A New Algorithm to Classify all the Rings Set in Graphs

Cerruela García, G.; López Espinosa, E.; Luque Ruiz, I. and Gómez-Nieto, M.A

Department of Computing and Numerical Analysis. University of Córdoba.
Campus Universitario de Rabanales, Edificio C2, Planta-3. E-14071 Córdoba (SPAIN)
gcerruela@uco.es :: mangel@uco.es

Abstract. In this paper a new algorithm to detect and classify the graph rings set is presented. The solution is based in the cyclical conjunction operator proposed in a previous work, the rings were classify taken in to account the number of nodes that compose it. The application was validate apply a set test graphs of varying complexity.

1 Introduction

They are many real computational problems that can be solved thought analysis techniques to the graph that represent the information involved in its. The graph rings analysis has been extensively used in the solution of diverse scientific and technical problems, for example, to solve electric networks problems [1], in the algorithms analysis [2], etc.

In chemistry, graphs are used to represent the molecular structure of compounds. By analyzing such graphs, a huge amount of information can be obtained for use in areas such as compound and fragment identification, the calculation of topological descriptors, complexity and similarity, etc. The ring systems of the molecular graph are key features in determining its shape and properties.

Rings play an especially significant role in pharmacological activity, can serve to properly position and orient functional groups that interact with a receptor and can also help to reduce the unfavorable loss of conformational entropy upon receptor binding. The manipulation of ring structure has long been a fundamental strategy in designing analogues to optimize the potency and selectivity of a lead molecule [3]. In [4], the ring analysis was used to characterize the most common structural frameworks in a database of known compounds as a guide for future compounds discovery.

Many efforts have been invested to find efficient methods and algorithms for the detection of rings in a graph [5-14], yet despite this effort, not many of the proposed algorithms are suitable for use in all types of graphs and to permit classify the cycles set; in most cases the computational cost is high, especially as graph complexity rises.

In this paper we propose an algorithm to detect and classify the all graph rings set, taken in to account the number of nodes that compose each ring. Section 2 details the algorithm to detect and classify the graph rings set. In section 3 the experimental results were analyzed, and in section 4 the principal conclusion are summarized.

2. Detecting and classify the Rings set

The process of extracting and classify the graph rings set, is based in the successive application of the cyclical conjunction operation proposed in [15] to the set of known initial rings, the process being repeated until no new rings are obtained or until reaching the theoretical maximum number of rings in the graph ($2^{(\text{edges-nodes+1})}$-1) this limit is seldom reached. The process (Figure 1) begins by obtaining an initial set of rings (***ICL***), applying cyclical conjunction to all possible pairs in order to arrive at a new list of rings (***NCL***).

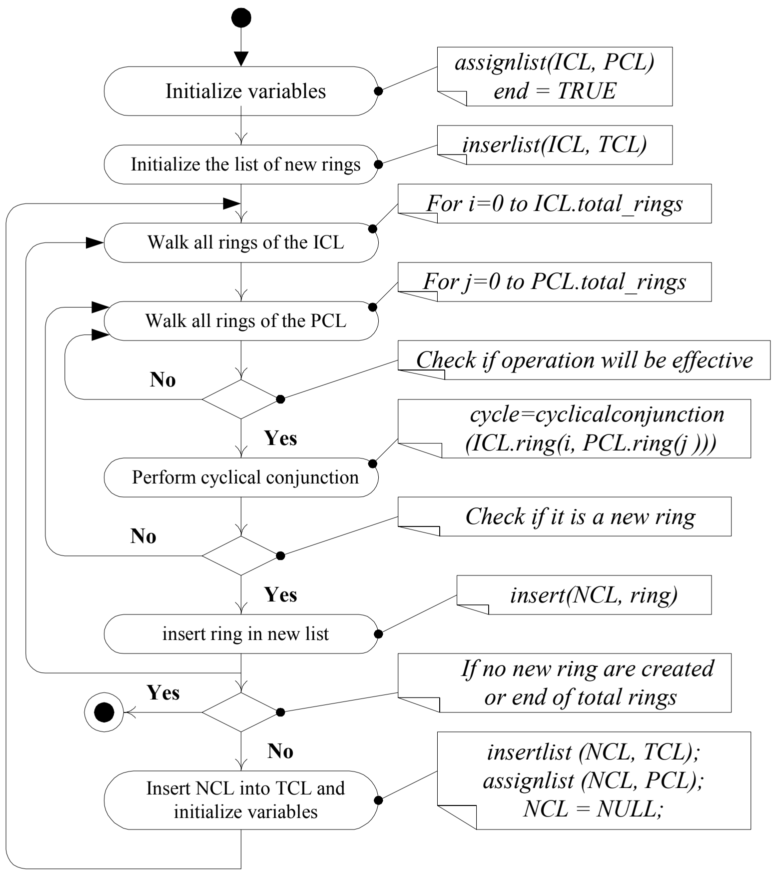

Fig. 1. Rings Detection Algorithm Diagram

The initial set of rings is extracted by an algorithm proposed in a previous work [16]. It is based on the adjacency matrix representing the graph ordered by node connectivity, and by means of a zigzag process a spanning tree is built, whose main feature is its shallowness but great breadth. Then, by choosing unvisited edges, a set of rings is obtained whose cardinality is equal to "*edges-nodes+1*". This algorithm avoids the use of path tracing and produces good results, with a computational cost of N number of nodes, at worst.

As Figure 1 show, once initial rings have been obtained, we can calculate all rings in the graph by applying cyclical conjunction to this set (***ICL***) and to the new sets of rings generated by this function. The process will terminate when all the required operations have been performed on the sets of known rings.

This algorithm works as follows:

1. The data structures used by the algorithm are initialized. Initial rings obtained are stored in the *Initial rings List* (***ICL***). The set of initial rings (***ICL***) is copied straight into the set of all the rings in the graph, represented by the *Total rings List* (***TCL***).
2. The set of initial rings is copied directly into the set of previous rings, represented by the *Previous rings List* (***PCL***). This list stores the rings generated during the prior step of the iteration.
3. Cyclical conjunction is performed on ring pairs from the ***ICL*** and ***PCL*** lists, for all pairs which satisfy the properties of this operator [15]. We test to see if the intersection of the rings might produce a new ring, and if this is not the case, the operation is rejected. This step determines the nodes in common between the two rings.
4. Rings are rearranged to be stored in the order established by the second stage of the algorithm. The cyclical conjunction operation is carried out on the rings, enacting the third stage of the algorithm.
5. A check is made to see if the new ring is present in the *New Rings List* (***NCL***), a list that stores the rings generated during each iterative pass of the algorithm. A check is made to see if the new ring is present in the total rings list (***TCL***). The ring is stored in the new rings list (***NCL***).
6. When the iteration step has concluded, the ***NCL*** is inserted into the ***TCL***, the ***PCL*** is initialized to the ***NCL*** and the ***NCL*** is erased, ready for the next pass of the iteration. The iterative process is halted when conjunction of the lists yields no further new rings, or when the ***TCL*** has a cardinality equal to $2^{(\text{edges-nodes+1})}-1$.

3. Results and Discussion

The proposed algorithm has been tested using a wide range of graphs. In Table 1 are detail the graph characteristics (number of nodes N, number of Edges E), the total number of Rings obtained, and the execution time. Figure 2 shows for the graph rings distribution taken into account the number of nodes in each ring detected.

In every case the proposed algorithm detect and classify, take into account the number of nodes in each ring, the total ring set in the problem graph. Computing time per ring applying the conjunction operation is not constant; this effect is mainly due to the computational cost of the second stage of the algorithm, where the determining factor is the number of nodes forming the rings involved in the cyclical conjunction algorithm.

Graph	N	E	Total Rings	Graph Rings Classification Number of Nodes							Time (Sec.)
				3	4	5	6	7	8	9	
GR-1	8	10	4	3	1	-	-	-	-	-	258×10^{-6}
GR-2	4	6	7	4	3	-	-	-	-	-	451×10^{-6}
GR-3	6	8	7	4	3	-	-	-	-	-	451×10^{-6}
GR-4	5	10	37	10	15	12	-	-	-	-	5.1×10^{-3}
GR-5	6	15	197	20	45	72	60	-	-	-	64.9×10^{-3}
GR-6	7	21	1172	35	105	252	420	360	-	-	1.01
GR-7	8	28	8018	56	210	672	1680	2880	2520	-	36.88
GR-8	9	30	9975	57	216	702	1800	3240	3240	720	56.92
GR-9	9	36	62814	84	378	1512	5040	12960	22680	20160	2856.5

Table 1. Graph caracteritics, rings and execution time

The properties of the cyclical conjunction operator allow substantial reduction of the number of cyclical conjunction operations required [15]. Although a few unnecessary cyclical conjunction operations might be carried out (those which fail to generate new rings), this algorithm offers good results.

While calculation time cannot be directly compared with that of other cited algorithms, since this depends on computing resources employed, it may be seen that graphs with fewer than 1000 rings take under one second to analyze (in the order of milliseconds for graphs of medium complexity, and microseconds for simple graphs), and only where graphs are highly complex, with say tens of thousands of rings, is computing time high. Many factors intervene here, such as the number of operations to be performed, the size of the data structures used to store the rings, and graph topology.

Implementation of the cyclical conjunction operator properties afforded a reduction of 60-90% of the number of operations required to extract all rings from a graph, greatly speeding up the algorithm, although this raised computational cost.

Tests showed calculation time for the extraction of graph rings is dependent on factors such as total number of rings and graph topology and complexity, in addition to the nature of the initial ring detected at the preprocessing phase, factors that are either unknown or excessively costly to determine a priori.

While computational cost is generally acceptable, it will clearly be high for complex graphs, suggesting the need for parallel or distributed processing techniques for a practical implementation of the proposed algorithm

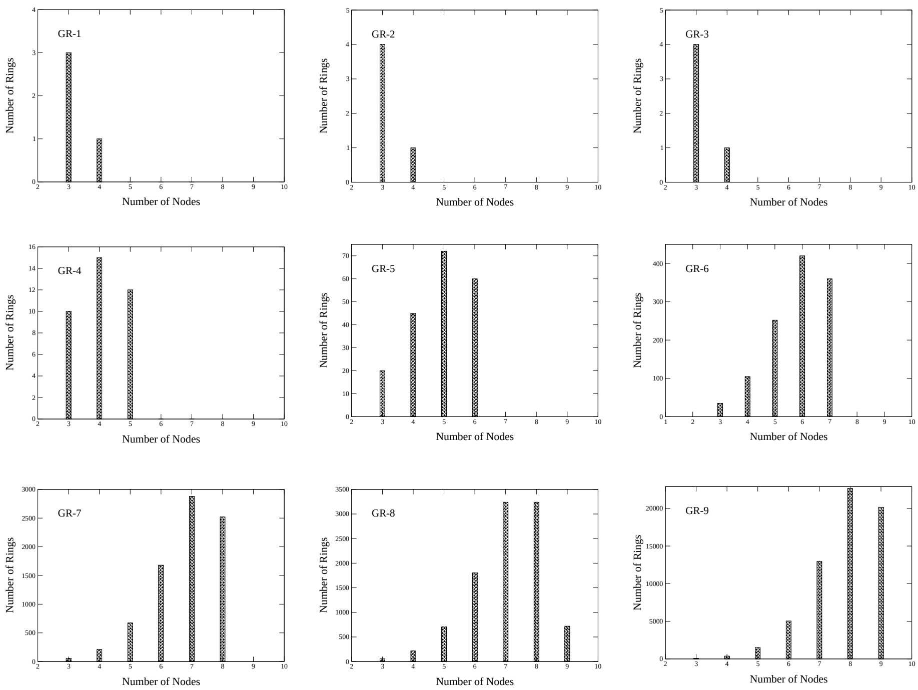

Figure 2. Graph Rings Distribution

4. Conclusions

This paper has presented an algorithm, to extract and classify all rings set. Rings are detected using a set of initial rings and an iterative process to perform cyclical conjunction operations on new rings detected in the foregoing iteration, to classify the rings we used the number of nodes in each ring.

The algorithm has been tested on a 9 graphs, and all existing rings were detected and classified. There was no instance of the generation of non-existent rings.

Implementation of the cyclical conjunction operator properties afforded a reduction of a great number of operations required to extract all rings from a graph, greatly speeding up the algorithm, although this raised computational cost.

Tests showed calculation time for the extraction of graph rings is dependent on factors such as total number of rings and graph topology and complexity, in addition to the nature of the initial rings detected at the preprocessing phase, factors that are either unknown or excessively costly to determine a priori.

References

[1] L. O. Chua and L. K. Chen. On optimally sparce cycle and coboundary basis for a linear graph. IEEE Trans. Circuit Theory, CT-20:495-503, 1973.
[2] D. E. Knuth. The Art of Computer Programing, 1. Addinson-Wesley : 363-370, 1968.
[3] C. G Wermuth. Ring Transformations. In *The Practice of Medicinal Chemistry*; Wermuth, C. G., Ed.; Academic Press: London, 1996; pp 239-260.
[3] G. W.Bemis, M. A. Murcko. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39*, 2887-2893.
[4] S. Basak, T. Balaban, G. Grunwald, and D. Brian. Topological indices: Their nature and mutual relatedness. *American Chemical Society*, **2000**, 40(4), 891–1150.
[5] M. Plotkin. Mathematical basis of ring - finding algorithms in CIDS. *Journal of Chemical Documentation*, **1970**, 11(1), 60–63.
[6] M. Bersohn. An algorithm for finding the synthetically important rings of a molecule. *Theory Graphs, American Mathematical Society Colloquium Publications*, **1973**, 38, 1239–1241.
[7] L.O. Chua and L. K. Chen. On optimally sparce cycle and coboundary basis for a linear graph. *IEEE Trans. Circuit Theory*, **1973**, 20, 495-503.
[8] D.E. Knuth. The Art of *Computer Programing, 1*. Addinson-Wesley : 363-370, 1968.
[9]E.J. Corey and G.A. Petersson. An algorithm for machine perception of synthetically significant rings in complex cyclic organic structures. *J. Am. Chem. Soc.*, **1972**, 94, 460–465.
[10]P. Mateti and N. Deo. On algorithms for enumerating all circuits of a graph. *SIAM J. Comput.*, **1976**, 5, 90–99.
[11] M. Randic. Random walks and their diagnostic value for characterization of atomic environment. *Journal of Computational Chemistry*, **1980**, 1(4), 386–399.
[12] Downs G. M., Gillet V. J., Holliday J. D., and Lynch M. F. Theoretical aspects of ring perception and development of the extended set of smallest rings concept. *American Chemical Society*, **1988**, 29(3), 187–206.
[13] T. F. Fan, J. Doucet, and A. Barbu. Ring perception. a new algorithm for directly finding the smallest rings from a connection table. *J. Chem Inf. Comput. Sci.*, **1993**, 33, 657–662.
[14] J. Figueras. Ring perception using breadth-first search. *J. Chem Inf. Comput. Sci.*, **1996**, 36, 986–991.
[15] G. Cerruela García, I. Luque Ruiz, M.A. Gómez-Nieto. Cyclical Conjunction: an Efficient Operator for the Extraction of Cycles from a Graph (to be submitted for publication to J. Chem. Inf. Comput. Sci.).
[16] G. Cerruela García, I. Luque Ruiz, M.A. Gómez-Nieto. Un algoritmo en zig-zag para la obtención de un conjunto de ciclos característicos de un grafo, **2000**, University of Córdoba, Internal Report UCO-ISCBD-JCVG93-00.