d'avoir que des points d'intérêt valides ayant au moins l'une des propriétés demandées par l'utilisateur. Le fondement mathématique de l'FCA permet de garantir l'exactitude des réponses délivrées. Dans les futurs travaux, nous comptons prendre en

Copyright © by the paper's authors. Copying permitted for private and academic purposes. Proceedings of the Spatial Analysis and GEOmatics conference, SAGEO 2015.

considération la nature dynamique du réseau routier et les requêtes en présence de facteurs de perturbation dans le réseau.

Bibliographie

- Azmeh, Z., Huchard, M., Tibermacine, C., Urtado, C., & Vauttier, S. (2008, November). Wspab: A tool for automatic classification & selection of web services using formal concept analysis. *In on Web Services, 2008. ECOWS'08. IEEE Sixth European Conference (pp. 31-40). IEEE.*
- Barbut M. & Monjardet B. (1970). *Ordre et Classification, Algèbre et Combinatoire. Hachette, Paris*
- Beckmann, N, Kriegel, H.P, Schneider, R. and Seeger, B. (1990) An efficient and robust access method for points and rectangles, *In Proc.ACMSIGMOD*, Atlantic City, NJ, USA, pp.47–57, 1990.
- Bendaoud, R., Toussaint, Y., & Napoli, A. (2010). L'analyse Formelle de Concepts au service de la construction et l'enrichissement d'une ontologie. *REVUE DES NOUVELLES TECHNOLOGIES DE L'INFORMATION*, 133-164.
- Berchtold, S., Böhm, C., Keim, D. A., & Kriegel, H. P. (1997, May). A cost model for nearest neighbor search in high-dimensional data space. *In Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (pp. 78-86). ACM.*
- Carpineto, C., & Romano, G. (2005). Using concept lattices for text retrieval and mining. *In Formal Concept Analysis (pp. 161-179). Springer Berlin Heidelberg.*
- Feng, F. and Watanabe, T. (2004) Search of continuous nearest target Objects along route on large hierarchical road network, *Proc. 14th Data Engineering Workshop (DEWS)*, Kaga city, Ishikawa, Japan, pp.145–151.
- Ferchichi, H., & Akaichi, J. (2013, May). A novel approach for the continuous search of the K nearest neighbors on road networks. *In Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference on(pp. 878-881). IEEE.*
- Ganter B. & Wille R. (1999). *Formal Concept Analysis. Springer, mathematical foundations edition*
- Godin, R., Missaoui, R., & Alaoui, H. (1995). Incremental concept formation algorithms based on Galois (concept) lattices. *Computational intelligence*, 11(2), 246-267.
- Godin, R., & Valtchev, P. (2005). Formal concept analysis-based class hierarchy design in object-oriented software development. *In Formal Concept Analysis (pp. 304-323). Springer Berlin Heidelberg.*
- Guttman, A. (1984) 'R-Trees: A Dynamic Index Structure For Spatial Searching', *In Proc. ACM SIGMOD*, Boston, Massachusetts, USA.
- Khayati, M., Akaichi, J. (2008). Incremental Approach for Continuous k-Nearest Neighbors Queries on Road, *Int. J. Intelligent Information and Database Systems, Inderscience Publishers.*

Copyright © by the paper's authors. Copying permitted for private and academic purposes. Proceedings of the Spatial Analysis and GEomatics conference, SAGEO 2015.

- Lakhal, L., & Stumme, G. (2005). Efficient mining of association rules based on formal concept analysis. In *Formal concept analysis* (pp. 180-195). Springer Berlin Heidelberg.
- Priss, U. (2005). Linguistic applications of formal concept analysis. In *Formal Concept Analysis* (pp. 149-160). *Springer Berlin Heidelberg*.
- R. Zhong, G. Li, K.-L. Tan, and L. Zhou. (2013) G-tree: an efficient index for knn search on road networks. In *CIKM*, pages 39–48.
- Song, Z and Roussopoulos. N, (2001). K-Nearest Neighbor Search for moving query Point, In *SSTD*, Heidelberg, Berlin.
- Tao. Y and Papadias. D, (2002) Time-Parameterized Queries in Spatio-Temporal Databases. In *Proc ACM SIGMOD*, Symp. on the Management of Data, pages 334-345.
- Tao. Y and Papadias. D, (2003). Time-Parameterized Queries in Spatio-Temporal Databases, In *ACM SIGMOD*, Midison, USA.
- Terry. D, Goldberg. D, Nichols. D, and Oki. B, (1992) Continuous Queries over Append-Only Databases. In *Proc. ACM SIGMOD Symp. on the Management of Data*.
- Tilley, T., Cole, R., Becker, P., & Eklund, P. (2005). A survey of formal concept analysis support for software engineering activities. In *Formal concept analysis* (pp. 250-271). *Springer Berlin Heidelberg*.