A New Algorithm for Exhaustive Ring Perception in a Molecular Graph

Th. Hanser,*,† Ph. Jauffret,‡ and G. Kaufmann

Laboratoire de Modèles Informatiques Appliqués à la Synthèse, URA 405 du CNRS, Université Louis Pasteur, 67000 Strasbourg, France

Received May 15, 1996⊗

A new fast and easy to implement algorithm for exhaustive ring perception is presented. This algorithm is based upon a progressive reduction (collapsing) of the path graph associated with the molecular graph studied. The path graph is an image of the molecular graph in which each vertex corresponds to a vertex of the molecular graph and each edge *a*-*b* describes an existing path between *a* and *b* in the molecular graph. During the reduction, nodes of the path graph are removed, and the information related to cycle occurrence is concentrated in the label of new edges between the remaining vertices. Each loop formed in the path graph during this collapsing process corresponds to a cycle in the molecular graph. Once the path graph has totally collapsed, all the rings in the molecular graph have been perceived.

INTRODUCTION

Ring perception has become a common feature in chemical computer systems. Evaluating the number and the kind of cycles in a chemical structure graph is required for many basic operations like

- structure classification
- compound naming
- aromaticity perception
- structure display optimization
- transforms descriptions

Although many algorithms have been developed to achieve ring perception in molecular graphs as described by G. M. Downs et al.,[1] cycle detection is still subject to further research.[2−3] Most algorithms aim to recognize only a subset of relevant cycles (e.g., chemically relevant rings) and fewer aim to perceive all the cycles in a graph. In both cases there are several methods to reach the goal. Some algorithms perceive directly the desired set of cycles; others derive it from a set of basic cycles (by composition) or from the set of all cycles (by selection). The most common sets of rings perceived in chemistry are known as SSSR (Smallest Set of Smallest Rings),[3−5] ESSR (Extended Set of Smallest Rings),[6] ESER (Essential Set of Essential Rings),[7] SER (Set of Elementary Rings).[2] The perception of all cycles in a graph is time consuming and most algorithms try to bypass this step if possible. However, for applications that require the perception of every cycle (e.g., display optimization,[8] computer generated chemical nomenclatures,[9] ESER perception based upon the set of all rings[7]) it can be more efficient to perceive all the cycles in one step rather than to derive them from a previous set of basic cycles. Several algorithms of the latter type have already been published and are still applied in computer chemistry.[8,10−11] In this paper we present a new algorithm, fast and easy to implement.

THEORY

The first step in our approach is to convert the molecular graph (**M-Graph**) into a path graph (**P-Graph**). This is done as follows: to each vertex in the **M-Graph** corresponds a vertex in the **P-Graph** that has the same label (this paper uses terminologies recommended by Downs et al.[1]). For each edge in the **M-Graph** there is an edge between the same vertices in the **P-Graph**. The label of an edge *a*-*b* in the **P-Graph** describes a possible path from *a* to *b* in the **M-Graph** and is defined by the set of edges in the **M-Graph** that are involved in the path. The initial label of the **P-Graph** edges thus corresponds to the associated **M-Graph** edges (Figure 1).

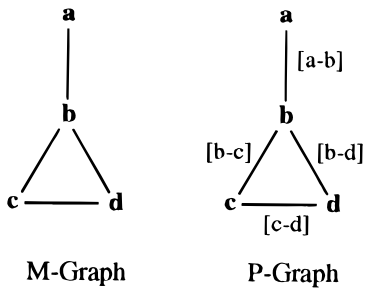

Figure 1. Molecular and path graph.

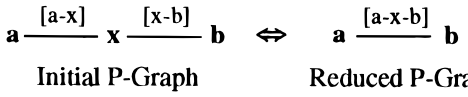

Figure 2. Path graph reduction.

The second step consists of reducing the **P-Graph**. The basic idea of this task is that a walk *a*-*x*-*b* in the **M-Graph** can be described in the **P-Graph** by a single edge between *a and b* labeled [a-x-b] (the notation [a-...-b] means a path from a to b). Accordingly, the vertex *x* can be removed from the initial **P-Graph** without losing any topological information (Figure 2).

Actually, to remove the vertex *x*, we need to delete the two edges *a*-*x* and *x*-*b* and to create a new edge *a*-*b*. The label of *a*-*b* is simply the concatenation of the labels of *a*-*x* and *x*-*b*.

To clarify the following, we introduce the notation p_{xy}, that defines an edge between vertices *x* and *y* in the **P-Graph** (i.e., describes a path from *x* to *y* in the **M-Graph)**. The reducing process can now be described with a single and general rule:

† E-mail: Thierry.Hanser@lmias6.u-strasbg.fr.
‡ E-mail: Philippe.Jauffret@lmias6.u-strasbg.fr.
⊗ Abstract published in *Advance ACS Abstracts,* August 15, 1996.

S0095-2338(96)00322-8 CCC: $12.00

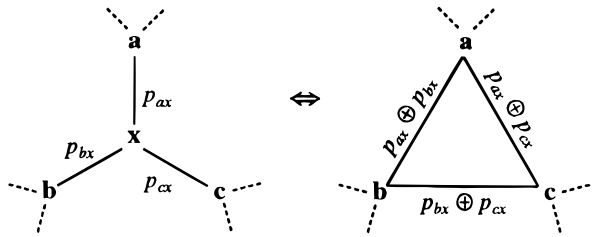

Figure 3. General reduction methodology.

To remove a vertex *x*:(1) create for each couple of edges p_{xy}, p_{xz} an edge $p_{yz} = p_{xy} \oplus p_{xz}$ where $\oplus$ is a concatenation operator; (2) remove the vertex *x* and all the edges involving *x*. Figure 3 shows an example of reduction.

This rule enables us to reduce the number of vertices in the **P-Graph** without losing any *connectivity* information. However, in order to detect only the cyclic elements of a graph, we discard edges that do not represent real paths (a path is a walk in which all vertices and edges are distinct). The previous rule can be refined so that only the couples of edges $p_{xy,\ x\neq y}$, $p_{xz,\ x\neq z}$, where $p_{xy} \otimes p_{xz} = \{x\}$ is true, lead to a new edge p_{yz} ($\otimes$ is an operator that extracts the set of common vertices between its operands). Let E and V be the set of edges and vertices in the **P-Graph**. The corresponding pseudo-code can be written as follows

REMOVE (x, V, E)

for each couple of paths $(p_{xy,\ x\neq y}, p_{xz,\ x\neq z}) \in E^2$

if $p_{xy} \otimes p_{xz} = \{x\}$ then

$p_{yz} \leftarrow p_{xy} \oplus p_{xz}$

$E \leftarrow E \cup \{p_{yz}\}$

for each path $p_{xy} \in E$

$E \leftarrow E - \{p_{xy}\}$

$V \leftarrow V - \{x\}$

This simple **REMOVE** function is used to perceive all the cycles in a **M-Graph**. The **P-Graph** can be totally reduced, removing one by one each vertex. This results into a general collapsing. During this process, the number of vertices decreases, and the number of edges may locally either increase or decrease depending upon the topology of the structure and the order in which vertices are removed. When a new edge between a vertex and itself (loop) is formed in the **P-Graph**, it corresponds to a cycle in the **M-Graph**. Indeed, the formation of a loop is equivalent to the closure of a path and thus the occurrence of a cycle. The label of the loop describes the detected cycle (Figure 4). When the **P-graph** is totally reduced ($V = \varnothing$, $E = \varnothing$), all the cycles have been detected.

The full algorithm for cycle perception follows exactly the two steps described above. First the molecular graph is converted in a path graph (here defined by V and E). Then a main loop takes care of removing one by one all the vertices of the **P-Graph**. The **CONVERT** ( ) function simply builds the sets V and E from the given **M-Graph**. The **REMOVE** () function is the same as above except that a special test has been included for loop detection. If a loop is found, the corresponding label (path) is added to the set of perceived rings (**RINGS**).

RINGS (M-GRAPH)

RINGS $\leftarrow \varnothing$

CONVERT(M-GRAPH, V, E)

while $V \neq \varnothing$

choose x in V

REMOVE (x, V, E, **RINGS**)

CONVERT (M-GRAPH, V, E)

$V \leftarrow \varnothing$, $E \leftarrow \varnothing$

for each vertex x in M-GRAPH

$V \leftarrow V \cup \{x\}$

for each edge *(x-y)* in M-GRAPH

$p_{xy} \leftarrow (x\text{-}y)$

$E \leftarrow E \cup \{p_{xy}\}$

REMOVE (x, V, E, **RINGS**)

for each couple of paths $(p_{xy,\ x\neq y}, p_{xz,\ x\neq z}) \in E^2$

if $p_{xy} \otimes p_{xz} = \{x\}$ then

$p_{yz} \leftarrow p_{xy} \oplus p_{xz}$

$E \leftarrow E \cup \{p_{yz}\}$

for each path $p_{xy} \in E$

if $x = y$ then **RINGS** $\leftarrow$ **RINGS** $\cup \{p_{xy}\}$

$E \leftarrow E - \{p_{xy}\}$

$V \leftarrow V - \{x\}$

Special notations:

$p_{xy} \otimes p_{xz}$: common vertices to the paths p_{xy} and p_{xz}

$p_{xy} \oplus p_{xz}$: concatenation of paths p_{xy} and p_{xz} (joined at vertex x)

REFINEMENT

The performance of the algorithm strongly depends upon the order in which vertices are removed from the **P-Graph**. Since it is better to begin to discard paths that might not lead to a cycle, we can assume that the appendage vertices can be directly removed (Figure 5). A more general rule

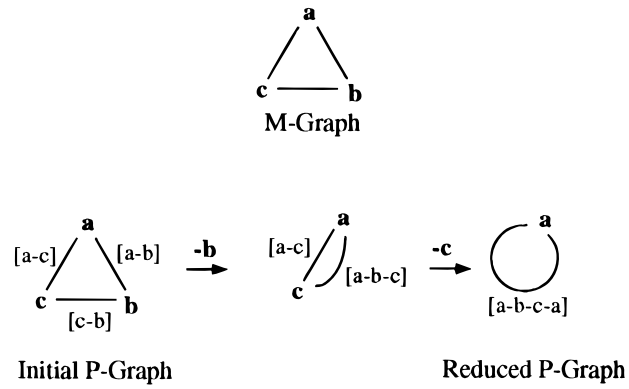

Figure 4. A loop centered on vertex "a" and corresponding to the cycle [a-b-c-a] appears if vertices "b" and "c" are removed.

would be to keep the connectivity of the remaining nodes in the **P-Graph** as low as possible; this would ensure a minimum number of cases where $p_{xy} \otimes p_{xz} \neq \{x\}$ is true in the algorithm described previously. A good selection criterion is therefore the connectivity of the vertices. Removing the vertices according to an increasing value of connectivity is probably the most efficient way to reduce the **P-Graph**. It is possible to select at each step the next vertex to be removed according to the new connectivity index induced by the previous step or to compute, for the initial **P-Graph** vertices, an extended connectivity index (based upon a Morgan like algorithm[12]) in order to sort the vertices and to guess a priori a good sequence. For the rest of this paper we used the first method.

Figure 5 shows a complete example of cycle perception. Note that there is always a reduced state of the **P-Graph** that corresponds to the Homeomorphically Reduced Graph *(HRG)* state, described by A. T. Balaban et al.[11] In the Collapsing **P-Graph** algorithm, the *HRG* is simply an intermediate state in a more general process.

COMPLEXITY

A general study of the algorithm presented in this paper would not be very meaningful for the limited subset of chemical graphs. Instead we propose beneath a statistical analysis of its performances.

Nevertheless, because the use of this algorithm is not restricted to chemical graphs, we demonstrate here its limit complexities; only the best and worst cases are studied. In both cases the complexity is independent from the order in which vertices are removed. The efforts required to perform the cycle perception are measured in terms of basic operations. The definition of a basic operation is as follows:

A basic operation occurs each time the content of the loop "for each couple of paths $(p_{xy,\ x\neq y},\ p_{xz,\ x\neq z}) \in E^2$ ") *is performed.*

Since the algorithm is strongly loop oriented, this definition gives a reliable measure of the **relative** complexity involved without taking into account any specific computer system or algorithm implementation.

The definitions for the complexity evaluation are as follows:

N = total number of vertices in the **M-Graph**

k = number of vertices removed

n_k = number of vertices in the **P-Graph***

l_k = number of edges between two vertices*

υ_k = number of distinct neighbors of a vertex*

a_k = number of edges starting from a vertex*

r_k = number of cycles detected*

b_k = number of basic operations performed*

* : *after k vertices have been removed*

B = total number of basic operations required

$$B = \sum_{k=1}^{N-1} b_k$$

Note that b_N is always equal to 0 since the only possible neighbor for the last node is itself (loop), thus for any edge x-y, $x = y$ and no basic operation will be performed when $k = N$. R = total number of cycles detected

$$R = \sum_{k=1}^{N-1} r_k$$

Best Case. The simplest instance of cycle perception is a graph with exactly one cycle of size N. In this case the connectivity of each vertex is constant and equal to 2 during the whole reducing process. For each vertex (except the last one) only one basic operation is required, the result is thus obvious:

$$B = N - 1 \quad R = 1$$

Worst Case. The most complex instance of a cycle perception (with no loop in the initial graph) is a clique, i.e., a graph where each vertex is connected to all other vertices as in Figure 6.

In the case of a clique, R and B can be evaluated in several steps.

$$\boldsymbol{n_k}: \quad n_k = n_0 - k$$

$$\boldsymbol{\upsilon_k}: \quad \upsilon_k = n_k - 1$$

$$\boldsymbol{a_k}: \quad a_k = \upsilon_k \times l_k$$

l_k. In order to walk from a vertex x to a vertex y using only one edge of the **P-Graph,** it is possible to use the initial

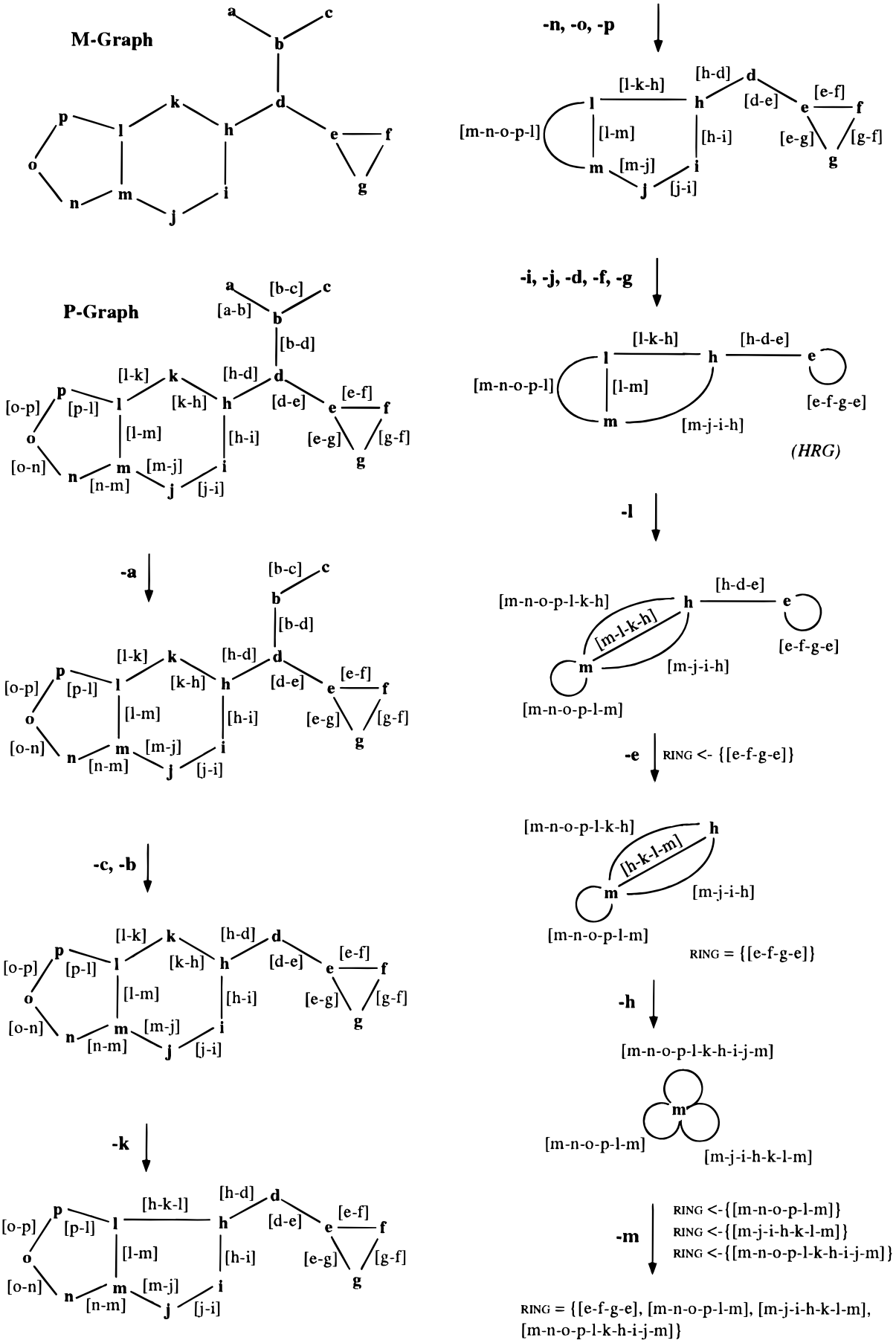

Figure 5. Example of perception using the collapsing **P-Graph** algorithm.

x-y edge (labeled [*x-y*]) or one of the edges describing an *x-(combination of the removed vertices)-y* path. For example, if $k = 3$ and *a*, *b*, *c* are the removed vertices, the different paths from *x* to *y* are

$i = 0$ [*x*-*y*]

$i = 1$ [*x*-*a*-*y*], [*x*-*b*-*y*], [*x*-*c*-*y*]

$i = 2$ [*x*-*a*-*b*-*y*], [*x*-*b*-*a*-*y*], [*x*-*a*-*c*-*y*], [*x*-*c*-*a*-*y*], [*x*-*b*-*c*-*y*], [*x*-*c*-*b*-*y*]

$i = 3$ [*x*-*a*-*b*-*c*-*y*], [*x*-*a*-*c*-*b*-*y*], [*x*-*b*-*a*-*c*-*y*], [*x*-*c*-*a*-*b*-*y*], [*x*-*c*-*b*-*a*-*y*], [*x*-*b*-*c*-*a*-*y*]

where i is number of intermediate vertices. This can be written as

$$l_3 = C_3^0 \times 0! + C_3^1 \times 1! + C_3^2 \times 2! + C_3^3 \times 3!$$

C_n^p = combination of p elements out of n equivalent to

$$l_3 = A_3^0 + A_3^1 + A_3^2 + A_3^3$$

A_n^p = arrangement of p elements out of n or in a more general way as

$$l_k = \sum_{i=0}^{k} A_k^i$$

***r*$_k$.** Determining r_k is achieved in the same way as for l_k, but in this case we are looking for paths from x to x. The first possible path between x and x (loop in P-graph) occurs for $k = 2$ (3 membered cycle). Furthermore the direction of those paths is not relevant. We can thus write

$$r_0 = r_1 = 0$$

$$r_k = \frac{1}{2}\sum_{i=2}^{k} A_k^i$$

$$R = \sum_{k=1}^{N-1} \frac{1}{2}\sum_{i=2}^{k} A_k^i$$

***b*$_k$.** In order to remove the kth vertex, we have to consider all the couples of edges starting from this vertex (see algorithm). For each couple there is one basic operation required. Thus

$$b_k = a_{k-1} \times (a_{k-1} - 1)$$

$$b_k = [\nu_{k-1} \times l_{k-1}] \times ([\nu_{k-1} \times l_{k-1}] - 1)$$

$$b_k = [\nu_{k-1} \times \sum_{i=0}^{k-1} A_{k-1}^i] \times ([\nu_{k-1} \times \sum_{i=0}^{k-1} A_{k-1}^i] - 1)$$

$$B = \sum_{k=1}^{N-1} [(N-k) \times \sum_{i=0}^{k-1} A_{k-1}^i] \times ([(N-k) \times \sum_{i=0}^{k-1} A_{k-1}^i] - 1)$$

Table 1 shows the R and B values for cliques of size $N = 3-8$. The results indicate that the number of basic operations required for the cycle perception in the case of a clique increases dramatically with the size of the clique. However this situation is extreme (the number of perceived cycles becomes very large) and occurs only with small sized cliques for molecular structure graphs.

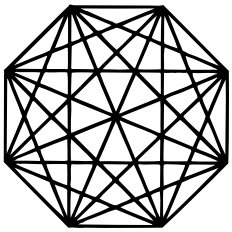

Figure 6. Example of clique, $N = 8$.

Table 1. Number of Cycles R and Pseudo-Complexity B for Cliques with $N = 3-8$

N	*R*	*B*	*N*	*R*	*B*
3	1	2	6	197	2719
4	7	19	7	1172	61047
5	37	186	8	8018	2136588

Table 2. Number of Basic Operations B Required According to the Permutation π of a Test Sequence

π	π_0	π_1	π_2	π_3	π_4	π_5	π_6	π_7	π_8
B	36	72	81	90	99	115	135	205	225

π	π_9	π_{10}	π_{11}	π_{12}	π_{13}	π_{14}	π_{15}	π_{16}
B	365	374	383	284	87	50	80	85

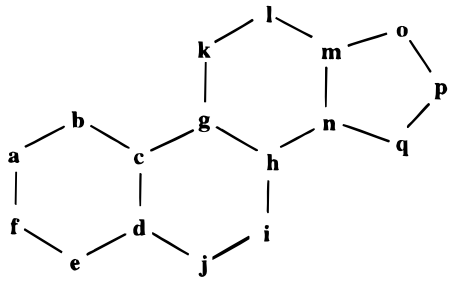

Figure 7. M-Graph corresponding to the results in Table 2.

RESULTS

Vertex Selection. The importance of the removing order of the vertices is studied. Table 2 shows the number of basic operations B required to perform a cycle perception on the graph of Figure 7. Each column corresponds to a different removing sequence. The first column (π_0) gives the number of basic operations required in case of an increasing connectivity selection mode (order π_0 = b, e, f, a, i, j, k, l, o, p, q, c, h, m, d, n, g). The next columns (π_{1-17}) correspond to successive cyclic permutations of the first sequence (π_1 = e, f, a, i, j,...g, b; π_2 = f, a, i, j, k,...,g, b, e; etc.). The molecular graph contains 17 vertices, 20 edges, and 10 condensed cycles.

The results show that a minimum number of basic operations is required when the vertices are removed according to an increasing value of connectivity. It is interesting to see how critical this ordering is. Indeed a factor up to 10 between the maximal (for π_{11}, starting with all vertices of connectivity index = 3) and minimal (for π_0, starting with all vertices of connectivity index = 2) value of B appears in this example.

Statistics. We undertook a statistical study on more than 34 000 cyclic structures randomly extracted from the Fine Chemical Directory (FCD) database. Figure 8 shows the

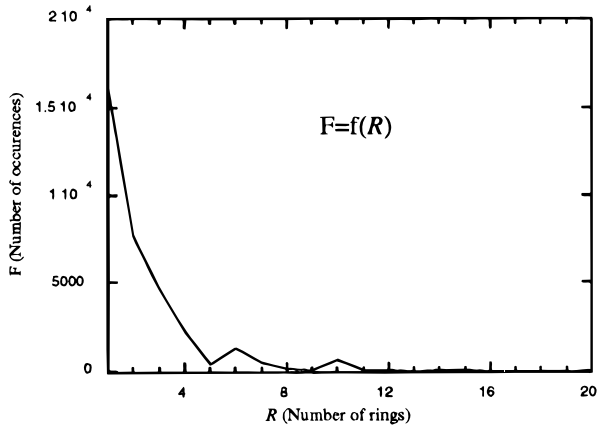

Figure 8. Distribution F of the structures according to their number of rings R.

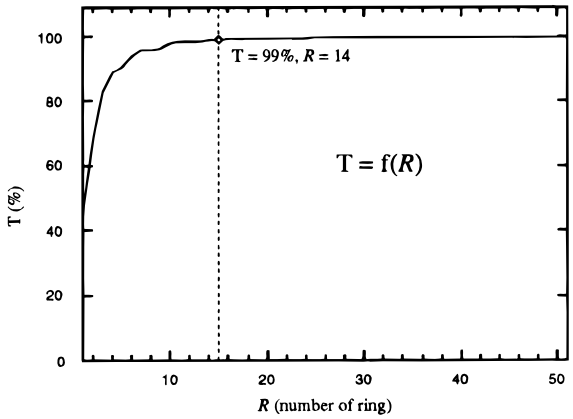

Figure 9. Distribution T in percent of the structures cumulated according to their number of rings R.

frequency F of structures containing a given number R of rings. This frequency decreases rapidly with the number of rings. In fact, 99% of the molecular graphs contain less than 15 rings as shown in Figure 9.

The number B of basic operations required for the ring perception according to the number R of rings in a structure is shown in Figure 10. According to these data we can conclude that for 99% of the structures, less than 110 basic operations are necessary. This amount can be transposed in very short perception times using any efficient implementation of the proposed algorithm.

DISCUSSION

The perception of all cycles in a graph is a time consuming task. It is important to optimize the speed of corresponding algorithms especially for systems applying cycle perception to a large number of graphs (e.g., Computer Assisted Synthesis). The algorithm presented in this paper has been designed to allow a speed effective implementation using sets as a main data structure. Although the results of the statistical analysis seem very interesting while dealing with mean values, it is important to note that the speed of a peculiar perception highly depends upon the type of structure studied, namely the "density" and relative position of the cycles (not only the number). Graphs involving a lot of condensed cycles may lead to vertices with a high connectivity index during the collapsing process and thus to lower performance. However, even for the complex structure shown in Figure 11, the number of basic operations required

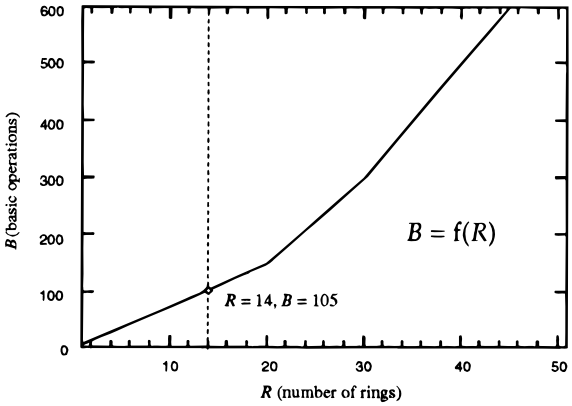

Figure 10. Number of basic operations B required for ring perception as function of the number of rings R in a structure.

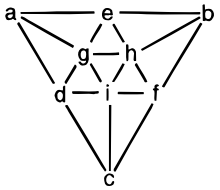

Figure 11. A complex graph containing 248 cycles.

Table 3. CPU Times for Some Examples

example	R	B	CPU (s)
Figure 5	3	11	0.0011
Figure 7	10	36	0.0014
Figure 11	248	4519	0.093
Figure 6	8018	2136588	1.5

remains fairly small ($B = 4519$) regarding the number of cycles perceived ($R = 248$).

The **Collapsing P-Graph** algorithm is very general and can be applied to any graph. Furthermore, there is no connexity requirement, each fragment in a nonconnex graph will be handled independently.

Originally designed for ring perception in molecular graphs, the same algorithm can also be used to enumerate all the paths between two vertices of a graph. Indeed these paths will be generated during the collapsing process when all but these two vertices have been removed. Each remaining edge corresponds to an existing path. Although this is probably not the most efficient way to enumerate paths between two vertices, it shows the potential of the *collapsing graph* methodology and allows us to consider new applications.

IMPLEMENTATION

The algorithm requires standard data structures; dynamic lists (for the edges of the **P-Graph**) and binary sets (for the labels of the edges) were used in our case. The exact implementation is not discussed here since the aim of this paper is to focus on the algorithm. Nevertheless it is important to note that main steps, in the algorithm, rely on set operations (intersections, unions,...) and thus support efficient implementations. The CPU times given in Table 3 correspond to some examples of this paper. The tests ran on a Hewlett-Packard 735 workstation, and the cycle perception code was written in C. The results indicate a fast perception though the data structures used can still be optimized to enhance performances.

CONCLUSION

The **Collapsing P-Graph** algorithm is a new approach to exhaustive cycle perception. The principle of this method is simple and can be implemented in a very efficient way. This algorithm is not specific to chemistry and can be applied to any graph. The concept of *collapsing graph* is a general and powerful tool that can be applied to other purposes such as path enumeration.

REFERENCES AND NOTES

(1) Downs, G. M.; Gillet, V. J.; Holliday, J. D.; Lynch, M. F. Review of Ring Perception Algorithms for Chemical Graphs. *J. Chem. Inf. Comput. Sci.* **1988**, *29*, 172−187.
(2) Takahashi, Y. Automatic Extraction of Ring Substructures from a Chemical Structure *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 167−170.
(3) Fan, B. T.; Panaye, A.; Doucet, J. P.; Bardu, A. Ring Perception: A New Algorithm for Directly Finding the Smallest Set of Smallest Rings from a Connection Table *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 657−662.
(4) Qian, C.; Fisanick, W.; Hartlzler, D. E.; Chapman, S. W. Enhanced Algorithm for Finding the Smallest Set of Smallest Rings. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 105−110.
(5) Gasteiger, J.; Jochum, C. An Algorithm for the Perception of Synthetically Important Rings. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 43−48.
(6) Downs, G. M.; Gillet, V. J.; Holliday, J. D.; Lynch, M. F. Theoretical Aspects of Ring Perception and Development of the Extended Set of Smallest Ring Concept. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 187−206.
(7) Fujita, S. A New Algorithm for Selection of Synthetically Important Rings. The Essential Set of Essential Rings for Organic Structures. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 78−82.
(8) Shelley, C. A. Heuristic Approach for Displaying Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 61−65.
(9) Kirby, G. H.; Polton, D. J. Systematic Chemical Nomenclatures in the Computer Age. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 560−563.
(10) (a) Corey, E. J.; Wipke, W. T.; Cramer, R. D.; Howe, W. J. Techniques for Perception by a Computer of Synthetically Significant Structural Features in Complex Molecules. *J. Am. Chem. Soc.* **1972**, *94*, 431−439. (b) Corey, E. J.; Petersson, G. A. An Algorithm for Machine Perception of Synthetically Significant Rings in Complex Cyclic Organic Structures. *J. Am. Chem. Soc.* **1972**, *94*, 460−465.
(11) Balaban, A. T.; Filip P.; Balaban, T. S. Computer Program for Finding All Possible Cycles in Graphs. *J. Comput. Chem.* **1985**, *6*, 316−329.
(12) Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures−A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107−113.

CI960322F