

Prise en compte de motifs et génération de structures de benzénoïdes*

Yannick Carissan¹ Denis Hagebaum-Reignier¹

Nicolas Prcovic² Cyril Terrioux² Adrien Varet²

¹ Aix Marseille Univ, CNRS, Centrale Marseille, ISM2, Marseille, France

² Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
{prénom.nom}@univ-amu.fr

Résumé

Les benzénoïdes sont une sous-famille d'hydrocarbures (molécules composées uniquement d'atomes d'hydrogène et de carbone) dont les atomes de carbone forment des hexagones. Ces molécules ont fait l'objet de nombreuses études en chimie théorique et peuvent posséder différentes propriétés physico-chimiques (résistance mécanique, conductivité électronique, ...) desquelles découlent de nombreuses applications concrètes. Ces propriétés peuvent notamment reposer sur l'existence ou l'absence de fragments de la molécule correspondant à un motif donné (certains motifs imposent la nature de certaines liaisons, ce qui a un impact sur la structure électronique totale). Générer des structures de benzénoïdes tout en maîtrisant la présence ou non d'un certain motif constitue donc une problématique importante en chimie théorique.

Dans cet article, nous montrons comment la programmation par contraintes peut aider les chimistes à répondre à différentes questions autour de cette problématique. Pour ce faire, nous proposons différentes modélisations dont une basée sur une variante du problème d'isomorphisme de sous-graphes et nous générons les structures souhaitées à l'aide du solveur Choco.

Abstract

Benzenoids are a subfamily of hydrocarbons (molecules that are only made of hydrogen and carbon atoms) whose carbon atoms form hexagons. These molecules are widely studied in theoretical chemistry and can have various physicochemical properties (mechanical resistance, electronic conductivity, ...) from which a lot of concrete applications are derived. These properties can rely on the existence or absence of fragments of the molecule corresponding to a given pattern (some patterns impose the nature of certain bonds, which has an impact on the whole electronic structure). Generating benzenoid

structures while controlling the presence or absence of a certain pattern is therefore an important problem in theoretical chemistry.

In this paper, we show how constraint programming can help chemists to answer different questions around this problem. To do so, we propose different models including one based on a variant of the subgraph isomorphism problem and we generate the desired structures using the Choco solver.

1 Introduction

Les hydrocarbures aromatiques polycycliques (HAP) sont des hydrocarbures dont les atomes de carbone forment des cycles de différentes tailles (de taille 6 dans le cas des benzénoïdes). Ils ont été très étudiés dans divers domaines (nanoélectronique moléculaire, synthèse organique, chimie interstellaire, ...) en raison de leur stabilité énergétique, de leurs structures moléculaires ou de leurs spectres optiques. Ils présentent une grande variété de propriétés physico-chimiques en fonction de leur taille et de leur structure. Par exemple, ils peuvent allier une forte résistance mécanique à une conductivité électronique élevée. Ces propriétés peuvent notamment reposer sur l'existence ou l'absence de fragments de la molécule correspondant à un motif donné. Certains motifs imposent la nature de certaines liaisons, ce qui impacte la structure électronique dans son ensemble. Par exemple, le pérylène (voir figure 2(c)) peut être vu comme deux triangles de trois cycles fusionnés se chevauchant avec pour conséquence que les liaisons centrales ont un caractère essentiellement simple. Ces dernières années, les motifs et les propriétés qui en découlent se retrouvent au cœur de nombreux travaux en chimie théorique [12, 4, 9]. Ainsi, être capable de générer des structures de benzénoïdes tout en maîtrisant

*Ce travail est soutenu par l'Agence Nationale de la Recherche dans le cadre du projet DEMOGRAPH (ANR-16-CE40-0028).

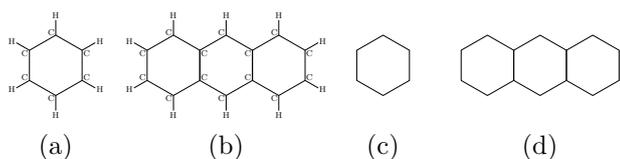


FIGURE 1 – Exemples de benzénoïdes : le benzène (a) et l’anthracène (b) accompagnés de leur représentation graphique (c) et (d).

la présence ou non d’un certain motif constitue donc une problématique importante en chimie théorique.

Dans la littérature, des approches ad-hoc ont été proposées pour générer des structures quelconques ou satisfaisant des propriétés particulières [1]. Elles s’avèrent très efficaces en pratique mais sont difficiles à adapter aux besoins des chimistes et ne considèrent que des propriétés portant sur l’ensemble de la molécule. Dans [2], nous avons proposé une nouvelle approche basée sur la PPC qui est plus flexible tout en étant compétitive. Cette approche ne traitant initialement que de propriétés globales, nous décrivons, dans ce papier, comment y intégrer la notion de motif. Plusieurs modélisations étant envisageables, nous les étudions sur la base des différentes questions se posant autour des motifs avant de les comparer expérimentalement.

Cet article est organisé comme suit. La section 2 introduit les notions nécessaires à la compréhension du papier. Puis, dans la section 3, nous rappelons comment générer des structures de benzénoïdes, notamment à l’aide de la PPC. Ensuite, nous formalisons la problématique qui nous intéresse dans la section 4 et abordons différentes questions pouvant en découler dans les sections 5 à 7. Enfin, nous comparons expérimentalement certaines modélisations, dans la section 8, avant de conclure dans la section 9.

2 Préliminaires

2.1 Chimie théorique

Le benzène (représenté à la figure 1(a)) est une molécule constituée de 6 atomes de carbone et de 6 atomes d’hydrogène. Ses atomes de carbone forment un hexagone (appelé *cycle benzénique*) et vont chacun être reliés à un atome d’hydrogène. Les *benzénoïdes* sont une sous-famille des *HAP* qui regroupe toutes les molécules obtenues en fusionnant des cycles benzéniques. La figure 1(b) présente, par exemple, l’anthracène qui est constitué de trois cycles benzéniques.

Les atomes établissent entre eux des liaisons qui peuvent notamment être simples ou doubles selon le nombre d’électrons impliqués dans la liaison. Dans un benzénoïde, chaque atome de carbone est relié soit à deux atomes de carbone et un atome d’hydrogène,

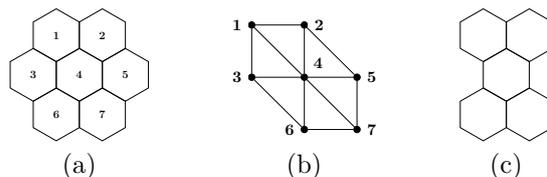


FIGURE 2 – Le coronène (a), son graphe d’hexagones (b) et le pérylène (c).

soit à trois atomes de carbone. Dans la suite du propos, les atomes d’hydrogène ne jouent aucun rôle (et leur présence peut être déduite si nécessaire). Aussi, ils peuvent être omis dans la représentation que nous allons considérer. Ainsi, un benzénoïde peut être représenté comme un graphe non orienté $B = (V, E)$ dans lequel chaque sommet de V correspond à un atome de carbone et chaque arête de E traduit l’existence d’une liaison entre les deux atomes de carbone correspondant. Ce graphe est connexe, biparti et planaire. Les figures 1(c) et (d) présentent les graphes correspondant au benzène et à l’anthracène.

Par ailleurs, les benzénoïdes pouvant se définir comme une combinaison de cycles benzéniques fusionnés, nous considérons, pour chaque benzénoïde B , un deuxième graphe appelé *graphe d’hexagones*. Ce graphe $B_h = (V_h, E_h)$ est un graphe non orienté dont chaque sommet correspond à un hexagone (c’est-à-dire à un cycle benzénique) de B et tel que deux sommets sont reliés par une arête si les hexagones correspondants partagent une arête dans le graphe B . La figure 2 présente le graphe correspondant au coronène ainsi que son graphe d’hexagones.

2.2 Programmation par contraintes

Une instance *CSP* (*Constraint Satisfaction Problem*) P se définit comme un triplet (X, D, C) . $X = \{x_1, \dots, x_n\}$ est un ensemble de n variables. À chaque variable x_i , on associe un domaine D_{x_i} , issu de $D = \{D_{x_1}, \dots, D_{x_n}\}$, qui contient les valeurs possibles pour x_i . $C = \{c_1, \dots, c_e\}$ désigne un ensemble de e *contraintes* qui traduisent les interactions entre les variables et définissent les combinaisons de valeurs autorisées. Résoudre une instance *CSP* $P = (X, D, C)$ revient à essayer de trouver une affectation de toutes les variables de X qui ne viole aucune contrainte de C . Il s’agit d’un problème NP-difficile.

De nombreux outils existent pour modéliser et résoudre efficacement des instances *CSP*. Par la suite, nous optons pour l’outil *Choco* [5]. Ce choix se justifie par la possibilité offerte par *Choco* de pouvoir définir des variables de graphe et de disposer de contraintes prédéfinies adaptées à ces variables (comme par exemple des contraintes de connexité ou de cyclicité). Ces va-

riables ont comme domaine des ensembles de graphes définis par une borne inférieure (un sous-graphe, noté *GLB*) et une borne supérieure (un super-graphe, noté *GUB*). De plus, Choco permet une modélisation assez simple et rapide d’un grand nombre de problèmes et le solveur est paramétrable à la guise de l’utilisateur.

3 Génération de benzénoïdes

La génération de benzénoïdes ayant certaines propriétés structurales (par exemple, ayant un nombre donné d’hexagones ou ayant une structure particulière du point de vue graphique) est un problème intéressant et important en chimie théorique [3, 7, 8, 10]. Ce problème constitue une étape préliminaire à l’étude de leurs propriétés chimiques. Il peut se définir formellement ainsi : Étant donné un ensemble de propriétés structurales \mathcal{P} , générer toutes les structures de benzénoïdes satisfaisant les propriétés de \mathcal{P} . Ces propriétés peuvent porter sur le nombre de carbones ou d’hexagones ou sur des formes particulières du graphe d’hexagones (arbre, rectangle, présence de « trous », ...).

Dans la littérature, des méthodes ad-hoc ont été proposées (par exemple [1]). Si elles s’avèrent souvent efficaces en pratique, elles possèdent l’inconvénient d’être difficilement adaptables aux besoins des chimistes. Dans [2], nous avons proposé une modélisation CSP de ce problème et montré combien il était facile de répondre aux souhaits exprimés par les chimistes en spécialisant ce modèle par simple ajout de variables et de contraintes. Par ailleurs, au-delà de sa flexibilité, cette approche se révèle relativement rapide du fait de l’efficacité du solveur Choco.

Nous rappelons maintenant la modélisation CSP permettant de générer toutes les structures ayant n hexagones. Elle repose sur la propriété que tout benzénoïde de n hexagones peut être placé dans un coronénoïde de taille au plus $k(n) = \lfloor \frac{n}{2} + 1 \rfloor$. Un coronénoïde de taille k est une molécule de benzène à laquelle on a ajouté successivement $k - 1$ couronnes d’hexagones. Le coronène (voir figure 2) est le coronénoïde de taille 2. La figure 3 présente les coronénoïdes de tailles 3 et 4. Par la suite, on note $B_h^{c(k(n))}$ le graphe d’hexagones du coronénoïde $c(k(n))$ de taille $k(n)$, n_c son nombre d’hexagones et m_c celui d’arêtes. Nous numérotions arbitrairement les hexagones de $B_h^{c(k(n))}$ à partir de 1. Tout d’abord, dans notre modèle (noté \mathcal{M}), nous considérons une variable de graphe x_G pour représenter le graphe d’hexagones de la structure souhaitée. Son domaine est l’ensemble de tous les sous-graphes entre le graphe vide et $B_h^{c(k(n))}$. L’intérêt d’utiliser des variables de graphe est de pouvoir exprimer certaines notions (comme la connexité) de façon plus aisée et plus compacte. Nous exploitons également un ensemble de n_c variables booléennes $\{x_1, \dots, x_{n_c}\}$. La variable

x_i vaut 1 si le i -ème hexagone de $B_h^{c(k(n))}$ est utilisé dans x_G , 0 sinon. De même, nous considérons un ensemble de m_c variables booléennes $y_{i,j}$. La variable $y_{i,j}$ vaut 1 si l’arête $\{i, j\}$ de $B_h^{c(k(n))}$ est utilisée dans x_G , 0 sinon.

Ensuite, nous modélisons les propriétés suivantes à l’aide de contraintes :

- *Lien entre x_G et x_i (resp. $y_{i,j}$)* : nous utilisons une contrainte **channeling** qui impose $x_i = 1 \iff x_G \text{ contient le sommet } i$ (resp. $y_{i,j} = 1 \iff x_G \text{ contient l'arête } \{i, j\}$).
- *x_G est un sous-graphe induit de $B_h^{c(k(n))}$* : Toute valeur de x_G n’est pas nécessairement un graphe d’hexagones valide. Pour garantir sa validité, il doit correspondre à un sous-graphe de $B_h^{c(k(n))}$ induit par les sommets appartenant à x_G . Ainsi, pour chaque arête $\{i, j\}$ de $B_h^{c(k(n))}$, on ajoute une contrainte $x_i = 1 \wedge x_j = 1 \iff y_{i,j} = 1$. En d’autres termes, l’arête $\{i, j\}$ existe dans x_G si et seulement si les sommets i et j apparaissent dans x_G .
- *La structure a n hexagones* : $\sum_{i \in \{1, \dots, n_c\}} x_i = n$.
- *Le graphe d’hexagones est connexe* : nous appliquons la contrainte de graphe **connected** sur x_G .
- *Six hexagones formant un cycle génèrent un hexagone (et non un trou)* : Pour chaque hexagone u , nous considérons l’ensemble $N(u)$ de voisins de u dans le graphe d’hexagones. Pour chaque sommet u , on pose la contrainte $\sum_{v \in N(u)} x_v = 6 \implies x_u = 1$.

Enfin, nous ajoutons plusieurs contraintes pour éviter les redondances. D’abord x_G doit avoir au moins un sommet sur le bord supérieur (resp. gauche) de $B_h^{c(k(n))}$ afin d’éviter les symétries par translation. Nous posons donc une contrainte qui spécifie que la somme des variables binaires x_i associées au bord supérieur (resp. gauche) est strictement positive. Ensuite, il faut s’assurer que le graphe décrit par x_G est le seul représentant de sa classe de symétrie. Il existe jusqu’à douze solutions symétriques : six symétries de rotation de 60 degrés combinées à une éventuelle symétrie axiale. Ces symétries sont cassées grâce à la contrainte **lex-lead**. Pour chacune des douze symétries, il faut ajouter n_c variables booléennes (chacune associée à une variable x_i) et un total de $3.n_c$ clauses ternaires.

Ce modèle peut facilement être mis en œuvre avec le solveur Choco. Il peut également être spécialisé pour prendre en compte les besoins des chimistes en ajoutant des variables et/ou des contraintes. Par exemple, générer des structures ayant une forme arborescente (appelées benzénoïdes *catacondensés*) nécessite simplement d’ajouter au modèle général la contrainte de graphe **tree** appliquée à x_G . D’autres propriétés ont été modélisées afin de générer des structures ayant une

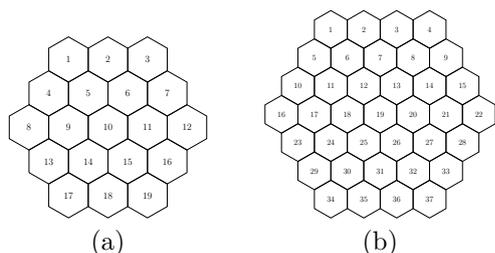


FIGURE 3 – Les coronenoïdes de taille 3 (a) et 4 (b).

forme rectangulaire, possédant un trou ou symétriques.

4 Prise en compte de motifs

Le modèle \mathcal{M} , présenté dans [2] et rappelé dans la section 3, permet d'exprimer le problème de génération de structures de benzénoïdes dans toute sa généralité. Si plusieurs spécialisations de ce modèle ont été proposées ensuite, toutes correspondent à des propriétés structurales portant sur l'ensemble de la molécule. Ces propriétés pourraient donc être qualifiées de *globales*. Toutefois, dans certains cas de figure, il peut être utile de raisonner en termes de propriétés *locales* que pourraient satisfaire ou non certaines parties (appelées *fragments*) des structures générées.

Les propriétés locales que nous considérons dans cet article peuvent être définies en « dessinant » une forme dont les briques de base sont des hexagones. Ces hexagones peuvent être de trois natures différentes :

- (i) Les hexagones *positifs* dont la présence est requise dans la propriété,
- (ii) Les hexagones *negatifs* dont l'absence est requise dans la propriété,
- (iii) Les hexagones *neutres* dont la présence ou l'absence n'ont pas d'influence sur la propriété.

Si l'utilisation des hexagones positifs est une évidence, on peut se poser la question de l'intérêt des hexagones négatifs ou neutres. Les hexagones négatifs (respectivement neutres) sont utiles, par exemple, pour indiquer qu'il n'y a rien entre deux hexagones positifs (resp. pour garantir un certain écart entre deux hexagones positifs). Afin de représenter les formes souhaitées, nous introduisons la notion de graphe d'hexagones étendu :

Définition 1 (graphe d'hexagones étendu) *Un graphe d'hexagones étendu est un graphe d'hexagones dont les sommets et les arêtes sont étiquetés par les symboles + (pour positif), - (pour négatif) et o (pour neutre) ainsi :*

- (i) Chaque sommet a pour étiquette la nature de l'hexagone qu'il représente.
- (ii) Une arête est étiquetée - si au moins un de ses sommets est étiqueté -. Sinon, elle est étiquetée +.

Comme pour les hexagones (ou les sommets), les étiquettes associées aux arêtes qualifient le statut que doit avoir l'interaction entre deux hexagones dans la propriété locale que l'on souhaite définir. Formellement une propriété locale peut être définie par un *motif* :

Définition 2 (motif) *Un motif M se définit par la donnée d'un triplet (M_+, M_-, M_o) et d'un graphe étendu d'hexagone M_h tels que :*

- (i) M_+ , M_- et M_o désignent respectivement l'ensemble des hexagones positifs, négatifs et neutres,
- (ii) ces trois ensembles sont deux à deux disjoints et
- (iii) M_h est un graphe connexe portant sur l'ensemble des hexagones de $M_+ \cup M_- \cup M_o$.

Son ordre k_M est la longueur maximale (exprimée en nombre d'arêtes) des plus court chemins de M_h séparant un hexagone négatif ou neutre d'un hexagone positif.

En d'autres termes, un motif est défini par une collection d'hexagones positifs, négatifs et neutres dont l'agencement est décrit par un graphe d'hexagones étendu. À titre d'exemple, la figure 4(a) présente un motif, appelé *baie profonde* [12], d'ordre 1 composé de quatre hexagones positifs et trois négatifs. Nous définissons enfin la notion d'inclusion de motif :

Définition 3 *Étant donné k un entier positif ou nul, on note B_h^k le graphe d'hexagones étendu représentant le benzénoïde B entouré de k couches d'hexagones négatifs (c'est-à-dire le graphe étendu de B augmenté de tous les hexagones négatifs situés à une distance au plus k d'un hexagone de B). Un fragment F^k d'ordre k d'un benzénoïde B est un sous-ensemble d'hexagones de B_h^k dont le graphe d'hexagones étendu est connexe. Il satisfait le motif M si $k = k_M$ et s'il existe une bijection qui à chaque hexagone positif de F^k associe un hexagone positif ou neutre de M et à chaque hexagone négatif de F^k associe un hexagone négatif ou neutre de M . Un benzénoïde B contient le motif M s'il possède un fragment d'ordre k_M satisfaisant M .*

Considérer B_h ou B_h^k ne change pas la nature du benzénoïde B . B_h^k permet simplement de matérialiser le vide qui l'entoure, ce qui s'avère nécessaire pour certaines propriétés. Par exemple, le benzénoïde de la figure 4(b) satisfait le motif « baie profonde » de la figure 4(a). Pour cela, nous sommes obligés de prendre en compte l'absence d'hexagone en bordure du benzénoïde pour identifier un fragment convenable.

Dans cet article, notre objectif est de générer des structures de benzénoïdes satisfaisant des propriétés locales exprimées à l'aide des motifs introduits ci-dessus. Ces propriétés locales peuvent revêtir différentes formes.

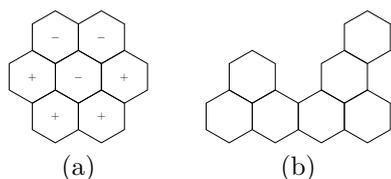


FIGURE 4 – Le motif « baie profonde »(a) et un benzénoïde satisfaisant ce motif (b).

La plus simple consiste à inclure un motif donné. Ensuite, on peut s'intéresser à généraliser l'approche en incluant plusieurs motifs différents ou un certain nombre de fois le même motif. À l'opposé, on peut également désirer exclure un motif. Les sections suivantes traitent de ces différentes questions. Dans tous les cas, l'idée est de générer des structures de benzénoïdes en partant de notre modèle général \mathcal{M} . Il s'ensuit qu'il est tout à fait possible de considérer à la fois des propriétés globales et des propriétés locales.

5 Structures incluant un motif

Soit un motif M portant sur n_M hexagones, ces hexagones pouvant être positifs, négatifs ou neutres. On numérote arbitrairement chaque hexagone du motif M de 1 à n_M . Les ensembles M_+ , M_- et M_o sont définis en conséquence. Dans cette section, nous souhaitons modéliser le problème qui consiste à générer toutes les structures de benzénoïdes ayant n hexagones et incluant le motif M . Nous considérons d'abord toutes les variables et les contraintes du modèle général \mathcal{M} auxquelles nous allons ajouter des variables et des contraintes pour exprimer le fait que le motif doit être présent dans les structures générées. À ce niveau, nous avons plusieurs possibilités selon le point de vue considéré.

5.1 Première modélisation

Nous partons du modèle \mathcal{M} et donc d'un coronénoïde de taille $k(n)$. Dans cette première modélisation (notée \mathcal{M}_{i_1}), nous identifions, en amont, tous les fragments possibles du motif M dans ce coronénoïde. Leur nombre étant en $O(c(k(n))) = O(n^2)$, ce calcul peut être réalisé efficacement à l'aide de rotations, de symétries axiales et de translations. Pour chacun de ces fragments F_i , nous définissons les ensembles F_{i+} , F_{i_o} et F_{i-} de ses hexagones positifs, neutres et négatifs. Nous associons à chaque fragment F_i une variable booléenne e_i . Puis, nous garantissons que si e_i est vraie alors le fragment F_i est présent dans la structure en cours de construction via la contrainte $e_i = 1 \Rightarrow \bigwedge_{j \in F_{i-}} x_j = 0 \wedge \bigwedge_{j \in F_{i+}} x_j = 1$.

Notons que, pour les motifs d'ordre non nul, il n'est pas nécessaire de considérer un coronénoïde de taille

supérieure. Le fragment peut être placé en bordure du coronénoïde avec des hexagones négatifs ou neutres se retrouvant hors de ce coronénoïde et, de ce fait, étant considérés comme absents. Dans ce cas, ces hexagones ne seront pas représentés dans F_{i-} , ni dans F_{i_o} , mais placés dans un ensemble F_{i*} . Enfin, nous posons la contrainte de somme $\sum_j e_j = 1$ pour garantir l'existence d'au moins un exemplaire du motif.

5.2 Deuxième modélisation

Dans cette seconde modélisation (notée \mathcal{M}_{i_2}), nous cherchons à exprimer l'existence d'un fragment correspondant au motif M en raisonnant sur le voisinage de chaque hexagone. Pour cela, partant du modèle \mathcal{M} , nous ajoutons une variable f_i par hexagone du coronénoïde de taille $k(n)$. Chaque variable f_i a pour domaine $\{0, 1, \dots, n_M\}$. La variable f_i est affectée à une valeur j positive si l'hexagone i du coronénoïde de taille $k(n)$ participe au fragment recherché en tant qu'hexagone occupant la position j dans M , 0 sinon. Ensuite, la génération des structures de benzénoïdes et la recherche d'un fragment se faisant simultanément, nous devons assurer leur concordance. Il faut, en particulier, garantir que les hexagones positifs (resp. négatifs) soient effectivement présents (resp. absents) dans la structure générée. Pour rappel, cette structure est notamment représentée par la variable de graphe x_G et par les variables booléennes x_i . Aussi, pour chaque hexagone i du coronénoïde de taille $k(n)$, on pose les contraintes suivantes (sous forme de clauses) :

- $x_i = 1 \Rightarrow f_i \in \{0\} \cup M_+ \cup M_o$,
- $x_i = 0 \Rightarrow f_i \in \{0\} \cup M_- \cup M_o$,
- $f_i \in M_+ \Rightarrow x_i = 1$,
- $f_i \in M_- \Rightarrow x_i = 0$.

Il nous faut ensuite définir la bijection permettant d'établir que le fragment construit satisfait le motif M . Autrement dit, il nous faut garantir qu'exactly n_M hexagones de la structure doivent correspondre aux n_M hexagones du motif M . Aussi, pour chaque hexagone $j \in \{1, \dots, n_M\}$ du motif, on pose la contrainte globale¹ $Count(\{f_1, \dots, f_{n_c}\}, \{j\}) = 1$ si $j \in M_+$ (≤ 1 sinon). La valeur 0 est obtenue dans le cas où un hexagone négatif ou neutre se trouve hors du coronénoïde de taille $k(n)$. En d'autres termes, le motif dépasse de ce coronénoïde mais uniquement pour des hexagones négatifs ou neutres (qui seraient alors absents). En procédant ainsi, cela évite d'introduire des variables supplémentaires (et les contraintes associées) pour représenter les k_M couches d'hexagones absents utilisés dans la définition formelle de fragment.

Il reste maintenant à définir le motif en lui-même. Pour cela, on va décrire les liens de voisinage entre

1. Pour rappel, la contrainte $Count(Y, V) \odot k$ est satisfaite ssi le nombre de variables de Y affectées avec une valeur dans V satisfait la condition vis-à-vis de l'opérateur \odot et de la valeur k .

chaque hexagone du motif. Un hexagone peut posséder jusqu'à six hexagones voisins. Pour un hexagone h donné, nous considérons ses voisins potentiels v_1 à v_6 dans le sens horaire en commençant par le voisin situé en haut à droite. À partir de là, nous listons les différentes configurations prises par les voisins selon que l'hexagone h participe au fragment ou non. Plus précisément, chaque configuration est un tuple composé d'un entier par voisin. Cet entier est une valeur j non nulle si le voisin participe au fragment en tant qu'hexagone j du motif, 0 sinon. Pour chaque position de l'hexagone h dans le motif M , nous considérons six configurations possibles afin de tenir compte des rotations de 60° du motif. Cela est nécessaire pour générer toutes les structures car le modèle \mathcal{M} impose l'existence d'hexagone(s) sur les bordures en haut et à gauche du coronénoïde considéré. Notons qu'à partir d'une configuration donnée, appliquer une rotation de 60° revient à effectuer une permutation circulaire au niveau du tuple. Par exemple, dans la table 1, nous listons toutes les configurations de voisinage possibles quand l'hexagone est en position 1 dans le motif « *baïe profonde* », la numérotation des hexagones étant celle de la figure 2(a). Pour les autres, nous n'en donnons qu'une par manque de place. Ces configurations vont être utilisées pour définir la relation associée à des contraintes de tables concises [11]. Nous considérons une telle contrainte par hexagone h du coronénoïde de taille $k(n)$ avec pour portée la variable f_h et chaque variable f_i associée à un voisin de h dans $B_h^{c(k(n))}$. Pour les hexagones en bordure du coronénoïde, on ne conserve que les lignes de la table dont les voisins participant au fragment correspondent à des hexagones (quelle que soit leur nature) à l'intérieur du coronénoïde ou à des hexagones négatifs ou neutres hors du coronénoïde. Puis, on fait une projection de ces lignes sur les voisins présents et la variable f_i .

5.3 Troisième modélisation

Un fragment d'ordre k d'un benzénoïde B correspond à un sous-graphe connexe de B_h^k . Aussi, déterminer s'il existe un fragment satisfaisant un motif M dans un benzénoïde B revient, d'une certaine manière, à déterminer s'il existe un sous-graphe dans $B_h^{k_M}$ isomorphe à M_h . Toutefois, il ne s'agit pas exactement du traditionnel problème d'isomorphisme de sous-graphes, mais, d'une de ses variantes prenant en compte l'étiquetage des sommets et des arêtes. Cela ne change en rien la complexité du problème de décision qui reste NP-complet. Fort heureusement, nous n'avons pas besoin de nous attaquer à ce problème car, dans notre approche, nous allons, par construction, produire directement des structures satisfaisant le motif.

Nous présentons maintenant notre modèle \mathcal{M}_{i_3} . Par-

f_i	f_{v_1}	f_{v_2}	f_{v_3}	f_{v_4}	f_{v_5}	f_{v_6}
0	*	*	*	*	*	*
1	0	2	4	3	0	0
1	2	4	3	0	0	0
1	4	3	0	0	0	2
1	3	0	0	0	2	4
1	0	0	0	2	4	3
1	0	0	2	4	3	0
2	0	0	5	4	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
3	1	4	6	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
4	2	5	7	6	3	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
5	0	0	0	7	4	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮
6	4	7	0	0	0	3
⋮	⋮	⋮	⋮	⋮	⋮	⋮
7	5	0	0	0	6	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮

TABLE 1 – La contrainte de table concise décrivant le voisinage pour le motif « *baïe profonde* ».

tant du modèle général \mathcal{M} , nous ajoutons une variable s_i par hexagone du motif M (quelle que soit sa nature). Chaque variable s_i a pour domaine $\{1, \dots, n'_c\}$ avec n'_c le nombre d'hexagones du coronénoïde de taille $k(n) + k_M$. Nous exploitons un coronénoïde de taille $k(n) + k_M$, et non $k(n)$, car nous avons besoin d'entourer le coronénoïde de taille $k(n)$ de k_M couronnes d'hexagones absents. Notons que cela n'a aucun impact sur la variable de graphe x_G ou sur les variables x_i car nous ajoutons des hexagones dont on sait qu'ils ne sont pas présents dans la structure considérée. La variable s_i a pour valeur j si le i -ème hexagone du motif M est le j -ème hexagone du coronénoïde de taille $k(n) + k_M$. Par convention, les valeurs de j comprises entre 1 et n_c correspondent à des hexagones présents dans le coronénoïde de taille $k(n)$. Nous ajoutons ensuite les contraintes suivantes pour exprimer la notion d'isomorphisme :

- *Injectivité* : Les hexagones participant au fragment doivent être tous différents deux à deux. Ceci est imposé grâce à la contrainte $\text{alldifferent}(\{s_1, \dots, s_{n_M}\})$. Cela assure également qu'on a bien n_M hexagones de x_G qui participent au fragment.
- *Préservation des arêtes* : On doit garantir qu'à deux sommets voisins de M_h correspondent deux sommets voisins dans le graphe d'hexagones du coronénoïde de taille $k(n)$. Aussi, pour chaque arête $\{i, i'\}$ de M_h (quelle que soit sa nature), on pose une contrainte de table portant sur s_i et $s_{i'}$ dont la relation contient tous les couples

(j, j') tels que $\{j, j'\}$ est une arête du graphe d'hexagones du coronénoïde de taille $k(n) + k_M$.

Cette partie du modèle est inspirée de la modélisation du problème d'isomorphisme de sous-graphes présenté dans [6]. Cependant, il faut noter que, dans notre cas, le graphe dans lequel le sous-graphe est recherché n'est pas connu à l'avance, car il s'agit du graphe que nous souhaitons construire. Aussi, dans notre modèle, nous contournons cette difficulté en considérant le graphe d'hexagones du coronénoïde de taille $k(n) + k_M$.

Concernant les étiquetages, par définition, l'étiquetage des arêtes découle de celui des sommets. Celui des sommets est directement pris en compte par définition des variables s_i . Il ne reste donc qu'à exprimer l'adéquation entre l'étiquetage des sommets et l'existence des hexagones grâce aux contraintes suivantes (exprimées sous forme de clauses) :

- $\forall i \in M_+, \forall j \in \{1, \dots, n_c\}, s_i = j \Rightarrow x_j = 1$,
- $\forall i \in M_+, \forall j \in \{1, \dots, n_c\}, x_j = 0 \Rightarrow s_i \neq j$,
- $\forall i \in M_-, \forall j \in \{1, \dots, n_c\}, s_i = j \Rightarrow x_j = 0$, et
- $\forall i \in M_-, \forall j \in \{1, \dots, n_c\}, x_j = 1 \Rightarrow s_i \neq j$.

Nous abordons maintenant les limites du raisonnement en termes d'isomorphisme de sous-graphes du point de vue de la chimie. Les figures 5(a)-(b) décrivent deux motifs basés sur trois hexagones positifs dont les graphes d'hexagone sont isomorphes. Il s'avère que les deux molécules correspondantes n'ont pas les mêmes propriétés chimiques. Or, si nous demandons à Choco de produire les structures correspondant à chacun de ces deux motifs sur la base du modèle \mathcal{M}_{i_3} nous obtiendrons les mêmes solutions. Aussi, pour pallier ce problème, nous ajoutons une phase de prétraitement du motif en amont de la génération de l'instance à résoudre. Cette phase consiste à rajouter des hexagones neutres de sorte à ce que toute arête du graphe d'hexagones du motif intervienne dans au moins un triangle (c'est-à-dire une clique de taille 3). Un triangle dans le graphe d'hexagones représente trois hexagones qui sont deux à deux voisins. Il caractérise donc une configuration unique (à une symétrie axiale ou à une rotation de 60° près). Ce prétraitement peut être mis en œuvre en parcourant une fois les hexagones du motif initial du haut vers le bas et de la gauche vers la droite. Par manque de place, nous ne détaillons pas cet algorithme. Les figures 5(c)-(d) présentent les motifs ainsi complétés associés aux motifs de (a) et (b). Notons qu'il n'est pas toujours nécessaire d'ajouter des hexagones neutres. Par exemple, le motif « baie profonde » reste inchangé car chaque arête de son graphe d'hexagones participe déjà à au moins un triangle.

6 Structures incluant plusieurs motifs

Dans cette section, nous nous intéressons à générer des structures contenant simultanément plusieurs mo-

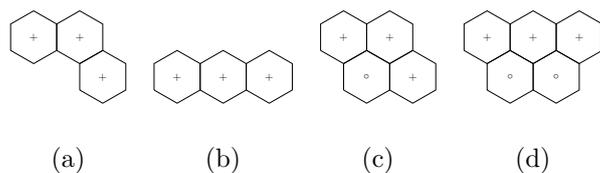


FIGURE 5 – Les limites de l'isomorphisme de sous-graphes (a) et (b). Les motifs (a) et (b) après prétraitement (c) et (d).

tifs. Soit $E_M = \{M^1, \dots, M^\ell\}$ l'ensemble de ces motifs. L'existence de plusieurs motifs pose la question de comment ils peuvent interagir entre eux. Nous listons ici trois cas ayant du sens du point de vue chimique :

- (1) Les motifs peuvent partager des hexagones (quelle que soit leur nature),
- (2) Les motifs ne peuvent partager que des hexagones absents (autrement dit du « vide »),
- (3) Les motifs sont disjoints deux à deux.

Une première approche naïve pour résoudre ce problème multi-motifs consiste à résoudre une collection de problèmes à un seul motif. Il faudrait, pour cela, énumérer tous les motifs uniques qu'on pourrait construire sur la base des motifs de E_M . Mais, vu la combinatoire, cette approche semble exclue. Nous proposons donc ci-dessous des modélisations adaptées sur la base des modèles vus précédemment.

6.1 Première modélisation

Nous partons du modèle général \mathcal{M} . Puis, nous ajoutons, pour chaque motif M^j de E_M , un ensemble de variables e_i^j équivalentes aux variables e_i pour un motif M dans le modèle \mathcal{M}_{i_1} ainsi que la contrainte de somme associée. Bien entendu, cela suppose d'avoir calculé, en amont, tous les fragments possibles de chaque motif de E_M . Cela définit le modèle $\mathcal{M}_{m_1}^1$.

Pour obtenir des motifs disjoints (modèle $\mathcal{M}_{m_1}^3$), il faut ajouter au modèle $\mathcal{M}_{m_1}^1$ des clauses d'exclusion mutuelle $e_i^j = 0 \vee e_{i'}^{j'} = 0$ pour chaque paire de fragments $\{F_i^j, F_{i'}^{j'}\}$ s'intersectant (c'est-à-dire tel que $(F_{i_+}^j \cup F_{i_-}^j \cup F_{i_o}^j \cup F_{i_*}^j) \cap (F_{i'_+}^{j'} \cup F_{i'_-}^{j'} \cup F_{i'_o}^{j'} \cup F_{i'_*}^{j'}) \neq \emptyset$).

Pour ne partager que du vide (modèle $\mathcal{M}_{m_1}^3$), nous ajoutons, au modèle $\mathcal{M}_{m_1}^1$, des contraintes de la forme $e_i^j = 0 \vee e_{i'}^{j'} = 0$ si F_i^j et $F_{i'}^{j'}$ peuvent partager un hexagone présent (c'est-à-dire si $(F_{i_+}^j \cap F_{i'_+}^{j'}) \cup (F_{i_+}^j \cap F_{i'_o}^{j'}) \cup (F_{i_o}^j \cap F_{i'_+}^{j'}) \neq \emptyset$). Sinon, s'ils partagent des hexagones neutres, ceux-ci doivent être absents de la structure, ce qui est assuré en posant la contrainte $(e_i^j = 1 \wedge e_{i'}^{j'} = 1) \Rightarrow x_h = 0$ pour chaque hexagone $h \in F_{i_o}^j \cap F_{i'_o}^{j'}$.

6.2 Deuxième modélisation

Nous partons du modèle général \mathcal{M} . Puis, nous ajoutons, pour chaque motif M^j de E_M , un ensemble de variables f_i^j équivalentes aux variables f_i pour un motif M dans le modèle \mathcal{M}_{i_2} ainsi que toutes les contraintes associées. Toutefois, dans chaque contrainte de table définissant le motif M^j , nous introduisons, dans sa portée, une variable booléenne t^j . Cette variable vaut 1 si la configuration associée est obtenue après l'application d'une symétrie axiale sur M^j , 0 sinon. Ainsi, la table va lister toutes les configurations valides obtenues à partir du motif M^j ou de son image par une symétrie axiale. La prise en compte des symétries axiales dans le cas de l'inclusion de plusieurs motifs est requise afin d'énumérer toutes les possibilités de combiner les motifs entre eux. Plusieurs axes de symétries sont possibles. Toutefois, il suffit d'en considérer un seul, les autres pouvant être obtenus par combinaisons avec des rotations de 60° . L'utilisation de la variable t^j au sein de chacune des contraintes de tables définissant M^j garantit que globalement, on exploite soit le motif M^j si t^j vaut 0, soit son image par symétrie axiale sinon. Cela évite donc de considérer des fragments erronés dont une partie correspondrait à M^j et une autre à son image par symétrie. Notons que, dans le cas d'un motif unique, l'utilisation de cette variable t^j ne ferait que rajouter des solutions équivalentes à celles déjà produites. Le modèle que nous venons de décrire correspond au cas (1). Nous le notons $\mathcal{M}_{m_2}^1$.

Ensuite, pour traiter le cas (2) autorisant le partage uniquement d'hexagones absents, nous reprenons le modèle $\mathcal{M}_{m_2}^1$ et ajoutons des contraintes d'exclusion mutuelle pour les hexagones présents. Cela revient à poser la contrainte suivante pour chaque hexagone h du coronénoïde de taille $k(n) : x_h = 1 \Rightarrow \text{Count}(\{f_h^1, \dots, f_h^\ell, \{1 \dots, n_{E_M}\}\}) \leq 1$ avec $n_{E_M} = \max_{M^j \in E_M} n_{M^j}$. Nous notons $\mathcal{M}_{m_2}^2$ ce modèle.

Enfin, pour pouvoir considérer des motifs disjoints (cas (3)), nous devons tenir compte des hexagones qui pourraient être partagés à l'extérieur du coronénoïde de taille $k(n)$. Pour cela, nous définissons l'ordre k_{E_M} de l'ensemble E_M comme l'ordre maximum d'un motif M^j de E_M . Puis, nous considérons le modèle $\mathcal{M}_{m_2}^1$ mais dans un coronénoïde de taille $k(n) + k_{E_M}$. Autrement dit, nous ajoutons à $\mathcal{M}_{m_2}^1$ une variable f_i^j par hexagone se trouvant à l'extérieur du coronénoïde de taille $k(n)$ et par motif M^j . Tous les hexagones étant représentés explicitement, les contraintes de tables sont définies en prenant en compte ces nouvelles variables et les contraintes *Count* de $\mathcal{M}_{m_2}^1$ pour des hexagones j' négatifs ou neutres du motif M^j sont désormais de la forme $\text{Count}(\{f_1^j, \dots, f_{n_c}^j, \{j'\}\}) = 1$. Pour terminer, nous ajoutons une contrainte d'exclusion mutuelle $\text{Count}(\{f_h^1, \dots, f_h^\ell, \{1 \dots, n_{E_M}\}\}) \leq 1$ pour chaque

hexagone h du coronénoïde de taille $k(n) + k_{E_M}$. Nous notons $\mathcal{M}_{m_2}^3$ ce modèle.

6.3 Troisième modélisation

Le principe est le même que pour la modélisation précédente. Pour chaque motif M^j de E_M , nous ajoutons au modèle \mathcal{M} un ensemble de variables s_i^j équivalentes aux variables s_i utilisées pour un motif M dans le modèle \mathcal{M}_{i_3} ainsi que toutes les contraintes associées. Ce faisant, nous obtenons le modèle $\mathcal{M}_{m_3}^1$ correspondant au cas (1). Le modèle \mathcal{M}_{i_3} dépendant de l'ordre du motif, les structures générées devront être inscrites dans un coronénoïde de taille $k(n) + k_{E_M}$. Bien entendu, comme dans \mathcal{M}_{i_3} , chaque motif doit être prétraité en amont.

Puis, nous pouvons étendre ce modèle au modèle $\mathcal{M}_{m_3}^3$ afin de tenir compte des motifs disjoints, en ajoutant la contrainte d'exclusion mutuelle $\text{alldifferent}(\{s_1^1, \dots, s_{n_{M^1}}^1\} \cup \dots \cup \{s_1^\ell, \dots, s_{n_{M^\ell}}^\ell\})$.

Enfin, le modèle $\mathcal{M}_{m_3}^2$ correspondant au cas (2) est obtenu à partir du modèle $\mathcal{M}_{m_3}^1$, en ajoutant les contraintes suivantes :

- $\text{alldifferent}(\{s_i^j | j \in \{1, \dots, \ell\}, i \in M_+^j\})$ qui exprime que les hexagones positifs dans x_G sont deux à deux disjoints,
- $\forall j, j' \in \{1, \dots, \ell\}, j < j', \forall i \in M_0^j, \forall i' \in M_0^{j'}, s_i^j = s_{i'}^{j'} \Rightarrow (s_i^j > n_c \vee \text{Element}(\{x_z | z \in \{1, \dots, n_c\}\}, s_i^j) = 0)$ ² qui exprime le fait que si deux hexagones neutres désignent le même hexagone de x_G , alors le sommet correspondant ne figure pas dans x_G .
- $\forall j, j' \in \{1, \dots, \ell\}, j \neq j', \forall i \in M_0^j, \forall i' \in M_+^{j'}, s_i^j \neq s_{i'}^{j'}$ qui interdit d'avoir le même hexagone de x_G pour un hexagone neutre et un hexagone positif de deux motifs différents.

7 Autres questions autour des motifs

Nous abordons maintenant quelques problématiques voisines autour des motifs. D'abord, nous traitons de l'exclusion d'un motif avant de montrer comment imposer des contraintes sur le nombre d'exemplaires d'un motif donné.

7.1 Structures excluant un motif

Nous souhaitons, à présent, générer toutes les structures à n hexagones ne contenant pas un motif M donné. Le raisonnement suivi pour les modèles \mathcal{M}_{i_2} et \mathcal{M}_{i_3} semble peu adapté car il faudrait garantir qu'il n'existe aucune numérotation f_i ou aucun sous-graphe

2. Pour rappel, la contrainte $\text{Element}(Y, j) \odot k$ est satisfaite ssi la valeur de la j -ème variable de Y satisfait la condition vis-à-vis de l'opérateur \odot et de la valeur k .

isomorphe convenable. Aussi, nous suivons ici le même raisonnement que celui du modèle \mathcal{M}_{i_1} . Plus précisément, nous partons du modèle \mathcal{M} et y ajoutons une variable e_i par fragment possible dans un coronénoïde de taille $k(n)$. Chaque variable e_i est vraie ssi la contrainte $\bigwedge_{j \in F_{i-}} x_j = 0 \wedge \bigwedge_{j \in F_{i+}} x_j = 1$ est satisfaite (c'est-à-dire le fragment présent dans la structure). Nous posons enfin une contrainte de somme $\sum_j e_j = 0$. Une formulation équivalente consiste à ne pas exploiter les variables e_i et à représenter directement chaque fragment F_i sous la forme d'un nogood $\bigvee_{j \in F_{i-}} x_j = 1 \vee \bigvee_{j \in F_{i+}} x_j = 0$.

7.2 Contraindre le nombre d'exemplaires

Certaines contraintes sur le nombre d'exemplaires d'un motif M sont faciles à modéliser. Par exemple, pour générer des structures de benzénoïdes ayant au moins k exemplaires disjoints du motif M , on peut utiliser les modélisations $\mathcal{M}_{m_1}^3$, $\mathcal{M}_{m_2}^3$ ou $\mathcal{M}_{m_3}^3$ et un ensemble E_M constitué de k fois le motif M . D'autres sont un peu plus délicates. Pour faciliter l'expression de telles contraintes, nous définissons une variable n_e qui représente le nombre d'exemplaires du motif M contenus dans la structure générée et sur laquelle nous poserons les contraintes adéquates selon les besoins des chimistes. La variable n_e a pour domaine $\{0, \dots, k_{max}\}$ avec k_{max} le nombre maximum d'exemplaires que peut ou doit contenir la structure. Par défaut, si aucune information n'est donnée en entrée sur k_{max} , on prend $k_{max} = \lfloor \frac{n}{|M|} \rfloor$.

Une nouvelle fois, l'approche suivie dans le modèle \mathcal{M}_{i_1} semble être la plus appropriée. Ainsi, partant du modèle \mathcal{M}_{i_1} , nous intégrons la variable n_e . En plus des variables e_i introduites pour chaque fragment possible, nous ajoutons une variable booléenne e'_i par fragment. La variable e'_i est vraie si le fragment F_i est présent dans le motif. Cela est assuré en ajoutant, pour chaque fragment F_i la contrainte $\bigwedge_{j \in F_{i-}} x_j = 0 \wedge \bigwedge_{j \in F_{i+}} x_j = 1 \Rightarrow e'_i = 1$. Ensuite, certains fragments partageant des hexagones, nous ajoutons des contraintes d'exclusion mutuelle, avec, pour chaque hexagone h , la contrainte $\sum_{i|h \in F_i} e'_i \geq 1 \Rightarrow \sum_{i|h \in F_i} e_i = 1$ (en considérant que $F_i = F_{i+} \cup F_{i-} \cup F_{i0} \cup F_{i*}$). Ainsi, cela nous garantit qu'un seul fragment est considéré comme présent pour un hexagone participant simultanément à plusieurs fragments. Enfin, la contrainte $\sum_j e_j = n_e$ nous permet alors de calculer le nombre d'exemplaires présents sur lequel nous pouvons ensuite poser aisément n'importe quelle contrainte arithmétique. Il est également possible d'utiliser cette variable afin de trouver les structures maximisant le nombre d'exemplaires du motif.

8 Expérimentations

Nous comparons ici nos différentes modélisations. Notre implémentation repose sur Choco (v. 4.10.7) lequel est utilisé avec ses réglages par défaut. Nous considérons les huit motifs issus de [12] et décrits dans les figures 4(a) et 6 et faisons varier le nombre n d'hexagones présents dans les structures du nombre d'hexagones positifs du motif à 9. Cela nous permet de produire 55 instances (resp. 135) du problème de génération de structures contenant un motif (resp. deux motifs). Les expérimentations sont réalisées sur des serveurs DELL PowerEdge R440 dotés de processeur Intel Xeon 4112 à 2,6 GHz et de 32 Go de mémoire. Le temps est limité à 2h.

D'après la figure 7, le modèle \mathcal{M}_{i_1} s'avère être légèrement plus performant pour générer les structures de benzénoïdes contenant un motif donné que les modèles \mathcal{M}_{i_2} et \mathcal{M}_{i_3} . Une explication possible réside dans le fait que les trois modèles reposent sur le modèle général \mathcal{M} auquel quelques variables et contraintes sont ajoutées. Au final, les trois modèles ont un nombre similaire de contraintes pour un nombre de variables plus important pour le modèle \mathcal{M}_{i_1} .

Pour deux motifs, le modèle $\mathcal{M}_{m_1}^3$ se révèle le meilleur (voir la figure 8), suivi par le modèle $\mathcal{M}_{m_3}^3$. Ce résultat semble lié au nombre de contraintes qui est deux fois plus important pour le modèle $\mathcal{M}_{m_2}^3$ tandis que les modèles $\mathcal{M}_{m_1}^3$ et $\mathcal{M}_{m_3}^3$ conduisent à un nombre similaire de contraintes.

9 Conclusion et perspectives

Nous avons présenté une approche basée sur la PPC permettant de générer des structures de benzénoïdes satisfaisant certaines contraintes autour de l'inclusion de motifs. Pour cela, plusieurs modélisations ont été considérées et comparées. La première modélisation, basée sur la liste des fragments possibles, se révèle globalement la plus robuste, mais aussi celle ayant le plus grand pouvoir d'expression (inclusion/exclusion de motif, nombre d'occurrences, ...).

Une première perspective de ce travail sera d'étudier ses retombées du point de vue de la chimie théorique. Concernant la modélisation, une étude plus poussée de la notion d'ordre d'un motif devrait permettre de l'affiner avec, à la clé, une réduction de l'espace de recherche à explorer. Enfin, plusieurs voies peuvent être explorées pour améliorer l'efficacité pratique de l'approche (définition de contraintes globales dédiées, étude des heuristiques de choix de variables, ...).

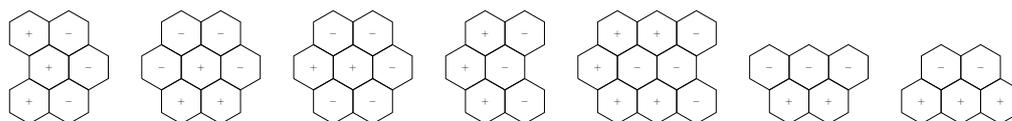


FIGURE 6 – Sept des huit motifs considérés pour les expérimentations.

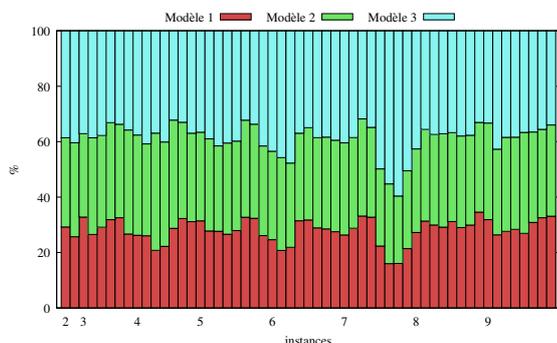


FIGURE 7 – Rapport en pourcentage du temps consommé par chaque modèle sur le temps cumulé des modèles \mathcal{M}_{i_1} , \mathcal{M}_{i_2} et \mathcal{M}_{i_3} en fonction des instances classées selon le nombre n d’hexagones souhaités.

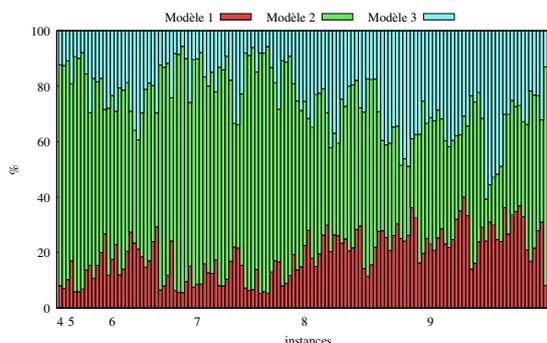


FIGURE 8 – Rapport en pourcentage du temps consommé par chaque modèle sur le temps cumulé des modèles \mathcal{M}_{m_1} , \mathcal{M}_{m_2} et \mathcal{M}_{m_3} en fonction des instances classées selon le nombre n d’hexagones souhaités.

Références

- [1] G. BRINKMANN, G. CAPOROSI et P. HANSEN : A Constructive Enumeration of Fusenes and Benzenoids. *Journal of Algorithms*, 45(2), 2002.
- [2] Y. CARISSAN, D. HAGEBAUM-REIGNIER, N. PROCOVIC, C. TERRIOUX et A. VARET : Using Constraint Programming to Generate Benzenoid Structures in Theoretical Chemistry. *In CP*, pages 690–706, 2020.
- [3] J. CYVIN, J. BRUNVOLL et B. N. CYVIN : Search for Concealed Non-Kekuléan Benzenoids and Coronoids. *J. Chem. Inf. Comput. Sci.*, 29(4):237, 1989.
- [4] T. DUMSLAFF, Y. GU, G. M. PATERNO, Z. QIU, A. MAGHSOUMI, M. TOMMASINI, X. FENG, F. SCOTOGNELLA, A. NARITA et K. MÜLLEN : Hexa-Peri-Benzocoronene with Two Extra K-Regions in an Ortho-Configuration. *Chem. Sci.*, 11:12816–12821, 2020.
- [5] J.-G. FAGES, X. LORCA et C. PRUD’HOMME : Choco solver user guide documentation. <https://choco-solver.readthedocs.io/en/latest/>.
- [6] C. LECOUTRE et O. ROUSSEL, éditeurs. *Proceedings of the 2018 XCSP3 Competition*, 2018.
- [7] S. MISHRA, D. BEYER, K. EIMRE, S. KEZILBIEKE, R. BERGER, O. GRÖNING, C. A. PIGNEDOLI, K. MÜLLEN, P. LILJEROTH, P. RUFFIEUX, X. FENG et R. FASEL : Topological frustration induces unconventional magnetism in a nanographene. *Nature Nanotechnology*, 15(1):22–28, 2020.
- [8] S. MISHRA, D. BEYER, K. EIMRE, J. LIU, R. BERGER, O. GRÖNING, C. A. PIGNEDOLI, K. MÜLLEN, R. FASEL, X. FENG et P. RUFFIEUX : Synthesis and Characterization of π -Extended Triangulene. *Journal of the American Chemical Society*, 141(27):10621–10625, 2019.
- [9] W. NIU, J. MA, P. SOLTANI, W. ZHENG, F. LIU, A. POPOV, J. WEIGAND, H. KOMBER, E. POLIANI, C. CASIRAGHI, J. DROSTE, M. HANSEN, S. OSELLA, D. BELJONNE, M. BONN, H. WANG, X. FENG, J. LIU et Y. MAI : A Curved Graphene Nanoribbon with Multi-Edge Structure and High Intrinsic Charge Carrier Mobility. *J. Am. Chem. Soc.*, 142:18293–18298, 2020.
- [10] Georges TRINQUIER et Jean-Paul MALRIEU : Predicting the Open-Shell Character of Polycyclic Hydrocarbons in Terms of Clar Sextets. *The Journal of Physical Chemistry A*, 122(4):1088–1103, 2018.
- [11] H. VERHAEGHE, C. LECOUTRE et P. SCHAUSS : Extending Compact-Table to Negative and Short Tables. *In AAAI*, pages 3951–3957, 2017.
- [12] N. WOHNER, P. K. LAM et K. SATTLER : Systematic energetics study of graphene nanoflakes : From armchair and zigzag to rough edges with pronounced protrusions and overcrowded bays. *Carbon*, 82:523–537, 2015.