

Exhaustive Generation of Benzenoid Structures Sharing Common Patterns

Yannick Carissan, Denis Hagebaum-Reignier, Nicolas Prcovic, Cyril Terrioux, Adrien Varet

▶ **To cite this version:**

Yannick Carissan, Denis Hagebaum-Reignier, Nicolas Prcovic, Cyril Terrioux, Adrien Varet. Exhaustive Generation of Benzenoid Structures Sharing Common Patterns. 27th International Conference on Principles and Practice of Constraint Programming, Oct 2021, Montpellier, France. 10.4230/LIPIcs.CP.2021.19 . hal-03402690

HAL Id: hal-03402690
https://amu.hal.science/hal-03402690

Submitted on 25 Oct 2021

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

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



Distributed under a Creative Commons Attribution 4.0 International License

Exhaustive Generation of Benzenoid Structures Sharing Common Patterns

Yannick Carissan
Aix Marseille Univ, CNRS, Centrale Marseille, ISM2, Marseille, France

Denis Hagebaum-Reignier
Aix Marseille Univ, CNRS, Centrale Marseille, ISM2, Marseille, France

Nicolas Prcovic
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France

Cyril Terrioux
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France

Adrien Varet
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France

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 both experimentally and theoretically 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). The exhaustive generation of families of benzenoids sharing the absence or presence of given patterns is an important problem in chemistry, particularly in theoretical chemistry, where various methods can be used to better understand the link between their shapes and their electronic properties.

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 Choco solver.

2012 ACM Subject Classification Computing methodologies → Artificial intelligence

Keywords and phrases Constraint satisfaction problem, modeling, pattern, application, theoretical chemistry

Digital Object Identifier 10.4230/LIPIcs.CP.2021.19

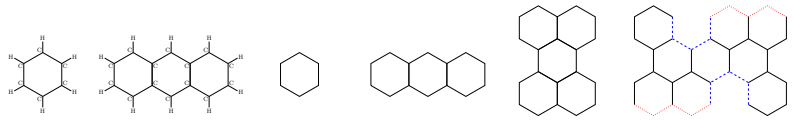
Supplementary Material *Software (Source Code)*: `https://github.com/AdrienVaret/BenzenoidApplicationReleases/releases/tag/latest-version`
archived at `swh:1:dir:8b64aa73cb6b96ad557006c28255fc8642cef0a7`

Funding This work has been funded by the Agence Nationale de la Recherche project ANR-16-CE40-0028.

1 Introduction

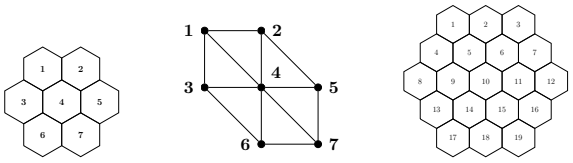
Polycyclic aromatic hydrocarbons (PAHs) are hydrocarbons whose carbons are forming cycles of different sizes. *Benzenoids* are a subfamily of PAHs for which all the cycles are of size 6. *Benzene*, represented in Figure 1(a), is the smallest one. It is made of 6 carbon atoms and 6 hydrogen atoms. Its carbon atoms form a hexagon (also called *benzenic cycle* or *benzenic ring*) and each of them is linked to a hydrogen atom. Benzenoids can also be seen as the molecules obtained by aggregating benzenic rings. For example, Figure 1(b) shows anthracene, which contains three benzenic rings. Atoms establish bonds between themselves

© Yannick Carissan, Denis Hagebaum-Reignier, Nicolas Prcovic, Cyril Terrioux, and Adrien Varet; licensed under Creative Commons License CC-BY 4.0



(a) (b) (c) (d) (e) (f)

Figure 1 Examples of benzenoids: benzene (a) and anthracene (b) with their graphical representation (c) and (d), perylene (e) and a benzenoid containing two instances of the pattern *deep bay* (in blue dashed) and two of instances of the pattern *zigzag bay* (in dotted red) (f).



(a) (b) (c)

Figure 2 Coronene (a), its hexagon graph (b) and the coronenoid of size 3 (c).

which can be single or double depending on the number of electrons involved in the bond. In a benzenoid, each carbon atom is linked either to two carbon atoms and one hydrogen atom, or to three carbon atoms. In the following, hydrogen atoms play no role (and their presence can be inferred if necessary). Also, they can be omitted in the representation. Thus, a benzenoid can be represented as an undirected graph $B = (V, E)$ in which each vertex of V corresponds to a carbon atom and each edge of E reflects the existence of a bond between the two corresponding carbon atoms. Note that the nature of bonds (simple or double) has no importance for our purpose. This graph is connected, bipartite and planar. Figures 1(c) and (d) show the graphs corresponding to benzene and anthracene. Moreover, since benzenoids can be defined as a combination of fused benzenic rings, we consider, for each benzenoid B, a second graph called the *hexagon graph*. This graph $B_h = (V_h, E_h)$ is an undirected graph in which each vertex corresponds to a hexagon (i.e. a benzenic ring) of B and such that two vertices are connected by an edge if the corresponding hexagons share an edge in the graph B. Figures 2(a)-(b) show the graph corresponding to coronene and its hexagon graph.

Benzenoids and more generally PAHs are well-studied in various fields (molecular nanoelectronics, organic synthesis, interstellar chemistry, ...) because of their energetic stability, molecular structures or optical spectra. They have a wide variety of physicochemical properties depending on their size and structure. For example, they can combine a strong mechanical resistance with high electronic conductivity. 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 impacts the whole electronic structure. For instance, perylene (Figure 1(e)) can be seen as two overlapping triangles of three fused rings with the consequence that the central bonds are essentially simple.

The controlled synthesis of PAH with tuned edges is very recent. This is a hot topic as shown by the number of recent publications on the synthesis of such compounds in high impact factor chemistry journals (e.g. [29, 19, 1, 6, 8, 14, 16, 31, 35]). This has motivated many theoretical studies to better understand the impact of the edge topology on their electronic properties. This chemistry leads to enhanced optoelectronic molecular properties [11, 20, 24, 25, 28, 36] or magnetic properties [23, 38], which are further improved and combined in mixed edge molecules [17, 26, 27, 37]. As recent examples, it was shown that the addition of two extra K-regions (or armchair edges, see Figure 5(a)) to hexabenzocoronene leads to an enhanced optical activity of the molecules with potential applications as organic

laser materials [11]. Furthermore, PAHs with armchair edges are semiconductors with high bands gaps whereas zigzag edges (see Figure 5(g)) lead to improved conductivity but are fairly unstable. Thus chemists intend to design molecules with zigzag patterns at the edges and stabilize the molecular structure with neighboring cove regions, which also lead to higher dispersibility in solution and improved optoelectronic properties [26]. Niu et al. [26] provide several benzenoids that chemists can synthesize and which contain such patterns. For instance, Figure 1(f) describes one of them which contains two instances of the pattern *cove* (depicted in blue dashed and called *deep bay* in [34]) and two instances of the pattern *zigzag* (depicted in dotted red and called *zigzag bay* in [34]). It is thus important to be able to exhaustively generate families of benzenoids sharing common given patterns on their edges for a given number of fused rings.

In the literature, bespoke approaches have been proposed to generate benzenoid structures satisfying or not some particular properties (e.g. [3]). They turn to be very efficient in practice but are difficult to adapt to the needs of chemists. Moreover, they only consider properties on the whole molecule. In [5], we have described a new approach based on constraint programming (CP) which is more flexible while being competitive. More precisely, we model the problem as an instance of the *Constraint Satisfaction Problem (CSP)*. Remember that a CSP instance can be defined as a triplet (X, D, C) where $X = \{x_1, \ldots, x_n\}$ is the set of *variables*, $D = \{D_{x_1}, \ldots, D_{x_n}\}$ is the set of domains, the domain D_{x_i} being related to the variable x_i, and $C = \{c_1, ..., c_e\}$ represents the set of the *constraints* which define the interactions between the variables and describe the allowed combinations of values. For sake of simplicity and expressivity, our model exploits *graph variables* (notably to represent the hexagon graph). Graph variables have as domain a set of graphs defined by a lower bound (a subgraph called GLB) and an upper bound (a super-graph called GUB). In [5], we used Choco solver [13] since this library supports *graph variables* and its graph-related constraints (e.g. the `connected` constraint [12]) and also the usual global constraints which make the modeling easier. However, this model only handles global properties. Hence, in this paper, we describe how to integrate the notion of pattern. Several models being possible, we study them on the basis of the different questions that arise about patterns before comparing them experimentally.

This paper is organized as follows. In Section 2, we recall how to generate benzenoid structures, in particular with the help of CP. Afterward, in Section 3, we formalize the problem we are interested in and address different questions that may be of interest for chemists in Sections 4 to 6. Finally, we assess experimentally some models in Section 7, before concluding in Section 8.

2 Generating Benzenoid Structures

Generating benzenoid structures which have certain structural properties (e.g. having a given number of hexagons or having a particular structure from a graph viewpoint) is an interesting and important problem in theoretical chemistry [9, 21, 22, 30]. This problem is a preliminary step to the study of their chemical properties. It can be formally defined as follows: Given a set of structural properties $\mathcal{P}$, generate all the benzenoid structures satisfying the properties of $\mathcal{P}$. These properties may concern the number of carbon or hexagon atoms or particular shapes (tree, rectangle, presence of "holes", ...).

In the literature, bespoke methods have been proposed (e.g. [3]). If they are often efficient in practice, they have the disadvantage of being difficult to adapt to the needs of chemists. In [5], we proposed a CP model of this problem and showed how easy it is to meet the wishes expressed by the chemists by simply adding variables and constraints. Moreover, beyond its flexibility, this approach is relatively efficient thanks to Choco solver.

We now recall the CP model allowing to generate all structures with n hexagons. It relies on the property that any benzenoid of n hexagons can be embedded in a coronenoid of size at most $k(n) = \lfloor \frac{n}{2} + 1 \rfloor$. A coronenoid of size k is a benzene molecule to which $k-1$ crowns of hexagons have been successively added. Coronene (see Figure 2(a)) is the coronenoid of size 2. Figure 2(c) shows the coronenoid of size 3. We can remark that the number of hexagons in the ith crown grows with i. Thereafter, we denote $B_h^{c(k(n))}$ (where $c(k(n))$ stands for coronenoid of size $k(n)$) the hexagon graph of the coronenoid of size $k(n)$, n_c its number of hexagons and m_c its number of edges. The hexagons and edges of $B_h^{c(k(n))}$ are arbitrarily numbered starting from 1. Figure 2 presents a possible numbering. First, in this model (denoted $\mathcal{M}$), we consider a graph variable x_G to represent the hexagon graph of the desired structure. Its domain is the set of all subgraphs between the empty graph and $B_h^{c(k(n))}$. The use of a graph variable makes it much easier to express the connectedness of the generated structures (as described below). We also exploit a set of n_c Boolean variables $\{x_1, \ldots, x_{n_c}\}$. The variable x_i is set to 1 if the i-th hexagon of $B_h^{c(k(n))}$ is used in x_G, 0 otherwise. Similarly, we also consider a set of m_c Boolean variables $y_{i,j}$. The variable $y_{i,j}$ is set to 1 if the edge $\{i,j\}$ of $B_h^{c(k(n))}$ is used in x_G, 0 otherwise.

Then, the following properties are expressed thanks to constraints:

- *Link between x_G and x_i (resp. $y_{i,j}$)*: a `channeling` constraint imposes that $x_i = 1 \iff$ *x_G contains the vertex i* (resp. $y_{i,j} = 1 \iff$ *x_G contains the edge $\{i,j\}$*).
- *x_G is an induced subgraph of $B_h^{c(k(n))}$*: Any value of x_G is not necessarily a valid hexagon graph. To guarantee its validity, it must correspond to a subgraph of $B_h^{c(k(n))}$ induced by the vertices belonging to x_G. Thus, for each edge $\{i,j\}$ of $B_h^{c(k(n))}$, one adds a constraint $x_i = 1 \wedge x_j = 1 \Leftrightarrow y_{i,j} = 1$. In other words, the edge $\{i,j\}$ exists in x_G if and only if the vertices i and j appear in x_G.
- *The structure has n hexagons*: $\sum_{i \in \{1,\ldots,n_c\}} x_i = n$.
- *The hexagon graph is connected*: It is achieved by applying the `connected` graph constraint on x_G [12].
- *Six hexagons forming a cycle generate a hexagon (and not a hole)*: For each hexagon u, let $N(u)$ denotes the set of the neighbors of u in the hexagon graph. Then, for each vertex u having 6 neighbors, the property is ensured by adding a constraint between x_u and the variables corresponding to its neighbors which imposes: $\sum_{v \in N(u)} x_v = 6 \Rightarrow x_u = 1$.

Finally, several constraints are added in order to avoid redundancies. First, x_G must have at least one vertex on the top (resp. left) edge of $B_h^{c(k(n))}$ in order to discard the symmetries by translation. This can be achieved by posting a constraint that specifies that the sum of the Boolean variables x_i associated with the top (resp. left) edge of $B_h^{c(k(n))}$ is strictly positive. Then, one must ensure that the graph described by x_G is the only representative of its symmetry class. There are up to twelve symmetric solutions: six 60 degree rotational symmetries combined with a possible axial symmetry. These symmetries are broken by the constraint `lex-lead` [10]. For each of the twelve symmetries, one needs to add n_c Boolean variables (one per variable x_i) and a total of $3.n_c$ ternary clauses.

This model can easily be implemented with Choco solver. It can also be specialized to take into account the needs of chemists by adding variables and/or constraints. For example, generating structures with a tree shape (called *catacondensed* benzenoids) simply requires the addition of the `tree` graph constraint on x_G to the general model. Other properties have been modeled in order to generate structures having a rectangular shape, possessing a hole or being symmetrical [5].

3 Considering Patterns

The model $\mathcal{M}$, presented in [5] and recalled in Section 2, allows to express the benzenoid structure generation problem in all its generality. If several specializations of this model have been proposed in [5], all of them correspond to structural properties concerning the whole molecule. These properties could thus be qualified as *global*. However, in some cases, it may be useful to reason in terms of local properties that may or may not be satisfied by some parts (called fragments) of the generated structures. In particular, among these local properties, it is important to be able to deal with the local properties related to the edge of the benzenoid structure.

The local properties we consider in this article can be defined by "drawing" a shape whose basic bricks are hexagons. These hexagons can be of three different natures:

(i) The *positive* hexagons whose presence is required in the property,
(ii) The *negative* hexagons whose absence is required in the property,
(iii) The *neutral* hexagons whose presence or absence has no influence on the property.

If the use of positive hexagons is obvious, one can ask the question of the interest of negative or neutral hexagons. Negative (respectively neutral) hexagons are useful, for example, to indicate that there is nothing between two positive hexagons or to model the edge of the benzenoid (resp. to guarantee a certain gap between two positive hexagons). In order to represent the desired shapes, we now introduce the notion of extended hexagon graph:

▶ **Definition 1** (extended hexagon graph). *An* extended hexagon graph *is a hexagon graph whose vertices and edges are labeled by the symbols $+$ (for positive), $-$ (for negative) and $\circ$ (for neutral) such that:*

(i) *Each vertex is labeled with the nature of the hexagon it represents.*
(ii) *An edge is labeled $-$ if at least one of its vertices is labeled $-$. Otherwise, it is labeled $\circ$ if at least one of its vertices is labeled $\circ$. Otherwise, it is labeled $+$.*

As for the hexagons (or vertices), the labels associated with the edges qualify the status that the interaction between two hexagons must have in the local property that we wish to define. Formally a local property can be defined by a *pattern*:

▶ **Definition 2** (pattern). *A* pattern *P is defined by giving a triplet $(P_+, P_-, P_\circ)$ and an extended hexagon graph P_h such that:*

(i) *P_+, P_- and $P_\circ$ denote the set of positive, negative and neutral hexagons respectively,*
(ii) *these three sets are pairwise disjoint and*
(iii) *P_h is a connected graph on the set of hexagons defined by $P_+ \cup P_- \cup P_\circ$.*

Its order k_P is the maximal length (expressed in terms of the number of edges) of the shortest paths of P_h separating a negative or neutral hexagon from a positive hexagon.

In other words, a pattern is defined by a collection of positive, negative and neutral hexagons whose arrangement is described by an extended hexagon graph. As an example, Figure 3(a) shows the pattern *deep bay* [34] of order 1 composed of four positive hexagons and three negative ones. The bonds (i.e. the edges of hexagons) which are at the interface between the positive hexagons and the negative ones allow us to handle the local property of the edge of the benzenoid depicted in blue in Figure 1(f). We finally define the notion of pattern inclusion:

▶ **Definition 3.** *Given a non-negative integer k, let B_h^k be the extended hexagon graph representing the benzenoid B surrounded by k layers of negative hexagons (i.e., the extended graph of B augmented by all negative hexagons located within distance k of a hexagon of B).*

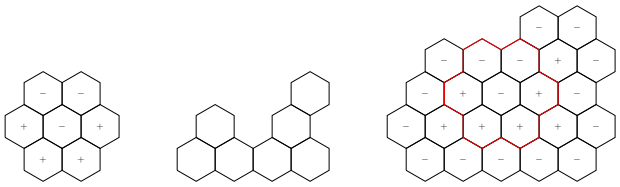


Figure 3 The pattern *deep bay* (a), a benzenoid satisfying this pattern (b) and the "extended" benzenoid related to its extended hexagon graph B_h^1 with the pattern in red (c).

A fragment F^k *of order* k *of a benzenoid* B *is a subset of hexagons of* B_h^k *whose extended hexagon graph is connected. It* satisfies *the pattern* P *if* $k = k_P$ *and if there exists a bijection that maps a positive or neutral hexagon of* P *to each positive hexagon of* F^k *and maps a negative or neutral hexagon of* P *to each negative hexagon of* F^k*. A benzenoid* B contains *(or* includes*) the pattern* P *if it has a fragment of order* k_P *satisfying* P*.*

Considering B_h or B_h^k does not change the nature of the benzenoid B. B_h^1 simply materializes the vacuum around it, which is necessary for some properties. For example, the benzenoid in Figure 3(b) (like the one in Figure 1(f)) satisfies the pattern *deep bay* of Figure 3(a). For this, we must take into account the absence of hexagon at the edge of the benzenoid to identify a suitable fragment, which is achieved thanks to its extended hexagon graph B_h^k (see Figure 3(c)).

In this paper, we aim to generate benzenoid structures satisfying local properties expressed thanks to the patterns introduced above. These local properties can take different forms. The simplest one is to include a given pattern. Then, one can also be interested in generalizing the approach by including several different patterns or a given number of times the same pattern. On the contrary, one may also wish to exclude a given pattern. The following sections deal with these different issues. In all cases, the idea is to generate benzenoid structures starting from the general model $\mathcal{M}$. By so doing, it follows that it is quite possible to consider both global and local properties.

4 Generating Structures Including a Pattern

Let P be a pattern involving n_P hexagons which can be positive, negative or neutral. We arbitrarily number each hexagon of the pattern P from 1 to n_P. The sets P_+, P_- and $P_\circ$ are then defined accordingly. In this section, we wish to model the problem of generating all benzenoid structures having n hexagons and including the pattern P. We first consider all the variables and constraints of the general model $\mathcal{M}$ to which we will add variables and constraints to express the fact that the pattern must be present in the generated structures. At this level, we have several possibilities depending on the point of view we consider. In the following, we explore three tracks. The first one consists in identifying all the possible locations of a fragment satisfying the pattern P. The second one considers the existence of a fragment by reasoning on the neighborhood of each hexagon. Finally, the third one exploits the proximity of our problem with the subgraph isomorphism problem.

4.1 First Model

We start with the model $\mathcal{M}$ and thus with a coronenoid of size $k(n)$. In this first model (denoted $\mathcal{M}_{i_1}$), we first identify all the possible fragments of the pattern P in this coronenoid. Their number being in $O(|c(k(n))|) = O(n^2)$, this computation can be efficiently performed

using rotations, axial symmetries and translations. For each of these fragments F_i, we define the sets F_{i+}, $F_{i\circ}$ and F_{i-} of its positive, neutral and negative hexagons. We associate to each fragment F_i a Boolean variable e_i such that the fragment F_i is present in the structure under construction if e_i is true. This is guaranteed via the constraint $e_i = 1 \Rightarrow \bigwedge_{j \in F_{i-}} x_j = 0 \wedge \bigwedge_{j \in F_{i+}} x_j = 1$. Note that for patterns whose order is strictly positive, it is not necessary to consider a larger coronenoid. Indeed, the fragment can be placed at the edge of the coronenoid with negative or neutral hexagons being outside this coronenoid and, therefore, being considered as absent. In this case, these hexagons will not be represented in F_{i-}, nor in F_i, but placed in a set F_{i*}. Finally, we set the sum constraint $\sum_j e_j = 1$ to guarantee the existence of at least one fragment satisfying the pattern P.

4.2 Second Model

In this second model (denoted $\mathcal{M}_{i_2}$), we express the existence of a fragment corresponding to the pattern P by reasoning on the neighborhood of each hexagon. To do so, starting from the model $\mathcal{M}$, we add a variable f_i per hexagon of the coronenoid of size $k(n)$. Each variable f_i has domain $\{0, 1, \ldots, n_P\}$. The variable f_i takes a positive value j if the hexagon i of the coronenoid of size $k(n)$ participates in the searched fragment as a hexagon occupying the position j in P, 0 otherwise. Thus, the variable f_i specifies whether the hexagon i is involved in the fragment and if so to which hexagon of the pattern P it corresponds. Then, since the generation of the benzenoid structures and the search for a fragment are done simultaneously, we need to ensure their concordance. In particular, we must guarantee that the positive (resp. negative) hexagons are indeed present (resp. absent) in the generated structure. As a reminder, this structure is represented by the graph variable x_G and by the Boolean variables x_i. Also, for each hexagon i of the coronenoid of size $k(n)$, we set the following constraints:

- $x_i = 1 \Rightarrow f_i \in P_+ \cup P_\circ \cup \{0\}$ (if the hexagon i is present in x_G, it is involved in the fragment as a positive or neutral hexagons or it does not participate in the fragment),
- $x_i = 0 \Rightarrow f_i \in P_- \cup P_\circ \cup \{0\}$ (if the hexagon i is absent, it is involved in the fragment as a negative or neutral hexagons or it does not participate in the fragment),
- $f_i \in P_+ \Rightarrow x_i = 1$ (if the hexagon i participates in the fragment as a positive hexagon, it is necessarily present),
- $f_i \in P_- \Rightarrow x_i = 0$ (if the hexagon i participates in the fragment as a negative hexagon, it is necessarily absent).

Next, we need to define the bijection that establishes that the constructed fragment satisfies the pattern P. In other words, we need to guarantee that exactly n_P hexagons of the structure must correspond to n_P hexagons of the pattern P. Also, for each hexagon j of the pattern, we add the global constraint[1] $Count(\{f_1, \ldots, f_{n_c}\}, \{j\}) = 1$ if $j \in P_+$ (≤ 1 otherwise). The value 0 is obtained in the case where a negative or neutral hexagon is outside the coronenoid of size $k(n)$. In other words, a part of the pattern overflows from this coronenoid, but only for negative or neutral hexagons (which would then be absent). By doing so, we avoid introducing additional variables (and associated constraints) to represent the k_P layers of absent hexagons used in the formal definition of fragment (see Definition 3).

The last step consists in defining the pattern itself. To do this, we consider the neighborhood links between each hexagon of the pattern. A hexagon can have up to six neighboring hexagons. For a given hexagon h, we consider its potential neighbors v_1 to v_6 in a clockwise

[1] As a reminder, the global constraint $Count(Y, V) \odot k$ is satisfied if the number of variables of Y assigned with a value in V satisfies the condition with respect to the operator $\odot$ and the value k.

Table 1 The compact table constraint describing the neighborhood for the pattern *deep bay*.

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
⋮	⋮	⋮	⋮	⋮	⋮	⋮

f_i	f_{v_1}	f_{v_2}	f_{v_3}	f_{v_4}	f_{v_5}	f_{v_6}
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
⋮	⋮	⋮	⋮	⋮	⋮	⋮

direction, starting with the neighbor at the top right. From there, we list the different configurations taken by the neighbors depending on which the hexagon h participates in the fragment or not. More precisely, each configuration is a tuple composed of one integer per neighbor. This integer is a non-zero value j if the neighbor participates in the fragment as the hexagon j of the pattern P, 0 otherwise. For each position of the hexagon h in the pattern P, we consider six possible configurations in order to take into account the 60° rotations of the pattern. This is necessary to generate all the structures because the model $\mathcal{M}$ imposes the existence of hexagon(s) on the top and left edges of the considered coronenoid. Note that from a given configuration, applying a 60° rotation is equivalent to performing a circular permutation at the tuple level. For example, in Table 1, we list all the possible neighborhood configurations when the hexagon is in position 1 in the pattern *deep bay*, the numbering of the hexagons being that of Figure 2. For the other positions, we give only one configuration by lack of space. These configurations will be used to define the relation associated with compact table constraints [32]. We consider one table constraint per hexagon h of the coronenoid of size $k(n)$ whose scope involves the variable f_h and each variable f_i associated with a neighbor of h in $B_h^{c(k(n))}$. For hexagons at the edge of the coronenoid, we keep only the rows of the table whose neighbors participating in the fragment correspond to hexagons (whatever their nature) inside the coronenoid or to negative or neutral hexagons outside the coronenoid. Then, we make a projection of these lines on the present neighbors and the variable f_h.

4.3 Third Model

A fragment of order k of a benzenoid B corresponds to a connected subgraph of B_h^k. Thus, determining whether there exists a fragment satisfying a pattern P in a benzenoid B is, in some way, the same as determining whether there exists a subgraph in $B_h^{k_P}$ isomorphic to P_h. However, this is not exactly the usual subgraph isomorphism problem, but one of its variants taking into account the labeling of vertices and edges. This does not change the complexity of the decision problem which remains NP-complete. Fortunately, we do not need to tackle this problem because, in our approach, we will, by construction, directly produce structures satisfying the pattern.

We now present our model $\mathcal{M}_{i_3}$. Starting again from the general model $\mathcal{M}$, we add one variable s_i per hexagon of the pattern P (whatever its nature). Each variable s_i has domain $\{1, \ldots, n'_c\}$ with n'_c the number of hexagons of the coronenoid of size $k(n) + k_P$. We exploit a coronenoid of size $k(n) + k_P$, instead of $k(n)$, because we need to surround the coronenoid of size $k(n)$ with k_P crowns of absent hexagons. Note that this has no impact on the graph

variable x_G or on the variables x_i because we are adding hexagons that are known not to be present in the structure under consideration. The variable s_i has value j if the i-th hexagon of the pattern P is the j-th hexagon of the coronenoid of size $k(n) + k_P$. By convention, values of j between 1 and n_c correspond to hexagons present in the coronenoid of size $k(n)$. We then add the following constraints to express the notion of isomorphism:

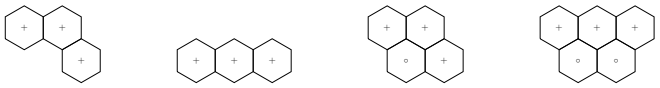
- *Injectivity*: The hexagons participating in the fragment must be pairwise different. This is imposed thanks to the global constraint `alldifferent`$(\{s_1, \ldots, s_{n_P}\})$. This also ensures that n_P hexagons of x_G participate in the fragment.
- *Edge preservation*: We must guarantee that two neighboring vertices of P_h correspond to two neighboring vertices in the hexagon graph of the coronenoid of size $k(n)$. Also, for each edge $\{i, i'\}$ of P_h (whatever its nature), we set a table constraint on s_i and $s_{i'}$ whose relation contains all pairs (j, j') such that $\{j, j'\}$ is an edge of the hexagon graph of the coronenoid of size $k(n) + k_P$.

This part of the model is inspired by the model of the subgraph isomorphism problem presented in [18]. However, it should be noted that, in our case, the graph in which the subgraph is searched is not known in advance, as it is the graph we wish to construct. Also, in our model, we circumvent this difficulty by considering the hexagon graph of the coronenoid of size $k(n) + k_P$.

Concerning the labeling, by definition, the labeling of the edges follows from that of the vertices. The labeling of the vertices is directly taken into account by definition of the variables s_i. It only remains to express the adequacy between the labeling of the vertices and the existence of the hexagons thanks to the following constraints:

- $\forall i \in P_+, \forall j \in \{1, \ldots, n_c\}, s_i = j \Rightarrow x_j = 1$ (if the positive hexagon i of P corresponds to the hexagon j in x_G, j must be present),
- $\forall i \in P_+, \forall j \in \{1, \ldots, n_c\}, x_j = 0 \Rightarrow s_i \neq j$ (if the hexagon j of x_G is absent, it cannot correspond to a positive hexagon i of P),
- $\forall i \in P_-, \forall j \in \{1, \ldots, n_c\}, s_i = j \Rightarrow x_j = 0$ (if the negative hexagon i of P corresponds to the hexagon j in x_G, j must be absent), and
- $\forall i \in P_-, \forall j \in \{1, \ldots, n_c\}, x_j = 1 \Rightarrow s_i \neq j$ (if the hexagon j of x_G is present, it cannot correspond to a negative hexagon i of P).

We now turn to the limitations of reasoning in terms of subgraph isomorphism from the perspective of chemistry. Figures 4(a)-(b) describe two patterns based on three positive hexagons and whose hexagon graphs are isomorphic. It turns out that the two corresponding molecules do not have the same chemical properties. However, if we ask Choco to produce the structures corresponding to each of these two patterns based on the model $\mathcal{M}_{i_3}$, we will obtain the same solutions. Also, to overcome this problem, we add a preprocessing step before the generation of the instance to solve. This step consists in adding neutral hexagons so that every edge of the hexagon graph of the pattern appears in at least one triangle (i.e. a clique of size 3). A triangle in the hexagon graph represents three hexagons which are pairwise adjacent. It thus characterizes a unique configuration (within one axial symmetry or 60° rotation). This preprocessing can be implemented by going through the hexagons of the initial pattern from top to bottom and from left to right. For lack of space, we do not detail this algorithm. Figures 4(c)-(d) present the patterns thus completed related to the patterns of Figures 4(a)-(b). Note that it is not always necessary to add neutral hexagons. For example, the pattern deep bay remains unchanged because each edge of its hexagon graph is already involved in at least one triangle.



(a) (b) (c) (d)

Figure 4 The limits of reasoning in terms of subgraph isomorphism with two different patterns (a) and (b) having isomorphic hexagon graphs. The patterns (a) and (b) after preprocessing (c)–(d).

5 Generating Structures Including Several Patterns

In this section, we are interested in generating structures containing several patterns simultaneously. Let $E_P = \{P^1, \ldots, P^\ell\}$ be the set of these patterns. The existence of several patterns raises the question of how they can interact with each other. We list here three cases that make sense from a chemical point of view:

(1) Patterns can share hexagons (regardless of their nature),

(2) Patterns can share only absent hexagons (i.e. it is allowed to share the vacuum),

(3) The patterns are pairwise disjoint.

A first naive approach to solve this multi-pattern problem is to solve a collection of single pattern problems. This would require enumerating all the single patterns that could be constructed on the basis of the patterns in E_P. But, given the combinatorics, this approach seems to be out of the question. Therefore we propose below to adapt the models we present in the previous section.

5.1 First Model

As usual, we start with the general model $\mathcal{M}$. Then, we add, for each pattern P^j of E_P, a set of variables e_i^j equivalent to the variables e_i for a single pattern P in the model $\mathcal{M}_{i_1}$ as well as the associated sum constraint. Of course, this assumes to have computed in advance all the possible fragments of each pattern of E_P. This defines the model $\mathcal{M}^1_{m_1}$.

In order to obtain pairwise disjoint patterns (model $\mathcal{M}^3_{m_1}$), one must add to the model $\mathcal{M}^1_{m_1}$ the mutual exclusion clauses $e_i^j = 0 \vee e_{i'}^{j'} = 0$ for each pair of overlapping fragments $\{F_i^j, F_{i'}^{j'}\}$ (i.e. fragments such that $(F^j_{i+} \cup F^j_{i-} \cup F^j_{i\circ} \cup F^j_{i*}) \cap (F^{j'}_{i'+} \cup F^{j'}_{i'-} \cup F^{j'}_{i'\circ} \cup F^{j'}_{i'*})$).

To share only vacuum (model $\mathcal{M}^3_{m_1}$), we add, to the model $\mathcal{M}^1_{m_1}$, constraints of the form $e_i^j = 0 \vee e_{i'}^{j'} = 0$ as soon as F_i^j and $F_{i'}^{j'}$ can share a present hexagon (i.e. if $(F^j_{i+} \cap F^{j'}_{i'+}) \cup (F^j_{i+} \cap F^{j'}_{i'\circ}) \cup (F^j_{i\circ} \cap F^{j'}_{i'+}) \neq \emptyset$)). Otherwise, if they share neutral hexagons, these hexagons must be absent from the structure, which is ensured by posting the constraint $(e_i^j = 1 \wedge e_{i'}^{j'} = 1) \Rightarrow x_h = 0$ for each hexagon $h \in F^j_{i'\circ} \cap F^{j'}_{i'\circ}$.

5.2 Second Model

We start with the general model $\mathcal{M}$. Then, we add, for each pattern P^j of E_P, a set of variables f_i^j equivalent to the variables f_i for a single pattern P in the model $\mathcal{M}_{i_2}$ as well as all the associated constraints. However, for each table constraint defining the pattern P^j, we introduce a Boolean variable t^j into its scope. This variable is set to 1 if the associated configuration is obtained after applying an axial symmetry on P^j, 0 otherwise. Thus, the table will list all valid configurations obtained from the pattern P^j or its image by an axial symmetry. Taking into account axial symmetries in the case of the inclusion of several patterns is required in order to list all the possibilities of combining the patterns with each other. Several axes of symmetries are possible. However, it is sufficient to consider only one,

as the others can be obtained by combining with 60° rotations. The use of the variable t^j within each table constraint defining P^j guarantees that globally, one exploits either the pattern P^j if t^j is set to 0, or its image by axial symmetry otherwise. This avoids considering erroneous fragments of which one part would correspond to P^j and another to its image by symmetry. Note that, in the case of a single pattern, the use of this variable t^j would only add equivalent solutions to those already produced. The model we have just described corresponds to case (1). We denote it $\mathcal{M}^1_{m_2}$.

Then, to deal with the case (2) allowing sharing only absent hexagons, we take the model $\mathcal{M}^1_{m_2}$ and add mutual exclusion constraints for the present hexagons. This amounts to posting the following constraint for each hexagon h of the coronenoid of size $k(n)$: $x_h = 1 \Rightarrow Count(\{f^1_h, \ldots, f^\ell_h\}, \{1 \ldots, n_{E_P}\}) \leq 1$ with $n_{E_P} = \max_{P^j \in E_P} n_{P^j}$. In other words, if the hexagon h is present, it can participate in at most one fragment. We denote $\mathcal{M}^2_{m_2}$ this model.

Finally, in order to consider pairwise disjoint patterns (case (3)), we need to consider hexagons that might be shared outside the coronenoid of size $k(n)$. To do this, we define the order k_{E_P} of the set E_P as the maximum order of a pattern P^j of E_P. Then, we consider the model $\mathcal{M}^1_{m_2}$ but in a coronenoid of size $k(n) + k_{E_P}$. In other words, we add to $\mathcal{M}^1_{m_2}$ a variable f^j_i per hexagon located outside the coronenoid of size $k(n)$ and per pattern P^j. Since all hexagons are represented explicitly, the table constraints are defined taking into account these new variables and the *Count* constraints of $\mathcal{M}^1_{m_2}$ for negative or neutral hexagons j' of the P^j pattern are now of the form $Count(\{f^j_1, \ldots, f^j_{n^j_c}\}, \{j'\}) = 1$. Finally, we add a mutual exclusion constraint $Count(\{f^1_h, \ldots, f^\ell_h\}, \{1 \ldots, n_{E_P}\}) = 1$ for each hexagon h of the coronenoid of size $k(n) + k_{E_P}$. We denote $\mathcal{M}^3_{m_2}$ this model.

5.3 Third Model

The principle is the same as for the two previous models. For each pattern P^j of E_P, we add to the model $\mathcal{M}$ a set of variables s^j_i equivalent to the variables s_i used for a single pattern P in the model $\mathcal{M}_{i_3}$ as well as all the associated constraints. By doing so, we obtain the model $\mathcal{M}^1_{m_3}$ corresponding to case (1). Since the model $\mathcal{M}_{i_3}$ depends on the order of the considered pattern, the generated structures will have to be embedded in a coronenoid of size $k(n) + k_{E_P}$. Of course, as in $\mathcal{M}_{i_3}$, each pattern must be preprocessed beforehand in order to remove any ambiguity.

Then, we can extend this model to the model $\mathcal{M}^3_{m_3}$ in order to take into account pairwise disjoint patterns, by adding the mutual exclusion constraint $\texttt{alldifferent}(\{s^1_1, \ldots, s^1_{n_{P^1}}\} \cup \ldots \cup \{s^\ell_1, \ldots, s^\ell_{n_{P^\ell}}\})$.

Finally, the model $\mathcal{M}^2_{m_3}$ corresponding to case (2) is obtained from the model $\mathcal{M}^1_{m_3}$ model, by adding the following constraints:

- $\texttt{alldifferent}(\{s^j_i | j \in \{1, \ldots, \ell\}, i \in P^j_+\})$ which expresses that the positive hexagons in x_G are pairwise disjoint,
- $\forall j, j' \in \{1, \ldots, \ell\}, j < j', \forall i \in P^j_\circ, \forall i' \in P^{j'}_\circ, s^j_i = s^{j'}_{i'} \Rightarrow s^j_i > n_c \vee Element(\{x_z | z \in \{1, \ldots, n_c\}\}, s^j_i) = 0$[2] which expresses the fact that if two neutral hexagons designate the same hexagon of x_G, then the corresponding vertex does not appear in x_G.
- $\forall j, j' \in \{1, \ldots \ell\}, j \neq j', \forall i \in P^j_\circ, \forall i' \in P^{j'}_+, s^j_i \neq s^{j'}_{i'}$ which prohibits having the same hexagon of x_G for a neutral hexagon and a positive hexagon of two different patterns.

[2] As a reminder, the constraint $Element(Y, j) \odot k$ is satisfied if the value of the j-th variable of Y satisfies the condition with respect to the operator $\odot$ and the value k.

6 Others Problems About Patterns

We now turn to some related problems around patterns. First, we deal with the exclusion of a pattern before showing how to constraint the number of occurrences of a given pattern.

6.1 Generating Structures Excluding a Pattern

We now aim to generate all the structures having n hexagons and not containing a given pattern P. The reasoning followed for the models $\mathcal{M}_{i_2}$ and $\mathcal{M}_{i_3}$ seems to be unsuitable because we would have to guarantee that there exists no suitable f_i numbering or isomorphic subgraph. Therefore, we follow here the same reasoning as for the model $\mathcal{M}_{i_1}$. More precisely, we start from the model $\mathcal{M}$ and add to it a variable e_i per possible fragment in a coronenoid of size $k(n)$. Each variable e_i is true if the constraint $\bigwedge_{j \in F_{i-}} x_j = 0 \wedge \bigwedge_{j \in F_{i+}} x_j = 1$ is satisfied (i.e. the fragment is present in the structure). Finally, we set a sum constraint $\sum_j e_j = 0$. This model is denoted $\mathcal{M}^1_{e_1}$. An equivalent formulation consists in representing directly each fragment F_i as a nogood $\bigvee_{j \in F_{i-}} x_j = 1 \vee \bigvee_{j \in F_{i+}} x_j = 0$, leading to a model denoted $\mathcal{M}^2_{e_1}$.

6.2 Generating Structures by Constraining the Number of Occurrences

Some constraints on the number of occurrences of a pattern P are easy to model. For example, to generate benzenoid structures with at least k pairwise disjoint occurrences of the pattern P, one can use any model among the models $\mathcal{M}^3_{m_1}$, $\mathcal{M}^3_{m_2}$ and $\mathcal{M}^3_{m_3}$ and a set E_P consisting of k times the pattern P. Others are a bit trickier. In order to make easier the expression of such constraints, we define a variable n_e which represents the number of occurrences of the pattern P contained in the generated structure and on which we will place the appropriate constraints according to the needs of the chemists. The variable n_e has the domain $\{0, \ldots, k_{max}\}$ with k_{max} the maximum number of occurrences that the structure can contain. By default, if no information is given as input on k_{max}, we take $k_{max} = \left\lfloor \frac{n}{|P_+|} \right\rfloor$.

Once again, the approach followed in the model $\mathcal{M}_{i_1}$ seems to be the most appropriate. Thus, starting from the model $\mathcal{M}_{i_1}$, we integrate the variable n_e. In addition to the variables e_i introduced for each possible fragment, we add a Boolean variable e'_i per fragment. The variable e'_i is true if the fragment F_i is present in the pattern. This is ensured by adding, for each fragment F_i the constraint $\bigwedge_{j \in F_{i-}} x_j = 0 \wedge \bigwedge_{j \in F_{i+}} x_j = 1 \Rightarrow e'_i = 1$. Then, as some fragments may share some hexagons, we add mutual exclusion constraints, with, for each hexagon h, the constraint $\sum_{i | h \in F_i} e'_i \geq 1 \Rightarrow \sum_{i | h \in F_i} e_i = 1$ (given that $F_i = F_{i+} \cup F_{i-} \cup F_{i\circ} \cup F_{i*}$). Thus, this guarantees that if a hexagon participates simultaneously in several fragments, only one of these fragments is considered as present. Finally, the constraint $\sum_j e_j = n_e$ allows us to compute the number of occurrences on which we can then easily put any arithmetic constraint. It is also possible to use this variable to find the structures maximizing the number of occurrences of the pattern.

7 Experiments

In this section, we assess empirically the different proposed models. To this end, we consider the eight patterns described in Figures 5(a)-(g) and Figure 3(a) from [34] and vary the number n of hexagons present in the structures from the number of positive hexagons in

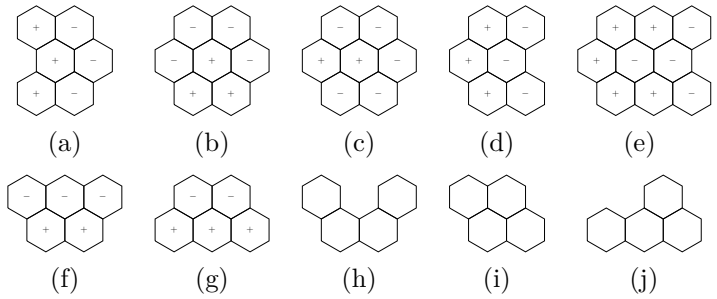


Figure 5 The patterns used as benchmarks in addition to the pattern *deep bay*: *armchair edge* (a), C_3H_3 *protusion* (b), C_4H_4 *protusion* (c), *shallow armchair bay* (d), *ultra deep bay* (e), *zigzag bay* (f), shortened to *zigzag* in [26], and *zigzag edge* (g). Two of the four benzenoid structures of four hexagons containing an instance of the pattern *armchair edge* (h) and (i). One of the three benzenoid structures of four hexagons containing no instance of the pattern *armchair edge* (j).

Table 2 The number of instances which are successfully processed (#I) and the related cumulative runtime in hours (Time) for each possible variable and value heuristics and for model $\mathcal{M}_{i_1}$, $\mathcal{M}_{i_2}$ and $\mathcal{M}_{i_3}$.

	$\mathcal{M}_{i_1}$				$\mathcal{M}_{i_2}$				$\mathcal{M}_{i_3}$			
	inc		*desc*		*inc*		*desc*		*inc*		*desc*	
	#I	Time	#I	Time	#I	Time	#I	Time	#I	Time	#I	Time
dom	55	1.44	55	1.42	55	1.51	55	1.49	55	1.54	55	1.52
dom/wdeg	47	23.58	48	17.71	47	22.07	48	16.90	48	20.93	48	17.40
$dom/wdeg^{ca.cd}$	50	12.56	55	4.85	50	15.47	50	15.35	49	16.47	50	14.76
CHS	43	27.06	41	28.97	47	23.04	48	20.69	48	20.42	48	17.43

the pattern to 9. This allows us to produce 55 instances (resp. 135) of the problem of generating structures containing/excluding one pattern (resp. containing two patterns). Our implementation is based on Choco (v. 4.10.7). We consider four state-of-the-art variable ordering heuristics namely *dom/wdeg* [2], $dom/wdeg^{ca.cd}$ [33], *CHS* [15] and *dom*. This latter chooses as next variable the first variable in the lexicographical ordering having the smallest domain. Regarding the value ordering heuristic, we use the heuristics *inc* and *desc* which choose respectively the smallest and the largest value first. The experiments are carried out on DELL PowerEdge R440 servers with an Intel Xeon 4112 2.6 GHz processor and 32 GB of memory. The runtime for processing an instance is limited to 2 hours. We do not compare our approach with an existing method because, to our knowledge, no such method has been proposed yet, probably due to the fact that this line of research has emerged recently.

First, from Table 2, we can observe that, whatever the model (among $\mathcal{M}_{i_1}$, $\mathcal{M}_{i_2}$ and $\mathcal{M}_{i_3}$) or the variable heuristic, the best results are generally obtained with the value heuristic *desc*. This can be explained by the fact that each model mainly involves Boolean variables. For instance, assigning 1 to a variable x_i amounts to create a hexagon and so allow us to exploit more quickly most of the constraints of the general model $\mathcal{M}$. Now, regarding the variable heuristic, the most sophisticated heuristics are not those leading to the best results. Indeed, whatever the model, the heuristic *dom* turns to be the more relevant for our problem. Moreover, as shown in Figure 6(a), *dom* performs better than $dom/wdeg^{ca.cd}$ (which is the best variable heuristic after *dom*) on all the considered instances. This may seem surprising, but, if we look closely at the definition of *dom*, we can note that it corresponds to start the search with the hexagons located on the top edge of the coronenoid. At the same time, the more sophisticated heuristics may be penalized by the uniformity of the problem. Finally, for

Table 3 The number of instances which are successfully processed (#I) and the related cumulative runtime in hours (Time) for each possible variable and value heuristics and for model $\mathcal{M}^3_{m_1}$, $\mathcal{M}^3_{m_2}$ and $\mathcal{M}^3_{m_3}$.

	$\mathcal{M}^3_{m_1}$				$\mathcal{M}^3_{m_2}$				$\mathcal{M}^3_{m_3}$			
	inc		*desc*		*inc*		*desc*		*inc*		*desc*	
	#I	Time	#I	Time	#I	Time	#I	Time	#I	Time	#I	Time
dom	135	6.82	135	6.85	129	44.37	135	17.76	135	7.68	135	7.78
dom/wdeg	100	100.16	113	74.69	105	92.48	108	77.77	106	82.99	113	63.06
$dom/wdeg^{ca.cd}$	121	51.45	134	25.64	93	101.77	105	81.44	96	102.07	105	79.44
CHS	85	116.19	83	120.62	86	106.24	70	136.93	108	75.84	110	66.61

given value and variable heuristics, we can note that the models often obtain close results. If we focus our attention on *dom* and *desc* (see Figures 6(b)-(d)), $\mathcal{M}_{i_1}$ turns out to perform slightly better than $\mathcal{M}_{i_2}$, which itself is better than $\mathcal{M}_{i_3}$. Maybe, this could be explained by the fact that all the models are based on the general model $\mathcal{M}$ to which some variables and constraints are added. Indeed, at the end, the models have similar numbers of constraints while the model $\mathcal{M}_{i_1}$ requires a few more variables than the other models.

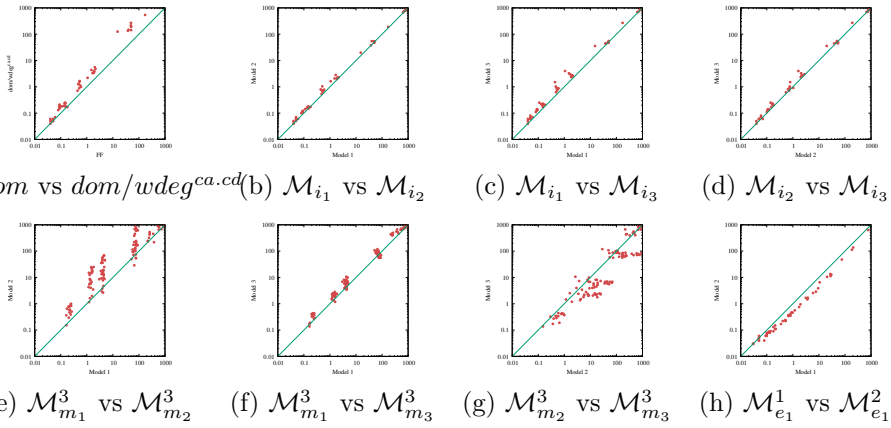
If we are now interested in the generation of structures containing two given patterns, we can note that the observed trends in Table 3 are quite similar to those obtained for a single pattern. Again, the value heuristic *desc* leads to the best results. Regarding the variable heuristic, *dom* is once more the most relevant and robust one. A slight difference from the single pattern case is that the efficiency of the other variables heuristic seems to depend on the model we consider. Beyond, we can observe that the differences between the models, whatever the variable heuristic, are more pronounced. Globally, the model $\mathcal{M}^3_{m_1}$ turns out to be the best one followed by the model $\mathcal{M}^3_{m_3}$ while the model $\mathcal{M}^3_{m_2}$ turns out to perform worst. This is clearly visible on Figures 6(e)-(g) when considering the heuristics *dom* and *desc*. This result seems to be correlated with the number of constraints which is twice as large for the model $\mathcal{M}^3_{m_2}$ than for $\mathcal{M}^3_{m_1}$ or $\mathcal{M}^3_{m_3}$.

Regarding the exclusion of a given pattern, the trends we observe for value and variable heuristics are similar to previous comparisons. By lack of place, we do not provide more details. If we compare the two models $\mathcal{M}^1_{e_1}$ and $\mathcal{M}^2_{e_1}$ (see Figure 6(h)), it appears that the latter is the most efficient. Using the heuristics *dom* and *desc*, both achieve the same exploration of the search space, but the model $\mathcal{M}^2_{e_1}$ does not consider additional variables w.r.t. the general model $\mathcal{M}$. Moreover, it only requires some clauses as additional constraints.

Finally, in Figure 7(b), we compare the average number of solutions (some of them are depicted in Figures 5(i)–(j) and 7(a)) with the number of benzenoid structures depending on the considered problem and the number n of hexagons. This figure first allows us to observe the growth of the number of structures with n. Then, we can also notice that we have to compute a large number of benzenoid structures. For instance, for the inclusion of a single pattern, we have to consider about 70% of all the benzenoid structures.

8 Conclusion and Perspectives

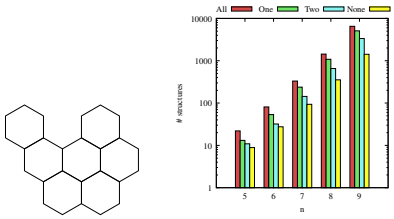
We have presented an approach based on CP to generate exhaustively benzenoid structures satisfying certain constraints around patterns. For this purpose, several models have been considered and compared. The model based on the identification of all the possible locations of a fragment turns out to be the most robust. It leads to an efficient solving while being able to deal with several questions about patterns (inclusion or exclusion, number of occurrences, ...).



(a) dom vs $dom/wdeg^{ca.cd}$ (b) $\mathcal{M}_{i_1}$ vs $\mathcal{M}_{i_2}$ (c) $\mathcal{M}_{i_1}$ vs $\mathcal{M}_{i_3}$ (d) $\mathcal{M}_{i_2}$ vs $\mathcal{M}_{i_3}$

(e) $\mathcal{M}^3_{m_1}$ vs $\mathcal{M}^3_{m_2}$ (f) $\mathcal{M}^3_{m_1}$ vs $\mathcal{M}^3_{m_3}$ (g) $\mathcal{M}^3_{m_2}$ vs $\mathcal{M}^3_{m_3}$ (h) $\mathcal{M}^1_{e_1}$ vs $\mathcal{M}^2_{e_1}$

Figure 6 Comparison of the variable heuristics dom and $dom/wdeg^{ca.cd}$ based on the runtime (in seconds) for $\mathcal{M}_{i_1}$ and the value heuristic $desc$. Pairwise comparison of models based on the runtime (in seconds) when using the variable heuristic dom and the value heuristic $desc$ (b)-(h).



(a) (b)

Figure 7 One of the twenty-five benzenoid structures of seven hexagons containing the patterns *armchair edge* and *deep bay* (a). Number of benzenoid structures (all) and average number of benzenoid structures containing a given pattern (one), two given patterns (two) or excluding a given pattern (none) (b).

In a way, we have proposed a modeling brick per issue which can be combined each other or with global properties (e.g. those defined in [5]) depending on the needs of chemists. To our knowledge, this work provides chemists with the first tool for generating benzenoid structures satisfying certain conditions on their edge topology. It would be useful in validating their theoretical models or identifying the most promising benzenoid structures before trying to synthesize them, what are currently hot topics in chemistry.

As a consequence, a first perspective of this work will be to study its repercussions from the viewpoint of chemistry (e.g. by generating all the structures containing some given patterns and applying to them some theoretical chemistry tools [7, 4]). Concerning the modeling, other forms of interaction between two patterns can be of interest to chemists (e.g. by sharing a positive hexagon only if the two patterns do not use the same bonds). Finally, several avenues can be explored to improve the practical efficiency of the approach.

References

1 M. R. Ajayakumar, Ji Ma, Andrea Lucotti, Karl Sebastian Schellhammer, Gianluca Serra, Evgenia Dmitrieva, Marco Rosenkranz, Hartmut Komber, Junzhi Liu, Frank Ortmann, Matteo Tommasini, and Xinliang Feng. Persistent peri-Heptacene: Synthesis and In Situ Characterization. *Angew. Chem. Int. Ed.*, 2021. doi:10.1002/anie.202102757.

2 Frédéric Boussemart, Fred Hemery, Christophe Lecoutre, and Lakhdar Sais. Boosting Systematic Search by Weighting Constraints. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence (ECAI)*, pages 146–150, 2004.

3 Gunnar Brinkmann, Gilles Caporossi, and Pierre Hansen:. A Constructive Enumeration of Fusenes and Benzenoids. *Journal of Algorithms*, 45(2), 2002. doi:10.1016/S0196-6774(02)00215-8.

4 Yannick Carissan, Chisom-Adaobi Dim, Denis Hagebaum-Reignier, Nicolas Prcovic, Cyril Terrioux, and Adrien Varet. Computing the Local Aromaticity of Benzenoids Thanks to Constraint Programming. In *Proceedings of the 26th International Conference on Principles and Practice of Constraint Programming (CP)*, pages 673–689, 2020. doi:10.1007/978-3-030-58475-7_39.

5 Yannick Carissan, Denis Hagebaum-Reignier, Nicolas Prcovic, Cyril Terrioux, and Adrien Varet. Using Constraint Programming to Generate Benzenoid Structures in Theoretical Chemistry. In *Proceedings of the 26th International Conference on Principles and Practice of Constraint Programming (CP)*, pages 690–706, 2020. doi:10.1007/978-3-030-58475-7_40.

6 Ying Chen, Chaojun Lin, Zhixing Luo, Zhibo Yin, Haonan Shi, Yanpeng Zhu, and Jiaobing Wang. Double π-Extended Undecabenzo[7]helicene. *Angew. Chem. Int. Ed.*, 60(14):7796–7801, 2021. doi:10.1002/anie.202014621.

7 Zhongfang Chen, Chaitanya S. Wannere, Clémence Corminboeuf, Ralph Puchta, and Paul von Ragué Schleyer. Nucleus-Independent Chemical Shifts (NICS) as an Aromaticity Criterion. *Chem Rev*, 105:3842–3888, 2005. doi:10.1021/cr030088.

8 Kwan Yin Cheung, Kosuke Watanabe, Yasutomo Segawa, and Kenichiro Itami. Synthesis of a zigzag carbon nanobelt. *Nat. Chem.*, 13(3):255–259, 2021. doi:10.1038/s41557-020-00627-5.

9 J. Cyvin, J. Brunvoll, and B. N. Cyvin. Search for Concealed Non-Kekuléan Benzenoids and Coronoids. *J. Chem. Inf. Comput. Sci.*, 29(4):237, 1989. doi:10.1021/ci00064a002.

10 Jo Devriendt, Bart Bogaerts, Maurice Bruynooghe, and Marc Denecker. Improved Static Symmetry Breaking for SAT. In *Proceedings of the 19th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 104–122, 2016. doi:10.1007/978-3-319-40970-2_8.

11 Tim Dumslaff, Yanwei Gu, Giuseppe M. Paternò, Zijie Qiu, Ali Maghsoumi, Matteo Tommasini, Xinliang Feng, Francesco Scotognella, Akimitsu Narita, and Klaus Müllen. Hexa-peribenzocoronene with two extra K-regions in an ortho-configuration. *Chem. Sci.*, 11(47):12816–12821, 2020. doi:10.1039/D0SC04649C.

12 Jean-Guillaume Fages. *Exploitation de structures de graphe en programmation par contraintes*. PhD thesis, École des mines de Nantes, France, 2014.

13 Jean-Guillaume Fages, Xavier Lorca, and Charles Prud'homme. Choco solver user guide documentation. https://choco-solver.readthedocs.io/en/latest/.

14 Kei Fujise, Eiji Tsurumaki, Kan Wakamatsu, and Shinji Toyota. Construction of Helical Structures with Multiple Fused Anthracenes: Structures and Properties of Long Expanded Helicenes. *Chemistry – A European Journal*, 27(14):4548–4552, March 2021. doi:10.1002/chem.202004720.

15 Djamal Habet and Cyril Terrioux. Conflict History based Search for Constraint Satisfaction Problem. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC)*, pages 1117–1122, 2019. doi:10.1145/3297280.3297389.

16 Sindhu Kancherla and Kåre B. Jørgensen. Synthesis of Phenacene–Helicene Hybrids by Directed Remote Metalation. *J. Org. Chem.*, 85(17):11140–11153, 2020. doi:10.1021/acs.joc.0c01097.

17 Ashok Keerthi, Carlos Sánchez-Sánchez, Okan Deniz, Pascal Ruffieux, Dieter Schollmeyer, Xinliang Feng, Akimitsu Narita, Roman Fasel, and Klaus Müllen. On-surface Synthesis of a Chiral Graphene Nanoribbon with Mixed Edge Structure. *Chem. – Asian J.*, 15(22):3807–3811, 2020. doi:10.1002/asia.202001008.

18 Christophe Lecoutre and Olivier Roussel, editors. *Proceedings of the 2018 XCSP3 Competition*, 2018. arXiv:1901.01830.

19 Junzhi Liu and Xinliang Feng. Synthetic tailoring of graphene nanostructures with Zigzag-Edged topologies: Progress and perspectives. *Angewandte Chemie International Edition*, 59:2–18, 2020. doi:10.1002/anie.202008838.

20 Max M. Martin, Frank Hampel, and Norbert Jux. A Hexabenzocoronene-Based Helical Nanographene. *Chem. – Eur. J.*, 26(45):10210–10212, 2020. doi:10.1002/chem.202001471.

21 Shantanu Mishra, Doreen Beyer, Kristjan Eimre, Shawulienu Kezilebieke, Reinhard Berger, Oliver Gröning, Carlo A. Pignedoli, Klaus Müllen, Peter Liljeroth, Pascal Ruffieux, Xinliang Feng, and Roman Fasel. Topological frustration induces unconventional magnetism in a nanographene. *Nature Nanotechnology*, 15(1):22–28, 2020. doi:10.1038/s41565-019-0577-9.

22 Shantanu Mishra, Doreen Beyer, Kristjan Eimre, Junzhi Liu, Reinhard Berger, Oliver Gröning, Carlo A. Pignedoli, Klaus Müllen, Roman Fasel, Xinliang Feng, and Pascal Ruffieux. Synthesis and Characterization of π-Extended Triangulene. *Journal of the American Chemical Society*, 141(27):10621–10625, 2019. doi:10.1021/jacs.9b05319.

23 Shantanu Mishra, Xuelin Yao, Qiang Chen, Kristjan Eimre, Oliver Gröning, Ricardo Ortiz, Marco Di Giovannantonio, Juan Carlos Sancho-García, Joaquín Fernández-Rossier, Carlo A. Pignedoli, Klaus Müllen, Pascal Ruffieux, Akimitsu Narita, and Roman Fasel. Large magnetic exchange coupling in rhombus-shaped nanographenes with zigzag periphery. *Nat. Chem.*, pages 1–6, 2021. doi:10.1038/s41557-021-00678-2.

24 Tadashi Mori. Chiroptical Properties of Symmetric Double, Triple, and Multiple Helicenes. *Chem. Rev.*, 121(4):2373–2412, 2021. doi:10.1021/acs.chemrev.0c01017.

25 Marvin Nathusius, Barbara Ejlli, Frank Rominger, Jan Freudenberg, Uwe H. F. Bunz, and Klaus Müllen. Chrysene-Based Blue Emitters. *Chemistry – A European Journal*, 26(66):15089–15093, 2020. doi:10.1002/chem.202001808.

26 Wenhui Niu, Ji Ma, Paniz Soltani, Wenhao Zheng, Fupin Liu, Alexey A. Popov, Jan J. Weigand, Hartmut Komber, Emanuele Poliani, Cinzia Casiraghi, Jörn Droste, Michael Ryan Hansen, Silvio Osella, David Beljonne, Mischa Bonn, Hai I. Wang, Xinliang Feng, Junzhi Liu, and Yiyong Mai. A Curved Graphene Nanoribbon with Multi-Edge Structure and High Intrinsic Charge Carrier Mobility. *J. Am. Chem. Soc.*, 142(43):18293–18298, 2020. doi:10.1021/jacs.0c07013.

27 Michele Pizzochero and Efthimios Kaxiras. Imprinting Tunable π-Magnetism in Graphene Nanoribbons via Edge Extensions. *J. Phys. Chem. Lett.*, 12(4):1214–1219, February 2021. doi:10.1021/acs.jpclett.0c03677.

28 Zijie Qiu, Cheng-Wei Ju, Lucas Frédéric, Yunbin Hu, Dieter Schollmeyer, Grégory Pieters, Klaus Müllen, and Akimitsu Narita. Amplification of Dissymmetry Factors in π-Extended [7]- and [9]Helicenes. *J. Am. Chem. Soc.*, 143(12):4661–4667, March 2021. doi:10.1021/jacs.0c13197.

29 Zijie Qiu, Akimitsu Narita, and Klaus Müllen. Carbon nanostructures by macromolecular design from branched polyphenylenes to nanographenes and graphene nanoribbons. *Faraday Discussions*, 2020. Publisher: The Royal Society of Chemistry. doi:10.1039/D0FD00023J.

30 Georges Trinquier and 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. doi:10.1021/acs.jpca.7b11095.

31 Mizuho Uryu, Taito Hiraga, Yoshito Koga, Yutaro Saito, Kei Murakami, and Kenichiro Itami. Synthesis of Polybenzoacenes: Annulative Dimerization of Phenylene Triflate by Twofold C-H Activation. *Angew. Chem.*, 132(16):6613–6616, 2020. doi:10.1002/ange.202001211.

32 Hélène Verhaeghe, Christophe Lecoutre, and Pierre Schaus. Extending Compact-Table to Negative and Short Tables. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3951–3957, 2017. URL: https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14359/14122.

33 Hugues Wattez, Christophe Lecoutre, Anastasia Paparrizou, and Sébastien Tabary. Refining Constraint Weighting. In *Proceedings of the 31st IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 71–77, 2019. doi:10.1109/ICTAI.2019.00019.

34 Natalie Wohner, Pui K. Lam, and Klaus 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. doi:10.1016/j.carbon.2014.11.004.

35 Zeming Xia, Sai Ho Pun, Han Chen, and Qian Miao. Synthesis of Zigzag Carbon Nanobelts through Scholl Reactions. *Angew. Chem. Int. Ed.*, 60(18):10311–10318, 2021. doi:10.1002/anie.202100343.

36 Xuan Yang, Frank Rominger, and Michael Mastalerz. Benzo-Fused Perylene Oligomers with up to 13 Linearly Annulated Rings. *Angew. Chem. Int. Ed.*, 60(14):7941–7946, 2021. doi:10.1002/anie.202017062.

37 Xuelin Yao, Wenhao Zheng, Silvio Osella, Zijie Qiu, Shuai Fu, Dieter Schollmeyer, Beate Müller, David Beljonne, Mischa Bonn, Hai I. Wang, Klaus Müllen, and Akimitsu Narita. Synthesis of Nonplanar Graphene Nanoribbon with Fjord Edges. *J. Am. Chem. Soc.*, 143(15):5654–5658, 2021. doi:10.1021/jacs.1c01882.

38 Cheng Zeng, Bohan Wang, Huanhuan Zhang, Mingxiao Sun, Liangbin Huang, Yanwei Gu, Zijie Qiu, Klaus Müllen, Cheng Gu, and Yuguang Ma. Electrochemical Synthesis, Deposition, and Doping of Polycyclic Aromatic Hydrocarbon Films. *J. Am. Chem. Soc.*, 143(7):2682–2687, February 2021. doi:10.1021/jacs.0c13298.