Application of a mathematical problem to the symmetry of fullerene C\textsubscript{60}

G. R. Vakili-Nezhaad and A. R. Ashrafi

Institute of Nanoscience and Nanotechnology, University of Kashan, 87317-51167, Kashan, Iran

E-mail: vakili@kashanu.ac.ir

Abstract. In an earlier paper, the second author of this paper proved an algorithm for computing the symmetry of molecules. He applied his algorithm to calculate the symmetry of the smallest fullerene C\textsubscript{20}. In this paper, we improve the mentioned algorithm to compute the automorphism group of the fullerene C\textsubscript{60} with connectivity and geometry of \textit{Ih} symmetry point group.

1. Introduction

By a fullerene, we mean a trivalent plane graph $\Gamma = (V,E,F)$ with only hexagonal and pentagonal faces. It follows easily from Euler’s Formula that each fullerene has exactly 12 pentagonal faces. The simplest fullerene is the graph of the dodecahedron with 12 pentagonal faces and no hexagonal faces. The fullerene era was started in 1985 with the discovery of the stable C\textsubscript{60} cluster and its interpretation as a cage structure with the familiar shape of a soccer ball [1]. An explosive growth in fullerene research was triggered in 1990 by the development of a method to produce fullerenes in bulk quantities [2]. Fulleren chemistry is nowadays a well-established field of both theoretical and experimental investigations. The initial fascinating appeal, coming from their beautiful symmetry shifted later to real chemistry [3]. On the other hand, orientational phenomena which have important role in the properties of fullerenes go back to the high symmetry of these carbon nanostructures [4].

In this letter, a problem in mathematical chemistry related to the symmetry of molecules has been considered. Here, symmetry means an automorphism of Euclidean graph of the molecule under consideration.

Balasubramanian [5-13], in some leading papers considered the Euclidean matrix of a chemical graph to find its symmetry. He proved that for computing the symmetry of a molecule, it is enough to solve the matrix equation $P^tEP = E$, where $E$ is the Euclidean matrix of the molecule under consideration and $P$ varies on the set of all permutation matrices with the same dimension as $E$. He computed the Euclidean graphs and its automorphism group for benzene, eclipsed and staggered forms of ethane, eclipsed and staggered forms of ferrocene, see [3] for details.

Ashrafi [14], introduced an efficient algorithm for computing the symmetry of molecules. Using this algorithm in [15], the authors computed the symmetry of the big fullerene C\textsubscript{80}. In this letter we improve this algorithm to compute the automorphism group of fullerenes. In this paper our notation is standard and taken mainly from [16-18]. Computations were carried out with the aid of GAP [17,18]. We encourage reader to consult [5] and [14] for discussion and background material about the symmetry properties of molecules.
2. Computational Details

In this section we describe some notations which will be kept throughout. For a permutation \( \sigma \) on \( n \) objects, the corresponding permutation matrix is an \( n \times n \) matrix \( P_\sigma \) is given by \( P_\sigma = [x_{ij}] \), where \( x_{ij} = 1 \) if \( i = \sigma(j) \) and 0 otherwise. It is easy to see that \( P_\sigma P_\tau = P_{\sigma \tau} \), for any two permutations \( \sigma \) and \( \tau \) on \( n \) objects, and so the set of all \( n \times n \) permutation matrices is a group isomorphic to the symmetric group \( S_n \) on \( n \) symbols.

It is well known to associate an Euclidean graph to a molecule, which is defined by a weighted graph with the adjacency matrix \( D = [d_{ij}] \), where for \( i \neq j \) \( d_{ij} \) is the Euclidean distance between the nuclei \( i \) and \( j \). In this matrix \( d_{ii} \) can be taken as zero if all the nuclei are equivalent. Otherwise, one may introduce different weights for different nuclei. We can assign a vector \( v_{0} \) to any vertex \( v \) as \( v_{0} = (t_{1}, t_{2}, ..., t_{k}) \), \( t_{1} \leq t_{2} \leq ... \leq t_{k} \) where \( t_{1}, t_{2}, ..., t_{k} \) are the Euclidean distances of the edges incident with the vertex \( v \).

An automorphism of an Euclidean graph \( G \) is a permutation \( g \) of the vertex set of \( G \) with the property that for every edge \( e, e \) and \( g(e) \) have the same Euclidean length. The set of all automorphisms of a graph \( G \), with the operation of composition of permutations, is a permutation group on \( V(G) \), denoted \( \text{Aut}(G) \). By symmetry we mean the automorphism group symmetry of a graph.

The automorphisms have other advantages such as in generating nuclear spin species, NMR spectra, nuclear spin statistics in molecular spectroscopy, chirality and chemical isomerism. There is also another important applications of automorphism groups of weighted graphs to fullerenes.

It is a well-known fact that a permutation \( \sigma \) of the vertices of a graph \( G \) belongs to its automorphism group if it satisfies \( P_\sigma^t AP_\sigma = A \), where \( A \) is the adjacency matrix of \( G \). Suppose \( \text{Aut}(G) = \{\sigma_{1}, ..., \sigma_{m}\} \). The matrix \( S_{G} = [s_{ij}] \), where \( s_{ij} = \sigma_{j}(i) \) is called a solution matrix for \( G \). Clearly, for computing the automorphism group of \( G \), it is enough to calculate a solution matrix for \( G \).

Lemma. Let \( G \) be an Euclidean graph and \( g \) be an automorphism of the graph \( G \). Then for any vertex \( v, v_{0} = g(v_{0}). \)

Using Lemma 1 and its corollary from [14] and previous Lemma, we can write a new MATLAB Program (version 6.1) to compute the automorphism group of the Euclidean graph of a molecule. This program computes a solution matrix of the equation \( A \) = \( P \ A \ (P)^t \) and is available upon the request from the authors.

To explain the program, we apply our MATLAB program to compute the symmetry of fullerene \( C_{60} \) Figure 1, with \( I_h \) point group symmetry. It should be mentioned that one does not have to work with exact Euclidean distances in that a mapping of weights into a set of integers would suffices as long as different weights are identified with different integers. In fact the automorphism group of the integer-weighted graph is identical to the automorphism group of the original Euclidean graph.

Not all \( 60! \) permutations of the vertices \( C_{60} \) belong to the automorphism group of its weighted graph since the weights of all the edges are not the same. For example, the permutation \((1,2,3,4)\) does not belong to the automorphism group since the resulting graph does not preserve connectivity. Let \( X \) denote the set of all solutions of matrix equation \( P A P = A \). Set \( Y = \{ \alpha \in S_{60} \mid P \alpha \in X \} \). Then \( Y \) is the automorphism group of Euclidean graph of \( C_{60} \). We now apply our MATLAB program to find a solution matrix for this group. After running this program, we can see that \( G \) has order 120. Using the solution matrix of \( C_{60} \) and a simple GAP program, we can find the structure of the automorphism group \( G \) of Euclidean graph of \( C_{60} \). Using this GAP program two proper non-trivial normal subgroups \( N = \text{R}[2] \) and \( M = \text{R}[3] \) is obtained which intersects trivially. Therefore, \( G \) is isomorphic to the direct product \( Z_{2} \times A_{5} \), where \( Z_{2} \) is a cyclic group of order 2 and \( A_{5} \) is the unique simple group of order 60. Our GAP program is as follows:

**A GAP Program for Computing the Structure of the Automorphism Group of the Euclidean Graphs of \( C_{60} \)**

```gaps
G := Group(X); Size(G);
R := NormalSubgroups(G);
```

We now consider the following permutations:

\[ a = (1,7,2,4,8)(3,17,5,10,12)(6,21,13,16,27)(9,26,11,18,28)(14,36,25,23,34)(15,37,19,22,33)(20,48,38,30,43)(24,44,29,35,42)(31,54,40,47,50)(32,55,41,46,51)(39,56,49,45,58)(52,60,53,59,57), \]

\[ b = (1,11)(2,5)(3,6)(4,19)(7,25)(8,13)(9,14)(10,29)(12,28)(16,20)(17,38)(18,40)(21,41)(22,39)(23,32)(26,42)(27,33)(30,31)(35,53)(36,49)(37,50)(44,51)(45,46)(47,52)(48,57)(54,58)(55,60)(56,59), \]

\[ d = (1,60)(2,59)(3,58)(4,57)(5,56)(6,54)(7,53)(8,52)(9,51)(10,49)(11,55)(12,45)(13,47)(14,44)(15,43)(16,50)(17,39)(18,41)(19,48)(20,37)(21,40)(22,38)(23,42)(24,34)(25,35)(26,32)(27,31)(28,46)(29,36)(30,33), \]

\[ e = (1,60)(2,57)(3,58)(4,59)(5,49)(6,51)(7,52)(8,53)(9,54)(10,56)(11,50)(12,39)(13,41)(14,43)(15,44)(16,55)(17,45)(18,47)(19,42)(20,34)(21,46)(22,35)(23,48)(24,37)(25,38)(26,31)(27,32)(28,40)(29,33)(30,36), \]

\[ f = (1,10,9)(2,18,15)(3,4,16)(5,21,24)(6,7,23)(8,22,14)(11,17,30)(12,35,20)(13,36,31)(19,26,46)(25,37,45)(27,47,29)(28,48,32)(33,55,39)(34,56,40)(38,44,52)(41,43,59)(42,54,53)(49,51,60)(50,58,57). \]

**Figure 1:** Fullerenes $C_{60}$ with $I_h$ Point Group Symmetry

By this program, $\{a, b\}$ is a generating set for $G$. Also, $N = <d>$ and $M = <e, f>$ are normal subgroups of $G$. Since $M$ is a simple group of order 60 and $A_5$ is the unique simple group of this order, $M \cong A_5$. Therefore, the automorphism group of the Euclidean graph of fullerene $C_{60}$ has order 120 and is isomorphic to $Z_2 \times A_5$. 
3. Conclusions

The algorithm introduced by Ashrafi [14] is working with small groups. Using our new algorithm, it is possible to compute the symmetry of $C_{150}$ fullerene less than 10 seconds. On the other hand, our calculations with GAP and calculations done by Balasubramanian suggest that the automorphism group of the Euclidean graph of every molecule is trivial or have an even number of elements.

References
[1] H.W. Kroto, J.R. Heath, S.C. O’Brien, R.F. Curl and R.E. Smalley 1985 Nature 318 162.
[2] W. Krätschmer, L.D. Lamb, K. Fostiropoulos and D. Huffman 1990 Nature 347 354.
[3] A. Hirsh and M. Brettreich 2005 Fullerenes (Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim) pp. 409-411.
[4] R. Moret 2005 Acta Cryst. A61 62.
[5] K. Balasubramanian 1995 Chem. Phys. Lett. 232 415.
[6] K. Balasubramanian 1980 J. Chem. Phys. 72 665.
[7] K. Balasubramanian 1982 Int. J. Quantum Chem. 21 411.
[8] K. Balasubramanian 1985 Chem. Rev. 85 599.
[9] K. Balasubramanian 1983 Stud. Phys. Theor. Chem. 23 149.
[10] K. Balasubramanian 1981 J. Chem. Phys. 75 4572.
[11] K. Balasubramanian 2004 J. Phys. Chem. 108 5527.
[12] K. Balasubramanian 2004 Chem. Phys. Lett. 391 64.
[13] K. Balasubramanian 2004 Chem. Phys. Lett. 391 69.
[14] A.R. Ashrafi 2005 Chem. Phys. Lett. 406 75.
[15] A.R. Ashrafi and M.R. Ahmadi 2005 CEJC 3 647.
[16] P.W. Fowler and D.E. Manolopoulos 1995 An Atlas of Fullerenes (Oxford University Press, Oxford).
[17] M. Schönert, H.U. Besche, Th. Breuer, F. Celler, B. Eick, V. Felsch, A. Hulpke, J. Mnich, W. Nickel, G. Pfeiffer, U. Polis, H. Theißen and A. Niemeyer, GAP, Groups, Algorithms and Programming, Lehrstuhl De für Mathematik, RWTH, Aachen, 1995.
[18] A.R. Ashrafi 2005 MATCH Commun. Math. Comput. Chem. 53 161.