HAL open science

Génération aléatoire d'un graphe spatio-temporel localement cohérent

Aurélie Leborgne, Marija Kirandjiska, Florence Le Ber

To cite this version:

Aurélie Leborgne, Marija Kirandjiska, Florence Le Ber. Génération aléatoire d'un graphe spatio-temporel localement cohérent. CNIA 2021 : Conférence Nationale en Intelligence Artificielle, Jun 2021, Bordeaux (en ligne), France. pp 76-83. hal-03321317

HAL Id: hal-03321317

https://hal.science/hal-03321317v1

Submitted on 17 Aug 2021

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

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

Génération aléatoire d'un graphe spatio-temporel localement cohérent

Aurélie Leborgne, Marija Kirandjiska, Florence Le Ber

Université de Strasbourg, CNRS, ENGEES, ICube UMR 7357, F-67000 Strasbourg
aurelie.leborgne@unistra.fr, florence.leber@engees.unistra.fr

Résumé

Dans cet article, nous présentons une approche pour générer des graphes spatio-temporels localement cohérents et incluant des motifs fréquents inexacts. Cette approche s'appuie sur un travail précédent, qui consistait à développer un générateur paramétrable de graphes spatio-temporels. Dans ces graphes, les arêtes spatiales et spatio-temporelles sont étiquetées avec les relations spatiales de la théorie RCC8. Pour vérifier la cohérence des relations au cours de la génération du graphe, nous utilisons la méthode de la chemin-cohérence, qui s'appuie sur la faible composition des relations. L'approche est décrite et des expérimentations sont présentées. L'objectif final de notre travail est de disposer d'un générateur pour tester les méthodes de fouille de graphes.

Mots-clés

Graphe spatio-temporel, RCC8, cohérence, génération de graphes.

Abstract

This paper presents an original approach for the generation of locally coherent spatio-temporal graphs embedding frequent inexact patterns. This approach relies on a previous work, that have implemented a configurable generator for spatio-temporal graphs. These graphs contains spatial an spatio-temporal edges that are labelled with the RCC8 topological relations. In order to check the consistency of these relations when building the graph, the path-consistency method, based on relation weak composition, was implemented within the graph generator. The approach is described and some experiments are detailed. Our final aim is to build a test generator for graph mining methods.

Keywords

Spatio-temporal graph, RCC8, consistency, graph generation.

1 Introduction

L'amélioration régulière des outils et techniques de recueil des données amènent de plus en plus souvent à modéliser et analyser des données qui ont une dimension spatiale mais également temporelle. Une manière naturelle de modéliser de telles données est d'utiliser les graphes spatio-temporels (graphes ST), qui permettent de représenter différents phénomènes naturels (évolutions des occupations du sol dans un territoire [11], le déplacement de dunes [7], *etc.*) ou biologiques (évolution de la connectivité cérébrale [10], *etc.*). Une manière d'analyser les données recueillies est de s'intéresser aux phénomènes récurrents dans le temps et/ou l'espace (ensembles d'entités ayant des relations spatiales et évoluant les unes par rapport aux autres dans le temps). Il faut alors mettre en évidence les régularités dans les phénomènes spatio-temporels étudiés pour faciliter la compréhension des experts des différents domaines d'application. À titre d'exemple, en urbanisme, un même schéma d'artificialisation se répète selon des temporalités variées : à court terme, une zone végétalisée (forêt, parcelle agricole, etc.) est rasée progressivement, remplacée par du sol nu, puis des infrastructures apparaissent (routes, ponts, etc. ainsi que des maisons individuelles. La figure 1 repésente un exemple d'une telle situation. À plus long terme, cette zone se restructure : les maisons devenues anciennes disparaissent au profit d'immeubles collectifs. Cette évolution peut être alors modélisée par un motif spatio-temporel caractéristique d'une urbanisation.

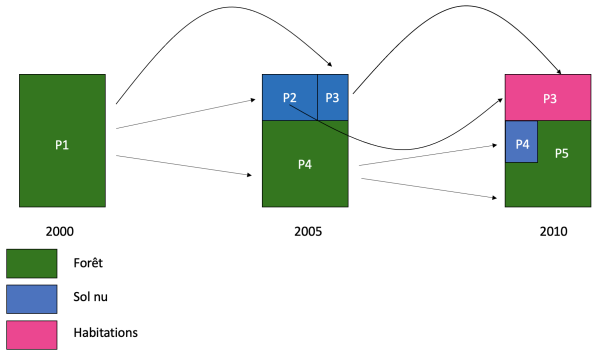


FIGURE 1 – Evolution des occupations du sol d'un territoire

Pour réaliser une telle extraction, il est nécessaire de mettre au point des algorithmes de recherche de motifs/sous-graphes fréquents dans des graphes ST. Cependant, pour développer de tels algorithmes, il est indispensable d'avoir une base de test de graphes ST annotés. Malheureusement, l'annotation de telles données est un travail très fastidieux et même impossible à réaliser précisément au vu de la quantité de données que nous manipulons. La seule solution per-

mettant d'obtenir une base de test est donc de développer un générateur de graphes spatio-temporels dans lesquels nous maitrisons les motifs fréquents présents dans ces graphes ST.
La génération de tels graphes a été décrite dans [12]. Dans cet article, nous nous intéressons à la manière de gérer la cohérence des graphes pour l'ensemble des relations considérées, à savoir, les relations spatiales qualitatives de la théorie RCC8 [14]. Ces relations permettent notamment de modéliser les évolutions des occupations du sol sur un territoire, comme dans l'exemple présenté ci-dessus.
L'article est organisé comme suit. La section 2 décrit les éléments théoriques sur lesquels notre approche est fondée, à savoir le modèle de graphes spatio-temporels, les relations RCC8 puis les réseaux de contraintes qualitatives. Les sections 3 et 4 présentent l'algorithme de génération puis les expérimentations menées. La section conclusive dresse quelques perspectives de ce travail.

2 Préliminaires

2.1 Simulation de graphes

La génération de graphes est une question importante dans de nombreux domaines, pour simuler des graphes réels, tester des algorithmes ou des applications permettant d'analyser, visualiser ou transformer des données [2]. Dans la plupart des cas, le but est de générer des graphes réalistes. De nombreux modèles ont été présentés en ce sens pour la génération de graphes complexes adaptés à la représentation de systèmes naturels ou humains (web sémantique, réseaux sociaux). Le modèle Barabási-Albert [8, 3] est un des plus connus. Ces approches sont fondées sur des distributions statistiques de propriétés des graphes (nombre de sommets, arêtes, degré des sommets, *etc.*). La question de la cohérence n'est pas traitée, dans la mesure où la sémantique des relations est peu prise en compte dans ces approches.
Dans [12], nous avons développé une approche permettant de générer, de manière aléatoire, des graphes sémantico-temporels, où les arêtes sont dotées d'une sémantique. Nous introduisons ci-dessous le modèle d'un tel graphe, inspiré de [7]. Nous utilisons pour exemple le graphe (figure 2) associé au schéma d'évolution de la figure 1. Il s'agit ici d'un graphe spatio-temporel, défini comme l'union de trois sous-graphes :

— **Le graphe des relations spatiales**, qui caractérise spatialement les interactions entre entités (dans l'exemple de la figure 1, les occupations du sol d'un territoire) à un moment donné. Il est composé des nœuds (en noir), et des arêtes pleines verticales (en vert) sur la figure 2.
— **Le graphe des relations spatio-temporelles**, qui se base sur les mêmes caractéristiques que le graphe des relations spatiales, mais en considérant des entités à des temps différents. Il est composé des nœuds (en noir), et des arêtes pleines (en rouge) sur la figure 2.
— **Le graphe des relations de filiation**, défini sur le concept d'identité. Il permet de caractériser la transmission de l'identité des entités à travers le temps. Il est composé des nœuds (en noir), et des arêtes pointillées (en bleu) sur la figure 2.

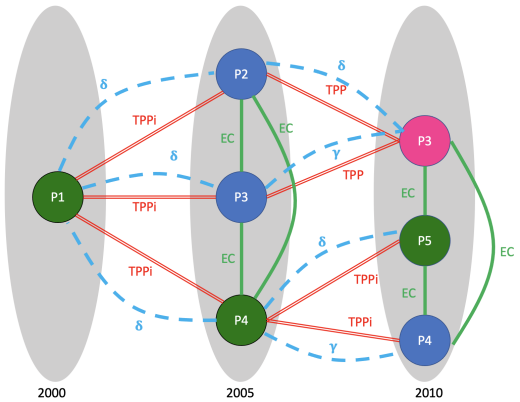


FIGURE 2 – Graphe ST modélisant l'évolution du territoire de la figure 1

De manière formelle, un graphe ST est défini ainsi. Soit $\mathcal{T} = \{t_1, t_2, ...t_n\}$, un domaine temporel, où t_i représente une instance de temps d'une granularité donnée et $t_i < t_{i+1}$ pour tout $i \in [1, n]$. Soit Δ un ensemble d'entités, $\{e_1, e_2, ..., e_m\}$. Soit également Σ, un ensemble de relations spatiales, et Φ, un ensemble de relations de filiation. Un graphe spatio-temporel $\mathcal{G}$ est un tuple (U, E_Σ, E_Φ), où U est l'ensemble des sommets $(e_l, t_i) \in \Delta \times \mathcal{T}$, E_Σ est l'ensemble des tuples $((e_i, t_i)T(e_j, t_j))$ où $(e_i, t_i), (e_j, t_j) \in U$, $t_i \leq t_j \leq t_{i+1}$, et $T \in \Sigma$, et E_Φ est l'ensemble des tuples $((e_i, t_i)\rho(e_j, t_{i+1}))$ où $(e_i, t_i), (e_j, t_{i+1}) \in U$, et $\rho \in \Phi$.
A l'origine, ce modèle de graphe a été introduit pour représenter l'évolution d'entités géographiques [7]. D'autres relations de filiation sont étudiées dans [6] ; parallèlement un algorithme a été proposé pour vérifier la cohérence de cette représentation avec les caractéristiques des données issues d'une base de données spatio-temporelles.

2.2 Relations RCC8

Les relations spatiales et spatio-temporelles que nous utilisons sont les relations de base $\mathbf{B}$ de la théorie RCC8 [14] sur le domaine spatial Δ. Ces relations définissent la position de deux régions : $DC(x, y)$ les régions x et y sont déconnectées ; $EC(x, y)$ elles sont connectées exterieurement ; $PO(x, y)$ elles se recouvrent partiellement ; $TPP(x, y)$ x est partie propre tangentielle de y ; $TPP_i(x, y)$ y est partie propre tangentielle de x ; $NTPP(x, y)$ x est partie propre non tangentielle de y ; $NTPP_i(x, y)$ y est partie propre non tangentielle de x ; $EQ(x, y)$ égalité de x et y (voir figure 3).
L'ensemble $2^{\mathbf{B}}$ représente l'ensemble de relations construit à partir des relations de base. Il est muni des opérations ensemblistes usuelles, l'union et l'intersection, de l'opération inverse et de la composition faible. Une relation de $2^{\mathbf{B}}$ s'écrit donc comme une union de relations de base, par exemple $R = \{DC, EC\}$ et s'interprète comme une

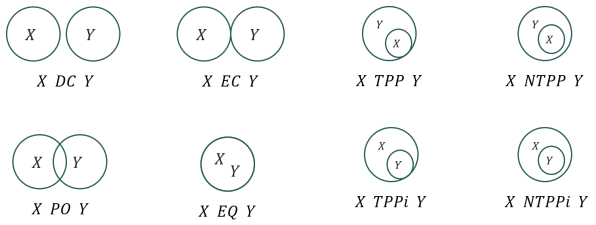


FIGURE 3 – Les relations de base de la théorie RCC8

disjonction. L'inverse (noté $\smile$) d'une relation est l'union des inverses de ses relations de base. La composition faible est notée $\diamond$ et définie ainsi : soient R et S deux relations de $2^{\mathbf{B}}$, $R \diamond S = \{b \in \mathbf{B} | b \cap (R \circ S) \neq \emptyset\}$ où $R \circ S = \{(x,z) \in \Delta^2 | \exists y \in \Delta, (x,y) \in R \,\mathtt{et}\, (y,z) \in S\}$. La composition faible des relations de base est représentée dans une table de composition [13], comme le montre la figure 7. Par exemple, supposons que trois régions x, y, z ont les relations $TPP(x,y)$ et $EC(y,z)$, on vérifie alors que $\{DC, EC\}(x,z)$ (voir figure 4).

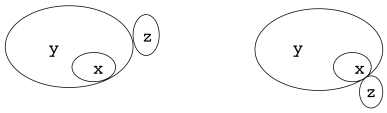


FIGURE 4 – Deux configurations possibles pour x et z connaissant $TPP(x,y)$ et $EC(y,z)$

2.3 Cohérence

La notion de cohérence d'un ensemble de relations spatiales ou temporelles reliant des régions a été étudiée dans le cadre des réseaux de contraintes qualitatives [4, 5]. Un réseau de contraintes qualitatives est un couple $N = (V, C)$ où V est un ensemble de variables sur un domaine continu $\mathcal{D}$ et C une application qui associe à chaque couple de variables (V_i, V_j) un ensemble C_{ij} de relations de base $\{r_1, ..., r_l\}$ prises dans une algèbre de relations. Cet ensemble représente la disjonction des relations possibles entre les deux entités représentées par les variables V_i et V_j. Une instanciation cohérente de N est une instanciation où chaque variable V_i prend une valeur $e_i \in \mathcal{D}$ telle que pour tout couple (V_i, V_j) la relation atomique vérifiée par les variables V_i et V_j appartienne à C_{ij}.

La vérification de la cohérence d'un réseau est un problème NP-complet dans le cas général [17]. Des méthodes locales ont été proposées pour vérifier des formes plus faibles de cohérence, dont la chemin-cohérence : un réseau de contraintes qualitatives N est dit chemin cohérent si pour toutes les variables $V_i, V_j, V_k \in V$, $C_{ij} \subseteq C_{ik} \circ C_{kj}$ [5].

La méthode de la chemin-cohérence consiste à réaliser l'opération de triangulation : $C_{ij} = C_{ij} \cap (C_{ik} \circ C_{kj})$ pour tout triplet jusqu'à obtention d'un point fixe. Le réseau final est chemin-cohérent et équivalent au réseau initial. Dans le cadre de RCC8, où on ne peut pas utiliser la composition, mais la composition faible $\diamond$, on parle de fermeture algébrique [15].

Différents algorithmes ont été proposés pour vérifier la chemin-cohérence. Leur temps de calcul est lié au nombre d'accès à la table de composition et donc à la taille de l'ensemble de relations $2^{\mathbf{B}}$ [1]. Dans [16] est présenté un algorithme permettant de vérifier de manière incrémentale la cohérence d'un réseau de contraintes qualitatives, qui croît par ajout d'entités spatiales ou temporelles ; l'algorithme exploite un graphe triangulé : quand une entité est ajoutée à l'étape t, elle est reliée à toutes les entités présentes à l'étape $t-1$ (triangulation). La méthode de la chemin-cohérence est ensuite appliquée sur le graphe. Notre approche est différente dans la mesure où nous cherchons à générer un graphe où chaque relation est atomique, et où il s'agit d'affecter à l'arête courante une relation qui soit cohérente avec l'existant, comme nous le détaillons ci-après.

3 Génération de graphes ST localement cohérents

Nous présentons ici une approche pour générer des graphes spatio-temporels localement cohérents et incluant des motifs (sous-graphes) fréquents inexacts. Nous nous appuyons pour cela sur la méthode présentée dans [12], qui permet de générer de tels graphes, incluant des motifs, mais sans traiter la question de la cohérence.

Nous décrivons ici succinctement l'algorithme de génération des graphes spatio-temporels et des motifs qui y sont intégrés, puis détaillons et justifions l'algorithme permettant de vérifier la cohérence locale des relations spatiales et spatio-temporelles générées. Notons que ces deux types de relations sont issues de l'ensemble $2^{\mathbf{B}}$ (voir en section 2.2).

L'algorithme décrit dans [12] permet de simuler des graphes spatio-temporels composés en partie de motifs *noyés* dans une génération stochastique uniforme de nœuds et d'arêtes. Il s'agit d'un algorithme entièrement paramétrable, exploitant la loi de Poisson, comme proposé dans [9], et dans lequel il est possible de choisir la taille du graphe généré, le nombre de relations spatiales, temporelles et filiation par nœud, ainsi que la taille des motifs-source à incruster et le nombre de leurs transformations. Dans cet article, nous considérons un algorithme modifié pour contrôler la part des motifs, en nombre de nœuds par rapport au graphe total. Les autres paramètres sont recensés dans le tableau 1.

Cet algorithme se déroule en trois étapes principales. Il calcule au départ le nombre de nœuds total (paramètre λ_n, tableau 1) du graphe à générer.

Étape 1 : génération et transformation de motifs-sources, selon des paramètres propres (voir ci-dessous) ; chaque motif est affecté à une temporalité du graphe, cette information est stockée dans le paramètre $patterns$.

Étape 2 : génération aléatoire des nœuds pour chaque temporalité dans le graphe (paramètres λ_r et $labels_n$). Le nombre de nœuds est ajusté en fonction du nombre de nœuds des motifs affectés à la temporalité courante.

Étape 3 : génération aléatoire des relations entre les nœuds (paramètres Λ_e et $labels_e$). Les nœuds de la temporalité courante sont reliés entre eux et avec les nœuds de la tem-

Paramètre	Description
λ_n	Espérance de la loi de Poisson zéro-tronquée pour le nombre total de nœuds dans le graphe
λ_r	Espérance de la loi de Poisson zéro-tronquée pour le nombre de nœuds par temporalité
Λ_e	Triplet des espérances des lois de Poisson pour le nombre de relations spatiales / spatio-temporelles / de filiation par nœud
$labels_n$	Liste des étiquettes disponibles pour les nœuds
$labels_e$	Tableau de taille 3 de listes d'étiquettes disponibles pour chaque type de relation
$patterns$	Une liste de tuples dont chaque tuple est composé d'un numéro de temporalité et du motif à insérer à cette temporalité

TABLE 1 – Paramètres pour la génération de graphes spatio-temporels

Paramètre	Description
p	Proportion de nœuds dans les motifs / nombre dans le graphe
$pnodes$	Intervalle pour le nombre de nœuds dans un motif-source
λ_r	Espérance de la loi de Poisson zéro-tronquée pour le nombre de nœuds par temporalité du motif-source
Λ_e	Triplet des espérances des lois de Poisson pour le nombre de relations spatiales / spatio-temporelles / de filiation par nœud du motif-source
$support$	Intervalle pour le nombre de répétitions d'un motif-source
λ_t	Espérance de la loi de Poisson pour le nombre de transformations à effectuer sur un motif-source

TABLE 2 – Paramètres pour la génération de motifs

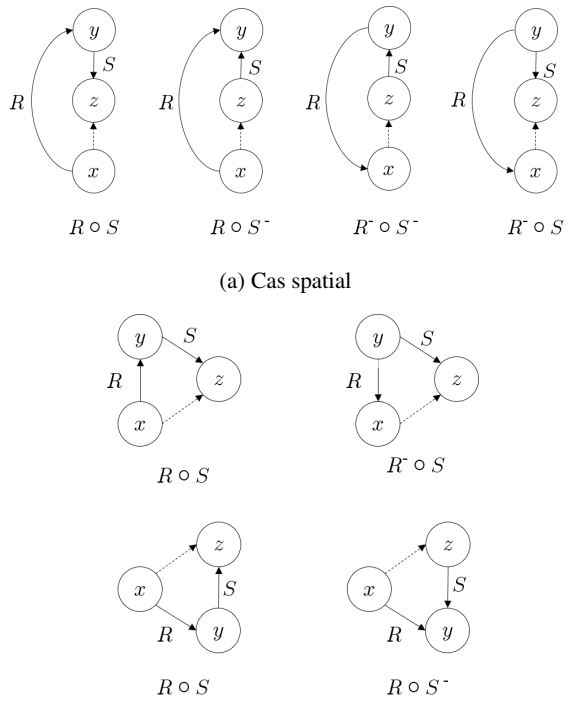
poralité précédente. Le nombre d'arêtes est ajusté pour les nœuds des motifs (s'ils en possèdent déjà). Chaque arête est étiquetée avec une relation atomique.
Lors de la première étape, la génération de motifs obéit aux paramètres suivants (voir tableau 2) : le nombre de motifs insérés dépend d'une proportion p (en pourcentage du nombre de nœuds) que ces motifs doivent représenter dans le graphe total. Le nombre de nœuds dans un motif-source est tiré au hasard dans un intervalle ($pnodes$). Les paramètres λ_r et λ_e ont le même rôle que pour le graphe complet. Chaque motif-source est répété selon une valeur (support) tirée aléatoirement dans l'intervalle $support$. Finalement chaque répétition donne lieu à des transformations (paramètre λ_t) de sorte à introduire une variation autour de chaque motif-source.
La complexité théorique de cet algorithme est en moyenne $O(\lambda_n \times \lambda_r)$, chaque nœud du graphe étant potentiellement relié à tous les nœuds de sa temporalité et de la temporalité précédente. Au pire, quand le nombre de temporalités diminue, la complexité tend vers $O(\lambda_n^2)$ [12].
L'objectif du travail présenté ici est de générer un graphe localement cohérent fondé sur le modèle de cohérence présenté dans la section 2.3. Plus précisément, cela consiste à générer des relations spatiales et spatio-temporelles tout en s'assurant qu'elles forment des triangles spatiaux ou temporels cohérents avec les arêtes préexistantes dans le graphe.

Définition 1 *Un **triangle cohérent** est une clique de 3 sommets dans laquelle les trois relations modélisées par des arêtes sont cohérentes entre elles, soit, pour e_i, e_j, e_k les sommets d'un tel triangle, $R_{ik} \subseteq R_{ij} \diamond R_{jk}$ et $R_{ij} \subseteq R_{ik} \diamond R_{kj}$.*

Définition 2 *Un sous-graphe constitué de trois nœuds x, y, z qui se trouvent dans la même temporalité est appelé un **triangle spatial**. Quatre configurations sont à considérer pour la composition, selon le sens des arêtes xy d'une part et yz d'autre part (figure 5a).*

Définition 3 *Un sous-graphe constitué de trois nœuds x, y, z qui se trouvent dans deux temporalités voisines est appelé **triangle spatio-temporel**. Quatre configurations sont à considérer pour la composition, selon le sens des arêtes xy d'une part et yz d'autre part (figure 5b).*

Le choix de se limiter à une cohérence locale (3-cohérence) est liée à des aspects pratiques : d'une part, les motifs inscrustés sont de petite taille (3-4 nœuds par temporalité, 2 ou 3 temporalités au plus) ; d'autre part, le nombre de relations par nœud est généralement faible (même si nous faisons des expérimentations avec des densités élevées, voir section 4). Finalement, nous cherchons à limiter la complexité de génération des graphes ST.
L'algorithme 1 décrit cette méthode. Une phase d'initialisation (l. 1) permet de créer une liste L avec l'ensemble des relations de **B**. Pour déterminer une relation entre les nœuds x et z, on recherche tous les nœuds y, qui ont une relation à la fois avec x et avec z (l. 2). Les différentes configurations (figure 5) sont étudiées. Pour chaque nœud, la liste est mise à jour en ne conservant que les relations possibles de L (l. 4, 6, 8, 10). Finalement, la relation entre x et z est assignée au hasard parmi les relations de L (l. 14). Si la liste est vide, aucune relation n'est assignée.
La complexité théorique de l'algorithme 1 est $O(\lambda_r)$ puisque, étant donnée la paire (x, z), il parcourt au pire tous les nœuds des temporalités courante et précédente. En revanche, comme les relations sont atomiques, un seul accès à la table de composition est nécessaire pour traiter chaque triangle.
Un exemple de graphe obtenu est présenté en figure 6 : les nœuds des motifs sont désignés par la lettre P, tandis que les nœuds génériques ont seulement un numéro. Des nœuds portant le même numéro portent la même étiquette. Sur cet exemple, on voit à la temporalité t_4 que P2 est relié à P1



(a) Cas spatial

(b) Cas spatio-temporel

FIGURE 5 – Différentes configurations pour la composition des arêtes reliant les nœuds x et y d'une part, et y et z d'autre part, dans un triangle spatial ou spatio-temporel.

par une arête spatiale $NTPP$, P2 est relié à P3 (t_5) par une arête spatio-temporelle DC, P1 et P3 sont reliés à P6 (t_5) respectivement par des arêtes portant les étiquettes PO et EC. Pour relier P1 et P3, l'algorithme doit donc examiner successivement les triangles (P1,P2,P3) et (P1,P6,P3) :

— Pour (P1,P2,P3), on a $NTPP^{\smile} \diamond DC = NTTP_i \diamond DC = \{DC, EC, PO, TPP_i, NTPP_i\}$, comme le montre la case bleue de la table de composition sur la figure 7.

— Pour (P1,P6,P3) on a $PO \diamond EC^{\smile} = PO \diamond EC = \{DC, EC, PO, TPP_i, NTPP_i\}$, comme le montre la case jaune de la table de composition sur la figure 7.

Finalement la relation entre P1 et P3 doit être choisie dans l'ensemble $\{DC, EC, PO, TPP_i, NTPP_i\}$, ici c'est TPP_i qui a été sélectionnée.

4 Expérimentations

Dans cette phase expérimentale nous avons étudié le comportement, en terme de temps de calcul, de l'algorithme de génération en faisant varier les différents paramètres (voir tableaux 1 et 2). Afin d'observer l'influence des différents paramètres sur la complexité de la génération des graphes spatio-temporels cohérents, nous avons fait varier ces paramètres un par un. Les valeurs par défaut des paramètres sont présentées au tableau 3.

Cette batterie de tests a été effectuée sur une machine

Algorithm 1 Génération d'une relation entre deux nœuds avec vérification de la cohérence locale

Input: sommets x, z
Output: relation entre x et z

```
L = B
for each sommet y tel que (x,y) et (y,z) ∈ E_Σ do
    if R(x,y) et S(y,z) then
        L ← L ∩ R ◇ S
    else if R(x,y) et S(z,y) then
        L ← L ∩ R ◇ S⌣
    else if R(y,x) et S(z,y) then
        L ← L ∩ R⌣ ◇ S⌣
    else
        L ← L ∩ R⌣ ◇ S
    end if
end for
if L ≠ ∅ then
    return relation au hasard dans L
else
    return pas de relation
end if
```

Paramètre	Valeur
Génération du graphe	
λ_n	10000
λ_r	100
Λ_e	[5,5,2]
Génération de motifs	
p	30
$pnodes$	[5,15]
λ_r	2
Λ_e	[5,5,2]
$support$	[10,20]
λ_t	taille moyenne des motifs / 2

TABLE 3 – Valeurs par défaut des paramètres lors des expérimentations

Ubuntu 18.04.4 LTS, 32 Go de RAM et 32 cœurs. Cependant l'algorithme n'a utilisé qu'un de ces cœurs.

4.1 Variation du nombre de nœuds

Dans cette première expérience, seul varie le paramètre λ_n (de 10000 à 10^6), qui règle le nombre total de nœuds dans le graphe. Le paramètre λ_r est fixé de sorte que le nombre de temporalités $\mathcal{T}$ ne change pas (le nombre de nœuds par temporalité varie proportionnellement au nombre de nœuds total, $\lambda_r = \lambda_n/100$) La figure 8 montre que le temps de génération des graphes croît exponentiellement avec le nombre total de nœuds et donc avec le nombre de nœuds par temporalité. De fait, pour chaque nœud créé, l'algorithme doit parcourir les nœuds de sa temporalité et de la précédente pour établir les relations : comme λ_r est le nombre moyen de nœuds par temporalité il y a donc en moyenne au plus $2\lambda_r \times \lambda_n \approx \lambda_n^2$ opérations, puisque $\mathcal{T}$ est fixé.

Dans un deuxième temps, afin d'examiner l'influence du nombre de nœuds par temporalité, nous avons fixé le nombre de nœuds total et avons fait varier le nombre de nœuds par temporalité (de 100 à 2000). Cette expérience

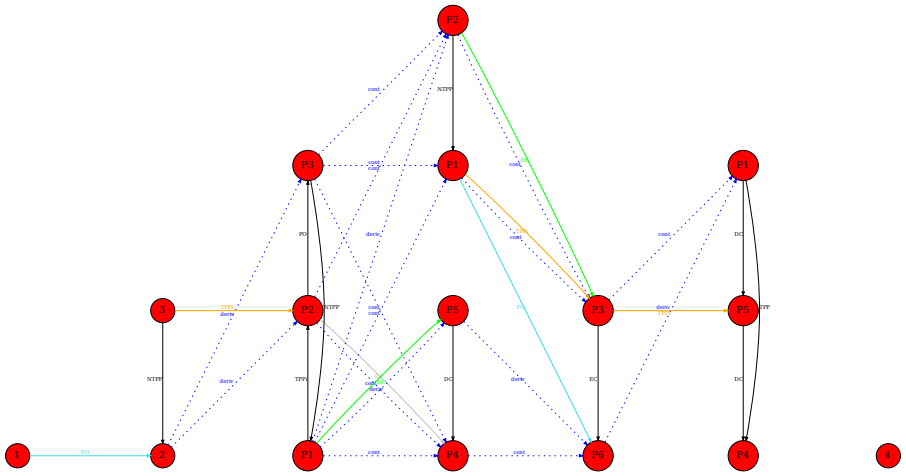


FIGURE 6 – Un graphe spatio-temporel cohérent incluant des motifs : les arêtes spatiales sont représententées en noir, les arêtes spatio-temporelles en couleur, les arêtes de filiation en traits tiretés

montre qu'il y a une relation linéaire entre le temps de génération des graphes et le nombre de nœuds par temporalité (voir figure 9), pour un nombre total de nœuds fixé. Tout étant fixé par ailleurs, seul varie le nombre de nœuds à parcourir pour établir les relations dans une temporalité et avec la temporalité précédente. Comme ci-dessus, le nombre d'opérations est en moyenne $2\lambda_n \times \lambda_r$. Le paramètre λ_n étant fixé, le temps de calcul est donc proportionnel à λ_r.

4.2 Variation du nombre d'arêtes

Nous nous intéressons ici à l'influence de la génération des arêtes sur le temps de génération des graphes. Pour ce faire, nous avons fait varier le nombre de relations par nœud : Λ_e varie de [0,0,0] à [200,200,200]. Dans cette expérience, le nombre total de nœuds et de nœuds par temporalité sont fixés. La figure 10 illustre cette expérience. Il existe une relation linéaire entre le nombre de relations par nœud et le temps de génération du graphe complet jusqu'à arriver à un plateau aux alentours de 100 relations par nœud. La partie linéaire s'explique de cette manière : pour chaque nœud nous avons en moyenne $\lambda_e = \Lambda_e[1] + \Lambda_e[2] + \Lambda_e[3]$ créations d'arêtes et $\Lambda_e[1] + \Lambda_e[2]$ vérifications de contraintes à réaliser. Pour la totalité du graphe, nous avons donc $\lambda_n \times \lambda_e$ opérations, nombre qui croît linéairement avec λ_e, λ_n étant fixé. La partie constante est due à la saturation du graphe, c'est-à-dire que chaque nœud a atteint son nombre maximal d'arêtes. En effet, dans cette expérience, une temporalité est composée en moyenne de 100 nœuds, qui ont chacun au plus une seule relation de chaque type avec un nœud de même temporalité ou de la temporalité précédente. Notons que le temps de calcul associé à la vérification de la cohérence des relations n'est liée qu'au nombre de triplets à examiner, puisque chaque arête porte une relation atomique.

4.3 Variation du nombre de motifs

Dans cette dernière expérience, nous faisons varier le nombre de motifs insérés dans les graphes (ou plus exactement la proportion de nœuds provenant de motifs, réglée par le paramètre p (qui varie de 0 à 100%), à taille de graphe constante. La figure 11 montre qu'en augmentant cette proportion, le temps de génération des graphes augmente linéairement. Ceci s'explique ainsi : le temps de génération et de transformation des motifs-sources est constant (les paramètres $pnodes$, λ_r, Λ_e, $support$ et λ_t sont fixés), la taille des motifs est constante (paramètre $pnodes$), seul varie donc le nombre de motifs-sources à générer pour atteindre une proportion donnée de nœuds par rapport au nombre total de nœuds dans le graphe.

Finalement, la complexité expérimentale observée est du même ordre que la complexité théorique. Elle est liée à l'algorithme principal de génération des nœuds et des relations entre les nœuds. Le temps nécessaire à la vérification de la cohérence est négligeable au regard de l'algorithme principal.

5 Conclusion

Cet article présente une méthode permettant de générer des graphes spatio-temporels dont les arêtes spatiales et spatio-temporelles sont localement cohérentes. Pour ce faire, nous nous sommes appuyées sur un algorithme de génération de graphes spatio-temporels aléatoires existant [12]. Cet algorithme a été modifié dans le sens où à chaque ajout d'une arête (spatiale ou spatio-temporelle) entre deux nœuds, la cohérence de cette arête avec les arêtes préexistantes reliant ces deux nœuds à un même nœud est vérifiée. La méthode de la chemin-cohérence est utilisée pour cela, dans le cadre de la théorie RCC8.

La particularité de l'algorithme est d'insérer dans le graphe généré des motifs fréquents inexacts. Ces motifs sont gé-

$\diamond$	DC	EC	PO	TPP	$NTPP$	TPP_i	$NTPP_i$	EQ
DC	$DC, EC, PO, TPP, NTPP, TPP_i, NTPP_i, EQ$	$DC, EC, PO, TPP, NTPP$	$DC, EC, PO, TPP, NTPP$	$DC, EC, PO, TPP, NTPP$	$DC, EC, PO, TPP, NTPP$	DC	DC	DC
EC	$DC, EC, PO, TPP_i, NTPP_i$	$DC, EC, PO, TPP, TPP_i, EQ$	$DC, EC, PO, TPP, NTPP$	$EC, PO, TPP, NTPP$	$PO, TPP, NTPP$	DC, EC	DC	EC
PO	$DC, EC, PO, TPP_i, NTPP_i$	$DC, EC, PO, TPP_i, NTPP_i$	$DC, EC, PO, TPP, NTPP, TPP_i, NTPP_i, EQ$	$PO, TPP, NTPP$	$PO, TPP, NTPP$	$DC, EC, PO, TPP_i, NTPP_i$	$DC, EC, PO, TPP_i, NTPP_i$	PO
TPP	DC	DC, EC	$DC, EC, PO, TPP, NTPP$	$TPP, NTPP$	$NTPP$	$DC, EC, PO, TPP, TPP_i, EQ$	$DC, EC, PO, TPP_i, NTPP_i$	TPP
$NTPP$	DC	DC	$DC, EC, PO, TPP, NTPP$	$NTPP$	$NTPP$	$DC, EC, PO, TPP, NTPP$	$DC, EC, PO, TPP, NTPP, TPP_i, NTPP_i, EQ$	$NTPP$
TPP_i	$DC, EC, PO, TPP_i, NTPP_i$	$EC, PO, TPP_i, NTPP_i$	$PO, TPP_i, NTPP_i$	EQ, PO, TPP, TPP_i	$PO, TPP, NTPP$	$TPP_i, NTPP_i$	$NTPP_i$	TPP_i
$NTPP_i$	$DC, EC, PO, TPP_i, NTPP_i$	$PO, TPP_i, NTPP_i$	$PO, TPP_i, NTPP_i$	$PO, TPP_i, NTPP_i$	$PO, TPP, NTPP, EQ, TPP_i, NTPP_i$	$NTPP_i$	$NTPP_i$	$NTPP_i$
EQ	DC	EC	PO	PO	$NTPP$	TPP_i	$NTPP_i$	EQ

FIGURE 7 – Table de composition des relations de base de la théorie RCC8

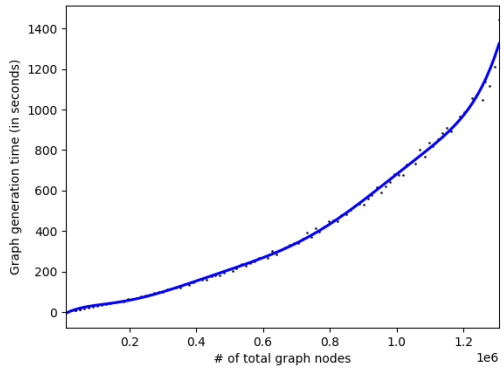


FIGURE 8 – Temps de génération des graphes en fonction du nombre de nœuds total

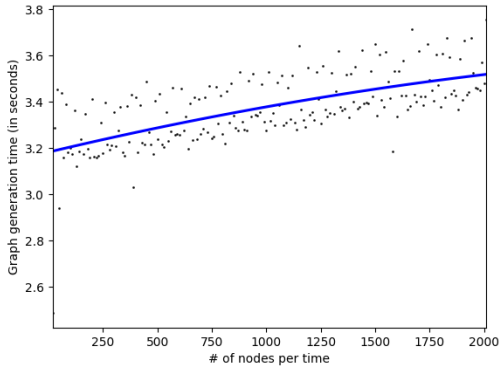


FIGURE 9 – Temps de génération d'un graphe en fonction du nombre de nœuds par temporalité

nérés et transformés en vérifiant également leur cohérence locale.
De plus, des tests de complexité ont été réalisés afin de mettre en évidence les étapes les plus coûteuses de la génération des graphes spatio-temporels. Les résultats de ces expérimentations se sont montrés cohérents avec la complexité théorique de l'algorithme. Une comparaison avec d'autres approches n'a pu être menée, car, à notre connaissance, il n'existe pas de méthode permettant de générer des graphes spatio-temprels comme nous le proposons.
Dans la suite du travail, une chemin-cohérence plus large pourra être développée, en s'inspirant des travaux de [16]. Au delà, nous voulons utiliser les graphes générés par notre approche pour tester diverses méthodes de recherche de motifs fréquents dans un graphe spatio-temporel. Pour comparer les résultats à ceux qui seraient obtenus sur une base de graphes réels, nous nous attacherons à paramétrer le générateur de sorte à produire des graphes similaires à des graphes réels, pour les applications que nous traitons, dans les domaines agricole et médical.

Remerciements

Ces travaux ont été réalisés dans le cadre du projet METEC-Graphe, financé par l'Idex de l'Université de Strasbourg.

Références

[1] C. Bessière. A simple way to improve path consistency processing in interval algebra networks. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, pages 375–380, 1996.

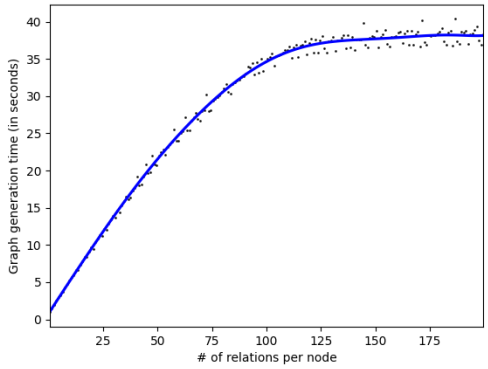


FIGURE 10 – Temps de génération des graphes en fonction du nombre de relations par nœud

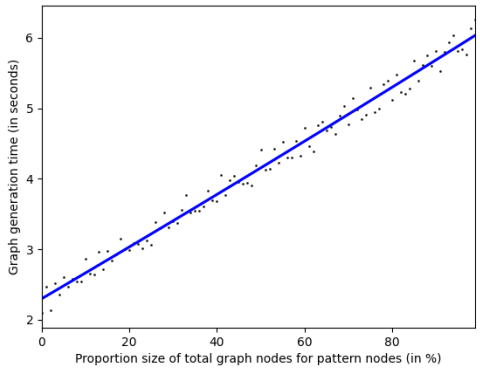


FIGURE 11 – Temps de génération des graphes en fonction de la proportion de motifs insérés (a) Paramètres (b) Courbe obtenue

[2] A. Bonifati, I. Holubová, A. Prat-Pérez, and S. Sakr. Graph generators : State of the art and open challenges. *ACM Comput. Surv.*, 53(2), April 2020.

[3] C. Campbell, K. Shea, and R. Albert. Comment on control profiles of complex networks. *Science*, 346(6209) :561–561, 2014.

[4] J.-F. Condotta. Problèmes de satisfaction de contraintes : algorithmes et complexité. Thèse de l'Université Toulouse 3, 2000.

[5] J-F. Condotta and Würbel E. Réseaux de contraintes temporelles et spatiales. In F. Le Ber, G. Ligozat, and O. Papini, editors, *Raisonnements sur l'espace et le temps*, chapter 7, pages 181–223. Hermès, 2007.

[6] G. Del Mondo, M. A. Rodríguez, C. Claramunt, L. Bravo, and R. Thibaud. Modeling consistency of spatio-temporal graphs. *Data & Knowledge Engineering*, 84 :59–80, 2013.

[7] G. Del Mondo, J. G. Stell, C. Claramunt, and R. Thibaud. A graph model for spatio-temporal evolution. *Journal of Universal Computer Science*, 16 :1452–1477, 2010.

[8] R. Ferrer i Cancho and R.V. Solé. Optimization in complex networks. In *Statistical mechanics of complex networks*, pages 114–126. 2003.

[9] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *Proceedings 2001 IEEE Int Conference on Data Mining*, pages 313–320, 2001.

[10] A. Leborgne, F. Le Ber, D. Niezgoda, C. Meillier, and S. Marc-Zwecker. Utilisation des graphes pour la représentation spatio-temporelle lors d'un examen d'irm fonctionnelle cérébrale. In *Journée Santé & IA 2020 (dans le cadre PFIA)*, Jun 2020.

[11] A. Leborgne, A. Meyer, H. Giraud, F. Le Ber, and S. Marc-Zwecker. Un graphe spatio-temporel pour modéliser l'évolution de parcelles agricoles. In *SAGEO*, Clermont-Ferrand, France, November 2019.

[12] A. Leborgne, J. Nuss, F. Le Ber, and S. Marc-Zwecker. An approach for generating random temporal semantic graphs with embedded patterns. In *Graph Embedding and Mining, ECML-PKDD 2020 Workshop Proc.*, 2020.

[13] D. A. Randell, A. G. Cohn, and Z. Cui. Computing transitivity tables : A challenge for automated theorem provers. In D. Kapur, editor, *Automated Deduction—CADE-11*, pages 786–790. Springer Berlin Heidelberg, 1992.

[14] D. A. Randell, Z. Cui, and A. G. Cohn. A Spatial Logic based on Regions and Connection. In *Proceedings 3rd International Conference on Knowledge Representation and Reasoning*, pages 165–176. Morgan Kaufmann Publishers, 1992.

[15] J. Renz and G. Ligozat. Weak composition for qualitative spatial and temporal reasoning. In Peter van Beek, editor, *Principles and Practice of Constraint Programming - CP 2005*, pages 534–548. Springer Berlin Heidelberg, 2005.

[16] M. Sioutis and J.-F. Condotta. Incrementally building partially path consistent qualitative constraint networks. In *AIMSA 2014 : Artificial Intelligence : Methodology, Systems, and Application*, pages 104–116, 09 2014.

[17] M. Vilain, H. Kautz, and P. V. Beek. Constraint propagation algorithms for temporal reasoning : A revised report. In D. S. Weld and J. De Kleer, editors, *Readings on Qualitative Reasoning about Physical Systems*, pages 373–381. Morgan Kaufmann, 1989.