This is a preprint version of the paper published in
DIMACS Series in Discrete Mathematics and Theoretical Computer Science
Vol. 51 (Discrete Mathematical Chemistry)
ISBN-13: 978-0-8218-0987-7

An abstract representation for molecular graphs

Philippe Vismara and Claude Laurenço

Abstract. The design of a synthesis strategy in Organic Chemistry rests on the perception of the target molecule. This one has to be perceived from several viewpoints: topology, stereochemistry, functionality... We propose an abstract representation of molecules which describes the topological viewpoint (cycles, carbon chains and their structural relationships). This representation is based on a polynomial algorithm that computes the set of relevant cycles in the molecular graph. This set is equal to the union of all the minimum cycle bases of the graph. The abstract representation is integrated in RESYN, a system for computer-aided organic synthesis planning.

Introduction

Since the early sixties, several computer programs have been developed in the field of synthetic organic chemistry [**ON92**]. Although the synthetic chemists extensively use database systems for molecule and reaction retrieval, they have not yet recognize any computer program as an essential tool for devising synthesis strategies.

The synthesis of organic molecules is a problem which is solved in finding an efficient way to construct a given target molecule, for instance a natural product, from available simple starting materials, by applying chemical reactions in a certain sequence. These successive steps define a synthetic pathway. Retrosynthetic analysis [**CC89**], a reasoning backward from the target structure, is often used to design a synthetic pathway and to find a suitable set of starting materials. Because thousands of transformations are known and thousands of potential starting materials are available, it is impossible to perform a systematic analysis. Therefore, synthesis design programs have to work out strategies in order to reduce the size of the search space.

Many factors have limited the development of computer-assisted synthesis design, whose LHASA is a representative program [**CLR85**]. The combinatorics complexity of synthesis design is not the only obstacle. Most of the existing programs have been developed for several decades and they generally do not integrate new computer technologies. These programs often produce too many results without explaining them, specially when heuristics have been applied to reduce the size of the search space. Moreover, creating and updating the knowledge bases of

The authors are associate researchers at the LIRMM (CNRS / Université Montpellier II).

such systems remain a major difficulty and are hard tasks for experts in organic synthesis.

The RESYN project have been developed taking the weakness of current programs into consideration. The aim of the project is to mix research into modelling of organic synthesis design [**Lau85, Gie98, DFHL98**] and computer science results in Graph Theory, Constraint Satisfaction Problems, Distributed Artificial Intelligence, Classification, Analogy and Case-based reasoning, Knowledge Acquisition and Representation [**Vis95, R95, Lie97, Py92**]. A system for computer-assisted synthesis design is currently under development.

The aim of this paper is not to present the whole RESYN project. We focus on the topological perception of the target molecule. This analysis is a crucial step in the definition of retrosynthetic strategies. These strategies are also based on the perception of stereochemistry and functional groups which are not presented in this paper.

The perception of a molecule involves recognizing it as a member of conceptual categories and provides diverse types of information such as properties of that molecule, its shape and parts which make it up. All this information is a base for action. If we recognize that a molecule belongs to a chemical family, we may interpret that in terms of known synthetic methods or starting materials for its construction. The process starts with the target structural formula whose information is interpreted to construct a new representation. This one describes at several abstract levels, in a structured way, multiple aspects of the molecule and gives a view, both global and detailed, of the problem which is to be solved. The topological perception of a molecular graph consists in the recognition of cyclic parts and the description of non-cyclic patterns as carbon chains, heteroatomic links .... The main part of this paper aims at the computation of the set of relevant cycles in the molecule. The second part presents an abstract representation of the molecular graph which describes the structural relationships between patterns (cycles, carbon chains,...).

Relevant cycles

Study of cycles in molecules is a very long and rich story. It deals with finding the better set of cycles to analyze the cyclic structure of a molecule. A quite complete review of papers dealing with cycles in molecule has been done by [**DGHL89a**]. Therefore, the aim of this part is to perform a more precise analysis of some works from a Graph Theory point of view.

This part is organized as follows: Section 2 presents some algorithms that search for a cycle basis of the cycle vector space associated with a molecular graph. Two major kinds of cycles basis are studied: *fundamental* basis (associated to a *spanning tree*) and *minimum* basis (often named SSSR). We show how successive papers have tried to optimize the search for a minimum basis, until [**Hor87**] that presents a polynomial algorithm.

Generally, cycle basis notion is not satisfactory to perform a good analysis of the cyclic structure of the molecule. Hence, we present in section 3 a few papers that deals with extended minimum cycle basis. We discuss on their interests and limits before trying to define a general notion of *Relevant Cycles*. This notion is described in section 4. We propose a polynomial time algorithm to compute the *Set of Relevant Cycles* (denoted by $\mathcal{C}_R$) as a polynomial number of cycles families

which define a partition of $\mathcal{C}_R$. Each family is represented by a single prototype from which all the other cycles of the family can be generated without backtracking.

1. Preliminaries

1.1. Definitions. A molecule is generally represented by an *undirected* graph $G = (V, E)$ without *loops* (where V denotes the set of *vertices* associated with the atoms and E defines the set of *edges* which correspond to bonds between atoms). We denote by n and m the numbers of vertices and edges respectively.

In an undirected graph, an edge is an unordered pair of vertices (v_1, v_2).

A *path* (or a *walk*) is a route between two vertices passing along edges [1]. A *simple* path must contain each edge at most one time. A path is *elementary* if no vertex appears more than once.

For any simple path μ, we define the *length* of the μ (denoted by $|\mu|$) the number of edges in μ. Define $d(x, y)$ to be the length of any shortest path from x to y.

For any vertex v in V, we denote by $d(v)$ the degree of v in G and by $\Gamma(v)$ the set of vertices adjacent to v (so $d(v) = |\Gamma(v)|$).

A *cycle* (or a *ring*) is a closed path (i.e. a path that starts and ends at the same vertex). Simple and elementary cycles are defined according to path definitions. In this paper, we consider a cycle as either a set of edges $\{(v_1, v_2), (v_2, v_3), ...\}$ or a closed path $(v_1, v_2, v_3, ..., v_1)$. A simple cycle C can also be represented by a vector of $\{0, 1\}^m$ such that $C[i] = 1$ if and only if edge i belongs to C. Then, the sum of two cycles C_1 and C_2 is defined by the boolean addition of their associated vectors (regarding cycles as sets of edges, this sum is equal to $(C_1 \cup C_2) \setminus (C_1 \cap C_2)$).One can define the *vector space of cycles* on the set of simple cycles closed by addition. All the elements of this vector space (denoted by $\mathcal{C}$) are simple cycles or sum of simple cycles. Hence, $\mathcal{C}$ is the set of all the subgraphs[2] of G in which the degree of any vertex is even.

1.2. The cyclomatic number. The main interest of the cycle vector space is that it introduces the notion of cycle basis. A cycle basis is a set of cycles such that any element of $\mathcal{C}$ can be obtained by a sum of cycles of this basis.The dimension of the vector space defines the minimum number of cycles needed to describe all cycles of the graph. This dimension is generally called the *cyclomatic number* and denoted by ν. For a graph with n vertices and m edges, ν is equal to $m - n + 1$.

This formula has a very long history. It was introduced by [**Eul52**] to determine the number of faces in a polyhedron. This result was generalized by [**Cau13**] and described on a different way by [**Frè39**] (which is generally cited as *Frerejacques number*). Ever since, some other formulas have been introduced that give the same result, like [**Elk84**].

1.3. 2-connected components. A graph is *k-connected* if any vertices v_1 and v_2 are linked by at least k distinct paths. So, a molecular graph is generally *connected* (i.e. 1-connected). A *2-connected* component of a graph G (sometimes called a *block* of G) is a 2-connected maximal subgraph of G. An important property is that all the vertices of any elementary cycle belong to the same 2-connected component. Hence, studying the cycle vector space can be independently performed on

[1] more formally, there is a path between vertices v_0 and v_k, passing through vertices v_1, v_2 ... and v_{k-1}, if each pair (v_i, v_{i+1}) is an edge of the graph (where $i = 0, 1, ..., k$ - 1).

[2] a subgraph of $G = (V, E)$ is a graph $G' = (V', U')$ such that $V' \subseteq V$ and $E' \subseteq E$

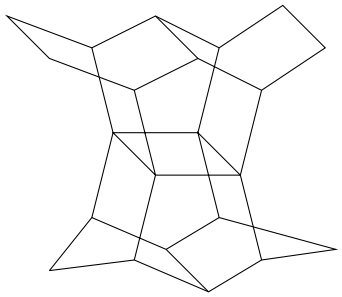


FIGURE 1. The 685 elementary cycles of this "pagodane" molecule can be generated from a cycle basis including only 11 cycles.

each 2-connected component. This decomposition of the graph generally provides a great improvement. Firstly because it eliminates all the vertices which do not belongs to any cycle (such vertices correspond to 2-connected components reduced to a single vertex). Secondly, because it divides the initial large problem into a few smallest ones.

Notion of 2-connectivity is seldom used by papers dealing with cycles in chemistry. Some of them, like [**LDP90**], suggest to recursively eliminate terminal atoms (i.e. vertices whose degree is 1). This reduction do not only preserves vertices belonging to cycles. It also conserves the vertices of any elementary path linking two independent cycles. To solve this problem, [**BSS91**] has proposed to eliminate vertices whose degree is 2. This method constructs a "crunched graph" on which the cycle search is performed. Some precautions must be taken to detect cycles that include vertices whose degree is 2. These reductions are not as efficient as 2-connectivity components. Hence, [**FPDB93**] has proposed to decompose the graph in *blocks* which correspond to 2-connected components. Unfortunately, the algorithm they presented is more complicated and has a greater time complexity than the one introduced by [**Tar72**] which can be performed in linear time.

2. Cycle basis

The set of all elementary cycles is not always relevant to analyze the cyclic structure of a graph. For example, the molecule presented in figure 1 includes 685 elementary cycles. Hence, it is often important to be able to describe the cycle vector space without enumerating all cycles. Of course, a cycle basis provides such an optimal representation of the vector space.

2.1. Fundamental basis. The first and simplest way used to construct a cycle basis consists in finding a *fundamental basis associated with a spanning tree*. This method has been introduced by [**Kir47**]. In a *spanning tree*[3] $T = (V, E')$ of a graph $G = (V, E)$, adding any edge from $E \setminus E'$ creates a single cycle. This cycle is called a *fundamental* cycle according to the spanning tree T. Because $|E \setminus E'| = \nu$ and all these cycles are independent[4], they form a basis of the cycle space. Many

[3]A spanning tree is a connected subgraph without cycles which covers all the vertices of the graph.

[4]We only add in T one edge of $E \setminus E'$ at the same time. So, each fundamental cycle includes an edge which cannot belong to any other cycle.

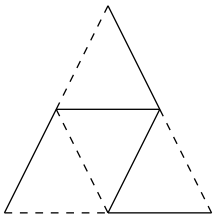


FIGURE 2. The minimum cycle basis of this graph consists of four 3-edges cycles. But no fundamental basis should include all of them.

papers deal with fundamental basis search [**Wel66, Pat69, Gib69**]. Most of them use this notion to enumerate all elementary cycles of the graph. Unfortunately, [**MD75**] has shown that such a method can generate 2^ν vectors a few of which actually correspond to elementary cycles. This difficulty can be surmounted using a search algorithm, as [**RT75**], which needs $O(n + m + nc)$ operations (where c is the number of elementary cycles). An other search algorithm, quite easy to implement, has been proposed by [**She83**]. For *planar graphs*[5], [**Sys81**] has shown that a vector space approach can be as efficient as a search algorithm.

Because cycle basis may be numerous, they are compared according to their lengths (the length of a cycle basis is defined by the sum of the lengths of all cycles in the basis). This notion defines a *minimum fundamental cycle basis*. [**DPK82**] showed that finding a minimum fundamental cycle basis is a NP-complete problem[6]. Note that a *minimum fundamental basis* must be associated to a spanning tree. But all cycle basis are not fundamental (see figure 2). We will see that finding a minimum cycle basis, not necessarily associated with a spanning tree, can be performed in polynomial time.

2.2. Minimum basis : SSSR. Since a few elementary cycles can be considered as "chemically relevant", the cycle vector space is generally described by a *Smallest Set of Smallest Rings*. This notion called SSSR is equivalent to a *minimum cycle basis* (not associated with a spanning tree). Hence, an SSSR contains ν independent cycles such that the sum of their lengths is minimum.

2.2.1. *SSSR computed from a fundamental basis.* The oldest method used to find an SSSR starts by searching a fundamental cycle basis. Then, this basis $\mathcal{B}$ is optimized using the following criterion: *for each cycles $C_1, C_2 \in \mathcal{B}$, if $C_1 + C_2$ is smaller than either C_1 or C_2, replace this fundamental cycle by $C_1 + C_2$ in $\mathcal{B}$.* The new set of cycles is a basis smaller than $\mathcal{B}$. This method, used by [**CP72**] and [**WD75**], does not necessarily find a minimum basis even if some other heuristics are employed. An other algorithm using fundamental basis notion has been proposed by [**HGT84**]. It directly finds a short fundamental basis that is not necessarily minimum.

2.2.2. *SSSR directly computed.* Since it is not easy to optimize a fundamental basis, many algorithms try to directly obtain a minimum basis. For "fundamental" algorithms, the main problem lies in the optimization of an initial basis. On the opposite, "direct" algorithms easily find a minimum set of independent cycles but

[5]a graph is *planar* if it can be represented on a plane in such a way that no two edges cross one another.

[6]It implicates that no polynomial time algorithm would probably be found to solve that problem

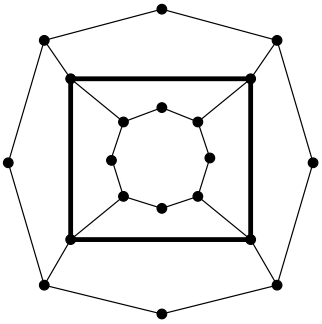


FIGURE 3. Any minimum cycle basis (SSSR) must include the 4-edges cycle. But this cycle never appears as a face of a planar representation of the graph.

have difficulties to obtain a complete basis. To minimize the basis length, most of the algorithms associate to any edge (or vertex) the shortest cycles that include it. But these cycles are not necessarily different or independent. The set finally obtained often includes less than ν cycles. Hence, general method for "direct" algorithms consists in applying a search heuristic that minimize length until ν independent cycles are obtained.

For example, [**Zam76**] has proposed to use spanning notions to find a cycle basis. The algorithm choose a vertex and search a smallest cycle that covers it. All vertices in that cycle are marked and the process goes on until every vertex is marked. If the cyclomatic number is not reached, a similar method is performed with edges and then with faces. However, the final set of cycles is not always an SSSR. The problem lies on the absence of direct relation between minimum cycles bases and faces of a planar representation of the graph. For such a representation of the molecule, the set of faces defines a cycle basis. Like fundamental ones, this basis is not necessarily minimum. It depends on the representation of the graph. And for some particular graphs, no planar representation defines a minimum basis (see figure 3).

An other interesting algorithm has been presented by [**GJ79**]. Like "fundamental" algorithms, it uses a spanning tree to obtain a set of "ring-closure edges". For any ring-closure edge, a shortest cycle is searched instead of a fundamental one. The problem is that two different closure edges can be associates with the same shortest cycle (see edges a and b in figure 4). Hence, Gasteiger has proposed to order the edges of the molecular graph according to the breadth first search that computes the spanning tree. Each cycle closure edge e is associated with the shortest cycle that only includes edges preceding e in this order. However, that simple but very powerful method can fail when a shortest cycle is only composed by ring-closure edges. For instance, in figure 5, the 3-edges cycle would not be found respecting edge orientation. Some heuristics has been proposed to solve that problem. But none of them guarantees that the final cycle basis is minimum.

All the algorithms we just have studied are failing in trying to directly find an SSSR. Whatever criterion is used, it is not possible to directly recognize a cycle which is member of a minimum basis. Hence, the solution consists in finding a set including more than ν cycles from which an SSSR can be extracted.

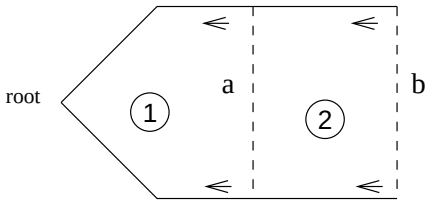


FIGURE 4. Edges a and b are associated with the same shortest cycle: Cycle 2. Using Gasteiger's heuristic, the edge a is associated with Cycle 1.

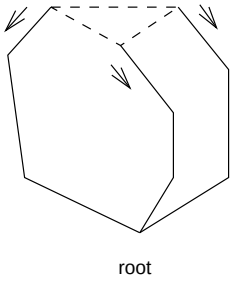


FIGURE 5. A counter-example for Gasteiger's heuristic.

2.2.3. *A polynomial algorithm to find an SSSR.* The weakness of the previous algorithms is due to the use of a unique spanning tree of the graph. Therefore, the solution consists in searching as many spanning tree as there are vertices in the graph. This idea has been introduced by [**Sor85**]. But he did not prove his algorithm or study its complexity. So, we will focus on the method presented by [**Hor87**]. It is considered as the first polynomial time algorithm that finds a minimal cycle basis.

Horton's algorithm is based on the following theorem :

THEOREM 2.1. *Let x be any vertex of any cycle C in a minimum cycle basis. There is an edge (y,z) in C such that C consists of a shortest path from x to y, a shortest path from x to z and the edge (y,z).*

Note that any cycle satisfying this property does not necessarily belong to a minimum basis. Therefore, Horton's algorithm extracts a cycle basis from an initial set of cycles (denoted by $\mathcal{C}_I$) verifying theorem 2.1:

1) $\forall a, b \in V$ find a shortest path $P(a,b)$ between a and b.
2) For all $v \in V$ do :
   For all $(x,y) \in E$ do :
   If $P(v,x) \cap P(v,y) = \{v\}$
   Then add in $\mathcal{C}_I$ the cycle $C = P(v,x) + P(v,y) + (x,y)$
3) Order by length the initial set of cycles $\mathcal{C}_I$
4) Use a greedy algorithm to extract a minimum cycle basis from $\mathcal{C}_I$.

First step of the algorithm chooses one path for every pair of vertices. Horton has proved that any choice can lead to an initial set $\mathcal{C}_I$ including a minimal cycle basis.

This proof consists in replacing the cycles of any minimum basis by cycles belonging to $\mathcal{C}_I$. In the second stage, a cycle C can be created as many times as it contains vertices. To avoid this duplication, Horton uses the method suggested by [**Tie70**]. Considering an order π on vertices, all cycles generated from a vertex v must only contain vertices that precede v in order π.

The two first stages of the algorithm can be implemented by the unique following one. From any vertex r in X, we perform a breadth first search through the graph. This search generate a spanning tree which includes the edges used to reach a vertex for the first time. For any vertex y, the subpath of this tree that links y to r is necessarily a shortest path. During the breadth first search, when a cycle closing edge (x, y) in found, a cycle is created if the paths linking x and y to r are disjoint.

Last step of the algorithm processes $\mathcal{C}_I$ in order of the length of the cycles. A new cycle is added to the k cycles of the current partial basis when the new induced set is independent. This test uses a gaussian elimination on the $m \times (k+1)$ boolean matrix corresponding to the set of cycle. This elimination can be performed in $O(m \times (k+1))$. Since $k \leq \nu$ and $\mathcal{C}_I$ contains at most νn cycles [7], the minimum cycle basis can be found in $O(m\nu^2 n)$ operations.

A quite similar method as Horton's one has been introduced by [**LDP90**]. Their algorithm can be summarized as follows: *from each cycle closure edge, search the shortest cycles of which it is a member. If all these cycles do not constitute a linearly independent set, find the next smallest cycles for each cycle closure edge. The process goes on until a minimum basis can be extracted from the current set of cycles.* The main difference between this method and Horton's algorithm is that Horton directly finds an unique and useful initial set of cycles. On the opposite, the method presented by Leach finds a sequence of ever growing sets whose biggest one is generally larger than the unique set used by Horton. Hence, Horton's algorithm is generally more efficient than the method presented by [**LDP90**].

3. Extended bases : Chemically relevant cycles

Although minimum cycles basis provides the smallest representation for the cyclic structure of the molecule, this representation is rarely the better one for chemical investigations. The problem lies in the following dilemma: on one hand, a cycle basis provides a too small set of cycles to perform a satisfactory analysis; on the other hand, the enumeration of all cycles does not allow evaluation in a reasonable time. Hence, an optimal set of cycles may be defined between these two extremes. Of course, such a definition depends on the kind of analysis performed.

3.0.1. *Including cycles of a given length.* On a synthetic analysis point of view, [**CP72**] suggests to include all rings containing 6 or fewer vertices. In the same way, [**WD75**] adds to the basis all cycles including at most 8 vertices (note that many works in graph theory deals with algorithms which enumerate all the cycles of a given length [**CNAO85, Ric86**]). The choice of the better upper bound for cycle selection is quite subjective and depends on many factors. We can say that the number of irrelevant cycles added to the set grows with the value of the upper bound.

[7] In step 2, a cycle is added to $\mathcal{C}_I$ when it consists of two distinct shortest paths. Hence, the number of cycles created from a vertex v is at most equal to the number of cycle closure edges.

3.0.2. *ESER.* An other extended basis has been defined by [**Fuj87**] as the *Essential Set of Essential Rings*. This definition is based on the notion of "tied" cycles. A tied cycle is a cycle which have a *chord*[8]. The algorithm used to build ESER starts by searching all the cycles of the molecular graph. Then, it extracts the set of "tied" and "multi-tied" cycles. A cycle C is *dependent* if it is the sum of tied cycles such that: **1.** each tied cycle is the same size or smaller than C **2.** the intersection of all these tied cycles involves no less than half of the edges of C **3.** each tied cycle includes no more *heteroatoms* (noncarbon and nonhydrogen atoms) than C.

This notion of ESER gives good results for a few examples of molecules. But it presents two inconvenients : firstly, ESER should includes a great number of cycles which are not chemically relevant (particularly when the molecule has no tied cycle); secondly, the algorithm that compute ESER is not efficient because it searches all the cycles and uses a high complexity criterion to detect dependent cycles.

3.0.3. *ESSR.* The extended SSSR introduced by [**DGHL89b**] is based on faces of a planar representation of the molecular graph. It defines the *Extended Set of Smallest Rings*. This set first includes all the *simple faces* corresponding to cycles without chords which are faces of a planar embedment of the graph. But some relevant cycles are not simple faces (see figure 3). Therefore, ESSR also includes some *cut faces* which corresponds to the other cycles without chords. A cut face C is named *primary* if it includes a least one edge e such that the simple face that contains e is the same size or smaller than C. So, ESSR includes all the simple faces and primary cut faces.

ESSR notion provides a complex but original definition of cycle relevance. Unfortunately, the algorithm that build this set also has a great time complexity. But the main weakness lies in the use of planar embedment. Although most of the simple or primary cut faces are relevant, a large part of them have no real interest. This is due to the fact that a planar embedment of the molecular graph is not a real projection of the molecule. Faces of an embedment are not equivalent to the real faces of the 3-dimensional structure of the molecule. That is the reason why using planar embedments of the molecular graph does not lead up to satisfactory definition of cycle relevance.

3.0.4. *SER.* The last extended SSSR we will study has been presented by [**Tak94**]. He defines the *Set of Elementary Rings* (where the word "elementary" does not correspond to the graph theory notion that we have mentioned in section 1.1). To build this set of cycles, the algorithm starts including all the cycles of a SSSR in SER. Then, for every pair of cycles C_1 and C_2 in the current SER, a new cycle $C_1 + C_2$ is added if C_1 and C_2 share at least 2 contiguous edges[9]. This process goes on until SER is stable.

According to its definition, the SER may include a lot of cycles a few of which are really relevant. For instance, figure 6 shows two unrelevant cycles added to the SER of the molecule presented in figure 1.

---

[8] a chord is an edge that links two vertices of a cycle and that does not belong to this cycle.

[9] more precisely, Takahashi uses the notion of θ-graph. It designates a connected graph such that every vertex has 2 adjacent edges, except for two unadjacent vertices whose valence is 3. Then, cycle $C_1 + C_2$ is added to SER if the subgraph induced by $C_1 \cup C_2$ is a θ-graph.

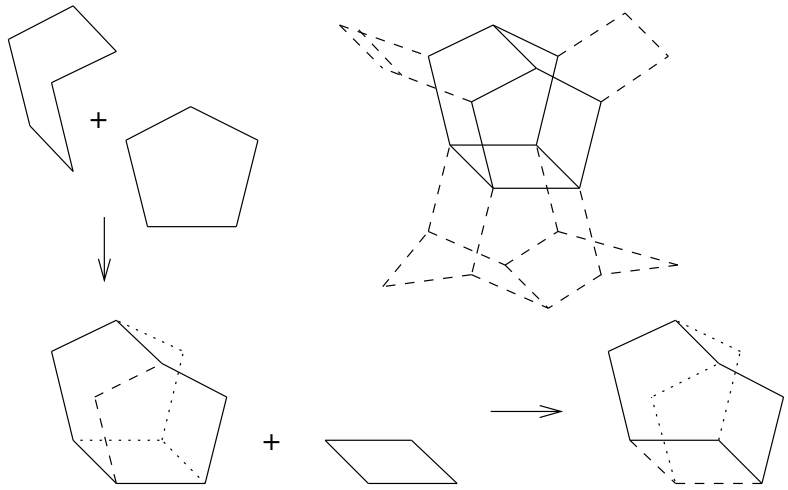


FIGURE 6. Some examples of cycle added to the SER corresponding to molecule in figure 1.

4. Set of Relevant Cycles

4.1. Definition.

4.1.1. *What is cycle relevance ?* Study of the previous definition of extended bases leads us to that obvious conclusion: whatever criterion is used, no satisfactory definition of cycle relevance has yet been found. Of course, all these works are interesting since they propose a formal definition of an abstract notion. According to particular applications, they should provide a better set of cycle than the minimum cycle basis one. But the problem of finding an universal definition of cycle relevance is still open. And it would probably never be solved.

One reason of this failure lies on the incompatibility of relevance notions coming from different aspects of chemistry. A given cycle may be relevant according to a particular point of view, but absolutely insignificant for an other one.

The other difficulty, and probably the most important, concerns the use of graphs to represent molecules. A molecular graph is not a molecule but an useful abstraction of very complicated 3-dimensional object. Trying to find, in a molecular graph, properties that depend on distance between atoms or spatial arrangement is doomed to failure. To be optimal, definition of cycle relevance probably needs to go beyond the scope of graph theory. It would require using spatial coordinates of atoms, atom interactions, chemical properties and so on. But such a definition of chemical relevant cycles would be very complex.

When we started the RESYN project, we had to choose a definition of relevant cycles. Because we are developing a system for computer-aided synthesis, this definition must satisfy some constrains. First, the definition of relevant cycles must be canonical, that is independant from the way it is computed. Then, it must be simple in order to be easy to understand by the users of the program. Finally, we need an efficient algorithm to compute the set of relevant cycles.

4.1.2. *A canonical definition.* In the scope of graph theory, notion of minimum cycle basis is absolutely unsatisfactory since different SSSR can be associated to the

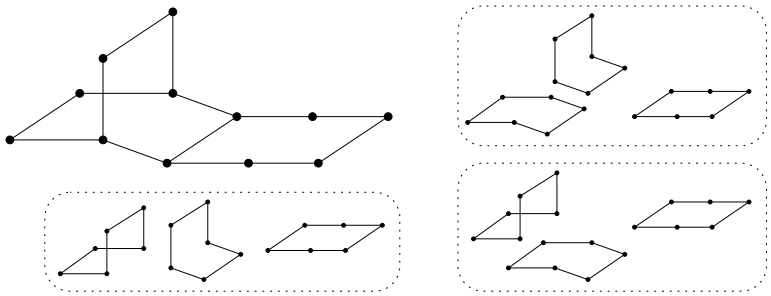


FIGURE 7. An example of a graph having several minimum bases.

same molecule. For example, the graph presented in Figure 7 has three different minimum bases.

Instead of trying to elaborate an other definition of relevance, we have tried to find the simplest canonical definition which is closed to the SSSR notion. Unifying previous definitions of extended bases leads us to that observation: *a cycle which is not the sum of smaller cycles is chemically relevant*. This definition has been suggested by Plokin [**Plo71**]. Hence, we will characterize *relevant cycles* according to that definition which can be formalized as follows (where $\mathcal{C}$ denotes the set of elementary cycles):

DEFINITION 4.1. $C \in \mathcal{C}$ is *relevant* **iff** $\nexists C_1, ..., C_k \in \mathcal{C}$ *s.t.* $\begin{cases} \forall i,\ |C_i| < |C| \\ and \\ C = C_1 + ... + C_k \end{cases}$

This characterization defines the *Set of Relevant Cycles* we denote by $\mathcal{C}_R$. This definition is very close to minimum basis notion. For example, in figure 7, $\mathcal{C}_R$ contains all the 6-edges cycles whereas each SSSR only includes three of them.

In fact, we should easily prove that:

LEMMA 4.2. *$\mathcal{C}_R$ is equal to the union of all the minimum cycle bases of G.*

Hence, a cycle is *relevant* if it belongs to at least one SSSR. Note that $\mathcal{C}_R$ is unique for a given molecule. This set is generally quite small for a molecular graph. For instance, in figure 1, $\mathcal{C}_R$ contains 15 cycles whereas any SSSR would include 11 of them.

On a graph theory point of view, the number of *relevant* cycles is not necessarily polynomial. If G is a complete graph [10], the size of $\mathcal{C}_R$ is equal to the polynomial number of 3-edges cycles. But there exists graphs, as the one of figure 8, which contain an exponential number of *relevant* cycles.

The existence of such graphs must not hide the interest of this definition. Note that the same problem occurs with any canonical definition based on SSSR. In organic chemistry, graphs with an exponential number of *relevant* cycles are hardly possible.

Till now, no algorithm has been proposed to compute $\mathcal{C}_R$. In this paper, we present a polynomial time algorithm to compute $\mathcal{C}_R$ as a polynomial number of parts. Each part corresponds to a *family* of cycles which can be directly listed from a particular relevant cycle. This cycle defines a *prototype* of its family. This compact

[10] in a *complete* graph, any vertex is adjacent to all the other ones.

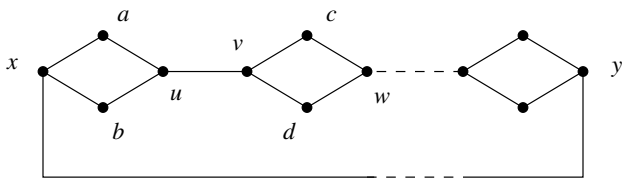


FIGURE 8. *According to the definition of relevant cycles, set* $\mathcal{C}_R$ *contains* $2^{\frac{n}{4}}$ *cycles in this graph* ($(x, b, u, v, d, w, ..., y, ..., x)$, $(x, a, u, v, d, w, ..., y, ..., x)$, *etc).*

representation of $\mathcal{C}_R$ gives a more precise description of the cycle vector space than a simple minimum basis. For instance, in the graph of Figure 8, apart from the $\frac{n}{4}$ 4-edges cycles that must be known, the other *relevant* cycles can be described by a single family. The prototype of this family is one of the $2^{\frac{n}{4}}$ elementary cycles with $\frac{3n}{4}$ vertices. Listing all these *relevant* cycles is not very meaningful for such a graph. But it can be very useful for a computer system to be able to perceive them and designate a single prototype.

For molecular graphs, the compact representation can be replaced by the complete enumeration of the relevant cycles.

The method we present to find the Set of Relevant Cycles is analogous to Horton's algorithm. We start by searching for an initial set of cycles (denotes by $\mathcal{C}'_I$) from which a subset of relevant prototypes is extracted. Then, all the others relevant cycles are directly generated from the set of relevant prototypes.

4.2. Partition of the set of relevant cycles. This section defines a partition of the set of relevant cycles such that the number of parts is polynomial.

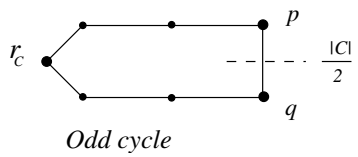
LEMMA 4.3. *If* μ *is a subpath of a relevant cycle* C *such that* $|\mu| \leq \frac{1}{2}|C|$ *then* μ *is a shortest path.*

PROOF. Assume that μ is not a shortest path. Then μ includes a subpath $\rho = (a \dots b)$ such that $|\rho| > d(a, b)$ and there is a shortest path ρ' from a to b such that ρ and ρ' are disjoint. Replacing ρ with ρ' in C will create a new cycle C' such that $|C'| < |C|$. Define C'' to be the cycle that consists of ρ and ρ'. Since $|\rho'| < |\rho| \leq \frac{1}{2}|C|$ we have $|C''| < |C|$. Because $C = C' + C''$, C cannot be relevant. □

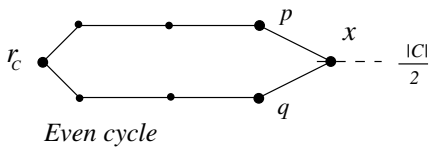
In the rest of this paper, we assume that the vertices are ordered by a numbering π that respects their degrees, i.e. $\forall x, y \in V,\ \pi(x) < \pi(y) \Rightarrow \deg(x) \leq \deg(y)$.

For any cycle C being considered, we denote by r_C the highest vertex in C according to the numbering π.

LEMMA 4.4. *any relevant cycle* C *consists of two disjoint shortest paths* $(r_C \dots p)$ *and* $(r_C \dots q)$ *having the same length and linked by the edge* (p, q) *if* $|C|$ *is* odd *or by the path* (p, x, q) *if* $|C|$ *is* even.



Odd cycle



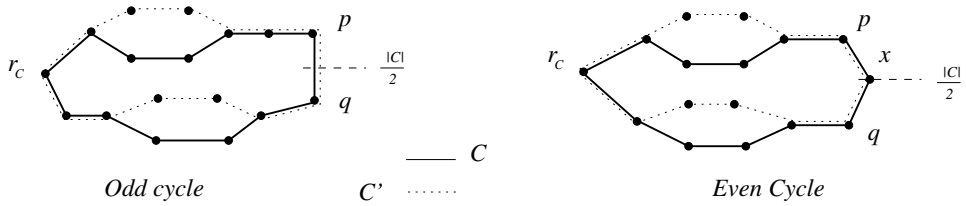
Even cycle

PROOF. For a given cycle C, we can define $(r_C \dots p)$ and $(r_C \dots q)$ to be the two longest subpaths of C whose length is strictly lower than $\frac{1}{2}|C|$. By lemma 4.3, these paths are shortest paths in the whole graph. □

The partition of the set of relevant cycles is based on lemma 4.4.

DEFINITION 4.5. the *cycle family* associated with a relevant cycle C, including the vertices r_C, p, q and eventually x as defined in lemma 4.4, is:

$$\mathcal{F}(C) = \left\{ C' \in \mathcal{C}_R \;\middle|\; \begin{array}{l} |C'| = |C| \text{ and } C' \text{ consists of :} \\ \bullet \text{ the vertex } r_C \text{ and the edge } (p,q) \text{ (or the path } (p,x,q)), \\ \bullet \text{ two shortest paths } (r_C,\dots,p) \text{ and } (r_C,\dots,q) \text{ passing only} \\ \text{through vertices lower than } r. \end{array} \right\}$$



Hence, two cycles C and C' belonging to $\mathcal{F}(C)$ only differ in the shortest paths from r_C to p and from r_C to q they include.

THEOREM 4.6. *The set of all the relevant cycle families defines a partition of* $\mathcal{C}_R$.

PROOF. By lemma 4.4 and definition 4.5, the cycle family associated with a relevant cycle C is unique for a given ordering π. So any relevant cycle belongs to exactly one family. Then, we define an equivalence relation such that two relevant cycles are equivalent if they belong to the same cycle family □

To describe the family $\mathcal{F}(C)$, we need a single cycle C which is a *prototype* of this family. All the other cycles in $\mathcal{F}(C)$ can be generated from C by replacing paths $(r_C, \dots p)$ and $(r_C, \dots q)$ with paths of the same length passing only through vertices smaller than r_C.

Hence, a cycle family is properly defined by a triple r_C, p, q for odd cycles or a quadruple r_C, p, q, x for even cycles.

The number of relevant cycle families depends on the order π used to define the families. However, we have the following result:

THEOREM 4.7. *the number of relevant cycle families is always polynomial.*

PROOF. The number of families whose prototype is an odd cycle is smaller than $n\,\nu$, since an odd cycle prototype is defined by a vertex r_C and a cycle closing edge (p,q). As for even cycle families, their prototype is defined by 4 vertices r_C, x, p and q such that p and q are adjacent to x. Hence, the number of even cycle families is smaller than:
$\sum_{r\in V} \sum_{x\in V,\, \pi(x)\le\pi(r)} \frac{\deg(x)^2}{2} \;\le \sum_{r\in V} \left(\deg(r) \times \sum_{x\in V,\, \pi(x)\le\pi(r)} \frac{\deg(x)}{2}\right) \;\le 2m^2$,
since order π respects vertex degrees ($\forall x \in V,\ \pi(x) \le \pi(v) \Rightarrow \deg(x) \le \deg(r)$) □

To compute the union of all the minimum cycle bases, we may find one prototype for each relevant cycle family.

4.3. A polynomial algorithm that finds the Set of Relevant Cycles. The algorithm we propose to compute cycle prototypes is based on the converse of lemma 4.4. This converse is not necessarily true but it gives a strong condition on cycle relevance.

The outline of the algorithm can be given as follows:

- firstly, we compute the set $\mathcal{C}'_I$ including one cycle for each triple r_C, p, q (or quadruple r_C, p, q, x) satisfying the condition of lemma 4.4,
- secondly, we use a greedy algorithm to extract relevant prototypes from $\mathcal{C}'_I$.

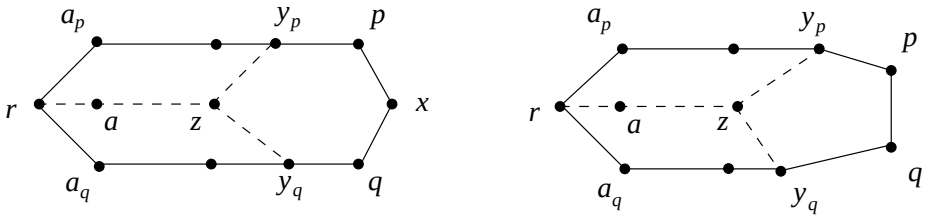
4.3.1. *Computation of $\mathcal{C}'_I$.* This set is generated using breadth first searches through the molecular graph G from every vertex r whose degree is higher than two[11].

Denote by V_r the set of vertices z such that $\pi(z) < \pi(r)$ and there is a shortest path in G from r to z that passes only through vertices which precede r in the ordering π.

DEFINITION 4.8. $\forall x \in V_r$, the set of *ancestors* of x according to r is defined by: $anc_r(x) = \{a \in \Gamma(r)$ *such that* $\exists$ *shortest path from* r *to* x *including* $a\}$

Now consider a cycle C that consists of the two shortest paths $(r, a_p, ..., p)$ and $(r, a_q, ..., q)$ linked by the edge (p, q) if $|C|$ is even or the path (p, x, q) if $|C|$ is odd.

LEMMA 4.9. *If* $anc_r(p) \cap anc_r(q) \neq \emptyset$ *Then* C *is not relevant*



PROOF. If there exists a vertex a such that $a \in anc_r(p) \cap anc_r(q)$, cycle C is the sum of three cycles[12] smaller than C. Note that the proof holds for degenerate cases when $a = a_p$ and/or $a = a_q$. □

Lemma 4.9 gives a strong condition to detect wrong cycles. For a given triple r_C, p, q (or quadruple r_C, p, q, x) we add a cycle in $\mathcal{C}'_I$ only if $anc_r(p) \cap anc_r(q) = \emptyset$.

To generate $\mathcal{C}'_I$, we also need to compute the folowing values, for any x in V_r:

- *fathers*$(x) = \{z \in \Gamma(x) \cap V_r$ *such that* $d(r, z) = d(r, x) - 1\}$
- *brothers*$(x) = \{z \in \Gamma(x) \cap V_r$ *such that* $d(r, z) = d(r, x)\}$
- *Path*(r, x) = a shortest path from r to x passing only through vertices in V_r

We can now consider the algorithm for the generation of $\mathcal{C}'_I$ as presented in Figure 9.

In the main While loop, lines 9-20 perform the breadth first search and compute the sets *fathers*(y), *brothers*(y) and $anc_r(y)$.

Note that we have the following property: *fathers*$(x) \neq \emptyset \Longrightarrow x \in V_r$.

[11] If $\nu > 1$, every cycle includes at least one vertex r such that $d(r) \leq 3$

[12] in the example: (r, a_p, y_p, z, a, r), (r, a_q, y_q, z, a, r) and (z, y_p, p, q, y_q, z)

[Algorithm 1]

1. For each $r \in V$ such that $d(r) \geq 3$ do :
2. | For each $x \in V$ do :
3. | | $level(x) \leftarrow +\infty$; $fathers(x) \leftarrow \emptyset$; $brothers(x) \leftarrow \emptyset$;
4. | **endfor**
5. | $fathers(r) \leftarrow \emptyset$; $anc_r(r) \leftarrow \emptyset$; $Path(r, r) \leftarrow (r)$;
6. | $FIFO \leftarrow \emptyset$; AddLast($r$, $FIFO$);
7. | $level(r) \leftarrow 0$; $explored \leftarrow \emptyset$;
8. | While $FIFO \neq \emptyset$ do :
9. | | $x \leftarrow$ ExtractFirst($FIFO$):
10. | | For each $y \in \Gamma(x)$ do :
11. | | | If $level(x) < level(y)$ Then :
12. | | | | If $\pi(y) < \pi(r)$ and ($fathers(x) \neq \emptyset$ or $x = r$) Then $fathers(y) \leftarrow fathers(y) \cup \{x\}$;
13. | | | | If $level(y) = \infty$ Then
14. | | | | | $level(y) \leftarrow level(x) + 1$; AddLast($y$, $FIFO$);
15. | | | | | If $x = r$ Then $anc_r(y) \leftarrow \{y\}$ Else $anc_r(y) \leftarrow anc_r(x)$;
16. | | | | Else $anc_r(y) \leftarrow anc_r(y) \cup anc_r(x)$;
17. | | | Else If $fathers(y) \neq \emptyset$ and $fathers(x) \neq \emptyset$ Then
18. | | | | $brothers(y) \leftarrow brothers(y) \cup \{x\}$
19. | | | **endif**
20. | | **endfor**
21. | | $\forall p, q \in fathers(x)$ such that $anc_r(p) \cap anc_r(q) = \emptyset$ and $(p, q) \notin E$ add even cycle "$Path(r, p) + x + Path^{-1}(r, q)$" to $\mathcal{C}_I^{'}$
22. | | If $fathers(x) \neq \emptyset$ Then $Path(r, x) \leftarrow Path(r, f) + x$, where $f \in fathers(x)$;
23. | | $\forall b \in brothers(x) \cap explored$ such that $anc_r(b) \cap anc_r(x) = \emptyset$ add odd cycle "$Path(r, x) + Path^{-1}(r, b)$" to $\mathcal{C}_I^{'}$
24. | | $explored \leftarrow explored \cup \{x\}$;
25. | **endwhile**
26. **endfor**

FIGURE 9. Algorithm that computes $\mathcal{C}_I^{'}$

Even and odd cycles are computed at line 21 and line 23 respectively, using a test on ancestors.

Line 22 chooses a shortest path form r to x, when $x \in V_r$.

LEMMA 4.10. *For molecular graphs, the set of cycles $\mathcal{C}_I^{'}$ can be computed with $O(m^2)$ operations and the cardinality of $\mathcal{C}_I^{'}$ is lower than $(2m^2 + \nu m)$.*

PROOF. Except for the computation of sets anc_r, the time complexity of lines 9-20 is the standard complexity of a breadth first search, that is $O(n + m)$ for each search from a vertex r.

Computation of $anc_r(y)$ only occurs for ring-closure edges whose extremities have different breadth levels (according to the breadth first search from r). For such edges, the modification is performed in $O(d(r))$. For molecular graphs, $d(r)$ is always bounded (by 5 or 6 in organic chemistry). So we can assume that each modification is performed in $O(1)$ if we use bit vectors to represent the sets anc_r.

Hence, time complexity of ancestor computation is $O(\nu n)$ for all the breadth first searches.

Line 21 is the critical step of the algorithm. The number of unordered pairs p, q checked by the program is lower than $\sum_{r \in V} \sum_{x \in V_r} \frac{1}{2} \deg(x)(\deg(x) - 1) < 2m^2$, since $\forall x \in V_r,\ \deg(r) > deg(x)$. If the sets anc_r are represented by bit vectors, each check is performed in O(1). So the complexity of Line 21 for all the breadth first searches is $O(m^2)$.

For the computation of odd cycles, the number of checks performed for each breadth first search is $O(\nu)$ since the number of "brothers" is lower than the number of ring-closure edges. So there are at most $O(\nu n)$ odd cycles.

Finally, the whole complexity is $O(m^2)$ for a molecular graph. □

LEMMA 4.11. *$\mathcal{C}_I^{'}$ includes one and only one prototype of each relevant cycle family.*

PROOF. Let $C = (r, \ldots, p, x, q, \ldots, r)$ be a relevant even cycle (if C is odd, the proof is quite similar). Assume that r is the highest vertex in C according to π and x is the unique vertex in C such that $d(r, x) = \frac{1}{2}|C|$.
Now consider the exploration of vertex x in algorithm 1. By lemma 4.4, $(r, \ldots, p)$ and $(r, \ldots, q)$ are shortest paths. So p and q belong to *fathers*(x). Define C' to be the cycle added to $\mathcal{C}_I'$ at line 21 for the pair of vertices p, q. By definition of the algorithm, C' is unique. Cycles C and C' only differ in the shortest paths $(r, \ldots, p)$ and $(r, \ldots, q)$ they include. The sum $C + C'$ defines a set of cycles which are smaller than C. Hence, cycle C' must be relevant since C is relevant. So, C' is the unique prototype of $\mathcal{F}(C)$ which belongs to $\mathcal{C}_I^{'}$ □

4.3.2. *Computation of a set of relevant prototypes.* By lemma 4.11, $\mathcal{C}_R \cap \mathcal{C}_I^{'}$ is a set of prototypes. So, we have to extract all the relevant cycles in $\mathcal{C}_I^{'}$.

As in Horton's algorithm, a minimum cycle basis $\mathcal{B}$ is calculated by processing the cycles of $\mathcal{C}_I^{'}$ in order of weight. During the processing of a cycle C, we denote by $\mathcal{B}_<$ and $\mathcal{B}_=$ the subsets of cycles in the current sub-basis whose lengths are less than $|C|$ and equal to $|C|$, respectively. By Definition 4.1, C is relevant when $\mathcal{B}_< \cup \{C\}$ is independent. When this check succeeds, cycle C is added to $\mathcal{B}_=$ if $\mathcal{B}_< \cup \{C\} \cup \mathcal{B}_=$ is also independent. The processing stops when $\mathcal{B}$ is complete and all the cycles having the same length as the greatest one in $\mathcal{B}$ have been tested.

The current minimal sub-basis of $\mathcal{B}$ is represented by a $(|\mathcal{B}_<|+|\mathcal{B}_=|)\times m$ boolean matrix. To test if $\mathcal{B}_< \cup \{C\}$ is independent, we perform a Gaussian elimination on the first rows of the matrix which are associated with $\mathcal{B}_<$. When this check succeeds, the Gaussian elimination follows on the other rows to test if $\mathcal{B}_< \cup \mathcal{B}_= \cup \{C\}$ is also independent.

Since $\mathcal{B}$ contains at most ν cycles, each check of a cycle is performed in $O(\nu m)$. By lemma 4.10, the cardinality of $\mathcal{C}_I'$ is in $O(m^2)$. So, the set of prototypes $\mathcal{C}_I' \cap \mathcal{C}_R$ can be generated in $O(\nu\, m^3)$ operations.

4.3.3. *Enumeration of $\mathcal{C}_R$.* $\mathcal{C}_R$ is completely described by a set of prototypes of the families defined in section 4.2. From this set of prototypes, the union of all the minimum cycle bases can easily be listed.

Consider a cycle prototype $C \in \mathcal{C}_R \cap \mathcal{C}_I^{'}$.To generate the cycle family $\mathcal{F}(C)$, we define a directed graph $D_r = (V_r, U_r)$ associated with the vertex r so that: $U_r = \{\ (y, z),\ \textit{where } z \in \textit{fathers}(y)\}$. Figure 10 gives an example of such a graph.

The computation of the digraph D_r can be directly performed in Algorithm 1.

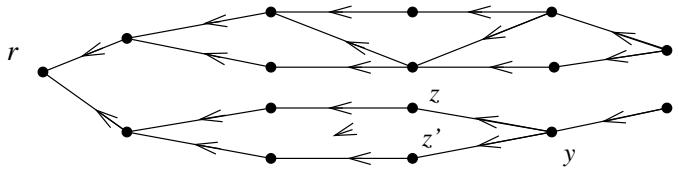


FIGURE 10. Digraph describing the shortests paths associated to the breadth first search from r

The digraph D_r has two major properties:

- It has no directed cycle,
- $\forall x \in V_r$, any directed path $(x, \ldots, r)$ in D_r corresponds to a shortest path in G.

To list all the paths from x to r in D_r, we use a backtracking function which is based on a depth first search from x. This recursive function can be summarized as follows:

```
Function List_Paths( x; current_path )
    add x at the head-end of current_path;
    If x = r Then Return({current_path})
    Else:
        Result ← ∅;
        For any z such that (x, z) ∈ U_r Do
            Result ← Result ∪ List_Paths(z, current_path);
        Return(Result)
    endif
```

To compute $\mathcal{F}(C)$, we first replace the path $(p, \ldots, r)$ in C by each path returned by the call List_Paths(p, $\emptyset$). Then, in any cycle generated this way, we replace the path $(q, \ldots, r)$ by each path resulting from List_Paths(q, $\emptyset$).

Each cycle in $\mathcal{F}(C)$ corresponds to a pair of paths $(p, \ldots, r)$, $(q, \ldots, r)$. Since the computation of each path takes a number of operations in the order of the path cardinality, the family $\mathcal{F}(C)$ can be listed in $O(\, n\, |\mathcal{F}(C)|\,)$.

So, the generation of $\mathcal{C}_R$ from subset $\mathcal{C}_R \cap \mathcal{C}_I^{'}$ is performed in $O(n\, |\mathcal{C}_R|)$ operations (including cycle enumeration).

Note that the use of prototypes substantially optimizes the generation of $\mathcal{C}_R$: if a cycle prototype is relevant, this implies that all cycles of the corresponding family are relevant. Therefore, the number of cycle relevance checks is polynomial since it is in $O(|\mathcal{C}_I^{'}|)$ instead of $O(|\mathcal{C}_R|)$.

Another interest of the prototypes lies in the possibility of computing the cardinality of $\mathcal{C}_R$ without enumerating all the cycles. This method is based on the digraph D_r from which we can easily evaluate the number of shortest paths from r to any vertex x, which passe only throught vertices in V_r. This method is discribed in [**Vis97**]. It also provides a way of calculating, in polynomial time, the number of relevant cycles of a given length which include a given vertex.

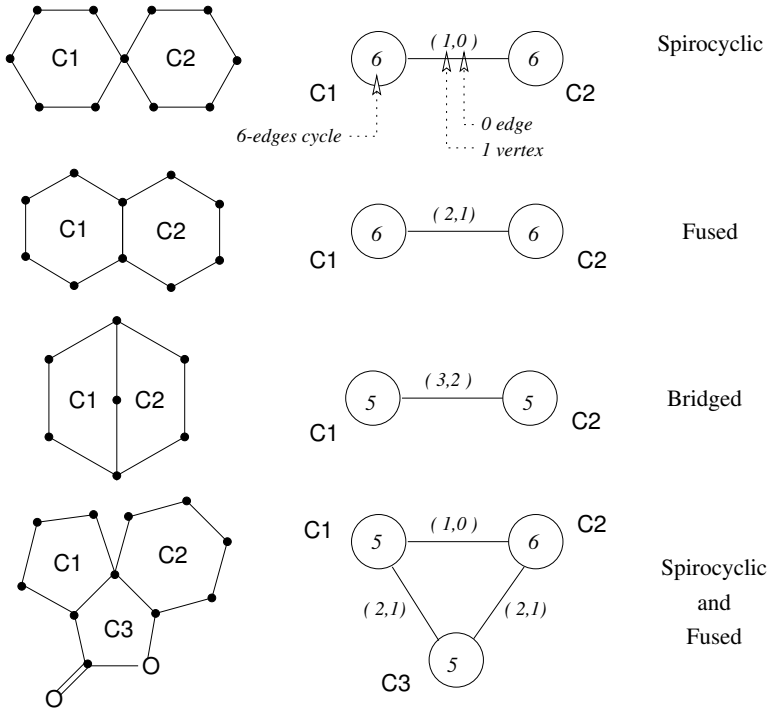


FIGURE 11. Graphs of the relevant cycles to describe cyclic systems

ABSTRACT REPRESENTATION OF A MOLECULAR GRAPH

This part presents an abstract representation of the molecular graph which describes the structural relationships between topological patterns as cycles, carbon chains,... We first define the "graph of relevant cycles" which aims at the description of the cyclic systems in the molecular graph. Then we give a definition of the major non-cyclic patterns as *carbon chains, cyclic links*, and *heteroatomic links*. Then we define the abstract representation.

5. Graph of the relevant cycles

To analyse the cyclic system of the molecule, we define a new graph in which each relevant cycle is associated to a vertex. This vertex is labeled by the size of the cycle. Two cycles are linked in the new graph if they share at least one atom. This link is labeled by a couple of values which correspond to the numbers of vertices and edges shared by the two cycles, respectively.

Figure 11 illustrates this *graph of the relevant cycles*.

With the label of the link joining two cycles, we can analyse the relationship between these cycles and characterize the corresponding cyclic system, according to the chemical nomenclature:

- a *spirocyclic* system corresponds to the label $(1, 0)$
- a *fused* system is labeled by $(2, 1)$
- other values represent *bridged* systems

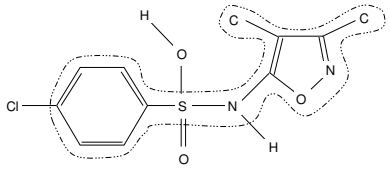


FIGURE 12. Skeleton of a molecular graph: terminal heteroatoms are recursivly removed

6. Non-cyclic topological patterns

The analysis of the non-cyclic parts of the molecule is quite easy to implement.

In the scope of organic synthesis, it is important to distinguish the *heteroatomic* parts from the *carbon* ones. The first step of the analysis consists in finding the "*skeleton*" of the molecule (Figure 12). The *skeleton* is obtained by recursively removing terminal heteroatoms (i.e. heteroatoms whose vertex degree is 1). Hence, the *skeleton* only includes heteroatoms which belong to a path linking two carbons. Note that the terminal heteroatoms are suppressed because they do not affect the topological analysis of a molecular graph. For a retrosynthesis analysis, the *skeleton* represents the main structural part of the target molecule on which topological strategies can be worked out.

The analysis of the non-cyclic parts is performed on the *skeleton* of the molecule. In order to define retrosynthetic topological stategies, the RESYN system detects the folowing components (see Figure 13):

- a *cyclic link* is a non-cyclic edge whose at least one extremity belongs to a cycle. The perception of *cyclic link* is used to detect the boundary between cyclic and non-cyclic parts. For instance, some retrosynthesis strategies consist in cutting *cyclic links* to isolate complex cyclic systems.
- an *heteroatomic link* is a maximal non-cyclic subgraph which consists of heteroatoms. The existence of *heteroatomic links* make easier the deconnection of the molecule.
- a *carbon chain* is a maximal non-cyclic subgraph which consists of carbons. A *carbon chain* is *linear* if all its vertex degrees are not greater than 2, otherwise the chain is *branched*.

The perception of these patterns can be performed in linear time using an algorithm based on a depth first search [**Vis95**].

7. Abstract representation

The abstract representation extends the graph of the relevant cycles. It also defines the relationship between the non-cyclic topological patterns. Figure 13 gives an example of abstract representation.

Abstract or reduced graph representations have been used for different purposes elsewhere. For example,[**LDP90**] describes such a representation used in automated conformational analysis.

Our abstract representation allows to define and describe strategic goals to be reached by the system. It can also be used to perform analogical reasoning. For instance, suppose we want to find a retrosynthetic pathway for the molecule of ibogamine described in Figure 14. Computing the abstract representation of

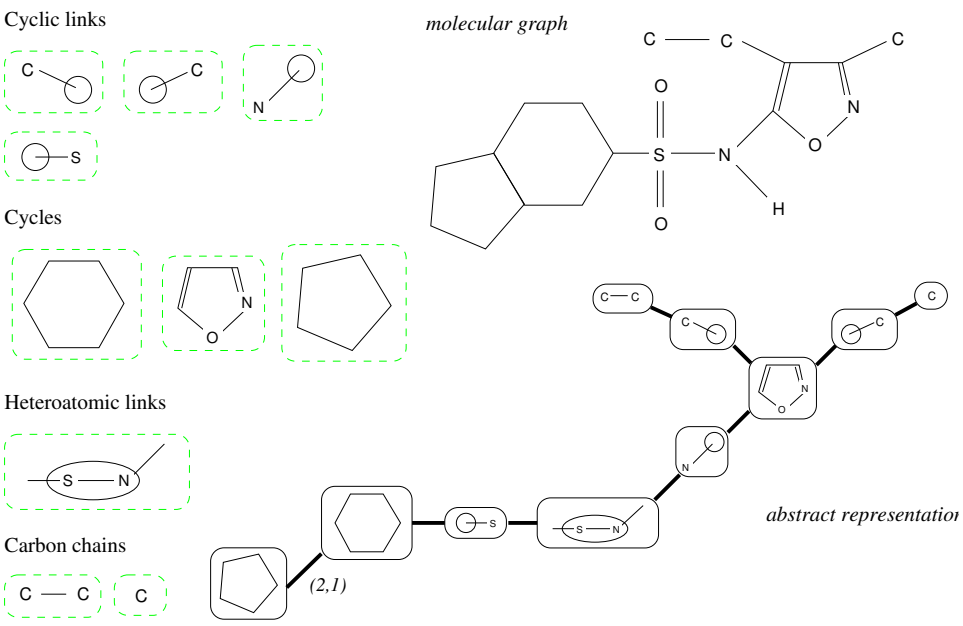


FIGURE 13. An example of abstract representation

ibogamine shows that it is very close to the representation of patchouli alcohol, whereas the molecular graphs are rather different. Then, if we now a retrosynthetic pathway for patchouli alcohol, based on the Diels-Adler reaction, we can try to use the same reaction in order to build the cyclic system of ibogamine. Without the abstract representation, the analogy would have been more difficult to find.

8. Conclusion

Perception of the target molecule is a main phase in solving an organic synthesys problem. From a structural formula, this process leads to a new representation which describes the molecule components and their relationships accordind to different viewpoints (topology, stereochemistry, functionality). This structured representation is used for defining strategies which will guide the retrosynthetic analysis of the target. To automate this process in a synthesis design system, it is necessary to precisily and formally define the chemical concepts used (cycles, functions,...) and to have efficient algorithms for recognizing them. The topological perception concerns the recognition of cycles, chains, links,... This perception is the subject of the work presented in this paper. We have particularly studied the problem of finding a set of relevant cycles.

We have seen, in the short review of papers dealing with SSSR, how difficult is the search of an efficient algorithm, even when the set of cycles to find is very well formalized. But minimum cycle basis do not provide a satisfactory notion to analyze the cyclic structure of a molecule. Hence, we have presented a few papers which define extended SSSR. None of them gives a perfect notion of cycle relevance. Since these definitions try to include all the relevant cycles, they also add, in their extended basis, cycles which have no real interest in chemistry. Naturally, the criterion used to select cycles may depend on the kind of analysis which is

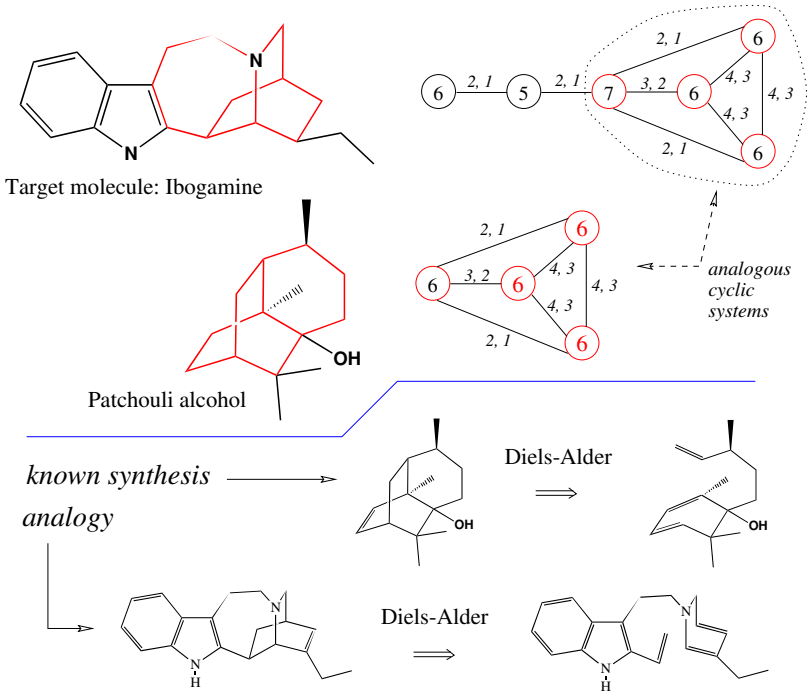


FIGURE 14. Using the abstract representation for analogy reasoning

performed. But cycle relevance should not be correctly defined in the scope of graph theory. It may resort to various other theories in chemistry which can explain why a given cycle is chemically relevant. Therefore, we chose the definition that corresponds to the smallest canonical set. According to this definition, the *Set of Relevant Cycles* (denoted by $\mathcal{C}_R$) is the *union of all the minimum cycles basis* of the graph. We have proposed a polynomial time algorithm that computes $\mathcal{C}_R$ in $O(\nu m^3)$ as a set of cycle families. From these families, all the relevant cycles can be generated in $O(n\,|\mathcal{C}_R|)$ (including output).

We have presented an abstract representation of molecular graphs for the definition of high level strategies allowing various kinds of reasoning (classification, analogy, induction, DAI,...). This work have been integrated in a prototype, the RESYN system, to experiment knowledge level modeling and reasoning through a constructive interaction with experts in chemistry.

References

[BSS91] L. Baumer, G. Sala, and G. Sello, *Ring perception in organic structures: A new algorithm for finding SSSR*, Computers Chem. **15** (1991), no. 4, 293–299.

[Cau13] A. L. Cauchy, *Recherche sur les polyèdres*, J. Ecole Polytechnique **9** (1813), no. 16, 68–86.

[CC89] E. J. Corey and X.-M. Cheng, *The logic of chemical synthesis*, Wiley-InterScience, New York, 1989.

[CLR85] E. J. Corey, K. A. Long, and S. D. Rubenstein, *Computer-assisted analysis in organic synthesis*, Science **228** (1985), 408–418.

[CNAO85] N. Chiba, T. Nishizeki, S. Abe, and T. Ozawa, *A linear algorithm for embedding planar graphs using PQ-trees*, Journal of computer and system sciences **30** (1985), 54–76.

[CP72] E. J. Corey and G. A. Petersson, *An algorithm for machine perception of synthetically signifiant rings in complex organic structures*, J. Am. Chem. Soc. **92** (1972), no. 2, 460–465.

[DFHL98] A. Dietz, C. Fiorio, M. Habib, and C. Laurenco, *Representation of stereochemistry using combinatorial maps*, this issue (P. Hansen et al, ed.), DIMACS workshop, AMS, 1998.

[DGHL89a] G. M. Downs, V. J. Gillet, J. D. Holliday, and M. F. Lynch, *Review of ring perception algorithms for chemical graphs*, J. Chem. Inf. Comput. Sci. **29** (1989), no. 3, 172–187.

[DGHL89b] G. M. Downs, V. J. Gillet, J. D. Holliday, and M. F. Lynch, *Theoretical aspects of ring perception and development of the Extended Set of Smallest Rings concept*, J. Chem. Inf. Comput. Sci. **29** (1989), no. 3, 187–206.

[DPK82] N. Deo, G. M. Prabhu, and M. S. Krishnamoorthy, *Algorithms for generating fundamental cycles in a graph*, ACM Trans. Math. Software **8** (1982), 26–42.

[Elk84] S. B. Elk, *Derivation of the principle of smallest set of smallest rings from Euler's polyhedron equation and a simplified technique for finding this set*, J. Chem. Inf. Comput. Sci. **24** (1984), no. 4, 203–206.

[Eul52] L. Euler, *Elementa doctrinae solidorum*, Novi. Comm. Acad. Sci. Imp. Petropol. **4** (1752), 109–140.

[FPDB93] B. T. Fan, A. Panaye, J-P. Doucet, and A. Barbu, *Ring perception. a new algorithm for directly finding the Smallest Set of Smallest Rings from a connection table*, J. Chem. Inf. Comput. Sci. **33** (1993), no. 5, 657–662.

[Frè39] M. Frèrejacque, *Condensation d'une molécule organique*, Bull. Soc. Chim. Fr. (1939), no. 6, 1008–1011.

[Fuj87] S. Fujita, *A new algorithm for selection of synthetically important rings : the Essential Set of Essential Rings for organic structures.*, J. Chem. Inf. Comput. Sci. **28** (1987), no. 2, 78–82.

[Gib69] N. Gibbs, *A cycle generation algorithm for finite undirected linear graphs*, J. ACM **16** (1969), 564–568.

[Gie98] O. Gien, *Modlisation des stratgies et des tactiques en synthse organique*, Thse de doctorat, Universit Montpellier II, 1998.

[GJ79] J. Gasteiger and C. Jochum, *An algorithm for the perception of synthetically important rings*, J. Chem. Inf. Comput. Sci. **19** (1979), no. 1, 43–48.

[HGT84] J. B. Hendrickson, D. L. Grier, and A. G. Toczko, *Condensed structure identification and ring perception*, J. Chem. Inf. Comput. Sci. **24** (1984), no. 4, 195–203.

[Hor87] J. D. Horton, *A polynomial-time algorithm to find the shortest cycle basis of a graph*, SIAM J. Comput. **16** (1987), no. 2, 358–366.

[Kir47] G. Kirchhoff, *Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird*, Poggendorf Ann. Physik **72** (1847), 497–508, English translation in *Trans. Inst. Radio Engrs.*, CT-5:4–7, 1958.

[Lau85] C. Laurenço, *Synthse organique assistée par ordinateur*, Thse de doctorat d'tat, Université Louis Pasteur (Strasbourg I), Septembre 1985.

[LDP90] A. R. Leach, D. P. Dolata, and K. Prout, *Automated conformational analysis and structure generation: Algorithms for molecular perception*, J. Chem. Inf. Comput. Sci. **30** (1990), no. 3, 316–324.

[Lie97] J. Lieber, *Raisonnement partir de cas et classification hirarchique. application la planification de synthses en chimie organique*, Thse de doctorat, Universit Nancy 1, octobre 1997.

[MD75] P. Mateti and N. Deo, *On algorithms for enumerating all circuits of a graph*, SIAM J. Comput. **5** (1975), 90–101.

[ON92] M.A. Ott and J.H. Noordik, *Computer tools for reaction retrieval and synthesis planning in organic chemistry. A brief review of their history, methods, and programs*, Recl. Trav. Chim., Pays-Bas **111** (1992), 239–246.

[Pat69] K. Paton, *An algorithm for finding a fundamental set of cycles of a graph*, Comm. ACM **12** (1969), 514–518.

[Plo71] M. Plotkin, *Mathematical basis of ring-finding algorithms in CIDS*, J. Chem. Doc. **11** (1971), 60–63.

[Py92] M. Py, *Un agent rationnel pour raisonner par analogie*, Thse de doctorat, Université Montpellier II, novembre 1992.

[RÝ5] J.-C. Régin, *Dveloppement d'outils algorithmiques pour l'Intelligence Artificielle et application la chimie organique.*, Thèse de doctorat, Universit Montpellier II, Dcembre 1995.

[Ric86] D. Richards, *Finding short cycles in planar graphs using separators*, Journal of Algorithms **7** (1986), 382–394.

[RT75] R. C. Read and R. E. Tarjan, *Bounds on backtrack algorithms for listing cycles, paths and spanning trees*, Networks **5** (1975), 237–252.

[She83] C. Shelley, *Heuristic approach for displaying chemical structures*, J. Chem. Inf. Comput. Sci. **23** (1983), 61–65.

[Sor85] E. Sorkau, *Ringerkennung in chemischen Strukturen mit dem Computer*, Wiss. Z. Tech. Hochsch. Leuna-Meuseburg **27** (1985), 765–770.

[Sys81] A. A. Syslo, *An efficient cycle vector space algorithm for listing all cycles of a planar graph*, SIAM Journal on Computing **10** (1981), 797–808.

[Tak94] Y. Takahashi, *Automatic extraction of ring substructures from a chemical structure*, J. Chem. Inf. Comput. Sci. **34** (1994), no. 1, 167–170.

[Tar72] R. E. Tarjan, *Depth-first search and linear graph algorithms*, SIAM J. Comput. **1** (1972), 146–160.

[Tie70] J. Tierman, *An efficient search algorithm to find the elementary circuits of a graph*, Comm. ACM **13** (1970), 722–726.

[Vis95] P. Vismara, *Reconnaissance et reprsentation d'lments structuraux pour la description d'objets complexes. application l'laboration de stratgies de synthse en chimie organique.*, Thse de doctorat, Universit Montpellier II, dcembre 1995.

[Vis97] P. Vismara, *Union of all the minimum cycle bases of a graph*, Electronic Journal of Combinatorics **4** (1997), no. R9, 15.

[WD75] W. T. Wipke and T. M. Dyott, *Use of ring assemblies in a ring perception algorithm*, J. Chem. Inf. Comput. Sci. **15** (1975), no. 3, 140–147.

[Wel66] J. Welch, *A mechanical analysis of the cyclic structure of undirected linear graphs*, J. ACM **13** (1966), 205–210.

[Zam76] A. Zamora, *An algorithm for finding the smallest set of smallest rings*, J. Chem. Inf. Comput. Sci. **16** (1976), 40–43.

Unité de Biométrie, ENSA-M, 2 place Pierre Viala, 34060 Montpellier Cedex 1
E-mail address: vismara@ensam.inra.fr

CCIPE, 141 rue de la Cardonille, 34094 Montpellier Cedex 5
E-mail address: cl@lirmm.fr