

Graphes de positions : représentations d'alignements et recherche de motifs

Rita-Nathalia Assaf¹, Frédéric Lardeux¹ Frédéric Saubion¹

¹ LERIA, Université d'Angers

prenom.nom@univ-angers.fr

Résumé

Dans cet article nous proposons un formalisme pour décrire des relations d'alignement entre des objets afin de définir un espace de positionnement au sein duquel nous cherchons la présence de motifs particuliers. La modélisation proposée s'appuie sur plusieurs graphes décrivant des aspects complémentaires de cet espace de positionnement. Afin de rechercher des motifs dans ces graphes, et donc d'identifier des structures particulières formées par les objets, nous proposons d'utiliser une approche de résolution par contraintes, s'appuyant sur un algorithme de filtrage existant et permettant de traiter le problème d'isomorphisme de sous-graphes. Des expérimentations nous permettent d'évaluer les avantages/inconvénients des représentations proposées. Une application potentielle qui a motivé ce travail est la recherche de structures de tableaux au sein de documents professionnels.

Mots-clés

Isomorphisme de sous-graphes - Modélisation - Algorithmes de filtrage

Abstract

In this paper, we propose a formalism for describing alignment relationships between objects in order to define a positioning space within which we search for specific patterns. The proposed modelling is based on several graphs that describe complementary aspects of this positioning space. To search for patterns in these graphs, and thus identify specific structures formed by the objects, we propose using a constraint-based solving approach. This approach builds on an existing filtering algorithm and enables us to address the subgraph isomorphism problem. We evaluate the advantages/disadvantages of the proposed representations by means of some experiments. A potential application that motivated this work is the search for table structures in professional documents.

Keywords

Subgraph isomorphism - Modelling - Filtering Algorithms

1 Introduction

La modélisation et l'analyse de relations spatiales entre objets [25] est une préoccupation cruciale notamment pour

la compréhension d'images et pour les systèmes d'information géographique [9]. Sur le plan plus formel, nous pouvons mentionner le système RCC-8 (*Region Connection Calculus*) qui permet un raisonnement qualitatif à partir d'un modèle de représentation spatiale et d'un ensemble d'axiomes [4], en s'appuyant notamment sur les intervalles de Allen [1]. Le raisonnement dans RCC-8 étant NP-complet en général [18], des approches efficaces ont été proposées dans la communauté de la programmation par contraintes pour aborder ces systèmes [16, 8]. En outre, il est intéressant de se rappeler que des travaux fondateurs en programmation par contraintes ont justement été motivés par l'analyse de scènes et d'images [15, 13].

Dans ce travail, nous souhaitons restreindre les relations spatiales à des relations d'alignement et nous focalisons notre problématique sur une recherche de motifs. La motivation principale de cette approche trouve son origine dans des travaux précédents que nous avons menés pour extraire automatiquement des informations à partir de documents structurés, et plus particulièrement de factures [22]. Au sein des diverses techniques utilisées, qui incluent de plus en plus d'approches basées sur les réseaux de neurones [20], des approches basées sur des modélisations utilisant des graphes ont été proposées [7, 19, 3]. La figure 1 illustre comment un document peut être modélisé par un graphe une fois son contenu analysé via des outils classiques (OCR, tokenisation...) qui permettent notamment d'obtenir les positions des différents éléments du document. Toutefois, nombre de ces approches, issues de l'apprentissage profond, nécessitent un entraînement coûteux et des jeux de données extrêmement importants. Cela nous a conduits à proposer une approche à base de programmation par contraintes qui nous a permis de traiter des jeux de données réels de manière efficace, dans un contexte réel de gestion électronique de documents [21]. Dans ce contexte, via une modélisation adéquate, la recherche d'un tableau au sein d'un document peut être ramenée à la recherche d'un sous-graphe (le tableau) dans un graphe (le document).

Le problème d'isomorphisme de sous-graphes (SIP) est connu pour être NP-complet dans le cas général [6]. Des algorithmes dédiés à la résolution des problèmes d'isomorphisme de sous-graphes ont ainsi été proposés depuis de nombreuses années [24, 5]. Actuellement des algorithmes exacts performants, basés sur des approches par fil-

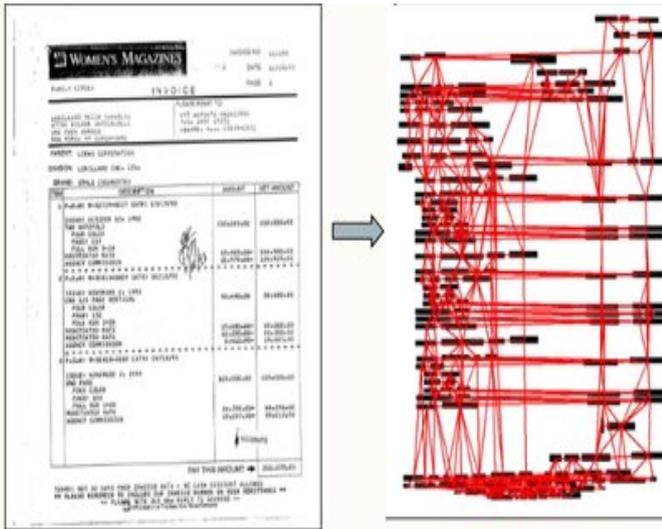


FIGURE 1 – Exemple de transformation d’une facture en un graphe après l’analyse de l’image initiale de la facture en entrée (par exemple un document numérisé). Les arêtes en rouge traduisent les relations d’alignement entre les entités élémentaires extraites. Notons que cette facture contient un tableau (illustration extraite de [3])

trage sont ainsi disponibles [26, 10, 14]. Ces dernières approches s’appuient sur les principes de la programmation par contraintes, notamment en tirant partie d’algorithmes de filtrage spécifiquement dédiés au SIP [27, 23], pouvant être vu alors comme une contrainte globale. En effet, les problèmes d’isomorphisme de sous-graphes peuvent aussi être formulés comme des CSP de manière assez directe, notamment en lien avec des contraintes globales de type *Alldiff*[17].

Dans ce contexte, les objectifs de ce travail exploratoire sont, d’une part de proposer un cadre général pour la modélisation d’espaces de positionnement sous forme de graphes représentant plusieurs points de vue complémentaires (alignement de tokens basiques, organisation en lignes et colonnes...) et, d’autre part, d’étudier comment cette modélisation peut faciliter la recherche de motifs au sein de ces espaces de positionnement. En nous concentrant sur le problème sous-jacent d’isomorphisme de sous graphes, nous nous appuyons sur un algorithme simple de filtrage [27], utilisant les spécificités de notre modélisation. Notre but n’est pas ici de proposer un nouvel algorithme de filtrage mais plutôt d’analyser comment se comportent différents modèles possibles, intégrant une contrainte globale dédiée, sur quelques cas d’études.

L’article est organisé comme suit : après une introduction qui motive l’étude des relations d’alignement entre objets et présente les objectifs et le contexte de ce travail, la section 2 définit un cadre formel pour décrire des espaces de positions et propose plusieurs représentations de ces espaces via des graphes, en précisant également la notion de motif dans chacun des cas et leurs liens avec le problème d’isomorphisme de sous-graphes. La section 3 propose une modélisation de la recherche de motifs sous la forme de problèmes

de satisfaction de contraintes et précise les mécanismes de filtrage pouvant être utilisés dans ce contexte. Enfin une section est consacrée à la présentation d’expérimentations visant à illustrer les différents modèles et concepts introduits.

2 Graphes de positions

2.1 Espaces et représentations

Afin de proposer un cadre général, nous nous basons sur des relations d’ordre partiel qui permettent de définir le positionnement relatif d’objets/items/tokens.

Définition 1 (Relation d’alignement). *Etant donné un ensemble de tokens \mathcal{T} , une relation d’alignement $\leq \subseteq \mathcal{T} \times \mathcal{T}$ est une relation d’ordre partiel telle que $\forall t, t', t'' \in \mathcal{T}, (t < t' \wedge t < t'') \Rightarrow (t' < t'' \vee t'' < t')$*

La condition introduite dans cette définition permet d’interdire la présence d’alignements incohérents de notre point de vue, c’est à dire correspondant à une division incongrue d’une même ligne. Nous considérons ici une relation d’ordre partiel stricte. On se donne à présent deux relations d’alignement définies sur $\mathcal{T} \times \mathcal{T}$: $<_h$ (alignement horizontal) et $<_v$ (alignement vertical). Le triplet $(\mathcal{T}, <_h, <_v)$ sera considéré comme un espace de positionnement qui définit ainsi l’organisation des tokens relativement à leurs alignements respectifs. Nous considérerons que tout élément de \mathcal{T} doit avoir un positionnement relatif par rapport à au moins un autre élément (soit vertical soit horizontal). Si on souhaite relâcher cette contrainte, il suffira de considérer des relations réflexives, ce qui ne change pas fondamentalement la suite.

Notons qu’un espace de positionnement peut ne pas être cohérent relativement aux relations $<_h$ et $<_v$ lorsqu’on les interprète en termes d’alignements verticaux et horizontaux des tokens. La cohérence d’un espace de positionnement peut être définie par le fait qu’il n’existe pas quatre tokens $t_1, t_2, t_3, t_4 \in \mathcal{T}$ tels que $t_1 <_h t_2 \wedge t_3 <_h t_4 \wedge t_1 <_v t_4 \wedge t_2 <_v t_3$ ou $t_1 <_v t_2 \wedge t_3 <_v t_4 \wedge t_1 <_h t_3 \wedge t_4 <_h t_2$. Dans la suite, nous ne manipulerons que des espaces de positionnement cohérents.

Notons ici que les graphes grilles (*grid or lattice graphs*) pourraient partager certaines similarités avec notre approche mais ces graphes sont en général réguliers et considèrent comme sommets des points de coordonnées entières et comme arêtes les paires de sommets partageant une coordonnée. Ainsi, nous cherchons ici une représentation plus spécifique et plus déclarative basée sur les alignements.

Afin d’illustrer les différentes représentations que nous allons proposer, nous considérons l’exemple suivant.

Exemple 1. *Soit $\mathcal{T} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$. Nous définissons notre espace de positionnement en posant : $1 <_h 2, 2 <_h 3, 3 <_h 4, 5 <_h 6, 6 <_h 7, 8 <_h 9, 9 <_h 10, 1 <_v 5, 5 <_v 8, 2 <_v 6, 6 <_v 9, 9 <_v 11, 3 <_v 7, 7 <_v 10$. Dans cet exemple, nous pouvons observer que l’élément 4 n’est aligné avec aucun autre élément selon $<_v$, ce qui mettra en avant une colonne avec un seul élément. Il en va de même pour l’élément 11 relativement à $<_h$.*

Définition 2 (Graphe de positions). *Étant donné un espace de positionnement $\Pi = (\mathcal{T}, <_h, <_v)$, on définit le graphe de positions induit GP_Π comme un graphe étiqueté orienté $(\mathcal{T}, \mathcal{A}, \mathcal{E})$ où $\mathcal{A} \subseteq \mathcal{T} \times \mathcal{T}$ et $\mathcal{E} : \mathcal{T} \times \mathcal{T} \rightarrow \{h, v\}$ tels que :*

- $\forall t, t' \in \mathcal{T}, t <_h t' \Leftrightarrow (t, t') \in \mathcal{A} \wedge \mathcal{E}((t, t')) = h$
- $\forall t, t' \in \mathcal{T}, t <_v t' \Leftrightarrow (t, t') \in \mathcal{A} \wedge \mathcal{E}((t, t')) = v$

Pour l'exemple 1, nous obtenons le graphe de la figure 2. Sur la figure 2, nous ne reportons pas les étiquettes des arcs car les arcs étiquetés 'h' seront horizontaux et naturellement les arcs étiquetés 'v' verticaux.

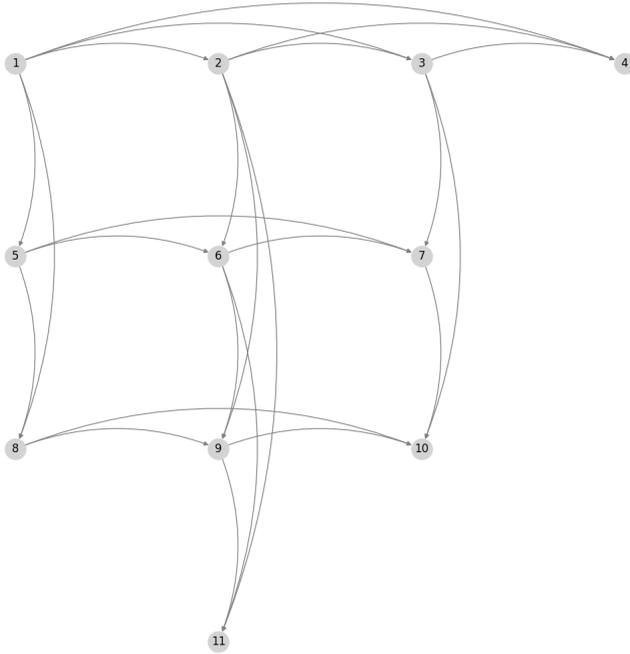


FIGURE 2 – Graphe de positions GP

Notons que les relations d'ordre partiel $<_h$ et $<_v$ sont transitives mais, à l'instar de ce qui est fait pour les diagrammes de Hasse, nous pouvons simplifier notre représentation en ne représentant pas la transitivité dans le graphe de positions.

Définition 3 (Graphe de positions réduit). *Étant donné un graphe de positions $GP_\Pi = (\mathcal{T}, \mathcal{A}, \mathcal{E})$ construit sur un espace de positionnement $\Pi = (\mathcal{T}, <_h, <_v)$, le graphe de positions réduit $GP_\Pi^{red} = (\mathcal{T}, \mathcal{A}^{red}, \mathcal{E}^{red})$ est obtenu à partir de GP_Π avec*

$$\mathcal{A}^{red} = \mathcal{A} \setminus (\{(t, t'') \in \mathcal{A} \mid \exists t' \in \mathcal{T} (t, t') \in \mathcal{A} \wedge (t', t'') \in \mathcal{A}\})$$

\mathcal{E}^{red} est naturellement la restriction de \mathcal{E} à \mathcal{A}^{red}

Nous avons considéré des relations d'ordre strict, si toutefois on prenait en compte un ordre réflexif, il conviendrait de retirer également l'ensemble $\{(t, t) \in \mathcal{A}\}$. Toujours avec l'exemple 1, le graphe réduit est représenté sur la figure 3.

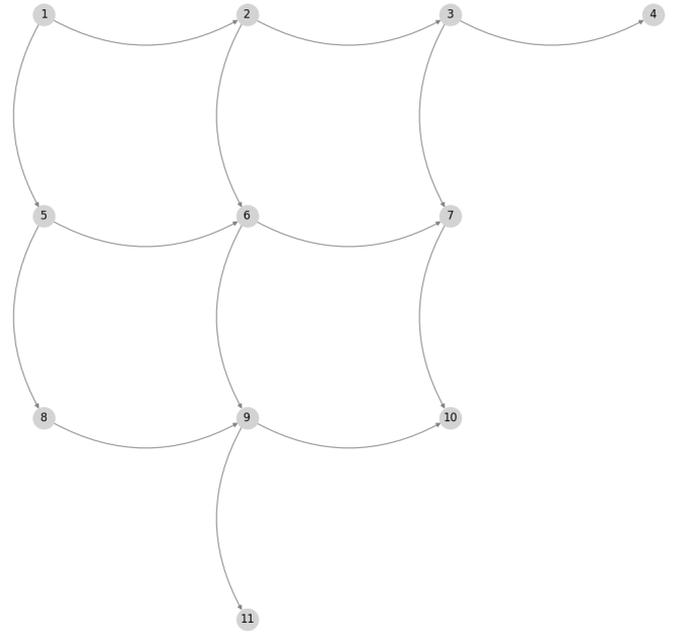


FIGURE 3 – Graphe de positions réduit GP^{red}

En utilisant les relations d'alignements $<_h$ (resp. $<_v$), nous définissons de manière naturelle la notion de ligne (resp. colonne).

Définition 4 (Lignes et colonnes). *Étant donné un espace de positionnement $\Pi = (\mathcal{T}, <_h, <_v)$, une ligne $L \subseteq \mathcal{T}$ (resp. une colonne $C \subseteq \mathcal{T}$) est un sous-ensemble totalement ordonné $(L, <_h)$ (resp. $(C, <_v)$). $\mathcal{L}_\Pi \subseteq 2^\mathcal{T}$ (resp. $\mathcal{C}_\Pi \subseteq 2^\mathcal{T}$) est l'ensemble des lignes (resp. des colonnes) associé à Π .*

Remarquons que $(\mathcal{L}_\Pi, \subseteq)$ et $(\mathcal{C}_\Pi, \subseteq)$ sont deux ensembles partiellement ordonnés selon la relation d'inclusion. Une ligne (resp. colonne) complète sera un élément maximal de $(\mathcal{L}_\Pi, \subseteq)$ (resp. $(\mathcal{C}_\Pi, \subseteq)$). Notons que l'ensemble des lignes (resp. colonnes) complètes \mathcal{L}_Π^{comp} (resp. \mathcal{C}_Π^{comp}) forment une partition de \mathcal{L}_Π (resp. \mathcal{C}_Π).

Définition 5 (Graphe des lignes/colonnes). *Étant donné un espace de positionnement $\Pi = (\mathcal{T}, <_h, <_v)$, $GLC_\Pi = (\mathcal{L}_\Pi \cup \mathcal{C}_\Pi, \mathcal{I})$ est le graphe non-orienté¹ des lignes/colonnes associées à Π , tel que $\mathcal{I} \subseteq \mathcal{L}_\Pi \times \mathcal{C}_\Pi$ vérifie $\forall (l, c) \in \mathcal{I}, l \cap c \neq \emptyset$. GLC_Π est un graphe biparti.*

Notons qu'il existe une arête du graphe non orienté GLC_Π , notée (l, c) qui correspond en fait à chaque token $t \in \mathcal{T}$ tel que $t \in c \wedge t \in l$. Le graphe de lignes/colonnes obtenu pour l'exemple 1 correspond à la figure 4. Nous pouvons remarquer que ce graphe possède considérablement plus d'arêtes

1. Comme nous allons manipuler des graphes orientés et non orientés, afin de pas alourdir le vocabulaire et les notations, nous noterons ici $(l, c) \in \mathcal{I}$ le fait que l'arête (l, c) est dans \mathcal{I} , de la même manière que pour les arcs. Le contexte, en général clair, permettra de distinguer cette nuance habituelle.

que le graphe de positions. Toutefois, ayant une structure bipartie, cela ne permet pas forcément de présumer de la difficulté de trouver un motif en son sein.

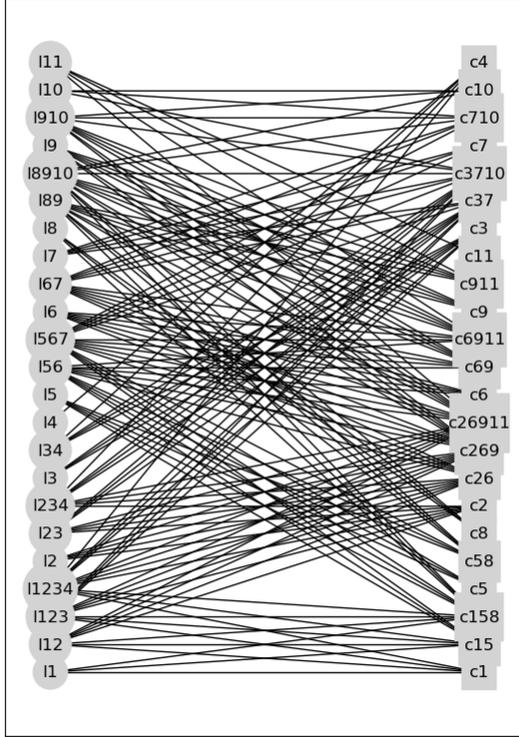


FIGURE 4 – Graphe GLC - Une ligne $\{5, 6\}$ est notée 156, une colonne $\{3, 7\}$ c37

Etant donné un graphe de lignes/colonnes $GLC_{\Pi} = (\mathcal{L}_{\Pi} \cup \mathcal{C}_{\Pi}, \mathcal{I})$, le graphe réduit de lignes/colonnes $GLC_{\Pi}^{red} = (\mathcal{L}_{\Pi}^{comp} \cup \mathcal{C}_{\Pi}^{comp}, \mathcal{I}^{comp})$ est obtenu en restreignant \mathcal{L}_{Π} (resp. \mathcal{C}_{Π}) aux lignes (resp. colonnes) complètes et en déduisant naturellement \mathcal{I}^{comp} .

Remarque : Comme les lignes et colonnes sont des sous-ensembles de \mathcal{T} , elles pourraient être également vues comme des hyperarêtes d'un hypergraphe sur \mathcal{T} . Nous ne développons pas cette autre modélisation ici.

Nous allons à présent établir des équivalences entre nos différentes représentations d'un même espace de positionnement.

Propriété 1 (Équivalence entre GP_{Π} et GLC_{Π}). *Etant donné un espace de positionnement $\Pi = (\mathcal{T}, <_h, <_v)$ et $GP_{\Pi} = (\mathcal{T}, \mathcal{A}, \mathcal{E})$, $GLC_{\Pi} = (\mathcal{L}_{\Pi} \cup \mathcal{C}_{\Pi}, \mathcal{I})$ ses représentations associées. Les propositions suivantes sont équivalentes, pour tout couple $(t, t') \in \mathcal{T} \times \mathcal{T}$:*

1. $t <_h t'$ (resp. $t <_v t'$)

2. $(t, t') \in \mathcal{A}$ et $\mathcal{E}((t, t')) = h$ (resp. $\mathcal{E}((t, t')) = v$)
3. $\exists l \in \mathcal{L}_{\Pi}, t \in l \wedge t' \in l$ (resp. $\exists c \in \mathcal{C}_{\Pi}, t \in c \wedge t' \in c$)

Nous noterons la relation d'équivalence induite entre les représentations par \sim_{Π} , i.e., $GP_{\Pi} \sim_{\Pi} GLC_{\Pi}$

Démonstration. L'équivalence entre 1. et 2. est assurée par la définition 2. Pour montrer l'équivalence entre 1. et 3., il faut utiliser la définition 4 des lignes et des colonnes. Nous avons en effet $t <_h t'$ (resp. $t <_v t'$) qui implique qu'il existe une ligne (resp. colonne) l (resp. c) qui contient t et t' . La réciproque est également évidente au vu de la définition des lignes et colonnes. \square

2.2 Isomorphisme de sous-graphes de positions

Afin d'alléger les notations, l'espace de positionnement $\Pi = (\mathcal{T}, <_h, <_v)$ sera supposé connu dans la suite et nous omettons donc les indices Π .

Définition 6 (Motif de positions). *Etant donné un ensemble T , un motif de positions est un graphe dirigé étiqueté $MP = (T, A, E)$, tel que $A \subseteq T \times T$ et $E : A \rightarrow \{h, v\}$.*

Définition 7 (Isomorphisme de sous-graphes de positions). *Soit un graphe de positions $GP = (T, \mathcal{A}, \mathcal{E})$ et un motif de positions $MP = (T, A, E)$. MP est isomorphe à un sous-graphe de GP si et seulement si il existe une injection $f : T \rightarrow T$ vérifiant :*

- $(t, t') \in A \Rightarrow (f(t), f(t')) \in \mathcal{A}$
- $(t, t') \in A \Rightarrow E((t, t')) = \mathcal{E}((f(t), f(t')))$

Chaque injection f définit donc un sous-graphe $f(MP) = (f(T), \mathcal{A}_{f(T)}, \mathcal{E}_{f(T)})$ de GP , où $\mathcal{A}_{f(T)} = \{(t, t') \in \mathcal{A} \mid t, t' \in f(T)\}$ et $\mathcal{E}_{f(T)}$ est la restriction de \mathcal{E} à $f(T)$. On notera alors $GP \triangleright_f MP$.

Sur l'exemple 1, nous pouvons rechercher, dans le graphe GP^{red} , le motif $M = (T = \{a, b, c, d\}, A = \{(a, b), (c, d), (a, c), (b, d)\}, E)$ avec $E((a, b)) = E((c, d)) = h$ et $E((a, c)) = E((b, d)) = v$. Nous aurons alors une solution illustrée par la figure 5.

Comme nos représentations sont équivalentes, nous allons étendre cette équivalence à la recherche de motifs. Nous définissons donc la notion de motif de lignes/colonnes.

2.3 Isomorphisme de sous-graphes lignes/colonnes

Définition 8 (Motif de lignes/colonnes). *Un motif de lignes/colonnes est un graphe non orienté $MLC = (TL \cup TC, I)$ défini sur deux ensembles TL, TC tels que $TL \cap TC = \emptyset$ et tel que $I \subseteq TL \times TC$.*

Définition 9 (Isomorphisme de sous-graphes lignes/colonnes). *Soit un graphe de lignes/colonnes $GLC = (\mathcal{L} \cup \mathcal{C}, \mathcal{I})$ et un motif de lignes/colonnes $MLC = (TL \cup TC, I)$. MLC est isomorphe à un sous-graphe de GLC si et seulement si il existe une injection $f : TL \cup TC \rightarrow \mathcal{L} \cup \mathcal{C}$ vérifiant :*

- $\forall l \in TL, f(l) \in \mathcal{L}, \forall c \in TC, f(c) \in \mathcal{C}$
- $\forall (l, c) \in I, (f(l), f(c)) \in \mathcal{I}$

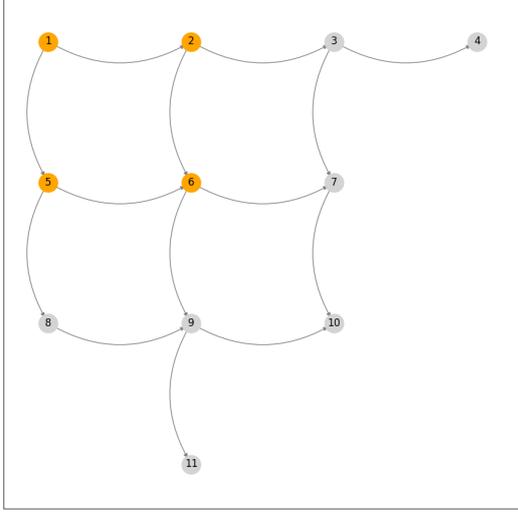


FIGURE 5 – Motif M sur GP^{red} (en orange)

Chaque injection f définit donc un sous-graphe $f(MLC) = (f(L) \cup f(C), \mathcal{I}|_{f(L) \cup f(C)})$ de GLC où $\mathcal{I}|_{f(L) \cup f(C)} = \{(f(l), f(c)) \in \mathcal{I}\}$. On notera alors $GLC \triangleright_f MLC$.

Propriété 2 (Equivalence de la recherche de motifs). Soit un graphe de positions $GP = (\mathcal{T}, \mathcal{A}, \mathcal{E})$, et un graphe de lignes/colonnes $GLC = (\mathcal{L} \cup \mathcal{C}, \mathcal{I})$, tels que $GP \sim GLC$. Soit un motif de positions $MP = (T, A, E)$ et un motif de lignes/colonnes $MLC = (TL \cup TC, I)$, tels que $TL \subseteq 2^T$ et $TC \subseteq 2^T$. Il existe une injection fP telle que $GP \triangleright_{fP} MP$ si et seulement si il existe une injection fLC telle que $GLC \triangleright_{fLC} MLC$ qui vérifient :

1. $\forall ml \in TL, \forall t \in T, (t \in ml) \Leftrightarrow fP(t) \in fLC(ml)$
2. $\forall mc \in TC, \forall t \in T, (t \in mc) \Leftrightarrow fP(t) \in fLC(mc)$
3. $\forall ml \in TL, \forall mc \in TC, \forall t \in T, ml \cap mc = t \Leftrightarrow fLC(ml) \cap fLC(mc) = fP(t)$

Démonstration. Supposons qu'il existe une injection $fP : T \rightarrow \mathcal{T}$ telle que $GP \triangleright_{fP} MP$. Soient $TL \subseteq 2^T$ et $TC \subseteq 2^T$ et une fonction $fLC : TL \cup TC \rightarrow \mathcal{L} \cup \mathcal{C}$ qui vérifie les trois conditions 1. 2. et 3. Soient $lc, lc' \in TL \cup TC$ telles que $lc \neq lc'$. Comme fP est injective et avec les conditions 1. et 2., on a forcément $fLC(lc) \neq fLC(lc')$, et donc fLC injective. Par ailleurs, nous savons que le motif $MP = (T, A, E)$ est isomorphe à un sous-graphe de GP noté $fP(MP)$ (voir définition 7). Comme $GP \sim GLC$, en utilisant la propriété 1, et les conditions 1., 2. et 3. liant fP et fLC , nous pouvons en déduire qu'il existe un sous-graphe de GLC construit comme $\{l \in \mathcal{L} | \exists fP(t) \in \mathcal{T}, fP(t) \in l\} \cup \{c \in \mathcal{C} | \exists fP(t) \in \mathcal{T}, fP(t) \in c\}, \{(l, c) \in$

$\mathcal{L} \times \mathcal{C} | \exists fP(t) \in \mathcal{T}, fP(t) \in l \cap c\}$ qui est isomorphe au motif de lignes/colonnes $(TL \cup TC, I)$ de telle sorte que $GLC \triangleright_{fLC} MLC$. La réciproque peut être obtenue en suivant le même raisonnement du fait des conditions imposées dans la propriété 2. \square

Le corollaire suivant découle de la propriété 2 et de la définition des équivalences entre modélisations de la définition 7.

Corollaire 1. Soit un graphe de positions GP et un graphe de lignes/colonnes GLC tels que $GP \sim GLC$, un motif de positions MP et un motif de lignes/colonnes MLC . S'il existe des injections fP et fLC telles que $GP \triangleright_{fP} MP \wedge GLC \triangleright_{fLC} MLC$ et qu'elles vérifient les conditions de la propriété 2, on a alors $fP(MP) \sim fLC(MLC)$

Ce corollaire nous permet d'établir qu'il est équivalent de rechercher un motif sur le graphe de position GP ou sur son graphe de lignes et colonnes associé, GLC .

La propriété 2 et son corollaire assurent une cohérence entre les motifs. Toutefois, il convient de remarquer que, d'un point de vue opérationnel, étant donné un motif de position MP , il faut construire un motif de lignes/colonnes correspondant MLC . Etant donné $GP \sim GLC$, nous aurons alors un ensemble de solutions $S_{MP}^{GP} = \{fP | GP \triangleright_{fP} MP\}$ et un ensemble de solutions $S_{MLC}^{GLC} = \{fLC | GLC \triangleright_{fLC} MLC\}$ tels que $\forall fP \in S_{MP}^{GP}, \exists fLC \in S_{MLC}^{GLC}, fP(MP) \sim fLC(MLC)$ et réciproquement. Toutefois, les ensembles de solutions ne sont pas directement équivalents. Revenons à notre exemple 1.

Le motif illustré par la figure 5 et défini sur $T = \{a, b, c, d\}$ va nous amener à rechercher un motif MLC défini sur $TL = \{\{a, b\}, \{c, d\}\}$ et $TC = \{\{a, c\}, \{b, d\}\}$ avec $I = \{(\{a, b\}, \{a, c\}), (\{a, b\}, \{b, d\}), (\{c, d\}, \{a, c\}), (\{c, d\}, \{b, d\})\}$. Si on considère la solution $fP = \{a \mapsto 1, b \mapsto 2, c \mapsto 5, d \mapsto 6\}$ présentée sur la figure 5, il existe donc plusieurs solutions pour ce motif dans le graphe GLC . Celle qui nous intéresse est $fLC = \{\{a, b\} \mapsto \{1, 2\}, \{c, d\} \mapsto \{5, 6\}, \{a, c\} \mapsto \{1, 5\}, \{b, d\} \mapsto \{2, 6\}\}$, mais $fLC' = \{\{a, b\} \mapsto \{5, 6\}, \{c, d\} \mapsto \{1, 2\}, \{a, c\} \mapsto \{2, 6\}, \{b, d\} \mapsto \{1, 5\}\}$ est également une solution au problème de recherche de motif. Toutefois cette solution fLC' ne vérifie pas les conditions de la propriété 2 et nous n'avons donc pas $fP(MP) \sim fLC'(MLC)$. Nous avons en effet perdu dans GLC la représentation directe de l'information sur les ordres entre tokens (positionnements relatifs verticaux et horizontaux) en passant d'un graphe orienté à un graphe non orienté.

Nous pouvons toutefois ajouter des contraintes lors de la recherche de motifs dans le graphe de lignes colonnes afin d'extraire de l'ensemble des solutions obtenues celles qui coïncident avec le motif de position initial. Pour des raisons de place, nous ne détaillons pas la formalisation de ce raffinement des solutions mais nous allons décrire de manière opérationnelle comment dans notre recherche de motifs nous prendrons en compte ces contraintes.

3 Recherche de motifs

Dans cette section, nous allons présenter une modélisation de la recherche de motifs sous la forme de CSPs et décrire également un algorithme de filtrage qui permet de traiter spécifiquement le problème d'isomorphisme de sous-graphes.

3.1 Motifs de positions

Soient un graphe $GP = (\mathcal{T}, \mathcal{A}, \mathcal{E})$ et un motif associé $MP = (T, A, E)$ tels que présentés dans la définition 7.

Nous introduisons, de manière naturelle en nous appuyant sur cette définition, le problème de satisfaction de contraintes $(\mathcal{X}, D, \mathcal{C})_P$ (variables, domaines, contraintes) relatif à la recherche de MP dans GP de la manière suivante. L'ensemble des variables de décision est défini par :

$$\mathcal{X} = \{u_t | t \in T\}$$

Nous associons donc à chaque élément du motif recherché une variable de décision dont le domaine est défini par :

$$\forall u_t \in \mathcal{X}, D(u_t) = \mathcal{T}$$

Il convient ensuite de faire le lien entre les arcs de GP et les arcs du motif.

$$\forall u_t, u_{t'} \in \mathcal{X}, (t, t') \in A \Rightarrow (u_t, u_{t'}) \in \mathcal{A}$$

Il reste à poser une contrainte également sur le type des arcs correspondant à chacune des relations d'alignement.

$$\forall u_t, u_{t'} \in \mathcal{X}, E((t, t')) = \mathcal{E}((u_t, u_{t'}))$$

Enfin, comme ce CSP modélise la recherche de la fonction fP de la définition 7 qui est injective, nous ajoutons une contrainte globale bien connue [17] sur l'ensemble des variables de décision $Alldiff(\mathcal{X})$.

3.2 Motifs de lignes/colonnes

Nous considérons à présent un graphe de lignes/colonnes $GLC = (\mathcal{L} \cup \mathcal{C}, \mathcal{I})$ et un motif de lignes/colonnes $MLC = (TL \cup TC, I)$ associé. De la même manière que précédemment, nous définissons un CSP $(\mathcal{X}, D, \mathcal{C})_{LC}$ relatif à la recherche de MLC dans GLC . De la même manière et pour modéliser la fonction fLC de la définition 9, nous posons donc :

$$\begin{aligned} \mathcal{X} &= \{u_{lc} | lc \in TL \cup TC\} \\ \forall u_{lc} \in \mathcal{X}, lc \in TL &\Rightarrow D(u_{lc}) = \mathcal{L} \\ \forall u_{lc} \in \mathcal{X}, lc \in TC &\Rightarrow D(u_{lc}) = \mathcal{C} \\ \forall u_{lc}, u_{lc'} \in \mathcal{X}, (lc, lc') \in I &\Rightarrow (u_{lc}, u_{lc'}) \in \mathcal{I} \\ Alldiff(\mathcal{X}) \end{aligned}$$

Notons bien que le graphe GLC étant non orienté la notation (lc, lc') désigne une paire de valeurs non ordonnée (comme déjà précisé). De plus, le fait que le graphe GLC soit biparti relativement à la partition $\mathcal{L} \cup \mathcal{C}$ de ses sommets nous permet de restreindre la définition des domaines de chaque variable. En lien avec les remarques de la fin de la

section 2.3, nous introduisons des contraintes supplémentaires afin de mieux établir les liens entre motifs de positions et motifs de lignes/colonnes.

Nous définissons deux relations d'ordre sur les lignes et les colonnes induites naturellement par les deux relations d'alignement $<_h$ et $<_v$ en premier lieu sur les lignes et colonnes complètes :

$$\begin{aligned} \forall l, l' \in \mathcal{L}^{comp}, l <_v l' &\Leftrightarrow \exists t \in l \wedge t' \in l', t <_v t' \\ \forall c, c' \in \mathcal{C}^{comp}, c <_h c' &\Leftrightarrow \exists t \in c \wedge t' \in c', t <_h t' \end{aligned}$$

Cet ordre s'étend alors de manière canonique aux lignes et colonnes par :

$$\begin{aligned} \forall l, l' \in \mathcal{L}, l <_v l' &\Leftrightarrow \exists l^c, l'^c \in \mathcal{L}^{comp}, l \subseteq l^c \wedge l' \subseteq l'^c \wedge l^c <_v l'^c \\ \forall c, c' \in \mathcal{C}, c <_h c' &\Leftrightarrow \exists c^c, c'^c \in \mathcal{C}^{comp}, c \subseteq c^c \wedge c' \subseteq c'^c \wedge c^c <_h c'^c \end{aligned}$$

Nous étendons alors le motif $MLC = (TL \cup TC, I)$ en un motif de lignes/colonnes ordonnées $MLC^O = (TL \cup TC, I, O)$ avec $O \subseteq (TL \times TL) \cup (TC \times TC)$, un ensemble de couples décrivant les relations d'ordres devant être respectées par les lignes et colonnes entre elles. Nous pouvons alors ajouter la contrainte d'ordre suivante à notre modèle :

$$\forall u_{lc}, u_{lc'} \in \mathcal{X}, (lc, lc') \in O \Rightarrow (u_{lc} <_v u_{lc'} \vee u_{lc} <_h u_{lc'})$$

3.3 Algorithme de filtrage

L'un de nos objectifs est de mesurer l'impact d'un algorithme de filtrage spécifique sur la recherche de motifs en fonction de nos différents modèles. Nous nous appuyons sur la version simple de l'algorithme *Iterative Labeling Filtering* (ILF) [27]. Cet algorithme de filtrage repose globalement sur un étiquetage des noeuds des graphes en fonction de relations avec les autres noeuds. Comme la plupart des algorithmes de filtrage le processus est itératif jusqu'à l'obtention d'un point fixe.

Nous allons préciser cet algorithme dans le cadre de la recherche d'un motif de positions. Par conséquent, on se donne en entrée un graphe $GP = (\mathcal{T}, \mathcal{A}, \mathcal{E})$, un motif associé $MP = (T, A, E)$ et le CSP associé $(\mathcal{X}, D, \mathcal{C})_P$ afin de pouvoir poser une contrainte globale d'isomorphisme de sous-graphes $SIP(GP, MP, \mathcal{X})$ telle que définie dans [27].

Pour étiqueter les sommets du graphe $GP = (\mathcal{T}, \mathcal{A}, \mathcal{E})$ on définit une fonction totale d'étiquetage (qui n'a pas de lien avec \mathcal{E}) $\alpha : \mathcal{T} \cup T \rightarrow \mathbb{L}$, où \mathbb{L} est un ensemble de labels.

GP et MP étant des graphes orientés, on définit classiquement pour tout sommet $t \in T \cup \mathcal{T}$, $deg^+(t)$ et $deg^-(t)$ son degré sortant et entrant. A l'aide de ces degrés, nous construisons alors un ordre partiel² $\preceq \subseteq \mathbb{L} \times \mathbb{L}$ défini ainsi pour $l = \alpha(t)$ et $l' = \alpha(t')$:

$$l \preceq l' \Leftrightarrow deg^-(t) \leq deg^-(t') \wedge deg^+(t) \leq deg^+(t')$$

2. D'autres mesures peuvent être utilisées pour définir cet ordre partiel. Toutefois, après plusieurs expérimentations basées sur les cycles et les distances, nous conservons le degré.

Cet étiquetage permettra ainsi de déterminer les couples d'associations compatibles entre les sommets du graphe motif et ceux du graphe de positions cible.

$$CC = \{(t, t') \in T \times \mathcal{T} \mid \alpha(t) \preceq \alpha(t')\}$$

Il est ainsi possible de filtrer les domaines des variables de \mathcal{X} à l'aide de CC , en supprimant du domaine $D(x_i)$ les valeurs $t' \in \mathcal{T}$ telles que $(t, t') \notin CC$. Dans l'algorithme ILF, cet étiquetage est ensuite itéré sur les voisins des sommets de $T \cup \mathcal{T}$ au moyen d'un ordre multi-ensemble et ce jusqu'à obtention d'un point fixe dans une séquence classique de filtrages successifs des domaines. Nous avons donc repris³ l'algorithme ILF [27] en y introduisant le fait que pour GP , le graphe est dirigé et pour GLC , le graphe est biparti, comme nous le verrons dans les expérimentations de la section suivante.

4 Analyse expérimentale

L'objectif de cette section est de comparer les deux modèles complémentaires proposés en étudiant la recherche d'un motif similaire dans les deux représentations. Nous souhaitons également mesurer l'impact de l'algorithme de filtrage ILF décrit plus haut.

4.1 Configuration des expérimentations

Nous proposons un générateur d'instances consistant à créer deux relations d'alignement à partir d'une matrice binaire M de dimension 10×10 . On définit alors nos tokens comme étant les valeurs non nulles de cette matrice et nos deux relations $<_h$ et $<_v$ en fonction des positions des tokens sur les lignes et colonnes de M . Un paramètre N correspondra au nombre de tokens présents sur les 100 positions possibles. Les graphes de positions GP^{red} et graphes lignes/colonnes GLC correspondant à l'espace de positionnement induit par M seront donc les graphes cibles pour nos expérimentations, au sein desquels nous chercherons différents motifs. Les paramètres des deux exemples choisis pour notre illustration expérimentale sont précisés dans la table 1. Nous avons choisi ici de restreindre cette expérimentation illustrative à des graphes de position réduits (i.e., sans considérer la transitivité des relations). Cela signifie que les motifs recherchés ne peuvent coïncider qu'avec des sous-graphes cibles ne faisant pas intervenir cette transitivité. Dans ce cas, nous pourrions utiliser pour GP^{red} un algorithme simple de recherche de motif en lieu et place de la PPC. Toutefois, notre objectif n'est pas ici la recherche de l'efficacité de la résolution mais l'utilisation des différents modèles de représentation dans un cadre unifié.

Nous nous limiterons ici à la présentation de deux matrices, avec des densités de tokens variées. Toutefois, des expériences similaires sur d'autres instances du même type révèlent des caractéristiques identiques.

Nous proposons alors de rechercher les motifs suivants :

Cible (GP/GLC)	# sommets	# arcs/arêtes
GP1	50	80
GLC1	311	3346
GP2	40	60
GLC2	210	1386

TABLE 1 – Paramètres des différents graphes cibles .

- carré : motif de taille 2×2 où tous les tokens sont présents et contenant donc deux lignes et deux colonnes (illustré sur la figure 5).
- pan : motif identique au motif carré avec un token supplémentaire sur la première colonne, constituant ainsi une troisième ligne réduite à un seul token. La figure 6 illustre ce motif pan présenté comme un motif de positions et comme un motif lignes/colonnes correspondant.
- gap : motif de dimension 3×2 comportant un token manquant situé à l'intersection de la deuxième ligne et de la première colonne.
- net : un motif de type carré de dimension 3×3 .

Nous avons sélectionné ces motifs car ils nous permettent d'exhiber différents cas pour les ensembles de solution calculés, notamment absence de solutions, en fonction des paramètres du graphe cible. Ils sont par ailleurs liés aux problèmes rencontrés lors des travaux sur l'extraction de tableaux dans des documents, d'où leurs formes assez spécifiques.

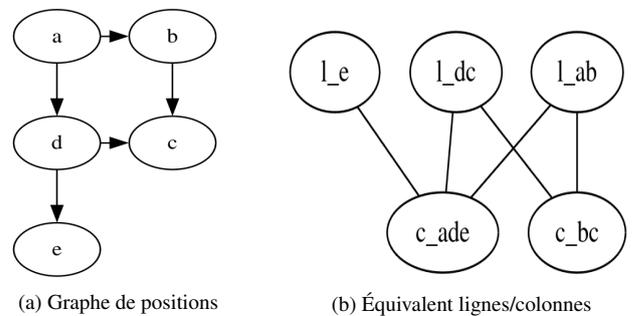


FIGURE 6 – Motif de positions pan 6a et motif lignes/colonnes induit 6b. Les labels des sommets correspondant aux lignes et colonnes utilisent la même convention que précédemment (type(l/c)) suivi du nom des sommets du motif qui sont dans la ligne/colonne correspondante.

Pour évaluer l'efficacité du solveur dans la résolution du problème, nous utilisons les mesures classiques suivantes [2] :

- Sollicitations effectives (effs) : le nombre de sollicitations d'un propagateur ayant eu un effet tangible sur la recherche.
- Affectations échouées (fails) : le nombre de fois où une affectation a été suivie d'un filtrage conduisant immédiatement à un conflit.

3. Pour des raisons d'espace, nous ne détaillons pas complètement cet algorithme ici et nous renvoyons le lecteur vers l'article original pour plus de précisions.

- Décisions erronées (Wrags) : cette mesure quantifie le nombre de choix incorrects effectués par le solveur. Notamment, cette mesure prend la valeur 0 si le solveur explore l'espace de recherche sans commettre d'erreur.

Toutes les expérimentations ont été menées en utilisant PyCSP3 [12] et le solveur ACE [11] dans une configuration de base.

4.2 Résultats

Motifs de positions : La table 2 correspond aux résultats obtenus pour la recherche de motifs sur les graphes de positions *GP1* et *GP2* avec nos quatre motifs. La colonne **ILF** indique l'utilisation (+) ou non (-) du filtrage ILF lors de la résolution. La colonne **Sol** correspond au nombre de solutions trouvées. Remarquons que les solutions trouvées ici correspondent bien directement aux motifs attendus (ce qui ne sera pas le cas pour les motifs lignes/colonnes). L'indication domaine vide (ILF) correspond à des situations où le filtrage initial permet de réduire directement certains domaines à vide et donc de montrer l'absence de solution.

Cible	Motif	ILF	Effs	Fails	Wrags	Sol	
GP1	pan	+	3227	12	28	5	
		-	31282	84	85	5	
	carré	+	3606	11	33	8	
		-	22541	81	82	8	
	net	+	domaine vide (ILF)				0
		-	26809	42	43	0	
	gap	+	6097	61	62	2	
		-	26188	69	70	2	
GP2	pan	+	344	1	1	5	
		-	5003	12	38	5	
	carré	+	527	5	7	5	
		-	4020	12	38	5	
	net	+	domaine vide (ILF)				0
		-	6329	32	33	0	
	gap	+	1470	19	20	0	
		-	3328	32	33	0	

TABLE 2 – Tableau des résultats sur les graphes de positions

Motifs de lignes/colonnes : La table 3 correspond aux résultats obtenus pour la recherche de motifs sur les graphes de lignes/colonnes *GLC1* et *GLC2* avec nos quatre motifs (dans leurs versions de motifs de lignes/colonnes). Les colonnes **ILF** et **Sol** ont naturellement les mêmes significations que précédemment. La colonne **Typage** correspond à l'utilisation (+) ou non (-), déjà évoquée, du type des sommets (ligne ou colonne) pour filtrer les domaines (spécificité du problème d'isomorphisme dans le cas d'un graphe biparti). La colonne **Ordre** correspond à l'ajout (+) ou non (-) de la contrainte d'ordre sur les lignes/colonnes (expliquée dans la sous-section 2.3). Nous observons tout naturellement que l'ajout de **Typage** et **Ordre** réduit le nombre de solutions en éliminant des solutions qui, en réalité, ne correspondent à l'un des motifs de positions calculé. A contra-

rio, en l'absence de **Typage** et **Ordre** certaines configurations ne peuvent être résolues. Dans ce cas, les champs des différentes colonnes affichent la mention "out of memory", indiquant que le solveur a dû faire face à un dépassement de capacité mémoire.

Analyse des résultats : Comme annoncé, ces expérimentations constituent une première analyse computationnelle de notre modélisation. Elles nous permettent toutefois d'observer les phénomènes suivants :

- **Impact d'ILF sur la recherche de motif dans GP :** Dans la table 2, nous observons clairement que l'utilisation du filtrage réduit considérablement la taille de l'arbre de recherche. Il est intéressant de noter que pour le motif net (pour lequel il n'y a pas de solution), ILF suffit à générer des domaines vides. En outre, nous avons observé que la prise en compte de l'orientation des arcs améliore la capacité de filtrage de l'algorithme.
- **Impact d'ILF sur la recherche de motif dans GLC :** Dans le cas des motifs lignes/colonnes, l'introduction des spécificités du graphe biparti dans le modèle (type des sommets et ordre comme mentionné dans la section 3.2) permet de réduire les domaines, ce qui rend l'utilisation d'ILF inutile.
- **Comparaisons des solutions obtenues sur GP et GLC :** Nous pouvons tout d'abord vérifier, que lorsqu'il n'y a pas de solutions, les deux modèles coïncident bien. Les différences notables entre les nombres de solutions obtenues pour un motif lignes/colonnes par rapport à un motif de positions correspondant est d'une part due à la notion d'ordre qui a déjà été illustrée sur notre exemple dans la section 2.3, mais également car le motif lignes/colonnes ne prend pas en compte la cardinalité des lignes et colonnes recherchées. Plus précisément certaines solutions obtenues correspondent à des lignes colonnes contenant plus (ou moins) de tokens que le motif de positions initial. La recherche d'un motif lignes/colonnes peut être vue comme une étape intermédiaire. Les solutions obtenues doivent ensuite être restreintes via une contrainte assurant une cohérence entre le nombre de tokens présents dans les lignes et colonnes du motif de positions et la cardinalité implicite des lignes et colonnes du motif lignes/colonnes. L'équivalence existentielle mise en avant par la propriété 2 ne nous garantit en effet pas directement, comme c'est le cas pour l'ordre des lignes et colonnes, une correspondance stricte des solutions. L'ajout d'une contrainte de cardinalité nous a permis de valider cette extension et constitue une piste de développement future pour l'amélioration du filtrage. Remarquons par ailleurs que, en utilisant les contraintes de typage et d'ordre sur les lignes/colonnes, si le nombre de solutions est important, l'arbre de résolution est toutefois plus compact que pour la recherche du motif de positions.

Cible	Motif	ILF	Typage	Ordre	Effs	Fails	Wrgrs	Solutions
GLC1	pan	+	-	-	out of memory			
			+	-	2573	42	58	4236
			+	+	1408	35	46	1972
		-	-	-	out of memory			
			+	-	out of memory			
			+	+	1408	35	46	1972
	carré	+	-	-	253173	148	168	2525552
			+	-	128873	107	128	1262776
			+	+	746	10	13	2471
		-	-	-	253173	148	168	2525552
			+	-	128873	107	128	1262776
			+	+	746	10	13	2471
	net	+	-	-	1300	154	155	0
			+	-	1	0	0	0
			+	+	1	0	0	0
		-	-	-	1325	155	156	0
			+	-	1	0	0	0
			+	+	1	0	0	0
gap	+	-	-	out of memory				
		+	-	2155	41	51	5068	
		+	+	811	13	18	1675	
	-	-	-	out of memory				
		+	-	out of memory				
		+	+	811	13	18	1675	
GLC2	pan	+	-	-	out of memory			
			+	-	581834	3205	3787	3377800
			+	+	451	34	38	186
		-	-	-	out of memory			
			+	-	585407	3347	3959	3377800
			+	+	453	34	38	186
	carré	+	-	-	67135	128	161	345024
			+	-	33302	62	67	172512
			+	+	361	22	23	395
		-	-	-	67135	128	161	345024
			+	-	33302	62	67	172512
			+	+	361	22	23	395
	net	+	-	-	875	100	101	0
			+	-	1	0	0	0
			+	+	1	0	0	0
		-	-	-	944	108	109	0
			+	-	1	0	0	0
			+	+	1	0	0	0
gap	+	-	-	out of memory				
		+	-	582308	2378	3029	3377800	
		+	+	252	7	7	192	
	-	-	-	out of memory				
		+	-	568372	3093	3656	3377800	
		+	+	252	7	7	192	

TABLE 3 – Tableau des résultats sur les graphes lignes/colonnes

5 Perspectives

Ce travail constitue une première étape dans la définition d'un formalisme de description d'espaces de positionnements d'objets via des relations simples d'alignements qui peut trouver diverses applications potentielles. La définition de plusieurs modèles nous a permis de présenter différentes approches possibles pour rechercher des motifs au sein de ces espaces de positionnement. En mettant en avant des correspondances entre ces représentations, nous avons également observé qu'elles ne visaient pas forcément, dans leurs versions initiales, les mêmes ensembles de solutions (certaines pouvant être vues comme des relaxations). Nous avons pu toutefois préciser les contraintes additionnelles nécessaires pour extraire précisément les structures recherchées.

Les perspectives sont nombreuses, tant du point de vue de l'extension du formalisme pour affiner les équivalences et les complémentarités entre les différents modèles de graphes, que du point de vue de la résolution du problème en insérant des contraintes telles que la cardinalité au niveau du filtrage et en définissant, par exemple, une approche de résolution multi-niveaux, faisant appel aux différents modèles de manière entrelacée.

Remerciements : Nous remercions les relecteurs pour leurs remarques et suggestions pertinentes qui ont contribué à identifier des pistes de recherche futures.

Références

- [1] James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11) :832–843, November 1983.
- [2] Christian Bessiere, Bruno Zanuttini, and Cèsar Fernández. Measuring Search Trees. In Hnich B. and Walsh T., editors, *Workshop on Modelling and Solving Problems with Constraints - ECAI'2004*, pages 31–40, Valencia, Spain, 2004. Hnich B. Workshop held in conjunction with the 16th European Conference on Artificial Intelligence (ECAI 2004).
- [3] Sanket Biswas, Pau Riba, Josep Lladós, and Uma-pada Pal. Graph-based deep generative modelling for document layout generation. *CoRR*, abs/2107.04357, 2021.
- [4] Anthony G. Cohn, Brandon Bennett, John Gooday, and Nicholas Mark Gotts. Qualitative spatial representation and reasoning with the region connection calculus. *GeoInformatica*, 1(3) :275–316, 1997.
- [5] Luigi P. Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(10) :1367–1372, 2004.
- [6] M. R. Garey and David S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [7] Andrea Gemelli, Emanuele Vivoli, and Simone Marinai. Graph neural networks and representation em-

- bedding for table extraction in PDF documents. In *26th International Conference on Pattern Recognition, ICPR 2022, Montreal, QC, Canada, August 21-25, 2022*, pages 1719–1726. IEEE, 2022.
- [8] Gael Glorian, Jean-Marie Lagniez, Valentin Montmairail, and Michael Sioutis. Une approche SAT incrémentale pour raisonner efficacement sur les réseaux de contraintes qualitatives. In *JFPC 2019 - Actes des 15es Journées Francophones de Programmation par Contraintes*, Albi, France, June 2019.
- [9] Bo Huang, Thomas Cova, Ming-Hsiang Tsou, Georg Bareth, Chunqiao Song, Yan Song, Kai Cao, and Elisabete Silva. *Comprehensive Geographic Information Systems*. 08 2017.
- [10] Lars Kotthoff, Ciaran McCreesh, and Christine Solnon. Portfolios of subgraph isomorphism algorithms. In Paola Festa, Meinolf Sellmann, and Joaquin Vanschoren, editors, *Learning and Intelligent Optimization - 10th International Conference, LION 10, Ischia, Italy, May 29 - June 1, 2016, Revised Selected Papers*, volume 10079 of *Lecture Notes in Computer Science*, pages 107–122. Springer, 2016.
- [11] Christophe Lecoutre. Ace, a generic constraint solver, 2024.
- [12] Christophe Lecoutre and Nicolas Szczepanski. Pycsp3 : Modeling combinatorial constrained problems in python, 2024.
- [13] Alan K. Mackworth. Interpreting pictures of polyhedral scenes. In Nils J. Nilsson, editor, *Proceedings of the 3rd International Joint Conference on Artificial Intelligence. Stanford, CA, USA, August 20-23, 1973*, pages 556–563. William Kaufmann, 1973.
- [14] Ciaran McCreesh, Patrick Prosser, and James Trimble. The glasgow subgraph solver : Using constraint programming to tackle hard subgraph isomorphism problem variants. In Fabio Gadducci and Timo Kehler, editors, *Graph Transformation - 13th International Conference, ICGT 2020, Held as Part of STAF 2020, Bergen, Norway, June 25-26, 2020, Proceedings*, volume 12150 of *Lecture Notes in Computer Science*, pages 316–324. Springer, 2020.
- [15] Ugo Montanari. Networks of constraints : Fundamental properties and applications to picture processing. *Inf. Sci.*, 7 :95–132, 1974.
- [16] Charalampos Nikolaou and Manolis Koubarakis. Fast consistency checking of very large real-world rcc-8 constraint networks using graph partitioning. *Proceedings of the National Conference on Artificial Intelligence*, 4 :2724–2730, 06 2014.
- [17] Jean-Charles Régin. A filtering algorithm for constraints of difference in cps. In Barbara Hayes-Roth and Richard E. Korf, editors, *Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, WA, USA, July 31 - August 4, 1994, Volume 1*, pages 362–367. AAAI Press / The MIT Press, 1994.
- [18] Jochen Renz and Bernhard Nebel. On the complexity of qualitative spatial reasoning : A maximal tractable fragment of the region connection calculus. *Artif. Intell.*, 108(1-2) :69–123, 1999.
- [19] Pau Riba, Anjan Dutta, Lutz Goldmann, Alicia Fornés, Oriol Ramos Terrades, and Josep Lladós. Table detection in invoice documents by graph neural networks. In *2019 ICDAR*. IEEE, 2019.
- [20] Mahmoud Salaheldin Kasem, Abdelrahman Abdallah, Alexander Berendeyev, Ebrahim Elkady, Mohamed Mahmoud, Mahmoud Abdalla, Mohamed Hamada, Sebastiano Vascon, Daniyar Nurseitov, and Islam Taj-Eddin. Deep learning for table detection and structure recognition : A survey. *ACM Comput. Surv.*, 56(12), October 2024.
- [21] Thomas Saout, Frédéric Lardeux, and Frédéric Saubion. A two-stage approach for tables extraction in invoices. In *35th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2023, Atlanta, GA, USA, November 6-8, 2023*, pages 10–15. IEEE, 2023.
- [22] Thomas Saout, Frédéric Lardeux, and Frédéric Saubion. An overview of data extraction from invoices. *IEEE Access*, 12 :19872–19886, 2024.
- [23] Christine Solnon. Alldifferent-based filtering for subgraph isomorphism. *Artif. Intell.*, 174(12-13) :850–864, 2010.
- [24] Julian R. Ullmann. An algorithm for subgraph isomorphism. *J. ACM*, 23(1) :31–42, 1976.
- [25] Yang Wang, Huilin Peng, Yiwei Xiong, and Haitao Song. Spatial relationship recognition via heterogeneous representation : A review. *Neurocomputing*, 533 :116–140, 2023.
- [26] Yiyuan Wang, Chenghou Jin, and Shaowei Cai. Pathlad+ : Towards effective exact methods for subgraph isomorphism problem. *Artif. Intell.*, 337 :104219, 2024.
- [27] Stéphane Zampelli, Yves Deville, and Christine Solnon. Solving subgraph isomorphism problems with constraint programming. *Constraints*, 15(3) :327–353, 2010.