Efficient computation of the Weiner Index for a class of planar graphs

Yuqing Lin and Nathan Van Maastricht

University of Newcastle
University Drive Callaghan
NSW 2308, Australia

Abstract

WRITTEN LAST. Computation of the Weiner Index for a graph on n vertices, using the definition, is of order n^2. We give an algorithm that applies to a certain class of planar graphs, which is of smaller order.

1 Introduction

The Weiner Index is an important metric in chemical graph theory as it closely correlates with the boilings points of alkane molecules, along with many other parameters of the molecule, such as density, surface tension and viscosity of its liquid phase [5]. Direct computation of the Weiner index from the definition, for a graph with n vertices, is of the order $O(nm)$, where n is the number of vertices and m is the number of edges in the graph. It is of practical and theoretical interest to find a more efficient algorithm for computing the Weiner index.

We consider a certain class of planar graphs, called partial cubes, the definition is well known, see for example [3], but will be reviewed in section 2. In section 3 we introduce an algorithm using cut methods to calculate the Weiner Index in linear time for this class of graphs.

2 Definitions

Before we talk about the algorithm, we need to introduce and recall some definitions. We need to talk about the class of graphs we are working with, and some terms used.

Of course, the most important thing to introduce is the Weiner Index itself. In 1971, Hosoya extended the original definition of the Weiner Index from trees to all connected Graphs G by the following definition[1].

Definition 1. *The Weiner Index of a graph, G, is given by*

$$W(G) = \sum_{u,v \in V(G)} d_G(u,v).$$

Definition 2. *[3] Let G be a connected graph, then G is a* **partial cube** *if its vertices u can be labeled with binary strings $l(u)$ of a fixed length, such that*

$$d_G(u,v) = H(l(u), l(v))$$

holds for any vertices u and v of G, where $H(l(u), l(v))$ is the Hamming distance of the binary strings $l(u)$ and $l(v)$.

The class of graphs to which we are apply our algorithm is the set of planar partial cubes. The algorithm is an example of a *cut method*, as described in [3]. There is a general notion of an elementary cut, which applies to arbitary graphs and is defined in terms of a relation called the *Djoković-Winkler* relation. We give a restricted definition of an elementary cut, which applies to our graphs of interest, and is equivalent to the more general definiition when restricted to our set.

Definition 3. *Let $G = (V, E)$ be a simple connected planar graph, all of whose cycles are of even length, which we assume to be embedded in the plane. An* **elementary cut** *is a set of edges $\{e_1, \ldots, e_k\}$ such that the alternating sequence of edges and faces:*

$$e_1 F_1 e_2 F_2 \cdots e_{k-1} F_{k-1} e_k$$

has the following properties:

- *Each e_i is an edge in the graph*
- *Each F_i is a face in the graph*
- *Edges e_i and e_{i+1} are antipodal edges belonging to the face F_i*
- *The first and last edges e_1 and e_{k+1} lie on the boundary of the embedding.*

It is a well known result (see [3]) that the Weiner index can be calculated on planar partial cubes by the following theorem.

Theorem 1. *Let G be a planar partial cube and C_i be an elementary cut. Let $n_1(C_i)$ and $n_2(C_i)$ be the number of vertices in the two connected components of $G - C_i$, that is, the graph created by removing from G the edges in the elementary cut C_i. Then the sum over all elementary cuts $C_1, \ldots, C_k$ is the Weiner index:*

$$W(G) = \sum_{i=1}^{k} n_1(C_i) n_2(C_i).$$

Cut methods have been used in the past to calculate the Weiner index of chemical graphs [2]. We present a faster way to calculate the Weiner index on a planar partial cube.

3 The Algorithm

The algorithm can be seperated into two main components.

1. Find the elementary cuts;
2. Count the number of vertices on either side of all the cuts.

The first of these can be achieved by algorithm 1.

Algorithm 1 Elementary Cut Finding Algorithm

for all edges in G **do**
    Join the edge to it's adjacent face.
    Increment a counter on the face by one, to keep track of the size of each cycle.
end for
for all boundary edges **do**
    Create a *cut* by tracing the unique path from edge to face to the antipodal edge, until another boundary edge is reached.
end for

The second of these can be acheived by algorithm 2.

Algorithm 2 Counting Vertices

for a **do**
    b
end for

4 Proof(s)

Nathan, write stuff here

Nathan, write stuff here

References

[1] H. Hosoya. Topological index. A newly proposed quantity characterizing the topological nature of structural isomers of saturated hydrocarbons. Bull. Chem. Soc. Japan. 44. pp 2332–2339. 1971.

[2] S. Klavžar and I. Gutman and A. Rajapaske. Wiener numbers of pericondensed benzenoid hydrocarbons. Croat. Chem. Inf. Comput. Sci. 70 pp 979–999. 1997.

[3] S. Klavžar and M. J. Nadjafi-Arani. Cut Method: Update on Recent Developments and Equivalence of Independent Approaches. Current Organic Chemisty. 19. pp 348–358. 2015.

[4] Y. Pan and M. Xie and F. Zhang. Binary Hamming Graphs in Archimedean Tilings.

[5] D. H. Rouvray and B. C. Crafford. The dependence of physical-chemical properties on topological factors. South African Journal of Science. 72. pp 47–51. 1976.

[6] H. Wiener. Structural determination of paraffin boiling points. JJ. Amer. Chem. Soc. 69. pp 17–20. 1947.