ROYAUME DU MAROC
Université Mohammed V - RABAT - جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم

CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre 3230

THESE

En vue de l'obtention du : ***DOCTORAT***

Structure de Recherche : Laboratoire de Recherche en Informatique et Télécommunications
Discipline : Sciences de l'ingénieur
Spécialité : Informatique et Télécommunications

Présentée et soutenue le 13/07/2019 par :

Meryam ZERYOUH

Efficient computational methods for two distance-based topological indices and applications on large networks

JURY

Ahmed HAMMOUCH	PES, École Normale Supérieure de l'Enseignement Technique, Université Mohammed V de Rabat	Président
Mohamed EL MARRAKI	PES, Faculté des Sciences, Université Mohammed V de Rabat	Directeur de Thèse
Dounia LOTFI	PH, Faculté des Sciences, Université Mohammed V de Rabat	Rapporteur/ Examinateur
Mohamed EL KAMILI	PH, Ecole Supérieure de Technologie, Université Hassan II de Casablanca	Rapporteur/ Examinateur
Noussaima EL KHATTABI	PH, Faculté des Sciences, Université Mohammed V de Rabat	Examinateur
Aziz ABOULMOUHAJIR	PES, Faculté des Sciences Aïn Chock, Université Hassan II de Casablanca	Examinateur
Mohamed ESSALIH	PA, Ecole Supérieure de Technologie Safi, Université Cadi Ayyad de Marrakech	Invité

Année Universitaire : 2018-2019

✉ Faculté des Sciences, avenue Ibn Battouta, BP. 1014 RP, Rabat –Maroc
☎ 00212 (05) 37 77 18 76 00212(05) 37 77 42 61; http://www. fsr.um5.ac.ma

I would like to dedicate my thesis
to my beloved parents
and my sister.

Acknowledgements

During the preparation of my thesis, I was granted a scholarship of excellence in research awarded by the National Center for Scientific and Technical Research (CNRST).
This thesis has been performed in the Laboratory of Research in Computer Science and Telecommunication (LRIT) of the Faculty of Sciences Rabat, Mohammed V University, under the supervision of Professor **Mohamed EL MARRAKI** and the co-supervision of Professor **Mohamed ESSALIH**.

First of all, I would like to express my profound gratitude to my supervisor, Professor **Mohamed EL MARRAKI**, Full Professor at Faculty of Sciences Rabat, for his availability, close collaboration, and his kind assistance. I am thankful to him for giving me a strong background to pursue my research in this field, for having fruitful discussions with him, as well as for his strong support and dedication during these years.

I would like to warmly thank Professor **Mohamed ESSALIH**, Assistant Professor at the High School of Technology of Safi, for his co-supervision, patience, and motivation. I want to thank him especially for letting me wide autonomy while providing appropriate advice. I hope to keep up our collaboration in the future.

I would like to convey my greatest honor and gratitude to Professor **Ahmed HAMMOUCH**, Full Professor in Higher School of Technical Education Rabat (ENSET), for agreeing to chair the committee of my thesis.

I would like to thank Professor **Dounia LOTFI**, Qualified Professor at Faculty of Sciences Rabat, for having agreed to report this work and to participate in the committee. I sincerely thank her for her availability and her precious remarks in order to ameliorate my modest work.

I am very fortunate and grateful to Professor **Mohamed EL KAMILI**, Qualified Professor at the High School of Technology of Casablanca, for having agreed to report this work and for his valuable comments that helped me to improve this manuscript.

I would also like to thank Professor **Aziz ABOULMOUHAJIR**, Full Professor at Faculty of Sciences Aïn Chock Casablanca, for his kind willingness to attend and judge this thesis.

I would also like to extend my warmest thanks to Professor **Noussaima EL KHATTABI**, Qualified Professor at Faculty of Sciences Rabat, for having taken the time to thoroughly evaluate this work.

I would like to express here my acknowledgments to every person who gave me support and contributed to the elaboration of this work.

Finally, my warmest thanks go to my lovely parents and sister for their endless support and encouragement through my entire life.

ABSTRACT

Many complex systems and real world phenomena can be modeled as networks and analyzed using techniques derived from graph theory. One of the used approaches to characterize the structural information of networks is a kind of quantitative measures called topological indices. The most considerable class of these measures is distance-based topological indices, which includes the oldest and the well-known measure called the Wiener index. In this thesis, we focus on two recent extensions of the Wiener index that are the Terminal Wiener index and the generalized Terminal Wiener index.

The ultimate objective of this work is to improve and develop some new methods that would compute efficiently the Terminal Wiener index and its generalization. First of all, we study some fundamental properties of these indices, and especially the maximal bounds that the Terminal Wiener index assumes for the class of trees. After that, we propose three methods to overcome the problem of calculating the Terminal Wiener index of some kinds of networks. However, the use of these methods in the case of some large and complex networks would be a difficult task. For this reason, we study a new powerful approach called the cut method and we use this technique to derive some new efficient formulas for computing the Terminal Wiener index and its generalization. Furthermore, we focus on an extension of the cut method that is based on an important concept called the canonical metric representation in order to reduce the computational complexity of these indices, as well as to develop a linear algorithm for some complex systems.

Keywords: Networks, Graph theory, Topological indices, Wiener index, Terminal Wiener index, Generalized Terminal Wiener index, Distance, Maximal bounds, Cut method, Complexity, Canonical representation.

Résumé

De nombreux systèmes complexes et phénomènes du monde réel peuvent être modélisés sous forme de réseaux et analysés à l'aide de certaines techniques dérivées de la théorie des graphes. L'une des approches utilisées pour la caractérisation des informations structurelles des réseaux est une sorte de mesures quantitatives appelées les indices topologiques. La classe la plus considérée de ces mesures est celle des indices topologiques basés sur la distance, qui inclut l'une des plus importantes et anciennes mesures appelé l'indice de Wiener. Dans cette thèse, nous allons nous concentrer sur deux extensions récentes de l'indice de Wiener notamment l'indice de Terminal de Wiener et l'indice de Terminal de Wiener généralisé.

L'objectif de ce travail est d'améliorer et de développer de nouvelles méthodes permettant le calcul efficace de l'indice de Terminal de Wiener et sa généralisation. Tout d'abord, nous étudions quelques propriétés fondamentales de ces indices, en particulier les limites maximales de l'indice de Terminal de Wiener pour le cas des arbres. Ensuite, nous proposons trois méthodes pour résoudre le problème de calcul de l'indice de Terminal de Wiener pour certains types de réseaux. Cependant, l'utilisation de ces méthodes dans le cas des réseaux larges et complexes serait une tâche difficile. Pour cette raison, nous étudions une nouvelle approche appelée la méthode de coupe **"cut method"** et nous utilisons cette technique pour dériver de nouvelles formules efficaces pour le calcul de l'indice de Terminal de Wiener et sa généralisation. De plus, nous nous concentrons sur une extension de la méthode de coupe basée sur un concept appelé la représentation canonique d'un réseau afin de réduire la complexité de calcul de ces indices et de développer un algorithme linéaire pour certains systèmes complexes.

Mots clés: Réseau, Théorie des graphes, Indices topologiques, L'indice de Wiener, L'indice de Terminal de Wiener, L'indice de Terminal de Wiener généralisé, Distance, Limites maximales, Méthode de coupe, Complexité, Représentation canonique.

Résumé Détaillé

Actuellement, les réseaux sont utilisés comme un outil fondamental pour représenter et modéliser tout système complexe et tout phénomène du monde réel. Par exemple, nous pouvons citer la structure chimique des molécules, les systèmes biologiques, les systèmes d'interconnexion, les systèmes sociaux, les systèmes écologiques et d'autres exemples issus de différents domaines. Alors, le terme réseau fait référence à une représentation abstraite d'un système. Il se compose d'un ensemble de sommets qui représentent les différents éléments du système et d'un ensemble d'arêtes entre les sommets qui décrivent les interactions entre les éléments de ce système. Par exemple, la structure chimique d'une molécule est un réseau, où les sommets étant les atomes et les arêtes interprètent les réactions chimiques entre ces atomes. En mathématiques, l'étude des réseaux est connue sous le nom de la théorie des graphes. Dans ce domaine de recherche, nous utilisons un objet mathématique appelé graphe pour modéliser toute situation réelle. La différence entre les termes graphe et réseau est que le premier terme décrit une situation réelle sous forme d'un objet purement mathématique, alors qu'un réseau peut contenir des informations supplémentaires, telles que les attributs liés à chaque sommet et arête. Dans cette thèse, nous allons utiliser les deux termes graphe et réseau de manière interchangeable.

L'une des questions fondamentales résultant de l'étude des réseaux est celle de la description des propriétés topologiques d'une structure de manières quantitatives. Pour cette raison, un certain nombre de mesures quantitatives, également appelées indices topologiques, ont été proposées et étudiées dans la littérature. Pour illustrer, les indices topologiques sont des valeurs numériques qui réalisent un codage de l'information structurelle d'un réseau et ils sont invariants dans le cas des réseaux isomorphes. Il existe plusieurs classes d'indices topologiques, certaines d'entre elles sont: les indices basés sur la distance et les indices basés sur les degrés des sommets. Le concept des indices topologiques a débuté en 1947, lorsque le physicien chimiste Harold Wiener a défini une mesure nommée l'indice de Wiener pour prédire le point d'ébullition des paraffines (Wiener, 1947). Plus tard, une forte corrélation entre cet indice et les propriétés chimiques d'une composante moléculaire a été trouvée, puis cette mesure est devenue l'une des descripteurs les plus fréquemment utilisés. Quelques années après l'introduction de l'indice de Wiener et ses applications en chimie, la même quantité a été étudiée par les mathématiciens sous divers noms, y compris gross status (Harary, 1959), la distance d'un graphe (Entringer et al., 1976), la transmission (Šoltés, 1991) et tout simplement comme étant la somme des distances (Yeh and Gutman, 1994). En outre, l'indice de Wiener a été appliqué dans plusieurs autres situations, tel qu'en informatique pour mesurer la performance moyenne d'un réseau, la cristallographie, l'emplacement d'installations et la cryptologie ; voir (Alsuwaiyel, 2010), (Quadras et al., 2016) et (Bonchev and Klein, 2002). Grâce au succès de l'indice de Wiener, un grand nombre d'indices topologiques ont été proposés dans la littérature (Xu et al., 2014). Dans cette thèse, nous allons nous concentrer sur les indices topologiques basés sur le concept de la distance et plus particulièrement à l'indice de Wiener et les deux plus récents indices nommés l'indice de Terminal de Wiener (Gutman et al., 2009a) et l'indice de Terminal de Wiener généralisé (Ilić and Ilić, 2013). Afin de permettre aux lecteurs de se familiariser avec les indices topologiques basés sur la distance, nous définissons les mesures précédemment citées comme suit. L'indice de Wiener d'un réseau G, noté $W(G)$, est obtenu de la matrice de distance de G, tel que nous calculons la somme de distances entre toutes les paires de sommets de G. Par contre, l'indice de Terminal de Wiener, noté $TW(G)$, est calculé à partir d'une matrice de distance réduite d'un réseau G. Plus précisément, il se définit comme étant la somme de distances entre les feuilles du réseau G. En outre, la généralisation de l'indice de Terminal de Wiener, noté $TW_K(G)$, est également obtenu à partir d'une matrice de distance réduite en considérant uniquement les sommets d'un degré donné $K \geq 1$.

Depuis l'émergence d'indices topologiques basés sur la distance, la majorité des recherches ont été concentrés sur la détermination des valeurs extrêmes qu'un indice admet dans les différentes classes de graphes, ainsi que le développement de la définition de base des indices pour faciliter leurs calculs dans le cas des réseaux étendus et complexes. Par conséquent, de nombreuses méthodes basées sur le concept de distance ont été proposées pour étudier certains types de réseaux possédant un grand nombre de sommets. Cependant, dans le cas de quelques structures complexes, l'application de ces méthodes sera dominée par le calcul des plus courts chemins entre les sommets. Donc, le problème en question est le développement de nouvelles techniques et d'algorithmes permettant le calcul de ces indices topologiques sans l'utilisation de la notion de distance. Ce problème a été posé dans (Mohar and Pisanski, 1988), où les auteurs de cet article ont introduit un algorithme pour calculer l'indice de Wiener des réseaux acycliques en un temps linéaire, ainsi ils ont motivé les autres chercheurs de continuer le travail dans cet axe de recherche et de traiter les autres classes de réseaux. En conséquence, une nouvelle approche appelée la méthode de coupe **"cut method"** a été proposée par Klavzar et al. (1995) pour calculer l'indice de Wiener d'une classe de réseaux, nommé les cubes partiels, sans passer par le calcul de la matrice de distance. Cette méthode a attiré l'attention de nombreux spécialistes ce qui a conduit à étendre cette technique pour les autres indices topologiques basés sur la distance, et de trouver d'autres extensions de cette méthode afin d'adapter son application pour les autres classes de réseaux.

L'objectif essentiel de cette thèse est de contribuer à l'étude de deux indices topologiques basés sur la distance, notamment l'indice de Terminal de Wiener et l'indice de Terminal de Wiener généralisé. D'une part, nous nous intéressons à l'étude de quelques propriétés mathématiques de ces deux indices, en particulier les valeurs extrêmes qu'un indice admet pour quelques classes de graphes. D'autre part, nous nous concentrons sur le développement et l'amélioration de nouvelles méthodes afin de réduire la complexité de calcul de

l'indice de Terminal de Wiener et sa généralisation dans le cas des réseaux larges et complexes.
Nos principales contributions sont résumées comme suit :

- **L'étude de quelques propriétés fondamentales de l'indice de Terminal de Wiener:** Afin d'extraire certaines caractéristiques fondamentales d'un indice topologique, nous devons déterminer ses limites extrêmes pour une classe de graphes définit auparavant. Dans cet axe de recherche, nous avons examiné les limites maximales de l'indice de Terminal de Wiener pour le cas des arbres, notamment la construction des arbres ayant la seconde valeur maximale de l'indice de Terminal de Wiener. Ce travail nous a permis de déterminer une transformation qui augmente et maximise la valeur de l'indice de Terminal de Wiener. En effet, cette première contribution à fait l'objet d'un article présenté dans une conférence internationale (Zeryouh et al., 2016).

- **La proposition de quelques méthodes pour faciliter le calcul de l'indice de Terminal de Wiener:** L'application direct de la définition des indices topologiques est une tâche difficile, en particulier dans le cas des réseaux contenant un grand nombre de sommets. Dans ce contexte, nous avons proposé trois méthodes pour résoudre le problème de calcul de l'indice de Terminal de Wiener. La première approche est une méthode de décomposition dédiée à un type de réseau construit à partir la concaténation de plusieurs copies de composantes, par exemple les Star-Trees et les Path-Trees. La deuxième méthode est basée sur l'utilisation d'un concept appelé le graphe de Thorn, dont l'objectif est de démontrer que l'indice de Terminal de Wiener d'un réseau peut être obtenu à partir de son réseau parent. La dernière méthode est une réécriture de la définition de l'indice de Terminal de Wiener. Les résultats obtenus dans cette contribution ont été publiés dans un journal international (Zeryouh et al., 2015b) et présentés dans deux conférences internationales (Zeryouh et al., 2015a),(Zeryouh et al., 2018a).

- **Étendre la méthode de coupe pour le cas de l'indice de Terminal de Wiener généralisé:** dans le cas des réseaux larges et complexes, la complexité des méthodes proposées dans la contribution précédente est dominée par le calcul des distances entre les sommets. Par conséquent, nous avons étudié une approche puissante, appelée la méthode de coupe **"cut method"**. L'objectif de cette méthode est de découper un réseau en un ensemble de petites composantes et d'assembler les indices de ces composantes pour générer la propriétée de la structure entière sans passer par l'utilisation de la notion de distance. Il existe deux versions pour cette technique : une méthode basée sur la partition des arêtes et une autre méthode basée sur la partition des sommets. Par la suite, nous sommes y arrivés à étendre et développer ces deux versions de la méthode de coupe pour améliorer le calcul de l'indice de Terminal de Wiener et sa généralisation. Cette contribution a donné lieu de deux publications présentées dans deux conférences internationales (Zeryouh et al., 2017) et (Zeryouh et al., 2018b).

- **Réduire la complexité de calcul de l'indice de Terminal de Wiener généralisé :** Finalement, nous présentons comme dernière contribution une extension de la méthode basée sur la partition des arêtes afin de réduire la complexité de calcul de l'indice de Terminal de Wiener et sa généralisation. Nous avons prouvé l'efficacité de cette technique en développant un algorithme rapide pour la calcul de l'indice de Terminal de Wiener généralisé d'un réseau hiérarchique appelé le Dendrimer graphe. Par la suite, nous avons présenté un algorithme qui calcul l'indice de Terminal de Wiener généralisé d'un système complexe en un temps linéaire. Cette dernière contribution a fait l'objet d'un article présenté dans une conférence internationale (Zeryouh et al., 2018a) et un autre accepté dans une revue internationale (Zeryouh et al., 2019).

Il faut noter que toutes les méthodes proposées dans cette thèse ont été appliquées sur un ensemble de réseaux larges et complexes pour prouver l'efficacité de nos méthodes et aussi afin d'analyser la structure topologiques de ces réseaux.

Contents

List of Figures

List of Tables

List of Algorithms

List of Publications

International journals

- Zeryouh, M., El Marraki, M., and Essalih, M. (2014). Wiener and terminal Wiener indices of some rooted trees. Applied Mathematical Sciences, 8(11), 4995-5002.

- Zeryouh, M., El Marraki, M., and Essalih, M. (2015). Terminal wiener index of star-tree and path-tree. Applied Mathematical Sciences, 9(39), 1919-1929.

- Zeryouh, M., El Marraki, M., and Essalih, M. (2019). An Efficient Computational Method for a Distance-based Measure of Graphenylene Network. International Journal of Advanced Trends in Computer Science and Engineering, 8(3), 778-785.

International conferences

- Zeryouh, M., El Marraki, M., and Essalih, M. (2015). Some tools of qsar/qspr and drug development: Wiener and terminal wiener indices. In 2015 International Conference on Cloud Technologies and Applications (CloudTech), pages 1-4. IEEE.

- Zeryouh, M., El Marraki, M., and Essalih, M. (2016). On the terminal wiener index of networks. In 5th International Conference on Multimedia Computing and Systems (ICMCS), pages 533-536. IEEE.

- Zeryouh, M., El Marraki, M., and Essalih, M. (2017). Studying the structure of some networks using certain topological indices. In International Workshop on Complex Networks and their Applications, pages 543-554. Springer.

- Zeryouh, M., El Marraki, M., and Essalih, M. (2018). A measure for quantifying the topological structure of some networks. In International Conference on Networked Systems, pages 371-381. Springer. [**Best student paper award**]

- Zeryouh, M., El Marraki, M., and Essalih, M. (2018). Quantifying the structure of some networks by using a vertex-cut method. In International Conference on Modern Intelligent Systems Concepts.

National conferences

- Zeryouh, M., El Marraki, M., and Essalih, M. (2014). Terminal Wiener index of composed trees. In the 2014 International Symposium on signal, Image, Video and Communications (ISIVC). Marrakesh.

- Zeryouh, M., El Marraki, M., and Essalih, M. (2014). Terminal Wiener index of Path-Tree. In International Workshop on WIreless Networks and mobile COMmunications (WinCom). Faculty of Science - Rabat.

- Zeryouh, M., El Marraki, M., and Essalih, M. (2015). Les modèles QSAR/QSPR et L'amélioration de calcul des indices topologiques. Les journées URAC du LRIT. l'Institut Scientifique - Rabat.

- Zeryouh, M., El Marraki, M., and Essalih, M. (2015). The Terminal Wiener index of Dendrimers. 4ème Edition des Journées Internationales Jeunes Chercheurs 2015 "JIJC 04" de chimie thérapeutique. Faculté des Sciences, Université Chouaïb Doukkali, El Jadida.

General Introduction

General context

Nowadays, networks are used as a fundamental tool to represent and model any complex system and real world phenomena. For instance, we can cite the chemical structure of molecules, biological systems, interconnection systems, social systems, ecological systems and many other examples from different fields. Therefore, the term network refers to an abstract representation of a system. It consists of a set of vertices, which represent the different elements of the system, and a set of edges between the vertices, which depict the interactions between the elements of this system. For example, the chemical structure of a molecule is a network, where the vertices represent the atoms and the edges are the chemical reactions between these atoms. In mathematics, the study of networks is known as graph theory. In this field of research, we use a mathematical object called graph to model any situation of interest in society. The difference between the terms graph and network is that the first term represents a real world situation as a pure mathematical object, whereas a network consists of some additional information like the attributes related to each vertex and edge. In this thesis, we will use the two terms graph and network interchangeably although they do not mean precisely the same thing.

One of the fundamental questions arising from the study of networks is that of describing the topological properties of the entire structure in quantitative terms. For this reason, a number of quantitative measures, also called topological indices, have been proposed and studied in the literature. To illustrate, the topological indices are numerical values encoding the important structural features related to a network and are invariant under graph isomorphism. These measures can be divided into many classes, some of them are: Distance-based measures, which are based on the shortest paths between pairs of vertices, and Degree-based measures, which are defined in terms of vertex degrees. The concept of topological indices began in 1947, when the physical chemist Harold Wiener used a measure called the Wiener index for predicting the boiling points of paraffin (Wiener, 1947). Later, strong correlation between this index and the chemical properties of a compound was found and then this measure became one of the most frequently used descriptors. Many years after the introduction of the Wiener index and its applications in chemistry, the same quantity has been studied by mathematicians as gross status (Harary, 1959), graph distance (Entringer et al., 1976), transmission (Šoltés, 1991) and simply as the sum of all distances (Yeh and Gutman, 1994). Moreover, The Wiener index was applied in many others situations, such as in theoretical computer science to measure the average performance of a network, crystallography, facility location, and cryptology; see (Alsuwaiyel, 2010), (Quadras et al., 2016) and (Bonchev and Klein, 2002). Due to the success of the Wiener index, a large number of other topological indices have been put forward in the literature (Xu et al., 2014). In this thesis, we will concentrate on distance-based topological indices. Particularly, to the Wiener index and the two most recent indices named the Terminal Wiener index (Gutman et al., 2009a) and the generalized Terminal Wiener index (Ilić and Ilić, 2013). In order to enable the readers to be familiar with the concerned distance-based topological indices, we define the previously cited measures as follows. The Wiener index, denoted by $W(G)$, is extracted from the distance matrix of a network G, such that we calculate the sum of distances between all unordered pairs of vertices of G. In contrast, the Terminal Wiener index, denoted by $TW(G)$, is obtained from a reduced distance matrix of a network G. More precisely, is defined as the sum of distances between all pairs of pendant vertices of G. In addition, the generalization of the Terminal Wiener index or the generalized Terminal Wiener index, denoted by $TW_K(G)$, is also obtained from a reduced distance matrix by considering pairs of vertices of some fixed degree $K \geq 1$.

Since the emergence of distance-based topological indices, the majority of the research have been concentrated on determining the extremal values that a topological index assumes in various classes of graphs, as well as on developing the basic definition of distance-based measures to facilitate the computation when dealing with large and complex networks. Therefore, a lot of methods based on the concept of distance have been proposed for some kind of networks having a large number of vertices. However, in the case of some complicated structures, the application of these methods will be dominated by computing the shortest paths between the corresponding vertices. Hence, the problem remains to develop some new techniques and algorithms that would calculate any distance-based topological index without using the concept of distance. This open problem was posed in (Mohar and Pisanski, 1988), where they introduced an algorithm to find the Wiener index of acyclic networks in linear time and motivated the researchers to work on this problematic for general networks. As a consequence, a new approach called the cut method was proposed by Klavzar et al. (1995) to compute the Wiener index of a class of networks, named partial cubes, without calculating the distance matrix. This method gained the attention of numereous scholars including us, which has led to extend this technique to some other distance-based topological indices and also to find some other extensions to make its application suitable for a wide range of networks.

Objectives and Contributions

The ultimate goal of this thesis is to contribute in the study of two distance-based topological indices, which are the Terminal Wiener index and the generalized Terminal Wiener index. In one hand, we concentrate on some mathematical properties of these indices, and particularly the extremal values that an index assumes in some classes of graphs. On the other hand, we focus on developing some new methods to reduce the computation of the Terminal Wiener index and its generalization in the case of large and complex networks. Our main contributions are summarized as follows:

- **Studying some fundamental properties of the Terminal Wiener index:** In order to extract some of the important features of a topological index, we need to determine its extremal bounds for a given class of graphs. In this direction of research, we discussed the maximal bounds of the Terminal Wiener index in the case of trees and especially the construction of trees that have the second maximal Terminal Wiener index. This work allowed us to determine a transformation that increases the value of the Terminal Wiener index. This first contribution was presented in an international conference (Zeryouh et al., 2016).

- **Developing some methods to facilitate the computation of the Terminal Wiener index:** The direct application of the definition of distance-based topological indices is a difficult task and especially when dealing with networks having a large number of vertices. In this context, we proposed three methods to overcome the problem of calculating the Terminal Wiener index. The first approach is a decomposition method dedicated to some networks constructed from several copies of components, such as Star-Trees and Path-Trees. The second method is based on an important concept called Thorny graph, where the objective is to show that the Terminal Wiener index of a network can be obtained from its parent network. The last method is a rewrite of the basic definition of the Terminal Wiener index. The obtained results of this contribution were published in an international journal (Zeryouh et al., 2015b) and presented in two international conferences (Zeryouh et al., 2015a),(Zeryouh et al., 2018a).

- **Extending the cut method to the case of the generalized Terminal Wiener index:** In the case of large and complex networks, the complexity of the methods that were proposed in the previous contribution is dominated by computing the distances between the corresponding vertices. Therefore, we studied a powerful approach called the cut method. The objective of this method is to cut the associated network into smaller components and assembling the indices of these components to generate the property of the whole structure without using the concept of distance. There exist two versions of the cut method: a method based on edge-partitions and a method based on vertex-partitions. As a result, we arrived to extend and ameliorate these two versions of the cut method for computing the generalized Terminal Wiener index. We note that this contribution was presented in two international conferences

(Zeryouh et al., 2017) and (Zeryouh et al., 2018b).

- **Reducing the computational complexity of the generalized Terminal Wiener index:** Finally, we present as a last contribution an extension of the edge-cut method that reduces the computational complexity of the Terminal wiener index and its generalization. We showed the efficiency of this technique by developing a fast algorithm for computing the generalized Terminal Wiener index of an hierarchical network called the Dendrimer graph. After that, we presented an algorithm that computes the generalized Terminal Wiener index of some complex systems in linear time complexity. This last contribution was presented in an international conference (Zeryouh et al., 2018a) and published in an international journal (Zeryouh et al., 2019).

We note that all the proposed methods were applied on some large and complex networks in order to show the effectiveness of our methods and also to quantify the topological structure of these networks.

Thesis Outline

The present thesis is organized into five chapters as follows:

Chapter 1: ***Basic concepts and definitions on Graph Theory.*** In this chapter, we present necessary background knowledge on graph theory, which are needed for the rest of this document and enable readers to be familiar with this interesting field of research.

Chapter 2: ***The concept of Topological Indices: Their Significance, Applications and Properties.*** In this chapter, we introduce briefly a new branch of graph theory and network science called the Quantitative Graph Theory, which is based on the structural quantification of information contained in networks by using some measures such as the topological indices. Also, we define the concept of topological indices, we give an outline of the existing classes of these indices and their applications in different fields.

Then, we study three distance-based topological indices, which are the Wiener index, the Terminal Wiener index and the generalized Terminal Wiener index. To be specific, for each index we report some extremal values that it assumes in various classes of graphs and especially in the case of trees. Furthermore, we show that the computation of these indices in the case of trees is much easier than that of an arbitrary graph. The main result of this chapter is our first contribution concerning the structure of trees that have the second maximal Terminal Wiener index and a transformation that increases the value of this index.

Chapter 3: ***Some methods for computing the Terminal Wiener Index of certain Networks.*** In this chapter, we show the proposed three methods to overcome the problem of calculating the Terminal Wiener index. In each section, we discuss a method and we apply the obtained results to treat and analyze some important structures, such as binomial trees and Dendrimer trees.

Chapter 4: ***On the Cut Method and its efficiency for quantifying the topological structure of some Complex Networks.*** Here, we investigate the cut method and its versions, which are the edge-cut method and the vertex-cut method. At first, we introduce the first technique and we discuss their two types that depend on the class of networks to which being applied. Then, we show the efficiency of the edge-cut method by analyzing the topological structure of a kind of large networks named Equilateral Triangular Tetra Sheet network and Hex Derived network. After that, we move to the second version based on the vertex-partitions and we bring an efficient formula based on this technique for computing the generalized Terminal Wiener index. Furthermore, we apply the proposed approach to obtain exact analytical expressions of the generalized Terminal Wiener index of some silicate networks, as well as to demonstrate its effectiveness.

Chapter 5: ***Reducing the Computational Complexity of the Generalized Terminal Wiener Index.*** In this last chapter, we reduce the computation of the terminal Wiener index and its generalization using an extension of the edge-cut method that is based on the concept of the canonical metric representation of a network. At first, we prove that the computation of the Terminal Wiener index or the generalized Terminal Wiener index of general networks can be reduced to the problem of computing the Wiener index of the appropriately quotient graphs, which are constructed by using the concept of the canonical metric. After that, we apply the obtained main result on a hierarchical network, called the Dendrimer graph, to demonstrate the efficiency of the method for a hand manipulation as well as to develop a powerful algorithm. After reducing the computation of the generalized Terminal Wiener index for general graphs, we move to a special class of networks called partial cubes and we reduce the computation of their generalized Terminal Wiener index. As a result, we present a linear algorithm for a system that full under partial cubes called the Graphenylene system and we prove its correctness. Finally, we show the ease of the discussed method to quantify the topological structure of some complicated partial cubes, such as the Graphenylene chain network and the Graphenylene sheet network.

To conclude, this dissertation is ended with a concluding chapter that summarizes our contributions and describes some perspectives.

Chapter

1 Basic concepts and definitions on Graph Theory

Contents

Graph theory is an important mathematical tool in different areas of society such as chemistry, physics, social science and computer science. Graph theory deals with the study of graphs in order to analyze and figure out any real world problem. In this chapter, we introduce briefly the historical background of graph theory and we present the basic definitions and concepts of graphs that are really needed for the rest of this manuscript. We refer the reader to the books (West et al., 2001), (Harris et al., 2008), (Balakrishnan and Ranganathan, 2012), (Benjamin et al., 2015) and (Rahman, 2017) which were used in preparing this chapter.

1.1 History Of Graph Theory

The foundation stone of graph theory was laid by Leonhard Euler in 1736 by solving a puzzle known as the "Königsberg Bridge Problem"; see (Euler, 1736) and (Biggs et al., 1976). The problem asks whether there is a continuous walk that crosses all the seven bridges of Königsberg only once, with the additional requirement that the trip ends in the same place it began. In 1736, Leonhard Euler resolved this question in terms of graph theory and proved that it was not possible to walk through the seven bridges exactly one time. He simplified the problem by constructing a mathematical model known as a graph, such that each of the four lands is represented by a node and each of the seven bridges is represented by a link, as illustrated in Figure 1.1. Euler reasoned that anyone standing on a land would have to have a way to get on and off. In other words, each land would need an even number of bridges. Therefore, since the underlying graph of the Königsberg Bridge Problem had four nodes with odd degree, there was no solution to the problem.

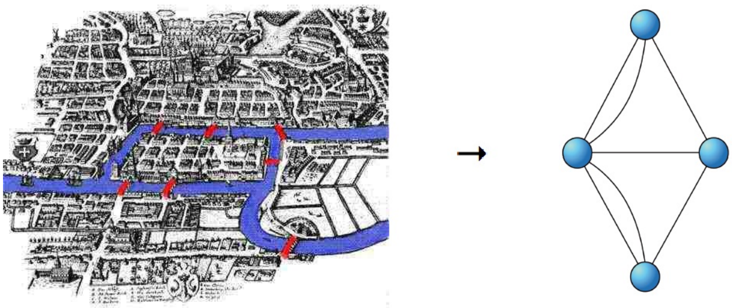


Figure 1.1: The Königsberg bridge problem. On the left a Map of Königsberg in Euler's time showing the actual city layout, and on the right the corresponding graph model.

After the work done by Euler, particularly since the mid-1800s, scientists began to realize that graphs could be used to model many things of interest in society. In 1845, the physicist Kirchhoff (1847) employed graph theory to understand electrical networks or circuit, leading to the well-known kirchhoff's circuit laws. Cayley (1857) studied a particular class of graphs called trees, which had many applications in theoretical chemistry and led to the invention of enumerative graph theory. In 1852, F. Guthrie posed another memorable problem named the Four Color Map. The problem asks if it is true that any map drawn in the plane may have its regions colored with four colors, in such a way that any two regions having a common border have different colors. The four color problem remained unsolved until 1976, when Appel and Haken (1977) used a computer to prove the conjecture. In the twentieth century, the field of graph theory began to blossom and more modeling possibilities were recognized. After that, the application of graph theory to systems of the real world has been pushed much further than to cross bridges in Königsberg.

1.2 Background On Graphs

Many systems of the real world can be schematically described by a collection of points and links between points. The points represent the different constituents of the system and the links depict the interactions between them. For example, if we want to describe the flow of some vehicular movement between geographic locations, the points are the geographic locations and the links represent the routes that exist between geographic locations. In the case of a social relationship, the points represent people and the links are the social interactions such as friendship. The mathematical objects that are required for describing this type of situations are called graphs. In this section, we briefly present some important notions and definitions for graphs that are really needed for the rest of this manuscript.

1.2.1 Definition of a Graph

Formally, a graph G is defined as an ordered pair of disjoint sets $(V(G), E(G))$, where $V(G)$ is the set of vertices and $E(G)$ is the set of edges formed by pairs of vertices. An edge $e = \{u, v\}$ joins the two vertices u and v and is often denoted by uv. The two vertices u and v associated with the edge e are called the end-vertices of e. Moreover, a loop is an edge whose end-vertices are the same and multiple edges are edges with the same pair of end-vertices. We denote by $N = |V(G)|$ the number of vertices of the graph G, also called the order of G, and $M = |E(G)|$ is the number of edges or the size of the graph G.

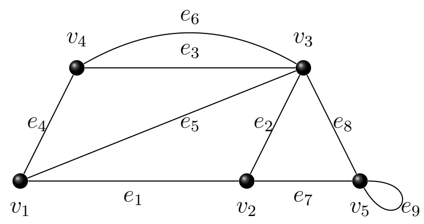


Figure 1.2: An example of a graph G

For example, Figure 1.2 represents a graph G, where each vertex of the set $V(G) = \{v_1, v_2, v_3, v_4, v_5\}$ is represented by a point and each edge of the set $E(G) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\}$ by a line segment or a curve between its two end-vertices. From this instance, we can see that the order and the size of the graph G are equal to $N = 5$ and $M = 9$, respectively. Moreover, the edge e_9 is a loop and $\{e_3, e_6\}$ are multiple edges.
If the graph G does not has any loop or multiple edges, then G is called a simple graph. Otherwise, the graph G is called a multigraph, which is the case in Figure 1.2. We say a graph G is a directed graph or digraph if each edge in $E(G)$ is associated with a direction, such that uv is distinct from vu, for $u, v \in V(G)$.
It may be noted that in this manuscript, we focus primarily on simple undirected graphs.

1.2.2 Adjacency and Vertex Degrees

When dealing with graphs, it is necessary to introduce common terms related to graph connectivity. One of the most basic notions of connectivity is that of adjacency. Two vertices $u, v \in V(G)$ are called adjacent if they are joined by an edge $e \in E(G)$ and the edge e is said to be incident to the vertices u and v. Similarly, two edges $e_1, e_2 \in E(G)$ are adjacent if they share a common end-vertex in $V(G)$. The neighborhood of a vertex $v \in V(G)$, denoted by $\mathcal{N}(v)$, is the set of vertices adjacent to v: $\mathcal{N}(v) = \{u \in V(G) \mid vu \in E(G)\}$. For instance, the graph shown in Figure 1.2, the vertices v_1 and v_3 are adjacent and the edge e_5 is incident to the vertices v_1 and v_3. The edges e_1 and e_2 are adjacent and the neighborhood of the vertex v_2 is $\mathcal{N}(v_2) = \{v_1, v_3, v_5\}$. If all the vertices of a graph G are pairwise adjacent, then G is called a complete graph. A complete graph of order N is denoted by K_N. It is trivial to see that K_N contains $\frac{N(N-1)}{2}$ edges. Figure 1.3 illustrates the first four complete graphs.

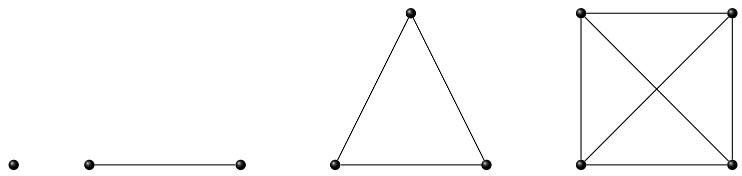


Figure 1.3: The first four complete graphs K_N, with $N = \{1, 2, 3, 4\}$.

The degree of a vertex v in a graph G, denoted by $deg(v)$, is the number of edges incident to v, with each loop at v counted twice. A vertex v with degree 1 is called a pendent vertex. The degree sequence of a graph G of order N is the sequence formed by arranging the vertex degrees in descending order. For the graph in Figure 1.2, the degree of the vertex v_4 is 3 and the degree sequence is $5, 4, 3, 3, 3$.

Since the degree of a vertex counts its incident edges, it is obvious that the summation of the degrees of all vertices in a graph is related to the number of edges in that graph. The following lemma, which is referred to as the "First Theorem of Graph Theory" or the "Handshaking Lemma", indicates that the sum of vertex degrees of a graph is twice the number of its edges.

Lemma 1.2.1. *(Euler, 1736) Let $G = (V(G), E(G))$ be a graph with M edges. Then*

$$\sum_{v \in V(G)} deg(v) = 2M$$

Proof. Every loop edge is counted twice in the degree of its incident vertex in G and every non-loop edge is incident to exactly two distinct vertices of G. Thus, each edge is counted exactly twice. □

The following lemma is a result of the degree-sum formula.

Lemma 1.2.2. *The number of vertices with odd degree in a graph is an even number.*

Proof. Since the degree-sum formula is an even number, it must be that the number of vertices with odd degree is even. □

The maximum degree of a graph G, denoted by $\Delta(G)$, is the maximum value among the degrees of all the vertices of G and mathematically is defined as: $\Delta(G) = max\{deg(v) \mid v \in V(G)\}$. Similarly, the minimum degree of a graph G, denoted by $\delta(G)$, is defined to be: $\delta(G) = min\{deg(v) \mid v \in V(G)\}$. For the graph in Figure 1.2, the maximum and the minimum degrees are $\Delta(G) = 5$ and $\delta(G) = 3$, respectively. If $\Delta(G) = \delta(G) = d$, then all the vertices have the same degree d. This type of graphs is called a regular graph or d-regular graph. Figure 1.4 shows a 2-regular graph known as a cycle graph, a 3-regular graph called a 3-dimensional hypercube and a portion of a 4-regular graph also called a Lattice.

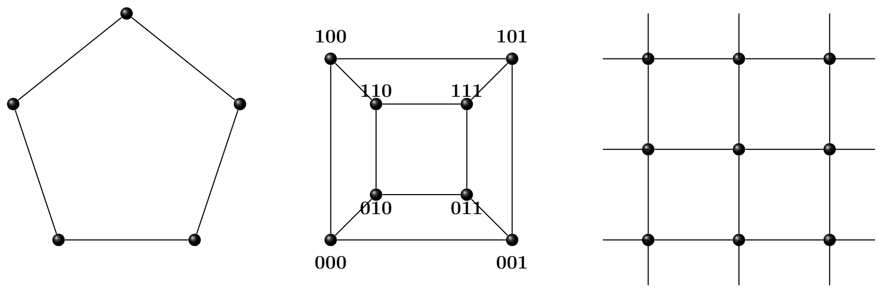


Figure 1.4: Some examples of d-regular graphs, with $d = \{2, 3, 4\}$.

The hypercube Q_d of dimension d is an important example of d-regular graphs. A d-dimensional hypercube has 2^d vertices, where each vertex is labeled by a binary string of length d and we say that two vertices are adjacent if their labeling differ in exactly one position.

1.2.3 Subgraphs

There are many problems in graph theory concerning the structure of graphs that lie within a given graph. These kinds of structures are called subgraphs. A graph H is called a subgraph of a graph G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. A subgraph H of a graph G is a proper subgraph of G if either $V(H) \neq V(G)$ or $E(H) \neq E(G)$. We can obtain subgraphs of a graph G by deleting some vertices and edges of G. We denote by $G - e$ the graph obtained by deleting the edge e from G. Generally, if F is a set of edges of G, we denote by $G - F$ the graph obtained by deleting all the edges in F. Similarly, we denote by $G-v$ the graph obtained by deleting the vertex v and all its incident edges from G. If X is a set of vertices of G, then $G-X$ is the graph obtained by deleting all the vertices in X and all the incident edges from G. See Figure 1.5.

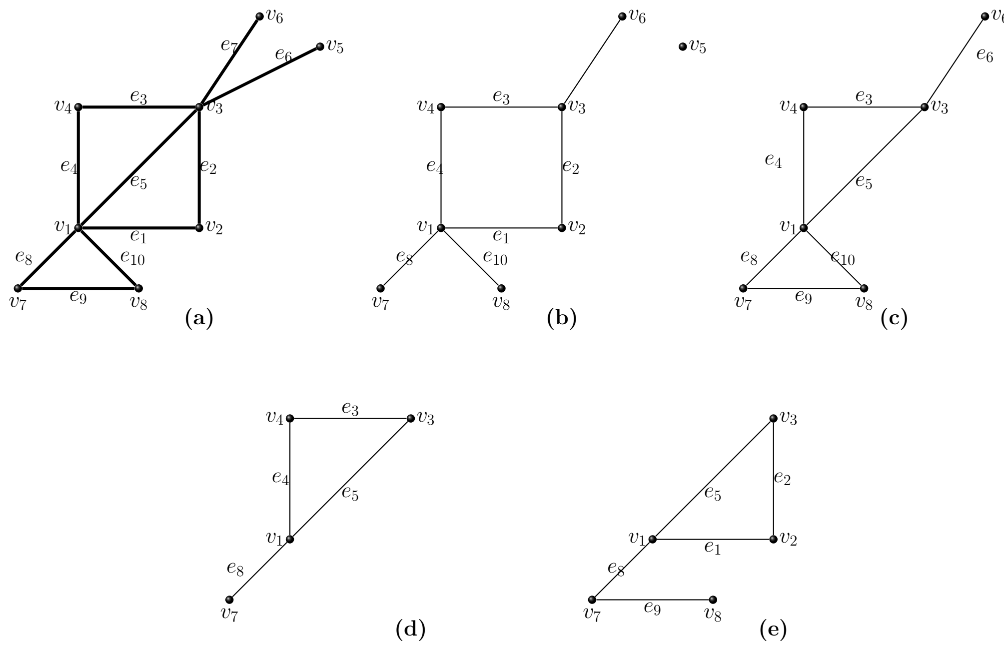


Figure 1.5: A graph G and its various subgraphs. (a): A graph G. (b): A subgraph $G-F$, with $F=\{e_5,e_7,e_9\}$. (c): A subgraph $G-X$, with $X=\{v_2,v_5\}$. (d): a subgraph of G induced by $X=\{v_1,v_3,v_4,v_7\}$. (e): a subgraph of G induced by $F=\{e_1,e_2,e_5,e_8,e_9\}$.

Let G be a graph and let X be a set of vertices of G. A subgraph H of G is a subgraph of G induced by X if $V(H)=X$ and $E(H)$ consists of all the edges e of G with the end-vertices in X. We say that, H is a subgraph of G induced by the set of edges F if $E(H)=F$ and $V(H)$ consists of all those vertices of G that are the end-vertex of some edges in F. See Figure 1.5.

1.2.4 Paths and Connectedness

Let consider the vertices of a graph G as locations and the edges as roads between pairs of locations, then a number of questions can be arise. Is there a way between two locations u and v? How can we get from location u to location v? In this subsection we investigate some concepts that will answer these questions and explain in detail the movement in a graph G.

Let us start at some vertex u of a graph G. If we proceed from u to a neighbor of u and then to a neighbor of that vertex and so on, until we finally come to a stop at a vertex v, then we have just described a walk from u to v in G. More formally, a $u-v$ walk in G is a finite sequence $(v_0 e_1 v_1 ... v_{k-1} e_k v_k)$, whose elements are alternatively the vertices and the edges of G, such that $v_0 = u$ and $v_k = v$. Each vertex v_{i-1} and v_i are the end-vertices of e_i, for $1 \leq i \leq k$. A $u-v$ walk in a graph in which no vertex is repeated is a $u-v$ path and is commonly specified by the sequence $P = \{u = v_0, v_1, ..., v_{k-1}, v_k = v\}$. Therefore, a $u-v$ path in a graph G describes a means to travel between two vertices u and v without repeating vertices along the way. A path P in a graph G for which the beginning and the ending vertices are the same is called a cycle. The length of a walk, a path, or a cycle is its number of edges. Thus, a path of N vertices has length equal to $N-1$, and a cycle of N vertices has length N. In the graph of Figure 1.2, $\{v_1 e_4 v_4 e_6 v_3 e_5 v_1 e_1 v_2 e_7 v_5\}$ is a walk but not a path, the sequence $\{v_1, v_4, v_3, v_2, v_5\}$ is a path and $\{v_1, v_2, v_3, v_4, v_1\}$ is a cycle. The length of these examples of walk, path and cycle are $\{5, 4, 4\}$, respectively.

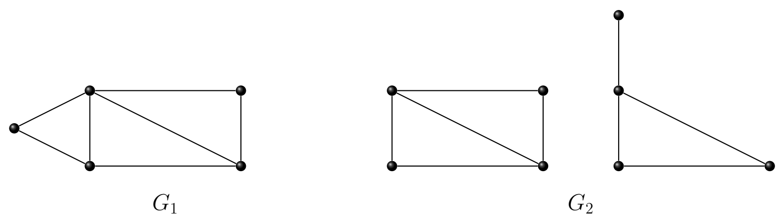


Figure 1.6: Connected and disconnected graphs.

A graph G is connected if there is a path between each pair of vertices in G. Otherwise, G is called a disconnected graph. The graph G_1 in Figure 1.6 is a connected graph, while the graph G_2 is disconnected. A disconnected graph contains some connected pieces called components. For example, in Figure 1.6 the graph G_2 contains 2 components. Formally, a component of a graph G is a connected subgraph of G that is not a proper subgraph of any other connected subgraph of G.

1.2.5 Distance in Graphs

Many questions involving distance have been arisen. What is the distance between two cities or two locations in general? Where is an ideal location to build an emergency facility? We can answer to these questions by considering distance in graphs. In this subsection we define the concept of distance and we study some distance parameters.

In the previous subsection 1.2.4, we mentioned that in a connected graph G there is at least one path between each pair of vertices. Actually, there may be several paths between two vertices in a graph G. For example, the graph in Figure 1.2 contains many paths between the two vertices v_1 and v_5, among them:

$$P = \{v_1, v_2, v_5\}$$
$$P = \{v_1, v_2, v_3, v_5\}$$
$$P = \{v_1, v_4, v_3, v_2, v_5\}$$

The first path between v_1 and v_5 has length 2, the second has length 3 and the last path has length 4. Therefore, the smallest length between $v_1 - v_5$ path is 2.
In a connected graph G, the distance from vertex u to vertex v is the length of a shortest $u - v$ path in G. We denote the distance by $d_G(u, v)$ or simply by $d(u, v)$. Consequently, the distance between v_1 and v_5 in the graph of Figure 1.2 is $d(v_1, v_5) = 2$. If there is no path from u to v, then $d(u, v) = \infty$. We assume that G is a connected graph, then the concept of distance satisfies the following properties.

Proposition 1.2.1. *For any connected graph G, then:*

1. $d(u, v) \geq 0, \quad$ *for all* $\{u, v\} \in V(G)$.

2. $d(u, v) = 0, \quad$ *if and only if* $u = v$.

3. $d(u, v) = d(v, u), \quad$ *for all* $\{u, v\} \in V(G)$.

4. $d(u, w) \leq d(u, v) + d(v, w), \quad$ *for all* $\{u, v, w\} \in V(G)$.

The distances between all pair of vertices in a graph G are described by a distance matrix D. Let l be the length of the shortest path between two vertices i and j, then the distance matrix of a graph G is the $n \times n$ matrix $D = d(i, j)$, where

$$d(i,j) = \begin{cases} l & \text{if } i \neq j, \\ 0 & \text{otherwise} . \end{cases}$$

The following matrix illustrates the distance matrix of the graph in Figure 1.2.

$$D = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 \\ 1 & 0 & 1 & 2 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 2 & 1 & 0 & 2 \\ 2 & 1 & 1 & 2 & 0 \end{bmatrix}$$

The definition of distance is simple and in the same time is a useful notion that has led to the determination of several graph parameters, such as the eccentricity, the diameter and the radius.

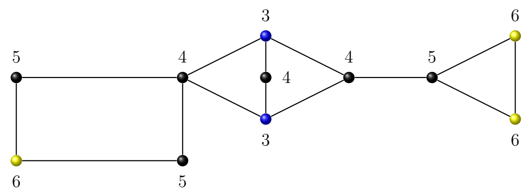


Figure 1.7: A graph G labeled by its eccentricity.

For a given vertex u in a connected graph G, the eccentricity of u, denoted by $ecc(u)$, is the greatest distance from u to any other vertex. More formally, $ecc(u) = max\{d(u, x) : x \in V(G)\}$. The maximum eccentricity among the vertices of G is called the diameter of G, which is denoted by $diam(G)$ or $D(G)$, and the minimum eccentricity is the radius of G and denoted by $rad(G)$. The center of the graph G is the set of vertices u such that $ecc(u) = rad(G)$. The periphery of G is the set of vertices u such that $ecc(u) = diam(G)$. For instance, the graph G in Figure 1.7 is labeled by its eccentricity. In this graph, the largest value of eccentricity is 6, then $diam(G) = 6$, and the smallest value is 3, which means $rad(G) = 3$. Moreover, the center and the periphery of G are shown with blue and yellow vertices, respectively. Later, we shall discuss and focus on some other graph parameters called the topological indices that are described in terms of distance.

1.2.6 Isomorphic Graphs

A graph can exist in different forms having the same number of vertices, edges, and also the same edge connectivity. Such graphs are called isomorphic graphs. Two graphs G_1 and G_2 are said to be isomorphic if there exist a bijective function $f : V(G_1) \to V(G_2)$, such that any two vertices u and v of G_1 are adjacent in G_1 if and only if $f(u)$ and $f(v)$ are adjacent in G_2. The mapping f is also called an isomorphism. When two graphs G_1 and G_2 are isomorphic, we say that $G_1 = G_2$ or that G_1 is G_2. For instance, Figure 1.8 shows two graphs that are isomorphic. The isomorphism could be described as follows: $\{(v_1, u_4), (v_2, u_3), (v_3, u_2), (v_4, u_6), (v_5, u_5), (v_6, u_1)\}$.

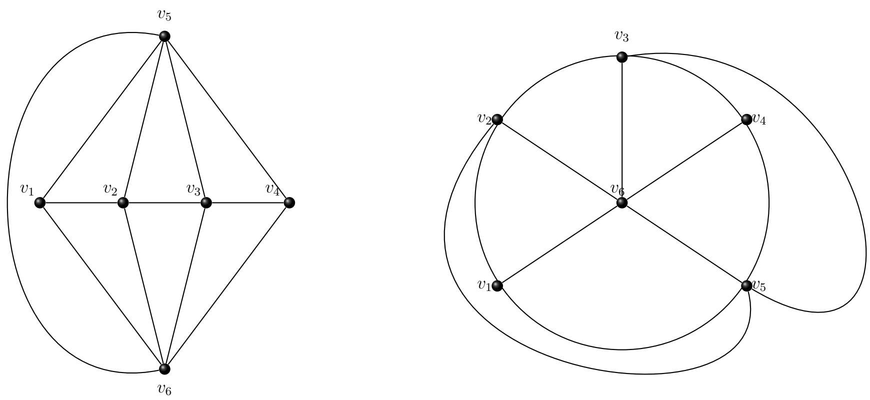


Figure 1.8: Isomorphic graphs.

Isomorphic graphs clearly have the same number of edges and vertices. On the other hand, the converse of this statement is not true. Another fact about isomorphic graphs is that if G_1 and G_2 are isomorphic, then the degree sequences must be identical and the converse of this statement is not true.

In general, determining whether two graphs are isomorphic is a difficult problem. A trivial approach would take all the possible permutations of the vertices to check whether any of these permutations induces an isomorphism. Clearly, this approach takes exponential time on the number of vertices.

1.2.7 Families of graphs

We have already defined some types of graphs, such as directed graphs, complete graphs and regular graphs. Here, we present some other families of graphs that are useful in graph theory.

We previously used the term of path to describe a kind of walks between two vertices u and v in a graph G. A path is also a type of graphs, whose vertices can be arranged in a linear sequence $v_1, v_2, ...v_N$ and in such a way that two vertices are adjacent if they are consecutive in the sequence. The degree of each vertex of the path graph is two except the two end-vertices v_1 and v_N, which have degree one. If we join the two end-vertices of a path graph we obtain a cyclic sequence that define an other kind of graphs called a cycle graph. The degree of each vertex of a cycle graph is two. A cycle graph of order $N \geq 3$ is often denoted by C_N, while a path graph of order N is denoted by P_N. Figure 1.9 depicts a path P_6 and a cycle graph C_{12}.

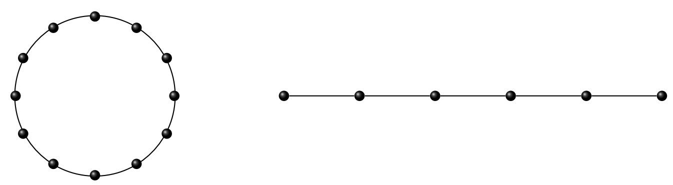


Figure 1.9: A Cycle graph of order 12 and a Path graph of order 6.

A bipartite graph is a graph $G = (V(G), E(G))$ such that the vertex set $V(G)$ may be partitioned into two disjoint sets V_1 and V_2. The two sets V_1 and V_2 are often called the partite sets of G. Each edge of a bipartite graph G must joins exactly one vertex of V_1 to exactly one vertex of V_2. For example, the two graphs in Figure 1.10 are bipartite.

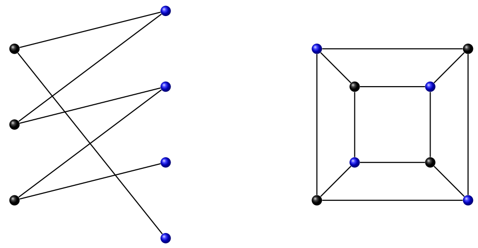


Figure 1.10: Two bipartite graphs.

Let G be a graph with a weight function ω that assigns positive real numbers to the vertices or the edges of G. Then, the graph G together with this weight function is called a weighted graph and denoted by (G, ω). In Figure 1.11, we show an example of two weighted graphs, such that we assigned weights to the edges and the vertices of the left graph and the right graph, respectively. The meaning of weights in a weighted graph depends on different applications. For example, in a communication network each weight represent the cost of transmitting data along a link and in a transportation network the weights may represent the traffic flow and so on.

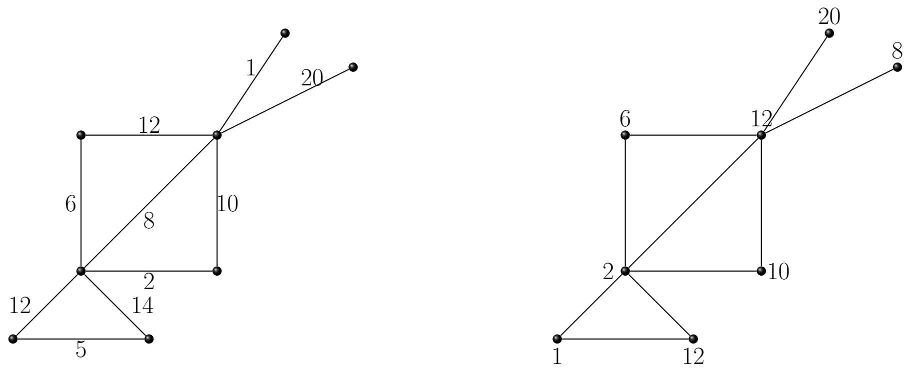


Figure 1.11: An example of weighted graphs

Another classes of graphs are called isometric subgraphs and convex subgraphs, which are defined as follows. let H be a subgraph of a graph G, if $d_H(u, v) = d_G(u, v)$ for all $u, v \in V(H)$, we say that H is an isometric subgraph. All isometric subgraphs are induced, but the converse is false. For example, the subgraph H (e) in Figure 1.5 is induced but not isometric since $d_G(v_1, v_8) \neq d_H(v_1, v_8)$. If every shortest G-path between vertices of a subgraph H lies entirely in H, we say that the subgraph H of G is convex. In Figure 1.11, the triangle at the bottom is a convex subgraph.

As a last example, we mention the class of planar graphs. A graph G is said to be planar if it can be drawn in the plane so that no two of its edges cross. Such drawing of a planar graph G is called a plane representation of G. For instance, the graph in Figure 1.11 is planar. A plane graph G partitions the plane into a number of connected regions called the faces of G. As an example, the planar graph shown in Figure 1.11 has four faces. Each plane graph has exactly one unbounded face called the outer face. One of the significant puzzles about planarity is the Three Houses and Three Utilities Problem in which there are three utilities: gas, water and electricity that need to be connected to three houses by gas lines, water mains and electrical lines. The problem asks whether there is a way to make all nine connections without any of the lines or mains crossing each other. The situation described in this problem can be modeled by the complete bipartite graph $K_{3,3}$; see Figure 1.12. The $K_{3,3}$ graph is a non-planar graph since it cannot be drawn in the plane without any of its edges crossing. From this fact, the solution of the Three houses and Three Utilities Problem is impossible.

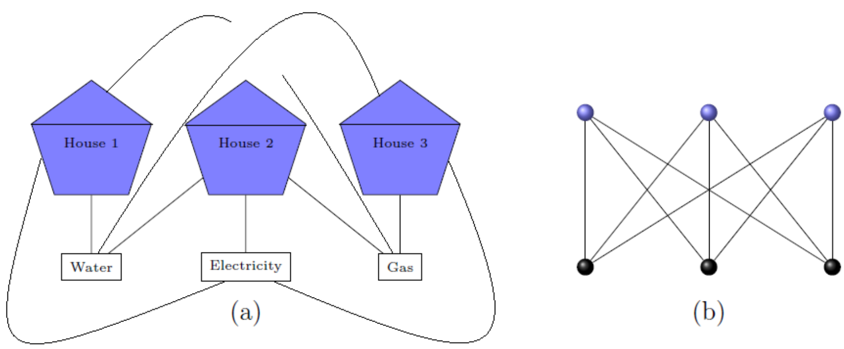


Figure 1.12: (a): The Three Houses and Three Utilities Problem. (b): The complete bipartite graph $K_{3,3}$.

There is a simple formula relating the numbers of vertices, edges and faces in a connected plane graph. This formula is called Euler's Formula.

Theorem 1.2.1. *Let G be a connected plane graph with N vertices, M edges and f faces. Then:*

$$N - M + f = 2.$$

Proof. By induction on the number of edges in the graph G, the result is obvious for $M = 0$ or 1.

Assume that $M \geq 2$ and the formula works for all connected plane graphs with no more than M edges. Let G be a tree (see the definition in section 1.5). Then, G has a vertex v of degree one. The connected plane graph $G - v$ has $N - 1$ vertices, $M - 1$ edges and $f = 1$ faces. So, by the inductive hypothesis, $(N - 1) - (M - 1) + f = 2$.

Let consider now the case when G is not a tree and e an edge on a cycle. Then, the connected plane graph $G - e$ has N vertices, $M - 1$ edges and $f - 1$ faces. So that, the formula is true. □

1.3 Representations Of Graphs

As we know, a graph can be schematically described by a diagram, where each vertex is represented by a point and each edge is represented by a line segment or a curve between its two end-vertices. This kind of representation is convenient for the visualization of a graph, yet it is unsuitable if we want to store a graph in a computer. For this purpose, matrix representation of a graph is needed. In this section, we show the widely used matrices for representing a graph in a computer, which are the adjacency matrix, the adjacency list and the incidence matrix.

1.3.1 Adjacency Matrix

Let G be a graph with the vertex set $V(G) = \{v_1, v_2, ..., v_N\}$ and the edge set $E(G) = \{e_1, e_2, ..., e_M\}$. The adjacency matrix of G is the $N \times N$ matrix $A = [a_{ij}]$, where

$$a_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is adjacent to } v_j \text{ in } G, \\ 0 & \text{otherwise .} \end{cases}$$

For example, the bellow matrix illustrates the adjacency matrix representation of the graph in Figure 1.13. It is easily to see that the adjacency matrices for simple graphs have all zeros on the main diagonal and are symmetric, while the adjacency matrix of a directed graph is usually not symmetric. Furthermore, from the adjacency matrix we can easily calculate the degree of a vertex v_i by the expression $\sum_{j=1}^{N} a_{ij}$.

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

An adjacency matrix uses $O(N^2)$ space to represent a graph of N vertices. From this fact, this kind of matrices is not suitable for performance reasons, especially when the number of edges in the graph is much less than the maximum possible $\frac{N(N-1)}{2}$. Actually, adjacency matrices are practical for small/medium graphs and for dense graphs. We refer to see (Cormen et al., 2009) for more details.

1.3.2 Adjacency List

Let G be a graph with the vertex set $V(G) = \{v_1, v_2, ..., v_N\}$ and the edge set $E(G) = \{e_1, e_2, ..., e_M\}$. The adjacency list *Adj* of G is an array of N lists, where for each v of G the adjacency list $Adj[v]$ contains all the vertices adjacent to v in G. The vertices in each list are stored in an arbitrary order. Figure 1.13 illustrates a simple graph with its adjacency list representation.

The space requirement for the adjacency list is $O(N+M)$. Thus, this representation is much more economical than the adjacency matrix. The only disadvantage of the adjacency list is that there is no quicker way to determine if a given edge uv exists or does not exist in the graph.

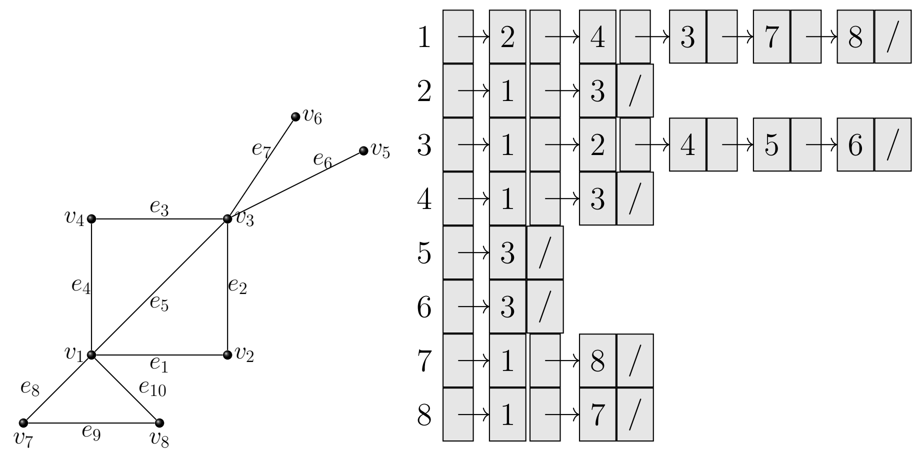


Figure 1.13: A simple graph and its adjacency list representation

1.3.3 Incidence Matrix

Let G be a graph with the vertex set $V(G) = \{v_1, v_2, ..., v_N\}$ and the edge set $E(G) = \{e_1, e_2, ..., e_M\}$. The incidence matrix of G is an $N \times M$ matrix $I = [b_{ij}]$ defined by the following conditions:

$$b_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is adjacent to } e_j, \\ 0 & \text{otherwise .} \end{cases}$$

The following matrix illustrates the incidence matrix of the graph in Figure 1.13. From the incidence matrix, we can extract some useful observations. First, since every edge is incident on exactly two vertices, each column of the incidence matrix has exactly two one's. Moreover, the number of one's in each row of the matrix equals the degree of the corresponding vertex.

$$I = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

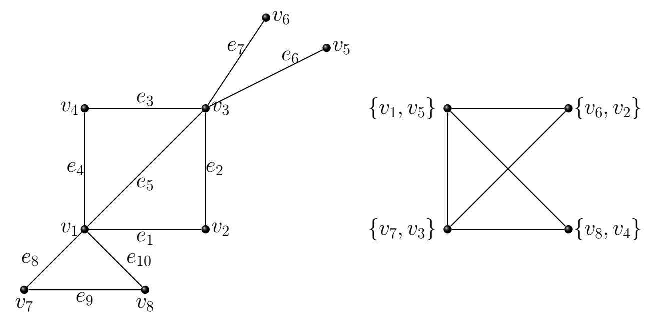
The space requirement for the incidence matrix is $O(N \times M)$. For a graph that contains much more number of edges compared to the number of vertices this space requirement is much higher than the adjacency matrix.

1.4 Operations On Graphs

A new graph can be obtained from an initial ones by applying some graph operations. We previously defined a graph operation in the subsection 1.2.3, which is based on deleting vertices or edges of a graph G. In this section, we discuss two other operations, which are the quotient graphs and the Cartesian product.

1.4.1 Quotient Graphs

The quotient graph of a graph G is a graph whose vertices are blocks of a partition of the vertices of G, with block A is adjacent to block B if some vertex of A is adjacent to a vertex of B. More formally, let $\sim$ be an equivalence relation on $V(G)$. The quotient graph of G with respect to $\sim$, denoted by $G/\sim$, is a graph whose vertex set is $V/\sim$ and edge set is $\{([u],[v]) \mid uv \in E(G)\}$. For example, let take the graph G shown in Figure 1.14(a) and let $\sim$ be a relation on $V(G)$ defined by: $v_i \sim v_j \Leftrightarrow 4 \mid (i-j)$, the quotient graph $G/\sim$ of G is demonstrated in Figure 1.14(b).



(a) (b)

Figure 1.14: A graph G and its quotient graph $G/\sim$.

1.4.2 Cartesian Product

The cartesian product of two graphs G and H is a graph, denoted as $G\Box H$, whose vertex set is $V(G) \times V(H)$. Two vertices (u,u') and (v,v') are adjacent in $V(G) \times V(H)$ if and only if either $u = v$ and $u'v' \in E(H)$ or $u' = v'$ and $uv \in E(G)$. Figure 1.15 shows the cartesian product $P_3 \square K_2$. There is an informal way for drawing the product $G \square H$, such that we replace each vertex v of G by a copy H_v of the graph H. If u and v are adjacent in G, then we join the corresponding vertices of H_u and H_v by an edge. Otherwise, we add no edges between H_u and H_v.

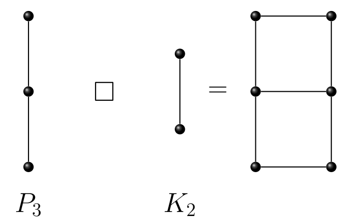


Figure 1.15: The cartesian product $P_3 \square K_2$.

1.5 Trees

Trees are considered as one of the most important classes of graphs that we saw during this chapter. Their significance has grown in view of their applications in various fields, especially in theoretical computer science and theoretical chemistry. In this section, we give the definition of trees and we look at some of their basic structural properties.

1.5.1 Definition of Trees and their Properties

A tree is a connected graph that contains no cycles. When dealing with trees, we often denote a tree by T rather than G. In a tree, an internal vertex is a vertex of degree at least two and a leaf, also called a pendent vertex, is a vertex of degree one. A collection of trees is called a forest. In other words, a forest is a graph with no cycle. Such a graph is also called an acyclic graph. Figure 1.16 illustrates a forest where each component is a tree of order 5.

There are many different properties of trees that can be taken as a definition of a tree. In the following theorem, we show some of these properties.

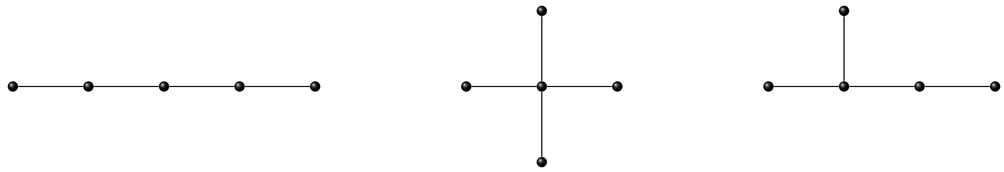


Figure 1.16: Trees on five vertices.

Theorem 1.5.1. *If T is a tree of order N. Then the following statements are equivalent.*

1. *T is a tree.*
2. *T has no cycles and has $N - 1$ edges.*
3. *Any two vertices are connected by exactly one path.*
4. *Every tree with two or more vertices has at least two leaves.*
5. *Removing any edge from T gives a graph $T - e$ which is not connected.*
6. *Adding a new edge to T gives a graph $T + e$ that has exactly one cycle.*

1.5.2 Rooted Trees

In many applications of trees, a particular vertex is designed as the root. In this case, the trees are called rooted trees. A rooted tree is a tree T in which only one vertex is stated as a root $r \in V(T)$ and every edge is directed away from the root. In other words, a rooted tree may be described in terms of generations or levels. The root is the 0-th generation and is drawn at the top, while the neighbors of the root form the first generation. In general, a vertex of distance k from the root form the k-th generation and the maximum level of all the vertices is called the height of the rooted tree. If a rooted tree is regarded as a directed graph in which each edge is directed from top to bottom, then every vertex u other than the root is connected by an edge from some other vertex p that is called the parent of u and in the same time u is considered as a child of the vertex p. In Figure 1.17 a rooted tree is shown, v_1 is the parent of v_4 and v_5, while v_8 is the child of v_5.

Afterward, we focus on some rooted trees in order to study their structural properties.

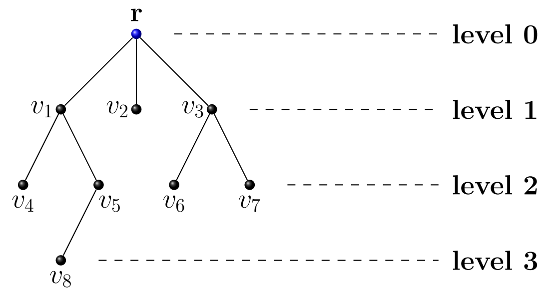


Figure 1.17: An example of a rooted tree.

1.6 Summary

Many mathematical puzzles and systems of the real world can be modeled and analyzed by using graphs. This chapter included a brief introduction to graph theory. At first, we presented a short history of this theory and some of the major concepts of graphs as the connectedness, the distance and some well-known families of graphs. Then, We showed the widely used matrices for representing a graph in a computer and some graph operations to generate new graphs from initial ones. Finally, we closed the chapter by presenting the class of trees and their properties.

In the next chapter, we study some graph measures called the topological indices that are extensively used to quantify and understand the structural information of networks and complex networks. To clarify, the topological indices will be used to analyze the topological structure of networks quantitatively.

Chapter

2 The concept of Topological Indices: Their Significance, Applications and Properties

Contents

The study of networks and particularly complex networks is a research topic with ongoing interest. One of the most fundamental questions arising from this area of research is to find suitable measures and use them to describe the topological properties of a structure in quantitative terms. Once such measures have been established, they can serve as a standard for exploring and analyzing networks or for comparing and designing new networks. For this reason, a large number of quantitative measures have been suggested as descriptors for networks and employed in several applications such as describing the chemical structure of molecules, the chemical reaction networks, ecosystems, the World Wide Web, social networks, etc. In chemical and biological applications these measures are referred to as topological indices. In this chapter, we start with an introduction of a new branch of graph theory and network sciences called the Quantitative Graph Theory (Dehmer et al., 2014), which relates to quantify the structural information of networks by using some measurements, such as the topological indices, as well as with an outline of the existing classes of topological indices and their applications. Then, we focus our attention to the oldest and the most well-known topological index called the Wiener index. We present the historical background of this index, the basic definition, and some of its main properties. The third section is dedicated to the Terminal Wiener index and its fundamental properties. In the same part, we present our first contribution concerning the construction of trees with the second maximal Terminal Wiener index and a transformation that increases the value of this measure. We note that this work was presented in an international conference (Zeryouh et al., 2016). In the last section, we introduce the generalized Terminal Wiener index and we report its extremal properties.

2.1 Quantitative Graph Theory And Graph Measures

Quantitative Graph Theory is considered as a modern branch of graph theory and network sciences (Dehmer et al., 2014). This graph-theoretical branch is based on the structural quantification of information contained in networks and particularly in complex networks by employing a measurement approach based on numerical invariants. In fact, quantitative approaches are compulsory for analyzing networks and have been examined from different perspectives in a variety of disciplines including computational biology, structural chemistry, medicinal chemistry and social network analysis; see (Junker and Schreiber, 2011), (Devillers and Balaban, 2000), (Wasserman and Faust, 1994) .
According to Dehmer et al. (2014), Quantitative Graph Theory can be divided into two main categories, namely, comparative graph analysis and graph characterization by using numerical graph invariants. In the following, we concentrate on the category of graph characterization. For more details on the first category and the Quantitative Graph Theory in general see the book (Dehmer et al., 2014).

2.1.1 Graph Characterization and Topological Indices

Graph characterization concerns the determination of graph complexity by using numerical graph invariants. The graph invariants are numerical values that play an important role in characterizing the structural features related to a network and are invariant under graph isomorphism. Such invariants can be classified in accordance with the type of information that we need to compute. We show here two classes of graph invariants, such that the first class quantifies local properties of graphs and the second class quantifies the topological information of the entire graph.

- **Local measures:** these graph invariants employ information from the vertex itself and can also use the topological information from the direct and the indirect neighborhoods of a vertex. For examples, the degree of a vertex, vertex eccentricities and vertex centrality (Harary, 1959).

- **Global measures:** this class of measures uses the entire network structure to be computed. For instance, distance-based topological indices (Xu et al., 2014), which are the focus of this thesis.

Many of global invariants have been proposed and described in the literature. In chemical and biological applications these measures are referred to as topological indices (Hosoya, 1971). The topological indices play a fundamental role to characterize molecular structures according to their size, degree of branching and overall shape (Todeschini and Consonni, 2008). These topological measures can be divided into the following classes.

- **Distance-based measures:** which are based on the shortest paths between pairs of vertices. In other words, this class can be derived from the distance matrix or some closely related distance-based matrices. Some of the most studied distance-based topological indices are the Wiener index (Wiener, 1947), the Hyper Wiener index (Randić, 1993), the Wiener polarity index (Wiener, 1947), the Terminal Wiener index (Gutman et al., 2009a) and the generalized Terminal Wiener index (Ilić and Ilić, 2013). Afterward, we discuss in details certain of these indices.

- **Degree-based measures:** which are defined in terms of vertex degrees. Some of these well-known measures are the Randic index (Randic, 1975), and the Zagreb index (Gutman and Trinajstić, 1972).

- **Eigenvalue-based measures:** are based on eigenvalues inferred from the adjacency matrix or the Laplacian matrix. Examples include Estrada index (Estrada, 2002), spectral radius (Von Collatz and Sinogowitz, 1957) and graph energy (Gutman et al., 2009b).

- **Information-theoretic measures:** are related to entropies of discrete probability measures on different components of a graph, such as Shannon's entropy (Dehmer and Mowshowitz, 2011).

There exist some other topological indices which can be assigned to two different classes since they use different information. For example, the degree distance index (Dobrynin and Kochetova, 1994) and the geometric-arithmetic indices (Fath-Tabar et al., 2010). Toward an overview on the topological indices, we refer to see Todeschini and Consonni (2008).

2.1.2 Applicability of Topological Measures

Topological indices have gained a lot of interest in a large variety of disciplines besides chemistry, such as biology, image analysis, linguistics, and so on. In this section, we outline a general overview about the existing applications in these fields of research.

Topological measures have been extensively used in chemistry and particularly in cheminformatics for the development of quantitative structure-activity relationships (QSARs) and quantitative structure-property relationships (QSPRs) (Devillers and Balaban, 2000). The QSAR and QSPR are mathematical models that attempt to correlate the chemical structure of a compound to its biological or physicochemical activity. The construction process of QSAR/QSPR model comprises of two main steps which are: the description of molecular structure by using the topological indices and the application of multivariate analysis for correlating the topological indices with the observed activities or properties. A schematic representation of QSAR/QSPR process is illustrated in Figure 2.1, and for more additional references see (Nantasenamat et al., 2009) and (Dehmer et al., 2012).

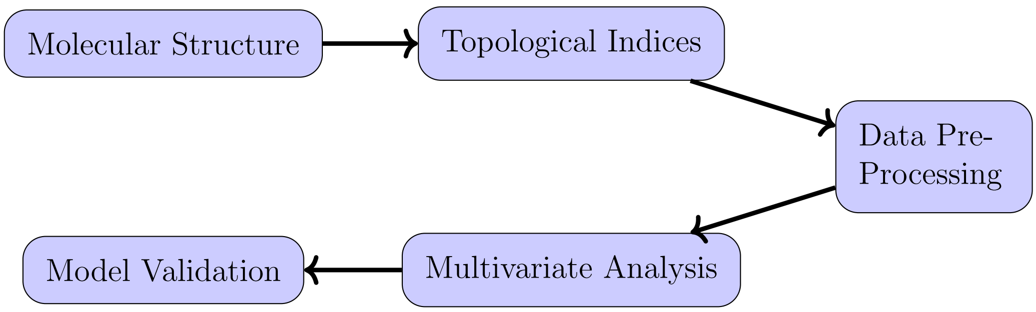


Figure 2.1: General process of developing a QSAR/QSPR model.

In computational biology, the structural graph measures have been employed in a variety of applications. For example, they have been used for extracting the characteristics of protein-protein interaction (PPI) networks and to study anatomical networks (Estrada, 2012). Outside chemistry and biology, the topological measures have been applied in the context of image analysis. Welk et al. (2014) presented the first exploration of an application of quantitative graph theory methods in discrimination of image textures. As a result, they proved the usefulness of selected topological descriptors in texture analysis.

Moreover, graph measures got an attention from computational linguistics. In this area of research, Abramov and Lokot (2011) investigated some topological indices in order to capture the linguistic variation among languages. They also motivated the use of information theoretic measures to study the properties of morphological derivation networks.

2.2 Wiener Index Of Graphs

During the recent years, a large amount of distance-based measures have been considered in the literature. In this section, we focus our attention to the most well-known topological index called the Wiener index. At first, we present the historical background of Wiener index and some of its main properties. Then, we expose some extensions of the Wiener index and its generalization.

2.2.1 Discovery and History of Wiener Index

The theory of topological indices began in 1947, when the physical chemist Harold Wiener used the Wiener index for predicting the boiling points of paraffin (Wiener, 1947). He mentioned in his article that the boiling point t_B of alkanes can be quite closely approximated by the following formula:

$$t_B = aW + bp + c,$$

where a, b and c are constants for a given isomeric group, W is the Wiener index, which equals to the sum of distances between all unordered pairs of vertices in the molecular graph, and p is the polarity number, which equals to the number of unordered vertex pairs at distance 3 in the molecular graph.

At first, the Wiener index did not attract the attention of the chemical community, but later, strong correlation between the Wiener index and the chemical properties of a compound was found and then this measure became one of the most frequently used molecular structure descriptors. It has found numerous applications for the development of quantitative structure-activity (structure-property) relationships (QSARs/QSPRs) (Devillers and Balaban, 2000) and to predict physical parameters, such as heats of vaporization, molar volumes and molar refractivity indices of alkanes (Klavzar and Gutman, 1997), (Quadras et al., 2016). Furthermore, the Wiener index has been studied for a number of chemical graphs, such as hexagonal systems (Dobrynin et al., 2002), phenylenes (Pavlović and Gutman, 1997), nanotubes (Ashrafi and Diudea, 2016), and so on. Many years after the introduction of the Wiener index and its applications in chemistry, the same quantity has been studied by mathematicians as gross status (Harary, 1959), graph distance (Entringer et al., 1976), transmission (Šoltés, 1991), and simply sum of all distances (Yeh and Gutman, 1994). In theoretical computer science, the quantity Wiener index is considered as one of the basic descriptors of fixed interconnection networks because it provides the average distance between any pairs of nodes of the network and consequently it is an effective measure of the average performance of a network (Alsuwaiyel, 2010), (Quadras et al., 2016). The Wiener index also was applied in many others situations such as crystallography, facility location, cryptology, etc (Bonchev, 2002). Due to the success of the Wiener index, a large number of other topological indices have been put forward in the literature (Xu et al., 2014).

2.2.2 Definition and Main Properties of Wiener Index

The Wiener index of an undirected connected graph G, denoted by $W(G)$, is defined as the sum of distances between all unordered pairs of vertices of G.

$$W(G) = \sum_{\{u,v\} \subseteq V(G)} d(u, v). \tag{2.1}$$

The Wiener index of a vertex $v \in V(G)$, also called farness or vertex transmission, is defined as the sum of distances between a chosen vertex v and the other vertices in G.

$$W(v) = \sum_{u \in V(G)} d(v, u). \tag{2.2}$$

The Wiener index is closely related to some quantities. For example, to the betweenness centrality of a vertex, which quantifies the number of times a vertex lies on the shortest path between two other vertices. This measure is widely used in the theory of social networks to quantify the ability of a person to have control over the communication between other people in a social network (Freeman, 1977). The betweenness centrality of a vertex $x \in V(G)$ can be computed as follows:

$$B(x) = \sum_{\substack{u,v \in V(G) \\ u \neq v \neq x}} \frac{\sigma_{u,v}(x)}{\sigma_{u,v}},$$

where, $\sigma_{u,v}$ denotes the total number of shortest $u - v$ paths in G and $\sigma_{u,v}(x)$ represents the number of shortest $u - v$ paths passing through the vertex x. Moreover, the sum of the betweenness centrality of all vertices of a graph G equals two times its Wiener index. The Wiener index is also related to the average distance $\mu(G)$ between the vertices of a graph G, which is given by:

$$\mu(G) = \frac{2}{N(N-1)} W(G).$$

This quantity is so much investigated in computer science to know the average distance traversed by a message in a network (Balakrishnan et al., 2008). Also, it was used to determine the class of small-world networks (Watts and Strogatz, 1998).

Mathematical properties of the Wiener index have been intensively studied over the past 55 years. One of these properties is the extremal values that the Wiener index assumes in various classes. By determining the lower and upper bounds of a given class of graphs, we can establish several important features of the underlying graph invariant. Here, we mention some basic bounds for the Wiener index that have been proved in many works; see (Entringer et al., 1976), (Gutman, 1997), (Schmuck, 2010) and (Essalih, 2013). It is well known that for trees on N vertices, the maximum Wiener index is obtained for the path P_N and the closed combinatorial expression for the Wiener index of P_N is:

$$W(P_N) = \binom{N+1}{3}. \tag{2.3}$$

On the other side, the tree with minimal Wiener index is the star S_N and it is known that:

$$W(S_N) = (N-1)^2. \tag{2.4}$$

Thus, for every tree T on N vertices, we have

$$(N-1)^2 \leq W(T) \leq \binom{N+1}{3}.$$

Among all simple and planar graphs with N vertices, the maximal planar graph ξ_N has the smallest Wiener index. To clarify, a planar graph G is called maximal if adding a new edge in G between any two non adjacent vertices gives a non planar graph. For example, the two graphs in Figure 1.8 belong to the class of maximal planar graphs. It was mentioned in (Essalih, 2013) that the Wiener index of ξ_N for $N \geq 3$ is:

$$W(\xi_N) = N^2 - 4N + 6. \tag{2.5}$$

Therefore, in the case of planar graphs, the extremal bounds of the Wiener index are:

$$N^2 - 4N + 6 \leq W(G) \leq \binom{N+1}{3}. \tag{2.6}$$

Concerning the case of general graphs on N vertices, the complete graph K_N has the smallest Wiener index, since the distance between any two distinct vertices is one. The Wiener index of the complete graph K_N is:

$$W(K_N) = \binom{N}{2}. \tag{2.7}$$

Thus, for any connected graph G of order N, the extremal bounds are:

$$\binom{N}{2} \leq W(G) \leq \binom{N+1}{3}. \tag{2.8}$$

Let G be a 2-connected graph of order N, such that for every vertex $v \in V(G)$ the graph $G - v$ is a connected graph. For this class of graphs, it was proved that the cycle C_N has the largest Wiener index and we have:

$$W(C_N) = \begin{cases} \frac{N^3}{8} & \text{if } N \text{ is even }, \\ \frac{N^3-N}{8} & \text{if } N \text{ is odd }. \end{cases} \tag{2.9}$$

Another fundamental property of the Wiener index is that its computation in the case of a tree is much easier than that of an arbitrary graph. An interesting formula can be found in Wiener's first paper (Wiener, 1947), in which the author shown that the Wiener index of a tree can be decomposed into easily calculable edge-contributions. In other words, this formula counts how often one has to pass each edge. Let $e = uv \in E(T)$ be an edge of a tree T. The forest $T - e$ is composed of two components or subtrees T_u and T_v that contains u and v, respectively. The order of the subtrees is denoted by $n_1(e) = |V(T_u)|$ and $n_2(e) = |V(T_v)|$. If T has N vertices, then $n_1(e) + n_2(e) = |V(T - e)| = N$ for all edges e in $E(T)$. Thus, we have the following theorem for trees.

Theorem 2.2.1. *Let T be a tree. Then*

$$W(T) = \sum_{e \in E(T)} n_1(e) n_2(e). \tag{2.10}$$

Proof. Since T is a tree, there is a unique path between a vertex $x \in V(T_u)$ and a vertex $y \in V(T_v)$ that must contain e. Thus, $n_1(e)n_2(e)$ is exactly the number of times that a particular edge e lies on the unique shortest path between two vertices of T. Thus, the sum of $n_1(e)n_2(e)$ over all edges of T must be the Wiener index of T. □

2.2.3 Wiener-type Graph Invariant

With the considerable success of the Wiener index, various extensions and generalizations of this graph invariant were recently put forward (Liu and Liu, 2013), (Gutman et al., 2004). In this subsection, we consider a class of extensions of the Wiener index expressed in term of Wiener-type invariant.

The Wiener-type graph invariant has found noteworthy applications in chemical literature and is defined as:

$$W_\lambda(G) = \sum_{\{u,v\}\subseteq V(G)} d(u,v)^\lambda, \tag{2.11}$$

where λ is a real or complex number.

Clearly, W_1 is just the Wiener index. For $\lambda = -2$ and $\lambda = -1$, the quantity W_λ is known as the Harary index (Plavšić et al., 1993) and reciprocal Wiener index (Diudea et al., 1997), respectively. These two indices have attracted much attention of chemical and mathematical researchers. Nowadays, many applications and properties on these topological indices have been studied and reported in the literature (Xu et al., 2015). The case of $\lambda = \frac{1}{2}$ was also analyzed in (Zhu et al., 1996).

The hyper Wiener index is another important graph invariant expressed in terms of the Wiener-type invariant. It was demonstrated that the hyper Wiener index is equal to $\frac{1}{2}W_1 + \frac{1}{2}W_2$ (Klein et al., 1995). For the mathematical properties of the hyper Wiener index, one can be referred to (Xu et al., 2014).

Nowadays, there exist a number of extensions expressed in terms of variable Wiener index. Here, we listed just the most important ones. In the next section, we investigate a recent extension of the Wiener index called the Terminal Wiener index, which is considered as the backbone of this thesis.

2.3 Terminal Wiener Index Of Graphs

The majority of distance-based topological indices have been derived from the distance matrix (Todeschini and Consonni, 2008). Recently, the so-called reduced distance matrix (Smolenskii et al., 2009), or terminal distance matrix (Horvat et al., 2008), (Randić et al., 2007) was proposed to serve as a source of novel topological indices and was used for other applications such as for modeling of amino acid sequences of proteins and genetic code (Randić et al., 2007), and to study the size of the Tree Bisection and Reconnection neighborhood of phylogenetic binary trees (Székely et al., 2011).

Let G be a graph with p pendent vertices, labeled by $v_1, v_2, ..., v_p$. Then, the terminal distance matrix of G, denoted by $TD(G)$, is the square matrix of order p whose $(i,j)-$entry is $d(v_i, v_j|G)$. Almost all researches on terminal distance matrix were concerned with trees. The reason of this is in the fact when the graph contains only one pendent vertex or two the information on the structure of G may be totally missing. It is important to know that the terminal distance matrix of a tree T determines the entire distance matrix of T, and thus completely determines the tree T itself (Zaretskii, 1965).

Based on the previous researches on the terminal distance matrix and its applications, the concept of the Terminal Wiener index was proposed as a recent extension of the Wiener index. In this section, we define the Terminal Wiener index and some of its fundamental properties. Then, we study some extremal bounds of this measure.

2.3.1 Definition and Main Properties of Terminal Wiener Index

The Terminal Wiener index was proposed by Gutman et al. (2009a). Somewhat later, but independently, Székely et al. (2011) arrived at the same idea. Let $V_p(G) \subseteq V(G)$ be the set of pendent vertices of the graph G. Then, the Terminal Wiener index of a connected graph G, denoted by $TW(G)$, is defined as the sum of distances between all pairs of pendent vertices of G. More formally,

$$TW(G) = \sum_{\{u,v\} \subseteq V_p(G)} d(u, v). \tag{2.12}$$

The Terminal Wiener index of a pendent vertex $v \in V_p(G)$ is defined as the sum of distances between a chosen pendent vertex v and all other pendent vertices in G. Then,

$$TW(v) = \sum_{u \in V_p(G)} d(v, u). \tag{2.13}$$

From the basic definition 2.12, we can set some fundamental properties of the Terminal Wiener index, which was studied in numerous papers. If the graph G has no pendent vertex or only one such vertex, then $TW(G) = 0$. If G has exactly two pendent vertices, then $TW(G) \geq 1$. The listed properties confirm that the Terminal Wiener index is powerful to extract the structural information of graphs with so many pendent vertices and especially the class of trees. From this fact, Gutman et al. (2009a) gave a formula similar to the Wiener's first theorem 2.2.1 to calculate efficiently the Terminal Wiener index of trees. Let $p_1(e)$ and $p_2(e)$ be the number of pendent vertices lying on the two sides of the edge $e \in E(T)$. Recall that if T is a tree with p pendent vertices, then for any edge e, we have $p_1(e) + p_2(e) = |V_p(T - e)| = p$. Thus, the Terminal Wiener index of a tree T is expressed as follows.

Theorem 2.3.1. *Let T be a tree of order N and with p pendent vertices. Then*

$$TW(T) = \sum_{e \in E(T)} p_1(e)p_2(e), \tag{2.14}$$

with the summation goes over all the edges of the tree T

Proof. We use the same idea of the proof of Wiener's first theorem.
Since T is a tree, there is a unique path between two pendent vertices u and v that must contains the edge e. Thus, $p_1(e)p_2(e)$ is exactly the number of times that a particular edge e lies on the unique shortest path between two pendent vertices of T. Then, the sum of $p_1(e)p_2(e)$ over all the edges of T must be the Terminal Wiener index of T. □

It should be noted that for all edges of a tree T, we have:

$$\begin{cases} p_1(e) + p_2(e) = p, \\ p_1, p_2 \geq 1, \\ p - 1 \leq p_1(e)p_2(e) \leq \left\lfloor \frac{p}{2} \right\rfloor \left\lceil \frac{p}{2} \right\rceil. \end{cases} \tag{2.15}$$

We previously pointed out that the Terminal Wiener index is a powerful measure only for graphs possessing so many pendent vertices. Therefore, until today, a number of results on extremal values for the Terminal Wiener index are established only for trees (Gutman et al., 2009a), (Chen and Zhang, 2013), (Chen and Zhang, 2015). We mention in the next subsections some basic bounds for this measure. We particularly focus on the maximal and the second maximal Terminal Wiener index that are defined via the Equation 2.14.

2.3.2 Trees with Minimal and Maximal Terminal Wiener Index

Our goal in this part, is to show the construction of trees with minimal and maximal Terminal Wiener index, which were solved and characterized in (Gutman et al., 2009a).We start with the minimal value of the Terminal Wiener index for any tree of order N.

In the case of a path tree P_N, there are 2 pendent vertices and for all the edges $p_1(e) = p_2(e) = 1$. It is obvious that 1 is the minimal value for the product $p_1(e)p_2(e)$ and because of the summation of the equation 2.14 goes over $N-1$ terms, it follows that $N-1$ is the minimal possible value that the Terminal Wiener index may assumes for trees of order N. Therefore, the tree of order N with a minimal Terminal Wiener index is the path P_N and the closed expression for the Terminal Wiener index of the path P_N is:

$$TW(P_N) = N - 1. \tag{2.16}$$

The characterization of trees with maximal Terminal Wiener index is more complex than the problem of finding the minimal value for this measure. Gutman et al. (2009a) solved this problem by restricting their consideration to the case of trees of order N and with fixed number of pendent vertices p. They proved that the Equation 2.14 can be rewritten as:

$$TW(T) = p(p-1) + \sum_{e'} p_1(e')p_2(e'), \tag{2.17}$$

where e' are the non-pendent edges of T, and there exist $N-1-p$ such edges.
Let T be a tree of order N, with $p \geq 4$ pendent vertices and the non-pendent edges e' verify the following condition:

$$p_1(e')p_2(e') = \left\lfloor \frac{p}{2} \right\rfloor \left\lceil \frac{p}{2} \right\rceil. \tag{2.18}$$

By using this condition, Gutman et al. (2009a) arrived to find the value of the maximal Terminal Wiener index and which is defined as follows.

$$TW(T) = p(p-1) + (N-p-1)\left(\left\lfloor \frac{p}{2} \right\rfloor \left\lceil \frac{p}{2} \right\rceil \right). \tag{2.19}$$

The construction of trees that satisfy the condition 2.18 and have the maximal Terminal Wiener index proceeds as follows:

- If p is even, $4 \leq p < N-1$, then the required tree is obtained from the path P_{N-p} by attaching to each of its terminal vertices $\frac{p}{2}$ new pendent vertices. This tree is unique.
- If p is odd, $5 \leq p < N-1$, then the required tree is obtained from the path P_{N-p} by attaching to each of its terminal vertices $\frac{p-1}{2}$ new pendent vertices, and by attaching one more pendent vertex to any vertex of P_{N-p}. There exist $\lceil \frac{N-p}{2} \rceil$ distinct trees of this kind.
- If $p = N-1$, then the respective tree is the star.

An example of trees with the maximal Terminal Wiener index is shown in Figure 2.2.

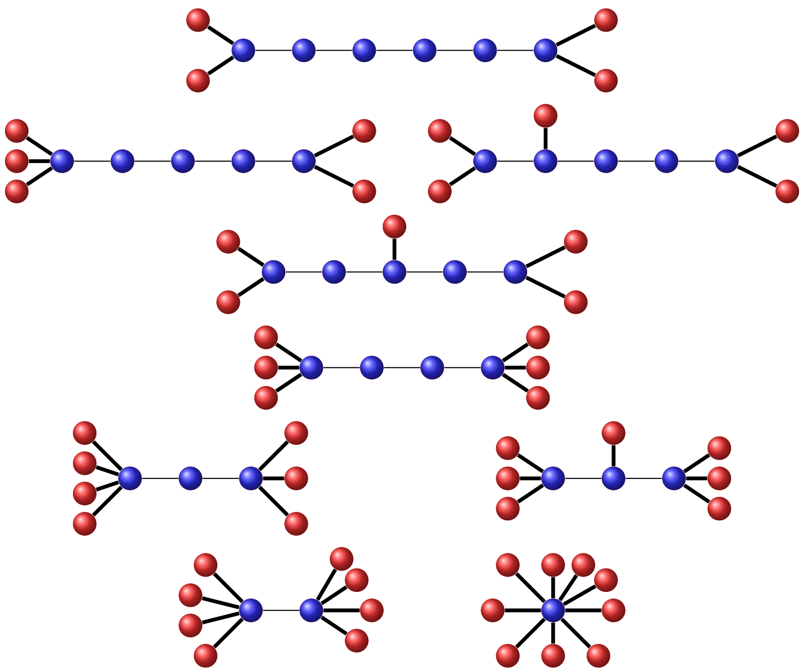


Figure 2.2: Trees of order $N = 10$ with a maximal Terminal Wiener index, for $p = \{4, 5, 6, 7, 8, 9\}$.

In the following lemma, we calculate the number L of trees with N vertices that have the maximal Terminal Wiener index.

Lemma 2.3.1. *(Zeryouh et al., 2016) Let $\mathcal{T}(N,p)$ be the set of all trees of order N. Then, we have:*

$$L = \begin{cases} \frac{1}{8}N^2 - \frac{1}{2}N + \frac{3}{8} & \textit{if } N \textit{ is odd,} \\ \\ \frac{1}{8}N^2 - \frac{1}{4}N - 1 & \textit{if } N \textit{ is even.} \end{cases} \tag{2.20}$$

Proof. From the construction process of trees with maximal Terminal Wiener index, we can get easily the result. □

2.3.3 Trees with Second Maximal Terminal Wiener Index and a Tree Transformation

In this subsection, we focus on characterizing the structure of trees having the second maximal Terminal Wiener index. Then, we propose a tree transformation that increases the value of the Terminal Wiener index.

Let $\mathcal{T}(N,p)$ be a set of trees of order N and with a fixed number p of pendent vertices. Let e' be the non-pendent edges of a tree T. Obviously, there exist $N-p-1$ such edges. The maximal value of the product $p_1(e').p_2(e')$ is $\lfloor \frac{p}{2} \rfloor \lceil \frac{p}{2} \rceil$ and the second maximal value of this product is $\lfloor \frac{p-2}{2} \rfloor \lceil \frac{p+2}{2} \rceil$. Then, we get the following conditions that should be satisfied in order to a tree T reaches the second maximal Terminal Wiener index.

- For $N-p-2$ non-pendent edges e', we should have:

$$p_1(e').p_2(e') = \left\lfloor \frac{p}{2} \right\rfloor \left\lceil \frac{p}{2} \right\rceil. \tag{2.21}$$

- One non-pendent edge e' should verifies:

$$p_1(e').p_2(e') = \left\lfloor \frac{p-2}{2} \right\rfloor \left\lceil \frac{p+2}{2} \right\rceil. \tag{2.22}$$

Therefore, we get the following main theorem of the second maximal Terminal Wiener index.

Theorem 2.3.2. *Let T be a tree of order N, with $p \geq 4$ pendent vertices and have the second maximal Terminal Wiener index. Then*

$$TW(T) = p(p-1) + (N-p-2)\left\lfloor \frac{p}{2} \right\rfloor \left\lceil \frac{p}{2} \right\rceil + \left\lfloor \frac{p-2}{2} \right\rfloor \left\lceil \frac{p+2}{2} \right\rceil. \tag{2.23}$$

Proof. We apply the definition of the Terminal Wiener index and both conditions (2.21) and (2.22). Then,

$$\begin{aligned}TW(T) =& \sum_{e} p_1(e/T)p_2(e/T)\\ =& p(p-1) + \sum_{e'} p_1(e'/T)p_2(e'/T)\\ =& p(p-1) + (N-p-2)\left\lfloor \frac{p}{2} \right\rfloor \left\lceil \frac{p}{2} \right\rceil + \left\lfloor \frac{p-2}{2} \right\rfloor \left\lceil \frac{p+2}{2} \right\rceil.\end{aligned}$$

The proof is complete. □

Now, we describe the construction of trees that have the second maximal Terminal Wiener index and evidently satisfy both conditions (2.21) and (2.22). The process of construction proceeds as follows:

- If p is even and $4 \leq p < N-1$, the tree is obtained from a path $v_1v_2v_3...v_l$ of length $l = N-p$ by attaching a total of $\frac{p}{2}$ pendent vertices to the left terminal vertex v_1 and a total of $\frac{p-2}{2}$ pendent vertices to the right terminal vertex v_l. Then, we add one more pendent vertex to the vertex v_{l-1}. We note that this tree is unique.
- If p is odd and $5 \leq p < N-1$, the tree is obtained from a path $v_1v_2v_3...v_l$ of length $l = N-p$ by attaching a total of $\left\lfloor \frac{p}{2} \right\rfloor$ pendent vertices to the left terminal vertex v_1 and $\left\lfloor \frac{p-2}{2} \right\rfloor$ pendent vertices to the right terminal vertex v_l. Then, we attach one more pendent vertex to the vertex v_{l-1}, and an other pendent vertex to any non-pendent vertex from the path $v_1v_2v_3...v_{l-1}$. We note that there exist $N-p-1$ such trees.

From the construction process of such trees, We get the following immediate result.

Lemma 2.3.2. *Let $\mathcal{T}(N,p)$ be the set of all trees of order N. Then, the number L of trees that have the second maximal Terminal Wiener index is:*

$$L = \begin{cases} \frac{1}{4}N^2 - 2N + \frac{15}{4} & \text{if } N \text{ is odd,} \\ \\ \frac{1}{4}N^2 - 2N + 4 & \text{if } N \text{ is even.} \end{cases} \tag{2.24}$$

Proof. We can get the result from the construction process of trees with second maximal Terminal Wiener index. □

We illustrate in Figure 2.3 an example of trees of order $N = 10$ and with $p = \{4, 5, 6, 7, 8\}$ pendent vertices that have the second maximal Terminal Wiener index.

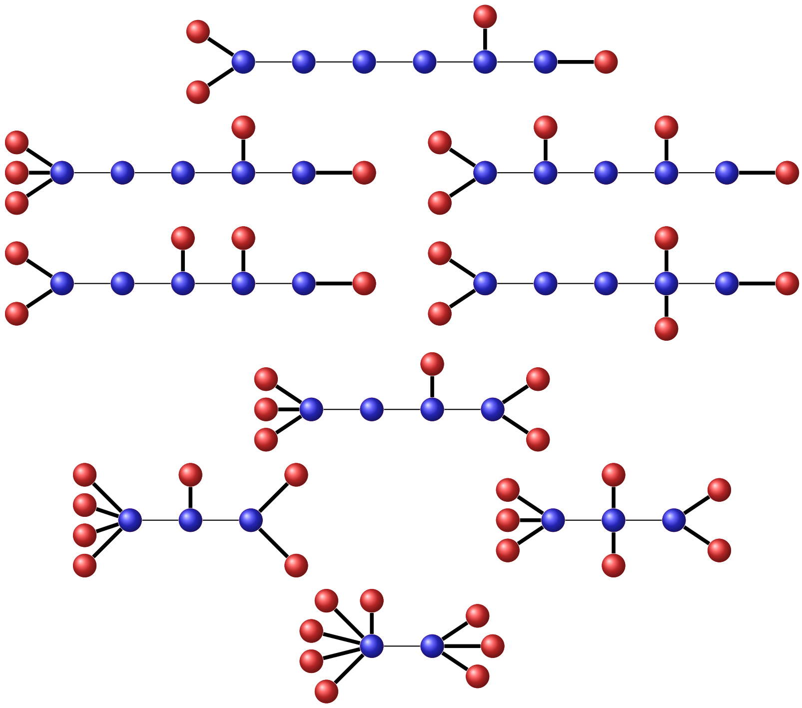


Figure 2.3: Trees with second maximal Terminal Wiener index, for $N = 10$ and $p = \{4, 5, 6, 7, 8\}$.

Now, we characterize a transformation that maximizes the Terminal Wiener index among all trees for a given order N.
Let T be a tree of order N and with p pendent vertices. u and v are two non pendent vertices, such as in the left side of the vertex u we have k pendent vertices, and the vertex v is attached to k' pendent vertices, labeled $\{a_1, a_2, .., a_{k'}\}$; see Figure 2.4 for more details.
Let T^* be a tree of order N and with p pendent vertices. The tree T^* is obtained from the transformation depicted in Figure 2.4 , such that we remove the edge va'_k and we add a new edge ua'_k. We should note that $p = k + k'$.

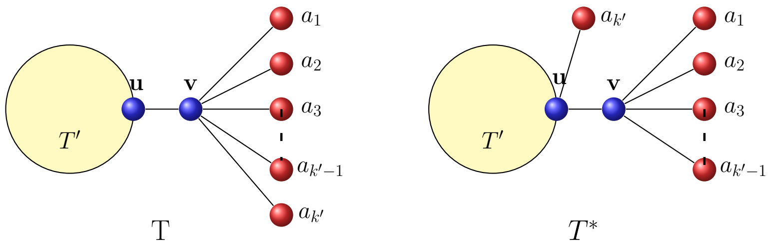


Figure 2.4: The transformation of T to T^*.

Then, the following theorem shows the main result of this transformation.

Theorem 2.3.3. *Let T and T^* be two trees of order N and with p pendent vertices. Then,*

$$TW(T) > TW(T^*) \qquad for \;\; k \geq k'. \tag{2.25}$$

$$TW(T) < TW(T^*) \qquad for \;\; k < k'. \tag{2.26}$$

and we have equality if $k = k' - 1$.

Proof. The tree T' has k pendent vertices, which are labelled by $\{x_1, x_2, ..., x_k\}$. Then,

$$\begin{aligned}
TW(T) = &\sum_{i=1}^{k} d(a_1, x_i) + ... + \sum_{i=1}^{k} d(a_k', x_i) + \sum_{1\leq i<j\leq k} d(x_i, x_j) + 2\sum_{i=1}^{k'-1}(k' - i)\\
=&k'\sum_{i=1}^{k} d(a_1, x_i) + \sum_{1\leq i<j\leq k} d(x_i, x_j) + 2\sum_{i=1}^{k'-1}(k' - i)\\
=&k'\sum_{i=1}^{k} \left[d(a_1, v) + d(v, x_i)\right] + \sum_{1\leq i<j\leq k} d(x_i, x_j) + k'^2 - k'\\
=&k'\sum_{i=1}^{k} d(v, x_i) + \sum_{1\leq i<j\leq k} d(x_i, x_j) + k'(k' + k - 1).
\end{aligned}$$

and

$$TW(T^*) = \sum_{i=1}^{k} d(a_{k'}, x_i) + (k' - 1)\sum_{i=1}^{k} d(a_1, x_i) + \sum_{1\leq i<j\leq k} d(x_i, x_j) + d(a_1, a_{k'}) + ...+$$

$$
\begin{aligned}
& d(a_{k'-1}, a_{k'}) + 2\sum_{i=1}^{k'-2}(k'-1-i) \\
=& \sum_{i=1}^{k} d(a_{k'}, x_i) + (k'-1)\sum_{i=1}^{k}[d(a_1, v) + d(v, x_i)] + \sum_{1 \le i < j \le k} d(x_i, x_j) + \\
& 3(k'-1) + k'^2 - 3k' + 2 \\
=& (k'-1)\sum_{i=1}^{k} d(v, x_i) + \sum_{i=1}^{k} d(u, x_i) + \sum_{1 \le i < j \le k} d(x_i, x_j) + k'(k'+k) - 1.
\end{aligned}
$$

By subtraction, we obtain $TW(T) - TW(T^*) = k - k' + 1$.

□

2.4 Generalization Of The Terminal Wiener Index

The concept of the Terminal Wiener index was recently generalized in (Ilić and Ilić, 2013) by considering pairs of vertices of some fixed degree $K \geq 1$. In this section, we introduce this new extension called the generalized Terminal Wiener index. Then, we report the extremal properties of this measure.

2.4.1 Definition of the Generalized Terminal Wiener Index

The generalized Terminal Wiener index is defined as the sum of distances between all pairs of vertices of degree $K \geq 1$. More formally,

$$
TW_K(G) = \sum_{\{u,v\} \subseteq V_K(G)} d(u, v), \tag{2.27}
$$

where $V_K(G)$ denotes the set of vertices with degree K.

The generalized Terminal Wiener index of a vertex $v \in V_K(G)$ is defined as the sum of distances between the vertex v and all other vertices of degree K in G. Then,

$$
TW_K(v) = \sum_{u \in V_K(G)} d(v, u). \tag{2.28}
$$

From the definition 2.27, we can set some remarks. For d-regular graphs, $TW_{K=d}(G) = W(G)$ and we have $TW_{K=d'}(G) = 0$ for $d' \neq d$. In the case of $K = 1$, the generalized Terminal Wiener index is simply the Terminal Wiener index.

In order to calculate efficiently the generalized Terminal Wiener index of trees, we can use a formula similar to Wiener's first theorem 2.2.1. Let $p_1^K(e)$ and $p_2^K(e)$ be the number of vertices with degree K lying on the two sides of the edge $e \in E(T)$. Therefore, the generalized Terminal Wiener index of a tree T is expressed as follows.

Theorem 2.4.1. *Let T be a tree of order N and with N_K vertices of degree K. Then*

$$TW_K(T) = \sum_{e \in E(T)} p_1^K(e) p_2^K(e), \tag{2.29}$$

with the summation goes over all the edges of the tree T.

Proof. We use the same proof of the Wiener index and the Terminal Wiener index. The product $p_1^K(e)p_2^K(e)$ is exactly the number of times that a particular edge e lies on the unique path between two vertices of degree K. Then, the sum of $p_1^K(e)p_2^K(e)$ over all the edges of T must be the generalized Terminal Wiener index of T. □

2.4.2 Extremal Properties of the Generalized Terminal Wiener Index

Extremal properties of the generalized Terminal Wiener index was also studied. In the first investigation of this index, Ilić and Ilić (2013) characterized the extremal trees that maximize the generalized Terminal Wiener index when $K = 3$. After that, Chen et al. (2016) proved the conjecture of Ilić and Ilić (2013) regarding the maximal value of the generalized Terminal Wiener index when $K \geq 4$. Here, we report some of the extremal results found in these two papers and we start with the case of $K = 2$.

For $K = 2$, the unique tree that maximizes TW_2 is the path P_N. This follows from the simple fact that in every tree there are at least 2 pendent vertices and at most $N - 2$ vertices of degree 2. Consequently, this construction is achieved only for the path P_N.

Let $C_{N,K,\rho}$ be the caterpillar obtained from a path of length $s+2 = N - \rho(K-2)$, by attaching $K-2$ pendent vertices to exactly ρ vertices of a path $P_{s+2} = v_0v_1...v_sv_{s+1}$, such that we start from the end vertices v_1 and v_s towards the middle successively and symmetrically; see Figure 2.5.

Lemma 2.4.1. *(Ilić and Ilić, 2013) The generalized Terminal Wiener index of $C_{N,K,\rho}$ is equal to:*

$$TW_K(C_{N,K,\rho}) = \begin{cases} \frac{1}{12}\rho\Big(3N\rho + 5\rho^2 - 3K\rho^2 - 2 - 6\rho\Big) & \text{if } \rho \text{ is even,} \\ \\ \frac{1}{12}(\rho+1)(\rho-1)\Big(3N + 5\rho - 3K\rho - 6\Big) & \text{if } \rho \text{ is odd.} \end{cases} \tag{2.30}$$

The following theorem demonstrates that the generalized Terminal Wiener index, in the case of $K = 3$, reaches its maximum value at the 3-bounded caterpillars, denoted by $C_{N,3,\lfloor \frac{N}{2} \rfloor - 1}$. We note that a caterpillar C is 3-bounded if all vertices of C have degree less than or equal to 3.

Theorem 2.4.2. *(Ilić and Ilić, 2013) Let T be a tree of order $N > 4$. Then*

$$TW_3(T) \leq TW_3(C_{N,3,\lfloor \frac{N}{2} \rfloor - 1}), \tag{2.31}$$

with the equality holds if and only if $T = C_{N,3,\lfloor \frac{N}{2} \rfloor - 1}$.

In the case of $K \geq 4$, we have the following result.

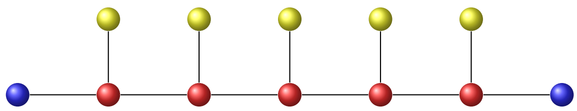
Theorem 2.4.3. *(Ilić and Ilić, 2013),(Chen et al., 2016) For $K \geq 4$, the generalized Terminal Wiener index reaches its maximum value at $C_{N,K,\rho}$ caterpillars.*

Based on Theorem 2.4.3, for any given N and K, the maximum value of the generalized Terminal wiener index can be determined if we find the caterpillar $C_{N,K,\rho}$ that maximizes the function $f(\rho) = TW_K(C_{N,K,\rho})$. Thus, Chen et al. (2016) determined the corresponding ρ value of the extremal graph. Briefly, they followed the following steps in order to get the maximum value of the generalized Terminal Wiener index of a tree with N vertices:

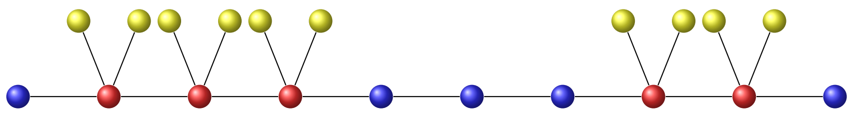
- Determine the parameters P^*_{even} and P^*_{odd}, which make $TW_K(C_{N,K,\rho})$ reaches its maximum value when ρ is even and odd, respectively.
- Determine which of $TW_K(C_{N,K,P^*_{even}})$ and $TW_K(C_{N,K,P^*_{odd}})$ is the grater one, and give the maximum value.
- Determine the structure of the extremal graph.

For more details, we refer to see (Chen et al., 2016).

We illustrate in Figure 2.5 an example of extremal trees that maximize the generalized Terminal Wiener index, for $K = 3$ and $K = 4$.



$C_{12,3,5}$



$C_{20,4,5}$

Figure 2.5: The extremal trees with maximal generalized Terminal Wiener index, for $K = 3$ and $K = 4$, among all trees of order $N = 12$ and $N = 20$, respectively.

2.5 Summary

Quantitative measures or topological indices are numerical values that play an important role in characterizing the structural properties of networks and appear in many applications, such as chemistry, biology, network science, and so on. In this chapter, we introduced the branch of Quantitative Graph Theory and the concept of topological indices. Then, we focused on some distance-based topological indices, such as Wiener index, Terminal Wiener index and generalized Terminal Wiener index. Our aim was to outline a general overview about the main properties of these distance-based measures, which have been proved in many works. We showed that the computation of these measures in the case of trees is much easier than that of an arbitrary graph. In addition, we reported some extremal values that these invariants assume in various classes of graphs and especially in the case of trees. Furthermore, the main important result in this chapter is our study concerning the second maximal Terminal Wiener index. In this work, we determined the structure of trees where the second maximal Terminal Wiener index is attained and we proved a transformation that increases the value of the Terminal Wiener index.

In the next chapter, we study some methods to overcome the problem of calculating the Terminal Wiener index of networks possessing a large number of vertices and we use the obtained results to analyze the structure of some well known structures, such as Binomial trees and Dendrimer trees.

Chapter

3 Some methods for computing the Terminal Wiener Index of certain Networks

Contents

The usual method for computing any distance-based topological index is the direct application of the index basic definition. When dealing with networks possessing a large number of vertices, the applicability of this classical method is a challenging computational problem. Therefore, the fundamental task would be to propose some new methods that reduce the computation of any distance-based topological index. In the literature, we can find a number of methods that have been proposed and studied for calculating certain topological indices. In this chapter, we overcome the problem of calculating the Terminal Wiener index by introducing three methods of computation. The first approach is a decomposition method dedicated to some networks constructed from several copies of components, such as Star-Trees and Path-Trees. The main idea of this technique is to generate the property of the whole structure by calculating the Terminal Wiener index of individual components . The second method is based on an important concept called the Thorny graph, where the objective is to obtain the Terminal Wiener index of a structure in terms of the Terminal Wiener index of its parent graph. The last method is an amelioration of the definition of the Terminal Wiener index. The advantage of this approach is in the fact that can be used even if we have a disconnected graph.

In each section of this chapter, we discuss a method and we apply the obtained results to treat and analyze some important structures. We note that the first method was published in an international journal (Zeryouh et al., 2015b) and we presented the obtained results of the second and the third method in two international conferences (Zeryouh et al., 2015a),(Zeryouh et al., 2018a).

3.1 Terminal Wiener Index Of Some Graph Compositions

Many important networks in different areas of applications are built from several copies of components. One of the efficient methods for computing the topological indices of this kind of networks is to assemble the indices of individual components to generate the property of the whole structure. A lot of works have been done in this direction for some topological indices. Mansour and Schork (2009) studied vertex PI index and Szeged index of an interesting graph that is built from several shapes called the bridge graph and in (Mansour and Schork, 2010), they calculated Wiener, hyper Wiener, detour and hyper-detour indices of bridge and chain graphs. In (Al Hagri et al., 2011), they calculated the Wiener index of star graph and chain graph, while in (El Marraki and Al Hagri, 2010), the authors studied the Wiener index of some composite trees called star-trees and path-trees. In this section, we focus on star-trees and path-trees, and we compute the Terminal Wiener index for them. The important special case where the star and path trees are built from several copies of the same graph is also studied. Then, we apply the obtained results to treat some particular examples.

3.1.1 Terminal Wiener Index of Star-Trees

The star-tree, denoted by $\mathcal{S}_L$, is a tree composed of L trees T_i of order N_i and with p_i pendent vertices, for $i = \{1, 2, ..., L\}$. All trees T_i are connected by a common vertex s. We restrict our consideration to the case where the common vertex s is a pendent vertex for all the trees T_i before connecting them. Figure 3.1 illustrates a theoretical example of two star-trees. In the case of a star-tree composed of two trees, we have the following result.

Theorem 3.1.1. *Let $\mathcal{S}_2$ be a star-tree composed of two trees T_1 and T_2. Then*

$$TW(\mathcal{S}_2) = TW(T_1) + TW(T_2) + (p_2 - 2) \sum_{u \in V_p(T_1)} d(s, u) + (p_1 - 2) \sum_{v \in V_p(T_2)} d(s, v), \quad (3.1)$$

where p_1 and p_2 are the number of pendent vertices in the two trees T_1 and T_2, respectively.

Proof. We apply the basic definition of the Terminal Wiener index. Then,

$$\begin{aligned}TW(\mathcal{S}_2) &= \sum_{\{u,v\}\subseteq V_p(\mathcal{S}_2)} d(u,v)\\ &= \sum_{\{u,v\}\subseteq V_p(T_1)} d(u,v) + \sum_{\{u,v\}\subseteq V_p(T_2)} d(u,v) + \sum_{u\subseteq V_p(T_1)}\sum_{v\subseteq V_p(T_2)} d(u,v).\end{aligned}$$

We have:

$$\sum_{u\subseteq V_p(T_1)}\sum_{v\subseteq V_p(T_2)} d(u,v) = \sum_{u\subseteq V_p(T_1)}\sum_{v\subseteq V_p(T_2)} \Big[d(u,s)+d(s,v)\Big].$$

Therefore,

$$\begin{aligned}TW(\mathcal{S}_2) = TW(T_1) - \sum_{u\subseteq V_p(T_1)} d(u,s) + TW(T_2) - \sum_{v\subseteq V_p(T_2)} d(s,v) +\\ \sum_{u\subseteq V_p(T_1)}\sum_{v\subseteq V_p(T_2)} d(u,s) + \sum_{u\subseteq V_p(T_1)}\sum_{v\subseteq V_p(T_2)} d(s,v)\\ = TW(T_1) + TW(T_2) + (p_2-2)\sum_{u\subseteq V_p(T_1)} d(u,s) + (p_1-2)\sum_{v\subseteq V_p(T_2)} d(s,v).\end{aligned}$$

The proof is complete. □

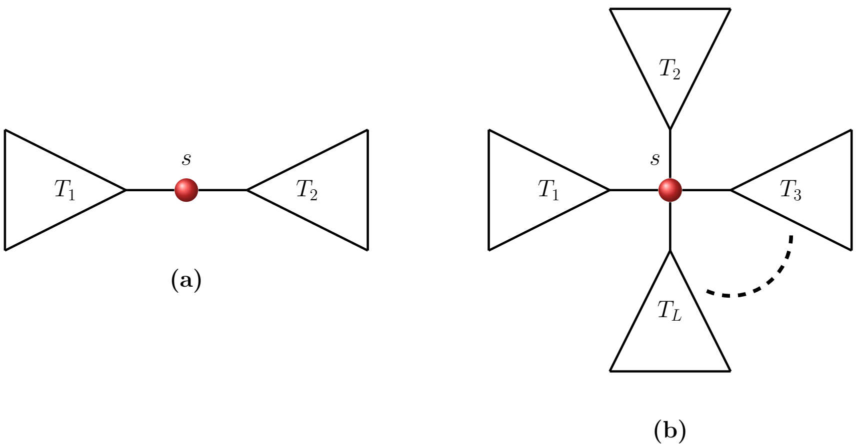


Figure 3.1: An example of two star-trees. (a): represents a star-tree $\mathcal{S}_2$ composed of two trees T_1 and T_2. (b): represents a star-tree $\mathcal{S}_L$ composed of L trees $T_1, T_2, ..., T_L$.

In the following theorem, we present a general formula for computing the Terminal

Wiener index of a star-tree composed of L trees.

Theorem 3.1.2. *Let $\mathcal{S}_L$ be a star-tree composed of L trees T_i of order N_i and with p_i pendent vertices, for $i = \{1, 2, ..., L\}$. Then*

$$\begin{aligned} TW(\mathcal{S}_L) =& \sum_{i=1}^{L} TW(T_i) + \sum_{i=1}^{L-1}\sum_{j=i+1}^{L}\left[p_i \sum_{u\in V_p(T_j)} d(s,u) + p_j \sum_{v\in V_p(T_i)} d(s,v)\right] - \\ & L\sum_{i=1}^{L}\sum_{u\in V_p(T_i)} d(s,u). \end{aligned} \tag{3.2}$$

Proof. We apply the definition of the Terminal Wiener index. Then,

$$\begin{aligned} TW(\mathcal{S}_L) &= \sum_{\{u,v\}\subseteq V_p(\mathcal{S}_L)} d(u,v) \\ &= \sum_{\{u,v\}\subseteq V_p(T_1)} d(u,v) + ... + \sum_{\{u,v\}\subseteq V_p(T_L)} d(u,v) + \sum_{u\subseteq V_p(T_1)}\sum_{v\subseteq V_p(T_2)} d(u,v) + \\ & ... + \sum_{u\subseteq V_p(T_1)}\sum_{v\subseteq V_p(T_L)} d(u,v) + \sum_{u\subseteq V_p(T_2)}\sum_{v\subseteq V_p(T_3)} d(u,v) + ... + \\ & \sum_{u\subseteq V_p(T_2)}\sum_{v\subseteq V_p(T_L)} d(u,v) + ... + \sum_{u\subseteq V_p(T_{L-1})}\sum_{v\subseteq V_p(T_L)} d(u,v) \\ &= \sum_{i=1}^{L} TW(T_i) + \sum_{i=1}^{L-1}\sum_{j=i+1}^{L}\Bigg(\sum_{u\subseteq V_p(T_i)}\sum_{v\subseteq V_p(T_j)} d(u,v)\Bigg) - \sum_{i=1}^{L}\Bigg(\sum_{u\subseteq V_p(T_i)} d(u,s)\Bigg). \end{aligned}$$

Using the fact that

$$\sum_{u\subseteq V_p(T_i)}\sum_{v\subseteq V_p(T_j)} d(u,v) = (p_j - 1)\sum_{u\subseteq V_p(T_i)} d(u,s) + (p_i - 1)\sum_{v\subseteq V_p(T_j)} d(s,v),$$

we obtain the Equation (3.2). □

Now, we generalize Theorem 3.1.2 to the special case where the star-tree is constructed from L copies of the same tree T.

Corollary 3.1.1. *Let $\mathcal{S}_L$ be a star-tree composed of L copies of the same trees T of order*

N and with p pendent vertices. Then,

$$TW(\mathcal{S}_L) = L\,TW(T) + L\Big[(L-1)p - L\Big] \sum_{u \in V_p(T)} d(s,u). \tag{3.3}$$

3.1.2 Computation of the Terminal Wiener Index of some Families of Star-Trees

Here, we apply the results derived in the previous subsection to compute the Terminal Wiener index of some families of star-trees.

3.1.2.1 The case of a star-tree composed of L paths

Let $\mathcal{T}_L$ be a star-tree composed of L paths P_N of order N, as shown in Figure 3.2. We want to determine the Terminal Wiener index of this structure.

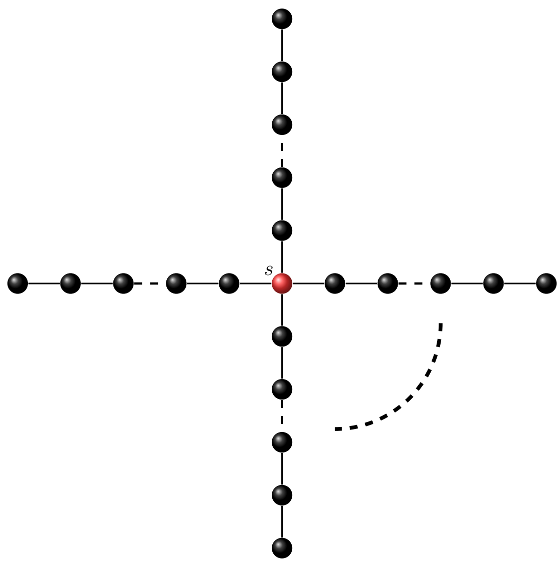


Figure 3.2: A star-tree composed of L paths P_N.

It is known that the closed expression for the Terminal Wiener index of a path P_N is:

$$TW(P_N) = N - 1.$$

The distance between the vertex of concatenation s and pendent vertices of P_N is simply the Terminal Wiener index of the path P_N. Thus, by applying the corollary 3.1.1, the

Terminal Wiener index of a star-tree composed of L paths P_N is given by:

$$TW(\mathcal{T}_L) = L(L-1)(N-1). \tag{3.4}$$

The following table illustrates some numerical values for the Terminal Wiener index of a star-tree composed of L paths P_N. We can see that the value of the Terminal Wiener index increases by changing the number of copies L and the order N.

	$TW(\mathcal{T}_L)$									
L	N=10	N=20	N=30	N=40	N=50	N=60	N=70	N=80	N=90	N=100
2	18	38	58	78	98	118	138	158	178	198
3	54	114	174	234	294	354	414	474	534	594
4	108	228	348	468	588	708	828	948	1068	1188
5	180	380	580	780	980	1180	1380	1580	1780	1980
6	270	570	870	1170	1470	1770	2070	2370	2670	2970
7	378	798	1218	1638	2058	2478	2898	3318	3738	4158
8	504	1064	1624	2184	2744	3304	3864	4424	4984	5544
9	648	1368	2088	2808	3528	4248	4968	5688	6408	7128
10	810	1710	2610	3510	4410	5310	6210	7110	8010	8910

Table 3.1: Some numerical results for the Terminal Wiener index of the star-tree $\mathcal{T}_L$.

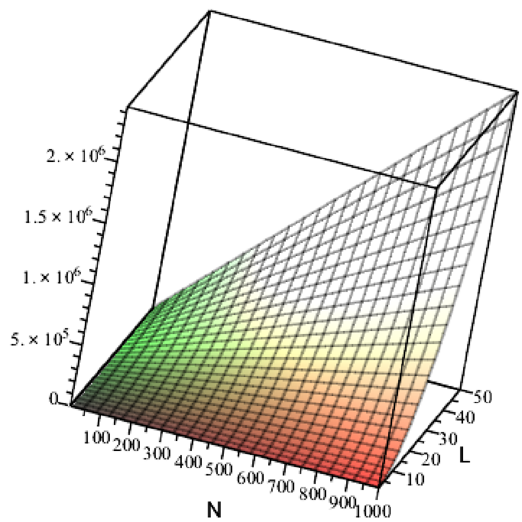


Figure 3.3: The graphical representation of the Terminal Wiener index of the star-tree $\mathcal{T}_L$.

Even in Figure 3.3, which represents a graphical representation of the Terminal Wiener index of the structure $\mathcal{T}_L$, we can observe the increase in the values of the Terminal Wiener index. The changes in the behavior of this index is well illustrated by the large variations in the perceptual appearance of the colors.

3.1.2.2 The case of a star-tree composed of L stars

Let $\mathcal{E}_L$ be a star-tree composed of L stars S_N of order N; see Figure 3.4. The Terminal Wiener index of a star S_N is equal to:

$$TW(S_N) = (N-1)(N-2).$$

and the distance between the vertex of concatenation s and all pendent vertices of S_N is defined as:

$$\sum_{u \in V_p(S_N)} d(s,u) = 2(N-2).$$

Therefore, by applying these two equations in Corollary 3.1.1, we have the following result.

$$TW(\mathcal{E}_L) = L(2L-1)(N-2)(N-1) - 2L. \tag{3.5}$$

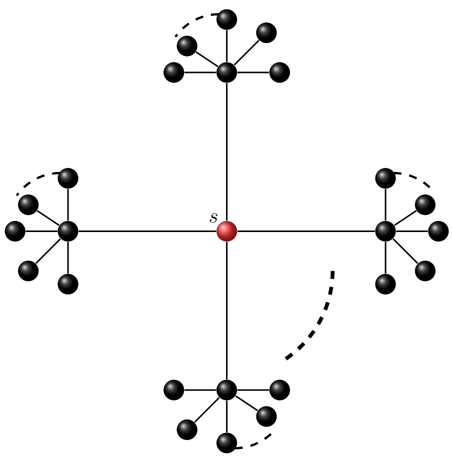


Figure 3.4: A star-tree composed of L stars S_N.

In the following table, we present some numerical results for the Terminal Wiener index of the star-tree $\mathcal{E}_L$. We obtained these outcomes by changing the order of the star S_N and the parameter L.

	$TW(\mathcal{E}_L)$								
L	N=10	N=20	N=30	N=40	N=50	N=60	N=70	N=80	N=90
2	428	2048	4868	8888	14108	20528	28148	36968	46988
3	1074	5124	12174	22224	35274	51324	70374	92424	117474
4	2008	9568	22728	4488	65848	95808	131368	172528	219288
5	3230	15380	36530	66680	105830	153980	211130	277280	352430
6	4740	22560	53580	97800	155220	225840	309660	406680	516900
7	6538	31108	73878	134848	214018	311388	426958	560728	712698
8	8624	41024	97424	177824	282224	410624	563024	739424	939824
9	10998	52308	124218	226728	359838	523548	717858	942768	1198278
10	13660	64960	154260	281560	446860	650160	891460	1170760	1488060

Table 3.2: Some numerical values for the Terminal Wiener index of the star-tree $\mathcal{E}_L$.

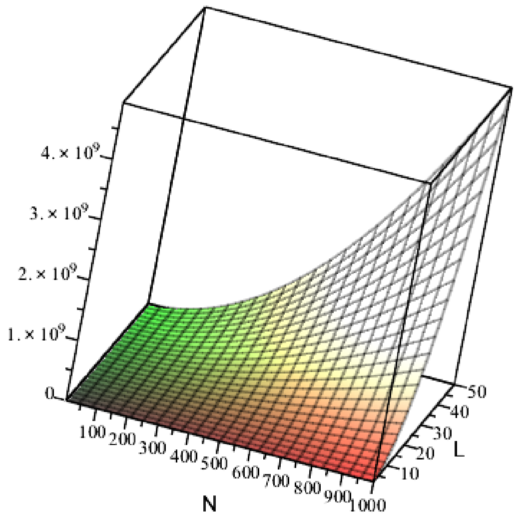


Figure 3.5: The graphical representation of the Terminal Wiener index of the star-tree $\mathcal{E}_L$.

The above table and Figure 3.5 show the increase of the Terminal Wiener index with the growth in the order N and the number of copies L. The large changes in the perceptual appearance of the colors in Figure 3.5 clarify the performance of the Terminal Wiener index for this structure.

3.1.3 Terminal Wiener Index of Path-Trees

The path-tree, denoted by $\mathcal{P}_L$, is a tree composed of L trees T_i of order N_i and with p_i pendent vertices, for $i = \{1, 2, ..., L\}$. The trees T_i are connected to each others by a set of edges $s_i s_{i+1}$, for $i = \{1, 2, ..., L-1\}$. Figure 3.6 illustrates a theoretical example of a path-tree.

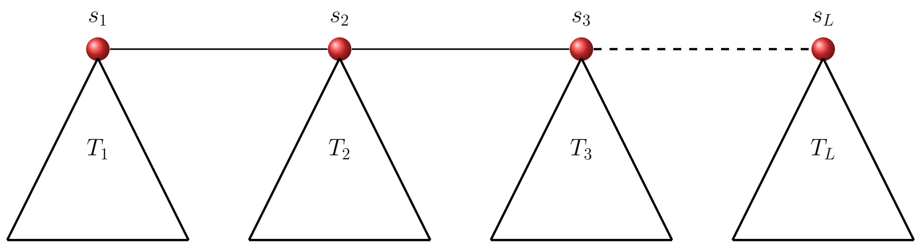


Figure 3.6: An example of a path-tree $\mathcal{P}_L$ composed of L trees $T_1, T_2, ..., T_L$.

The vertex s_i can be a pendent vertex or a non-pendent vertex. In this study, we consider the both cases to calculate the Terminal Wiener index of path-trees.

Theorem 3.1.3. *Let $\mathcal{P}_2$ be a path-tree composed of two trees T_1 and T_2. Then*

- *If s_i is a non-pendent vertex, for $i = \{1, 2\}$*

$$TW(\mathcal{P}_2) = TW(T_1) + TW(T_2) + p_2 \sum_{u \in V_p(T_1)} d(s_1, u) + p_1 \sum_{v \in V_p(T_2)} d(s_2, v) + p_1 p_2. \tag{3.6}$$

- *If s_i is a pendent vertex, for $i = \{1, 2\}$*

$$\begin{aligned} TW(\mathcal{P}_2) =& TW(T_1) + TW(T_2) + (p_2 - 2) \sum_{u \in V_p(T_1)} d(s_1, u) + (p_1 - 2) \sum_{v \in V_p(T_2)} d(s_2, v) + \\ & (p_1 - 1)(p_2 - 1). \end{aligned} \tag{3.7}$$

Proof. For the both cases, we apply the definition of the Terminal Wiener index.
If s_i is a non-pendent vertex, then

$$\begin{aligned} TW(\mathcal{P}_2) &= \sum_{\{u,v\} \subseteq V_p(\mathcal{P}_2)} d(u, v) \\ &= \sum_{\{u,v\} \subseteq V_p(T_1)} d(u, v) + \sum_{\{u,v\} \subseteq V_p(T_2)} d(u, v) + \sum_{u \subseteq V_p(T_1)} \sum_{v \subseteq V_p(T_2)} d(u, v). \end{aligned}$$

We have

$$\sum_{u \subseteq V_p(T_1)} \sum_{v \subseteq V_p(T_2)} d(u,v) = \sum_{u \subseteq V_p(T_1)} \sum_{v \subseteq V_p(T_2)} \Big[d(u,s_1) + d(s_1,s_2) + d(s_2,v) \Big].$$

Thus, by a simple calculation we get the Equation 3.6 and we follow the same proof for the second case. □

For the general case when the path-tree is composed of L trees, we have the following results.

Theorem 3.1.4. *Let $\mathcal{P}_L$ be a path-tree composed of L trees T_i of order N_i and with p_i pendent vertices, for $i = \{1, 2, ..., L\}$. Then,*

- *If s_i is a non-pendent vertex, for $i = \{1, 2, ..., L\}$*

$$TW(\mathcal{P}_L) = \sum_{i=1}^{L} TW(T_i) + \sum_{i=1}^{L-1} \Big[p_i \sum_{u \in V_p(T_{i+1})} d(s_{i+1}, u) + p_{i+1} \sum_{u \in V_p(T_i)} d(s_i, u) + p_i p_{i+1} \Big] + \sum_{i=1}^{L-2} \sum_{j=i+2}^{L} \Big[p_j \sum_{u \in V_p(T_i)} d(s_i, u) + p_i \sum_{u \in V_p(T_j)} d(s_j, u) + (j-i) p_i p_j \Big]. \tag{3.8}$$

- *If s_i is a pendent vertex, for $i = \{1, 2, ..., L\}$*

$$TW(\mathcal{P}_L) = \sum_{i=1}^{L} TW(T_i) + \sum_{i=1}^{L-1} \Big[(p_i - 1) \sum_{u \in V_p(T_{i+1})} d(s_{i+1}, u) + (p_{i+1} - 1) \sum_{u \in V_p(T_i)} d(s_i, u) + (p_i - 1)(p_{i+1} - 1) \Big] + \sum_{i=1}^{L-2} \sum_{j=i+2}^{L} \Big[(p_j - 1) \sum_{u \in V_p(T_i)} d(s_i, u) + (p_i - 1) \sum_{u \in V_p(T_j)} d(s_j, u) + (j-i)(p_i - 1)(p_j - 1) \Big] - \sum_{i=1}^{L} \sum_{u \in V_p(T_i)} d(s_i, u). \tag{3.9}$$

Proof. We follow the same procedure of Theorem 3.1.3. □

Now, we generalize Theorem 3.1.4 to the special case where the path-tree is built from L isomorphic trees.

Corollary 3.1.2. *Let $\mathcal{P}_L$ be a path-tree composed of L copies of the same tree T of order N and with p pendent vertices. Then,*

- *If s_i is a non-pendent vertex, for $i = \{1, 2, ..., L\}$*

$$TW(\mathcal{P}_L) = L\,TW(T) + pL(L-1)\sum_{u\in V_p(T)} d(s_1, u) + \frac{L(L^2-1)}{6}p^2. \tag{3.10}$$

- *If s_i is a pendent vertex, for $i = \{1, 2, ..., L\}$*

$$TW(\mathcal{P}_L) = L\,TW(T_N) + [L(L-1)(p-1) - L]\sum_{u\in V_p(T)} d(s_1, u) + \frac{L(L^2-1)(p-1)^2}{6}. \tag{3.11}$$

3.1.4 Computation of the Terminal Wiener Index of some Families of Path-Trees

In this subsection, we study some interesting graphs that can be treated as a path-tree. Evidently, we apply the previously derived results to compute the Terminal Wiener index of these families of path-trees.

3.1.4.1 The case of a path-tree composed of L stars

Let $\mathcal{A}_L$ and $\mathcal{A}'_L$ be two path-trees composed of L stars S_N of order N. The difference between the two structures $\mathcal{A}_L$ and $\mathcal{A}'_L$ is that in the first one the vertices of concatenation s_i are non-pendent vertices and for the second graph the vertices s_i are pendent. See Figure 3.7 for illustration.

Our aim is to find the Terminal Wiener index of these two path-trees. We previously defined the Terminal Wiener index of a star S_N of order N and the distance between the pendent vertices of S_N and the vertex of concatenation s_i. Thus, by using Corollary 3.1.2, we get the Terminal Wiener index of the path-trees $\mathcal{A}_L$ and $\mathcal{A}'_L$.

$$TW(\mathcal{A}_L) = L(N-1)(N-2) + L(N-1)^2\frac{L^2+6L-7}{6}. \tag{3.12}$$

$$TW(\mathcal{A}'_L) = L(N-2)(N-3) + L(N-2)^2\frac{L^2+12L-13}{6}. \tag{3.13}$$

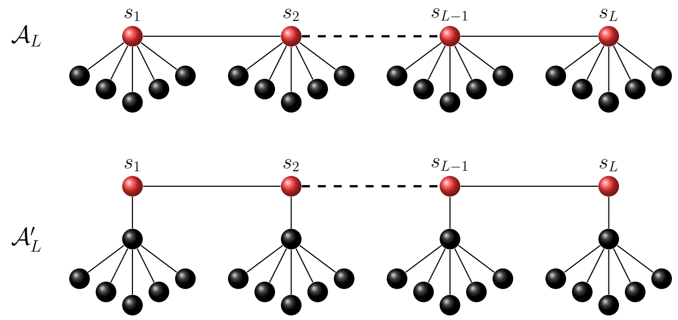


Figure 3.7: The two path-trees $\mathcal{A}_L$ and $\mathcal{A}'_L$.

In the following table, we show some numerical results for the Terminal Wiener index of the two path-trees $\mathcal{A}_L$ and $\mathcal{A}'_L$. The results are obtained by changing the order of the star S_N and the number of copies L.

	$TW(\mathcal{A}_L)$				$TW(\mathcal{A}'_L)$			
L	N=10	N=20	N=30	N=40	N=10	N=20	N=30	N=40
2	387	1767	4147	7527	432	2232	5432	10032
3	1026	4636	10846	19656	1192	6102	14812	27322
4	2070	9310	21750	39390	2400	12240	29680	54720
5	3600	16150	37700	68250	4120	20970	50820	93670
6	5697	25517	59537	107757	6416	32616	79016	145616
7	8442	37772	88102	159432	9352	47502	115052	212002
8	11916	53276	124236	224796	12992	65952	159712	294272
9	16200	72390	168780	305370	17400	88290	213780	393870
10	21375	95475	222575	402675	22640	114840	278040	512240

Table 3.3: Numerical results for the Terminal Wiener index of path-trees $\mathcal{A}_L$ and $\mathcal{A}'_L$

From the above table and Figure 3.8, we notice that the Terminal Wiener index of the two structures $\mathcal{A}_L$ and $\mathcal{A}'_L$ shows an increasing change with the growth of the order N and the number of copies L.

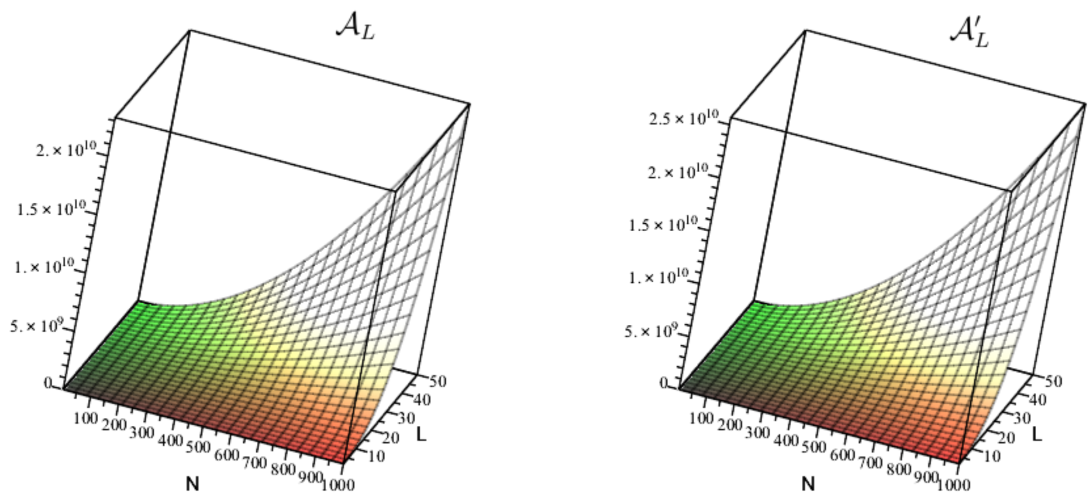


Figure 3.8: The graphical behavior of the Terminal Wiener index of the two structures $\mathcal{A}_L$ and $\mathcal{A}'_L$.

3.1.4.2 The case of a path-tree named Binomial Tree

The binomial tree is a kind of recursive networks, which looks like a path-tree composed of two isomorphic structures. Let $\mathcal{B}_k$ be a binomial tree, where k denotes the number of iterations. The $\mathcal{B}_k$ can be build in the following iterative way. Initially, $\mathcal{B}_0$ consists of a single vertex. $\mathcal{B}_1$ is obtained from two binomial trees $\mathcal{B}_0$, such that we connect them by introducing an edge $s's$. In general, for any $k \geq 1$, the binomial tree $\mathcal{B}_k$ is constructed from two binomial trees $\mathcal{B}_{k-1}$ by connecting them with an edge $s's$. The Figure 3.9 illustrates an example of binomial trees. A binomial tree $\mathcal{B}_k$ with k iterations has 2^k vertices.

In order to compute the Terminal Wiener index of the binomial tree $\mathcal{B}_k$, we set the distance between the root s and the pendent vertices of $\mathcal{B'}_{k-1}$ as $d_{\mathcal{B'}_{k-1}}(s)$. Then, by using this notation, we have

$$d_{\mathcal{B'}_k}(s) = 2^{k-2} + d_{\mathcal{B'}_{k-1}}(s) + d_{\mathcal{B'}_{k-2}}(s) + ... + d_{\mathcal{B'}_1}(s), \tag{3.14}$$

and

$$d_{\mathcal{B'}_{k-1}}(s) = 2^{k-3} + d_{\mathcal{B'}_{k-2}}(s) + d_{\mathcal{B'}_{k-3}}(s) + ... + d_{\mathcal{B'}_1}(s). \tag{3.15}$$

The two equations 3.14 and 3.15 yield

$$d_{\mathcal{B}'_k}(s) = 2^{k-3} + 2d_{\mathcal{B}'_{k-1}}(s) = (k+2)2^{k-3}.$$

From the construction method of a binomial tree, we can see that

$$\sum_{u \in V_p(\mathcal{B}_{k-1})} d(s,u) = \sum_{u \in V_p(\mathcal{B}'_{k-1})} d(s',u) = d_{\mathcal{B}'_k}(s) - 2^{k-2} = k2^{k-3}. \tag{3.16}$$

Whereof, by applying the equations 3.6 and 3.16, we obtain

$$TW(\mathcal{B}_k) = \begin{cases} 3 & \text{for } k = 2, \\ \\ 2TW(\mathcal{B}_{k-1}) + 2^{2k-4}(k+1) & \text{for } k \geq 3. \end{cases} \tag{3.17}$$

Solving the above equation, we found that the Terminal Wiener index of a binomial tree $\mathcal{B}_k$ is given by:

$$TW(\mathcal{B}_k) = k2^{2k-3} - 2^{k-2} \qquad \forall k \geq 2. \tag{3.18}$$

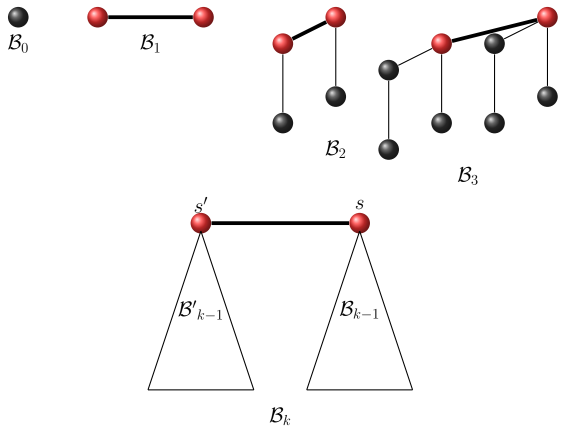


Figure 3.9: The construction method of binomial trees, such that $\mathcal{B}_{k-1} = \mathcal{B}'_{k-1}$.

In the following table, we present some values of the Terminal Wiener index of the binomial tree $\mathcal{B}_k$. We remark that the values of the Terminal Wiener index increase by increasing the number of iterations k.

k	$TW(\mathcal{B}_k)$
2	3
3	22
4	124
5	632
6	3056
7	14304
8	65472
9	294784
10	1310464
11	5766656
12	25164800
13	109049856
14	469757952
15	2013257728
16	8589918208
17	36507189248
18	154618757120
19	652834897920
20	2748778807296

Table 3.4: Some values of the Terminal Wiener index of the binomial tree $\mathcal{B}_k$.

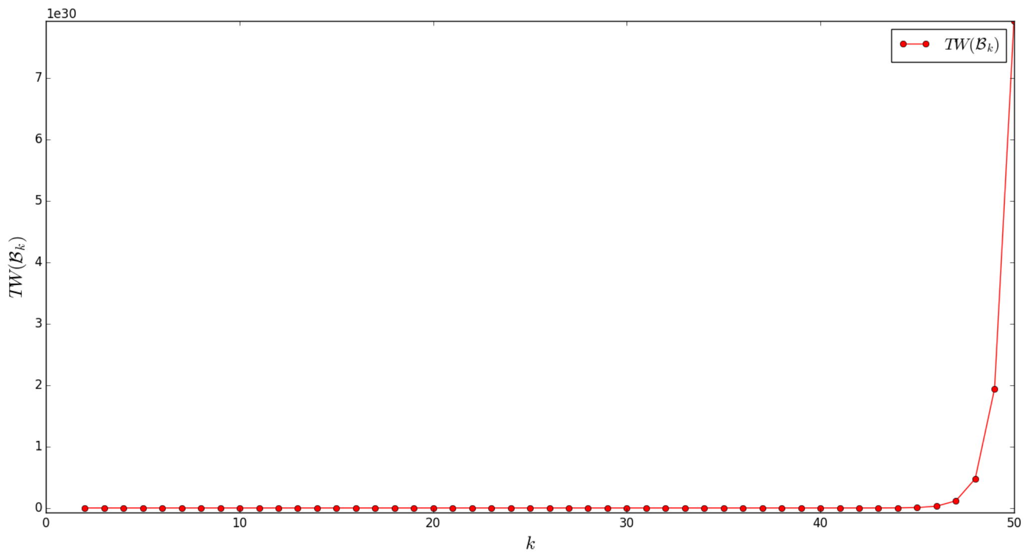


Figure 3.10: The graphical representation of the Terminal Wiener index of the binomial tree $\mathcal{B}_k$.

Also from the graphical representation of the Terminal Wiener index of the binomial tree $\mathcal{B}_k$, which is shown in Figure 3.10, we can observe that the behavior of this index increases and shows a dominant change in some iterations k.

3.2 The Thorny Graph Concept For Calculating The Terminal Wiener Index Of Some Structures

In the previous section, we investigated a kind of important networks that are built from several copies of components and discussed the efficient method for computing the topological indices of such structures. Here, we introduce another type of compulsory networks that are constructed from their subgraph, also called parent graph, by adding some new vertices. The first idea that occurs in order to calculate the topological indices of these networks is to delete some vertices and compute the topological indices of the remaining subgraph. In other words, we generate the property of the whole structure in terms of the topological indices of its parent graph. This efficient method is based on a great concept called the thorny graph, which was introduced by Gutman (1998) to reduce the computation of the Wiener index of some graphs. Later, this concept was found a variety of applications, such as in studying the relationship between the topological indices of a two famous types of molecular graphs named Kenograph and Plerograph (Gutman et al., 2013b), (Gutman et al., 2013a); see Figure 3.12. In this section, we focus on the application of the concept thorny graph to reduce the computation of the Terminal Wiener index of some structures.

3.2.1 Definition of Thorn Graphs

Let G be a graph of order N. The thorn graph G^* is the graph obtained from G by attaching p_i new pendent vertices to each vertex $v_i \in V(G)$, for $i = 1, 2, ..., N$. The p_i new vertices attached to the vertex v_i are called thorns of v_i and G is called the parent graph of G^*. Figure 3.11 illustrates an example of a thorn graph G^* and its parent graph G.

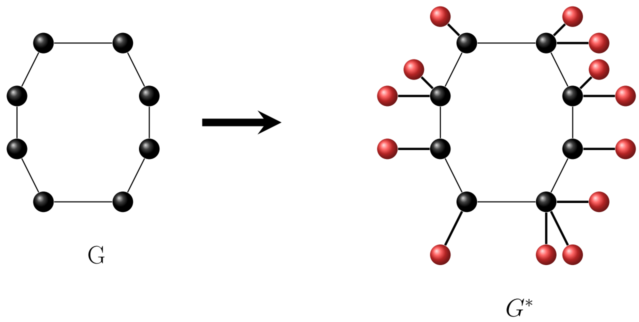


Figure 3.11: A thorn graph G^* and its corresponding parent graph G.

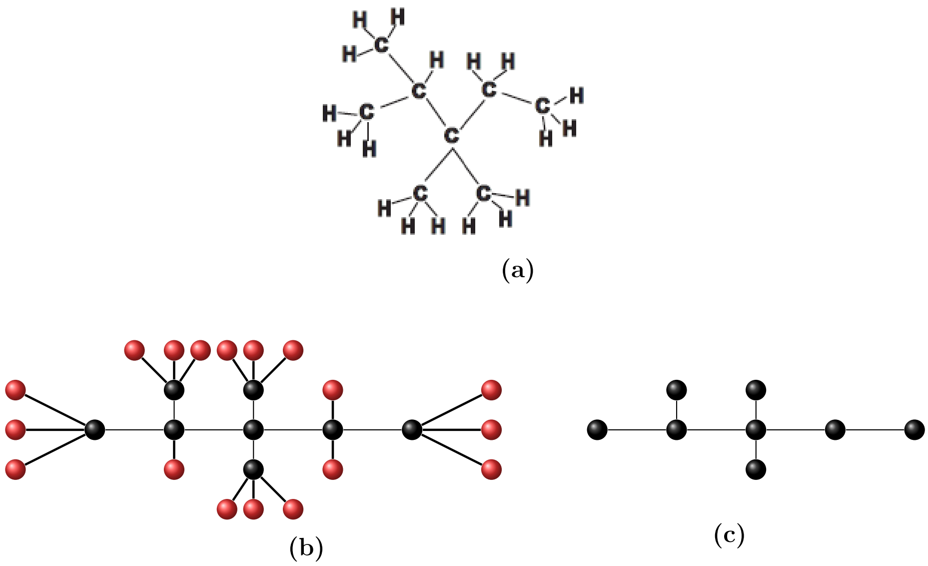


Figure 3.12: Plerograph and Kenograph representations of a molecule. **(a):** The molecule formula of 2,3,3-trimethylpentane. **(b):** The Plerograph representation of 2,3,3-trimethylpentane that is viewed as a 4-thorn graph. **(c):** The Kenograph representation of 2,3,3-trimethylpentane, which is considered as a parent graph.

Special attention has paid to a kind of thorn graphs determined by the condition $p_i = d_{max} - d_i$, where d_{max} is a constant and d_i is the degree of the i^{th} vertex in G. Then, the vertices of the thorn graph G^* are either of degree d_{max} or of degree one. In chemistry, this thorn graph G^* is called a Plerograph; a graph in which every atom is represented by a vertex, and the parent graph G is referred to as a Kenograph; a graph obtained from a Plerograph by suppressing hydrogen atoms (Cayley, 1874). For more clarification about these two molecular representations see Figure 3.12. In general, the thorny graphs with a uniform degree d of all non-terminal vertices are called d-thorn graphs.

3.2.2 Terminal Wiener Index of Thorn Graphs

Since the first attempt done by Gutman (1998), in which he derived a formula that relates the Wiener index of a thorn graph G^* with the Wiener index of its parent graph G, a lot of studies on different topological indices of thorn graphs like modified Wiener index (Vukičević and Graovac, 2004) and Hosoya polynomial (Walikar et al., 2006) have been considered. The aim of this subsection is to report some results done on the Terminal Wiener index of thorn graphs (Heydari and Gutman, 2010).

Theorem 3.2.1. *(Heydari and Gutman, 2010) Let G^* be a thorn graph obtained from a graph G of order N by attaching p_i new pendent vertices to the vertex v_i of G, for $i = 1, 2, ..., N$. Then,*

$$TW(G^*) = 2\sum_{i=1}^{N} \binom{p_i}{2} + \sum_{1 \le i < j \le N} p_i p_j \Big[d_G(v_i, v_j) + 2 \Big]. \tag{3.19}$$

In the case of $p_1 = p_2 = ... = p_N = p > 0$, we have the following generalization.

Corollary 3.2.1. *(Heydari and Gutman, 2010) Let G^* be a thorn graph obtained by attaching p pendent vertices to all the vertices of G. Then,*

$$TW(G^*) = p^2 W(G) + pN(pN - 1). \tag{3.20}$$

From Equation 3.20, we can see that the Terminal Wiener index of a thorn graph is related to the Wiener index of the parent graph. Now, we show the relation between the Terminal Wiener index of a thorn graph and the Terminal Wiener index of its parent graph.

Theorem 3.2.2. *(Heydari and Gutman, 2010) Let G be a graph of order N and with k pendent vertices. Let G^* be a thorn graph obtained by adding p new pendent vertices to* *each terminal vertex of G. Then*

$$TW(G^*) = p^2 TW(G) + pk(pk-1). \tag{3.21}$$

A graphical illustration of the two relations 3.20 and 3.21 is presented in Figure 3.13. We generated all trees T of order $N = 10$ and we calculated the Terminal Wiener index of all thorn graphs T_1^* and T_2^* that are obtained by adding $d = 3$ new pendent vertices to all vertices and terminal vertices of the trees T, respectively. The red data points represent the relation between the Wiener index of T and the Terminal Wiener index of the corresponding thorn graph T_1^* by applying the Equation 3.20. This graphical representation shows that there exist an exact linear relation between $W(T)$ and $TW(T_1^*)$.

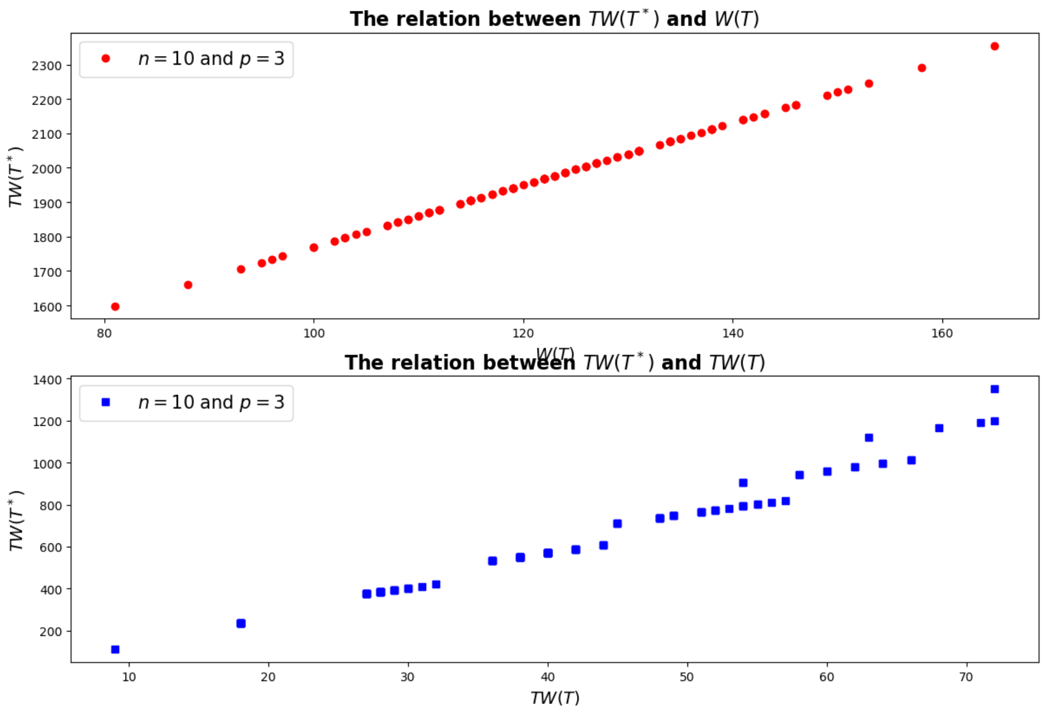


Figure 3.13: The relation between Wiener index (Terminal Wiener index) of a parent graph and the The Terminal Wiener index of the corresponding thorn graphs in the case of $N = 10$ and $d = 3$.

The blue data points represent the relation between the Terminal Wiener index of T and the Terminal Wiener index of the corresponding thorn graph T_2^* by applying the

Equation 3.21. The first detail that is noticed from this graphical representation is that $TW(T)$ and $TW(T_2^*)$ are not linearly correlated. Moreover, we can see that the blue data points are grouped into several clusters, such that each cluster corresponds to a particular value of pendent vertices k. For example, the single blue data point on the most left-hand side of Figure 3.13 corresponds to a tree T with $k = 2$ pendent vertices, the second cluster on the left-hand side corresponds to trees T with $k = 3$, the third cluster corresponds to trees T with $k = 4$ and so on until the last cluster on the most right-hand side that corresponds to a tree T with $k = 9$ pendent vertices.

3.2.3 On the Terminal Wiener Index of some Structures as Thorn Graphs

In this subsection, we use the mentioned results on the calculation of the Terminal Wiener index of thorn graphs to present our work concerning the method for computing the Terminal Wiener index of a kind of structures represented as thorn graphs. Also, we analyze the behavior of the Terminal Wiener index of these networks using graphical representations.

Let G be a cycle graph of order N and let G_0^* be the thorn graph obtained by attaching $p_i = 1$ new pendent vertex to each vertex of the parent graph G. Figure 3.14 illustrates an example of a thorn graph G_0^*.

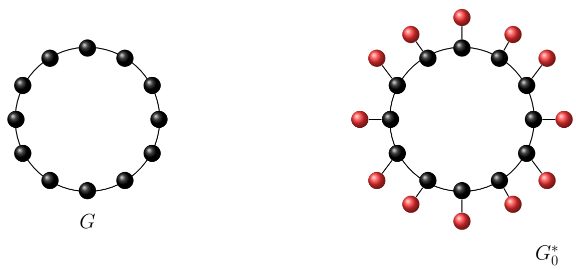


Figure 3.14: The cycle graph G and the thorn graph G_0^*

Starting from the thorn graph G_0^*, we construct a large structure G_h^* that is obviously represented as a thorn graph. The construction method of this structure is defined as follows. Let h be a positive integer, the thorn graph G_h^* is obtained by attaching $p_i = p$

new pendent vertices to each pendent vertex of G^*_{h-1}. An example of a thorn graph G^*_1 that is obtained from the parent graph G^*_0 is depicted in Figure 3.15.

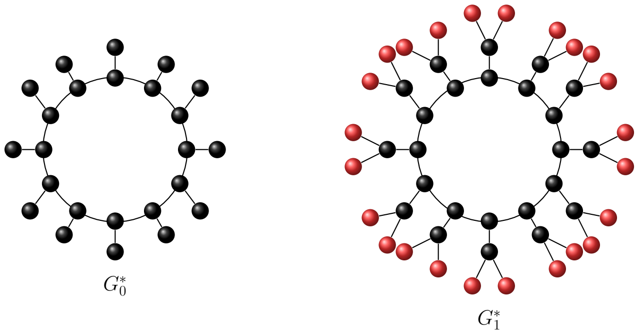


Figure 3.15: The thorn graph G^*_1 and its parent graph G^*_0.

In order to calculate the Terminal Wiener index of the structure G^*_h, with $h \geq 1$, we start by defining the Terminal Wiener index of the graph G^*_0.

Lemma 3.2.1. *Let G^*_0 be the thorn graph obtained from the cycle graph G of order N. Then,*

$$TW(G^*_0) = \begin{cases} \frac{1}{8}N^3 + N(N-1); & \textit{if } N \textit{ is even,} \\ \\ \frac{1}{8}N^3 + \frac{1}{8}N(8N-9); & \textit{if } N \textit{ is odd.} \end{cases} \tag{3.22}$$

Proof. By using the Equation 3.20, which relates the Terminal Wiener index of a thorn graph to the Wiener index of the parent graph, and the value of the Wiener index of a cycle graph that is defined in Equation 2.9, we can easily get the result. □

In the following theorem, we calculate the Terminal Wiener index of the structure G^*_h.

Theorem 3.2.3. *Let G^*_h be the thorn graph obtained from the parent graph G^*_{h-1}. Then,*

$$TW(G^*_h) = \begin{cases} Np^{2h}\left[\frac{1}{8}N^2 + N(1+h) - 1\right] - \frac{p^h-1}{p-1}Np^h; & \textit{if } N \textit{ is even,} \\ \\ Np^{2h}\left[\frac{1}{8}N^2 + N(1+h) - \frac{9}{8}\right] - \frac{p^h-1}{p-1}Np^h; & \textit{if } N \textit{ is odd.} \end{cases} \tag{3.23}$$

Proof. By using the recursive relation between the Terminal Wiener index of a thorn graph and the Terminal Wiener index of the parent graph, we have:

$$TW(G_h^*) = p^2 TW(G_{h-1}^*) + pk_{h-1}^*(pk_{h-1}^* - 1),$$

where k_h^* is the number of pendent vertices of the graph G_h^*, and we have:

$$k_h^* = pk_{h-1}^* = p^h k_0^*.$$

By recurrence, we obtain:

$$TW(G_h^*) = p^{2h} TW(G_0^*) + hp^{2h} k_0^{*2} - \frac{k_0^* p^h (p^h - 1)}{p - 1}.$$

which yields Equation 3.23. □

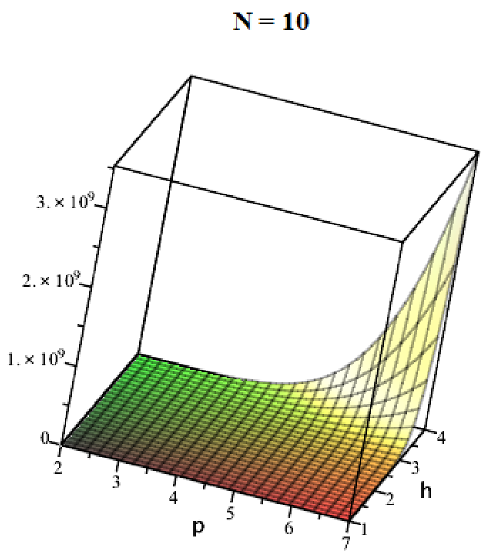


Figure 3.16: The graphical behavior of the Terminal Wiener index of the graph G_h^*.

An analysis of the behavior of the Terminal Wiener index of the structure G_h^* is given in Figure 3.16. We have fixed the order of the cycle graph G on the value $N = 10$. In the X-axis and Y-axis, we have the number of the thorns p and the number of iterations h, respectively. Along the Z-axis the values of the Terminal Wiener index are shown. From this graphical representation, we notice that the Terminal Wiener index of this structure is controlled by the parameters p, h and N. In other words, the Terminal Wiener index increases with a high increase in the number of thorns p and the number of iterations h. In general, the Terminal Wiener index of the structure G_h^* shows a dominant change if we consider a cycle graph G with a high order N.

Now, we consider another structure represented as thorn graphs and we calculate its Terminal Wiener index. Let $\mathcal{G}_0^*$ be the graph shown in Figure 3.17, such that $\frac{N}{2}$ pendent vertices are attached to a cycle G of order N. We should note that the parity of the order N is always even. For positive integer h, the graph $\mathcal{G}_h^*$ is obtained by attaching p new pendent vertices to each pendent vertex of $\mathcal{G}_{h-1}^*$. In Figure 3.17, we represent the graph $\mathcal{G}_1^*$ that is obtained from the parent graph $\mathcal{G}_0^*$ by attaching to each pendent vertex $p = 2$ new pendent vertices.

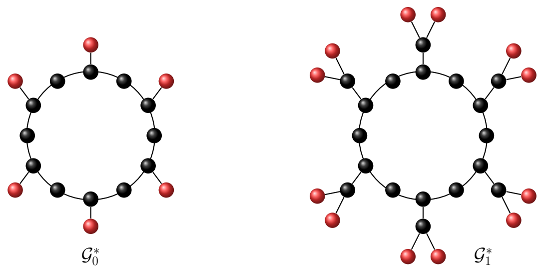


Figure 3.17: The thorn graph $\mathcal{G}_1^*$ and its parent graph $\mathcal{G}_0^*$.

At first, we calculate in the following lemma the Terminal Wiener index of the graph $\mathcal{G}_0^*$.

Lemma 3.2.2. *Let $\mathcal{G}_0^*$ be the graph obtained from a cycle graph G of order N. Then*

$$TW(\mathcal{G}_0^*) = \begin{cases} \frac{1}{2}N\left[\frac{1}{16}N^2 + \frac{1}{2}N - 1\right]; & \text{if } \frac{N}{2} \text{ is even,} \\ \\ \frac{1}{32}N^3 + \frac{1}{4}N^2 - \frac{5}{8}N; & \text{if } \frac{N}{2} \text{ is odd.} \end{cases} \tag{3.24}$$

Proof. We express the Terminal Wiener index of $\mathcal{G}_0^*$ in terms of the distance between the vertices of degree 3 in $\mathcal{G}_0^*$. Then,

$$\begin{aligned} TW(\mathcal{G}_0^*) &= \sum_{\substack{\{u,v\}\subseteq V(\mathcal{G}_0^*)\\ deg(u)=deg(v)=3}} \Big[d(u,v)+2\Big] \\ &= \sum_{\substack{\{u,v\}\subseteq V(\mathcal{G}_0^*)\\ deg(u)=deg(v)=3}} d(u,v) + \frac{N}{2}\left[\frac{N}{2}-1\right]. \end{aligned}$$

By calculation, we get

$$\sum_{\substack{\{u,v\}\subseteq V(\mathcal{G}_0^*)\\ deg(u)=deg(v)=3}} d(u,v) = \begin{cases} \frac{1}{32}N^3 & \text{if } \frac{N}{2} \text{ is even,} \\ \\ \frac{1}{32}N^3 - \frac{1}{8}N & \text{if } \frac{N}{2} \text{ is odd.} \end{cases} \tag{3.25}$$

In the above relation of the Terminal Wiener index of $\mathcal{G}_0^*$, we replace the left term of the equation by its value, and then the proof is completed. □

Finally, we calculate the Terminal Wiener index of the structure $\mathcal{G}_h^*$, with $h \geq 1$.

Theorem 3.2.4. *Let $\mathcal{G}_h^*$ be the thorn graph obtained from the parent graph $\mathcal{G}_{h-1}^*$. Then,*

$$TW(\mathcal{G}_h^*) = \begin{cases} \frac{1}{2}Np^{2h}\left[\frac{1}{16}N^2 + \frac{1}{2}N(1+h) - 1\right] - \frac{1}{2}\frac{p^h-1}{p-1}Np^h & \textit{if } \frac{N}{2} \textit{ is even,} \\ \\ \frac{1}{4}Np^{2h}\left[\frac{1}{8}N^2 + N(1+h) - \frac{5}{2}\right] - \frac{1}{2}\frac{p^h-1}{p-1}Np^h & \textit{if } \frac{N}{2} \textit{ is odd.} \end{cases} \tag{3.26}$$

Proof. By using the recursive relation between the Terminal Wiener index of a thorn graph and the Terminal Wiener index of the parent graph, we have:

$$TW(\mathcal{G}_h^*) = p^2 TW(\mathcal{G}_{h-1}^*) + pk'_{h-1}(pk'_{h-1} - 1).$$

The number of pendent vertices of $\mathcal{G}_h^*$ is equal to:

$$k'_h = pk'_{h-1} = p^h k'_0.$$

By solving the recurrence relation, we get

$$TW(\mathcal{G}_h^*) = p^{2h}TW(\mathcal{G}_0^*) + hp^{2h}k_0'^2 - \frac{k_0'p^h(p^h-1)}{p-1}.$$

Therefore, we obtain the result. □

Figure 3.18 shows the behavior of the Terminal Wiener index for the structure $\mathcal{G}_h^*$. We represent along the X-axis and Y-axis the values of the p thorns and the number of iterations h, respectively. We considered an example of a cycle graph of order $N = 30$. Similarly, we figure out from the graphical representation that the Terminal Wiener index is increasing by changing the values of the parameters p and h, and shows a dominant change if we consider cycle graphs with high order N.

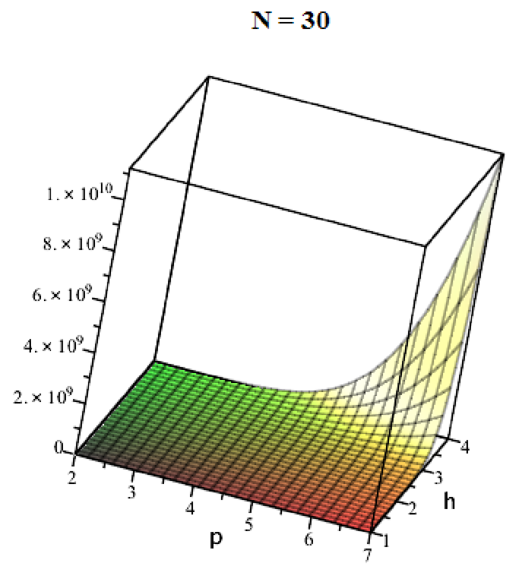


Figure 3.18: The graphical behavior of the Terminal Wiener index of the graph $\mathcal{G}_h^*$.

3.3 A Computation Method Based On A Re-formula Of The Terminal Wiener Index

From the beginning of this chapter, we mentioned that in the case of networks with a large number of vertices the direct application of the definition of topological indices is a challenging task. Another method to overcome the problem of calculating any distance-based topological index is to propose an amelioration of the basic definition of the corresponding index. In this section, we define a concept that can be used to reformulate the definition of some topological indices and we focus on the rewrite of the Terminal Wiener index. Then, as an application of the re-formula of the Terminal Wiener index, we analyze the topological structure of a well known network called the Dendrimer tree.

Let $d_G(k)$ be the number of vertex pairs of the graph G that are at distance k. By using this notation, we can rewrite some distance-based topological indices (Brückler et al., 2011). For example, the Wiener index can be expressed via the term $d_G(k)$ as:

$$W(G) = \sum_{k\geq 1} k d_G(k). \tag{3.27}$$

In (Essalih et al., 2011) and (Essalih, 2013), the Equation 3.27 was applied to calculate the Wiener index of some graphs with different diameters, such as the Fan Graph, Butterfly graph, Sunflower graph and Double Crystal graph. In some papers (Alizadeh et al., 2014), the Equation 3.27 was used to study the Wiener dimension of some families of graphs, which is defined as the number of different distances of its vertices. An other advantage of this re-formula is that the quantity $d_G(k)$ is well defined even if we have a disconnected graph.

In order to ameliorate the definition of the Terminal Wiener index, let $d_p(k)$ denotes the number of pendent vertex pairs of the graph G that are at distance k. Then, the following theorem represents a re-formula of the Terminal Wiener index.

Theorem 3.3.1. *Let G be a graph of order N and with $p \geq 2$ pendent vertices. let $D > 2$ denotes the diameter of G. Then,*

$$TW(G) = p(p-1) + d_p(3) + 2d_p(4) + ... + (D-2)d_p(D). \tag{3.28}$$

Proof. The Terminal Wiener index can be expressed via the term $d_p(k)$ as:

$$TW(G) = \sum_{k\geq 1} k d_p(k).$$

Evidently, the graph G have $\binom{p}{2}$ pairs of pendent vertices. Then,

$$d_p(2) + d_p(3) + ... + d_p(D) = \binom{p}{2}.$$

and

$$d_p(2) = \frac{p(p-1)}{2} - d_p(3) - ... - d_p(D).$$

which gives the Equation 3.28. □

3.3.1 Application of the Re-formula on Dendrimer Trees

In this subsection, we apply the rewrite of the Terminal Wiener index to analyze the topological structure of a kind of hierarchical networks named Dendrimer tree.

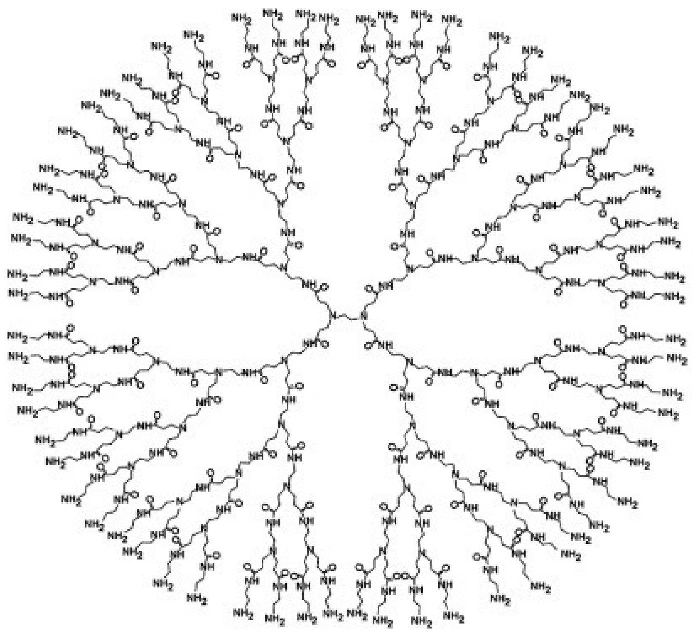


Figure 3.19: An example of the most commonly studied Dendrimer in biomedical applications.

The Dendrimer trees are hyper branched macromolecules, which are suitable for a wide range of biomedical and industrial applications (Malkoch et al., 2012), such as drug delivery and gene delivery. In Figure 3.19, we depicted an example of a dendrimer tree that is widely used in biomedical applications. We can also found the dendrimer topology in other fields of applications, such as to analyze communication networks (Xiao and Parhami, 2006). In this case of applications the dendrimer topology is called a Cayley tree.

Let $\mathcal{T}_{d,h}$ $(d \geq 3, h \geq 0)$ be a Dendrimer tree with two additional parameters: fixed maximum degree d and the number of iterations or depth h. The Dendrimer tree can be built in the following iterative way. Initially, $\mathcal{T}_{d,0}$ consists of only a central vertex that is the core of the Dendrimer tree. $\mathcal{T}_{d,1}$ is obtained by attaching d vertices to the central vertex. For any $h > 1$, we obtain $\mathcal{T}_{d,h}$ from $\mathcal{T}_{d,h-1}$ by attaching $d-1$ new vertices to the pendent vertices of $\mathcal{T}_{d,h-1}$. Figure 3.20 illustrates an example of Dendrimer trees with $d = 3$ and $h = \{0, 1, 2, 3\}$. Every internal vertex of the Dendrimer tree has degree d, and the iterations h denote the distance between all pendent vertices (red vertices) and the core vertex.

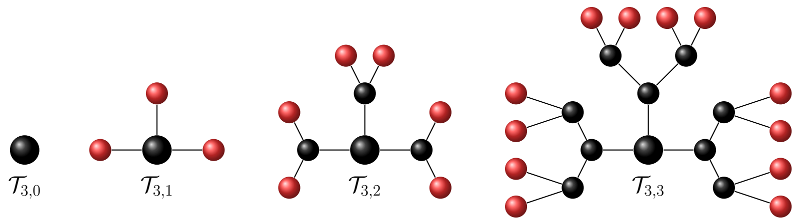


Figure 3.20: Dendrimer trees $\mathcal{T}_{d,h}$ with $d = 3$ and $h = \{0, 1, 2, 3\}$

From the construction method of the Dendrimer tree $\mathcal{T}_{d,h}$, we can extract the following structural properties:

- The number of pendent vertices of $\mathcal{T}_{d,h}$ is:

$$p_h = d(d-1)^{h-1}, \qquad \textit{with } h \geq 1$$

- The number of vertices of $\mathcal{T}_{d,h}$ is:

$$N_h = 1 + d\frac{(d-1)^h - 1}{d-2}.$$

- The diameter D_h of $\mathcal{T}_{d,h}$ is equal to:

$$D_h = 2h.$$

In order to apply the Equation 3.28, we start with the following lemma that is defined as follows:

Lemma 3.3.1. *For any Dendrimer tree $\mathcal{T}_{d,h}$. The $d_p(k)$, for $k = \{4, 6, 8, ..., D_h\}$, is given by:*

$$d_p(k) = \begin{cases} \frac{d(d-2)}{2}(d-1)^{\frac{2h+k-4}{2}} & \text{if } k \leq 2(h-1), \\ \\ \frac{d}{2}(d-1)^{k-1} & \text{if } k = 2h. \end{cases} \tag{3.29}$$

Proof. We have the distance between pendent vertices of the Dendrimer tree $\mathcal{T}_{d,h}$ is always even. In the first iteration, d vertices are attached to the central vertex, and in the next iterations, each pendent vertex v_i is attached to $(d-1)$ vertices. Obviously, with some calculations and due to the symmetry of this structure, we can get the result. □

Theorem 3.3.2. *Let $\mathcal{T}_{d,h}$ be a Dendrimer tree with $h \geq 1$ and $d \geq 3$. Then,*

$$TW(\mathcal{T}_{d,h}) = d(d-1)^{h-1}\left[(d-1)^{h-1}\left(hd - \frac{d-1}{d-2}\right) - 1 + \frac{d-1}{d-2}\right]. \tag{3.30}$$

Proof. By using Theorem 3.3.1, Lemma 3.3.1 and the structural properties of $\mathcal{T}_{d,h}$, we get:

$$\begin{aligned} TW(\mathcal{T}_{d,h}) &= p_h(p_h - 1) + 2d_p(4) + 4d_p(6) + ... + (D_h - 2)d_p(D_h) \\ &= d(d-1)^{h-1}[d(d-1)^{h-1} - 1] + \frac{d(d-2)}{2}\sum_{i=1}^{h-2} 2i(d-1)^{h-1+i} + \\ &\quad d(h-1)(d-1)^{2h-1}. \end{aligned}$$

which yields the Equation (3.30). □

We show in Figure 3.21 the behavior of the Terminal Wiener index of the dendrimer tree $\mathcal{T}_{d,h}$. Along the X-axis and Y-axis, we have the maximum degree d and the number of iterations h, respectively. In the Z-axis the values of the Terminal Wiener index of $\mathcal{T}_{d,h}$

are determined. From this graphical representation, we observe that the Terminal Wiener index of dendrimer trees shows a dominant change with the high increasing values of the two parameters d and h.

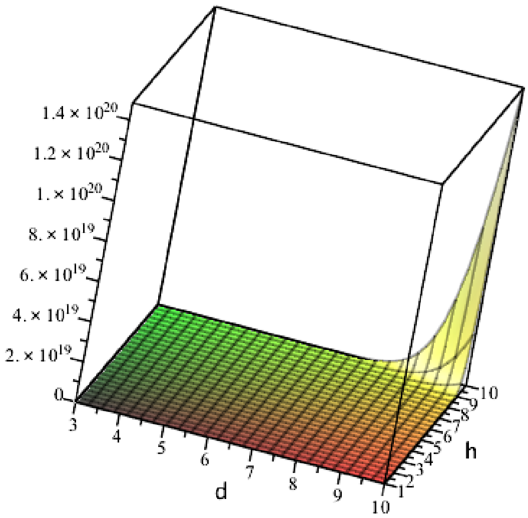


Figure 3.21: The graphical representation of the Terminal Wiener index of the Dendrimer tree $\mathcal{T}_{d,h}$

3.4 Summary

The calculation of the Terminal Wiener index by using the basic definition is a difficult computational problem and especially when dealing with networks of a large number of vertices. In this chapter, we discussed three methods for facilitating the computation of the Terminal Wiener index. the first technique is a decomposition method to treat a kind of graphs called star-trees and path-trees, which are built from several copies of components. The objective of this technique was to prove that we can generate the property of a star-tree or a path-tree by calculating the Terminal Wiener index of the individual components. We also considered the important special case where the star-trees and path-trees are built from several copies of the same graph. Then, we used the obtained main results to calculate the Terminal Wiener index of some particular examples, such as a star-tree composed of N paths, a star-tree composed of N stars, a path-tree composed of N stars and binomial trees. For the second method, which is based on an important concept called the thorny graph, we showed that the Terminal Wiener index of a thorn graph can be obtained from the Terminal Wiener index of its parent graph. Then, we applied this relation to derive the Terminal Wiener index of some structures represented as thorn graphs. The last method was an amelioration of the definition of the Terminal Wiener index. We proved that the Terminal Wiener index of a graph G can be calculated in terms of the number of pendent vertex pairs of G that are at distance k. After that, we used this re-formula to extract analytical expression for the Terminal Wiener index of a well known structure called the dendrimer tree or Cayley tree.

Briefly, the considered methods of this chapter are able to overcome the problem of calculating the Terminal Wiener index of a kind of networks, such as path-trees, star-trees, structures as thorn graphs and dendrimers. We can see that the common point between these three approaches is the application of the concept of distance between a set of vertices. In the case of some other complicated structures, such as hexagonal systems and silicate networks, the complexity of these methods will be dominated by computing the shortest paths between the corresponding vertices. Based on this limitation, we investigate in the next chapters some effective methods for calculating any distance-based topological index by avoiding the computation of distances.

Chapter

4 On the Cut Method and its efficiency for quantifying the topological structure of some Complex Networks

Contents

In the last chapter, we proposed three methods to facilitate the computation of the Terminal Wiener index, which are the decomposition method of graphs containing several copies of components, the use of the concept of thorny graphs and a rewrite of the definition of the Terminal Wiener index. All of these methods are based on the use of the concept of distance, which makes their complexity dominated by computing the shortest paths between the corresponding vertices. Hence, the fundamental purpose of this chapter would be to propose some new techniques to calculate the Terminal Wiener index and its generalization efficiently by avoiding the computation of shortest paths.

A powerful approach called the cut method was initiated by Klavzar et al. (1995) to facilitate the computation of the Wiener index of a class of graphs called partial cubes. The objective of this method is to cut the associated network into smaller components and assembling the indices of components to generate the property of the whole structure.

There exist two versions of the cut method: a method based on edge-partitions and a method based on vertex-partitions. Our main focus is to extend and ameliorate these two versions of the cut method for computing the generalized Terminal Wiener index. At first, we introduce the first technique based on edge-partitions, also called edge-cut method. We give an overview on the basic notions that are required to figure out this technique. Then, we discuss two types of the edge-cut method that depend on the class of graphs to which being applied. After that, we move to the second version based on vertex-partitions, also called vertex-cut method. We explore in depth this approach and we bring a nice formula based on vertex-partitions for computing the generalized Terminal Wiener index. For each technique, we demonstrate its efficiency and significance by computing the generalized Terminal Wiener index and other measures of some complex networks, such as Equilateral Triangular Tetra Sheet networks, Hex Derived networks and Silicate networks. We note that the obtained results were presented in two international conferences (Zeryouh et al., 2017), (Zeryouh et al., 2018b).

4.1 Edge-Cut Method And Its Versions

The edge-cut method, also called edge-partitions, is a powerful technique that facilitates the computation of distance-based topological indices of large and complex networks. For a given structure G, the main idea of this method is described as follows. At first, we partition the edge set of G into classes $E_1, ..., E_q$, called cuts, such that the graphs $G - E_i$, $i = 1, ..., q$, consist of two or more connected components. After that, we assemble the indices of the components to generate the property of the whole structure G.

The edge-cut method is classified into two versions based on the class of networks to which it is being applied. A standard cut method is applied to a type of networks that fall under partial cubes and we have extended cut methods that are applied to classes larger than partial cubes. For additional information on the edge-cut method, we refer to the survey by Klavzar and J Nadjafi-Arani (2015). In the following subsection, we give an overview on the basic notions and concepts that are required to figure out and apply the edge-cut method.

4.1.1 The Relation Θ and Partial Cubes

In Chapter 1, we presented all the important concepts of graphs. Here, we recall the definition of some important notions needed later. We introduce the class of partial cubes and its characterizations. Then, we define the famous relation Θ and its basic properties.

We start with the definition of convex subgraphs and isometric subgraphs. A subgraph H of a graph G is called a convex subgraph if for any two vertices u and v in H, every shortest $u-v$ path in G belongs completely to H. If $d_H(u, v) = d_G(u, v)$ for any two vertices u and v of H, then H is an isometric subgraph of G. We defined the hypercube of dimension n to be the graph Q_n whose vertex set consists of all binary strings of length n, such that two strings are adjacent if they differ in exactly one position. Figure 4.1 illustrates some examples of hypercubes. A partial cube is a connected graph obtained from isometric subgraphs of hypercubes Q_n. Figure 4.2 shows a simple example of a partial cube H, such that H is an isometric subgraph of an hypercube Q_3. Partial cubes were first investigated by Graham and Pollak (1971) as a model for communication networks. Afterwards, they have been used in many applications, such as in mathematical chemistry (Klavzar, 1998), (Klavzar et al., 1995) and computational biology (Bryant and Moulton, 2004). Partial cubes constitute a large variety of graphs, for example, trees, benzenoid systems, square systems, graphenylenes systems, phenylenes, even cycles and median graphs. Later, we discuss in details certain of these systems. For more information about partial cubes see (Hammack et al., 2011).

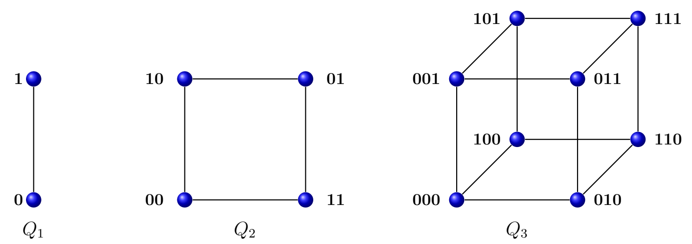


Figure 4.1: An example of some hypercubes.

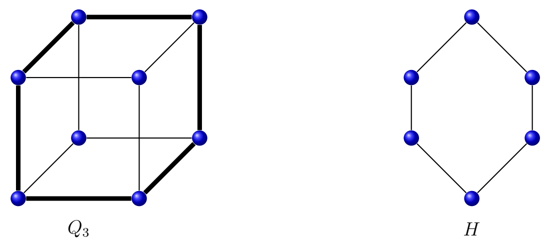


Figure 4.2: A simple example of partial cubes.

Now, we introduce an important concept as well as a key notion of the edge-cut method called the relation Θ. This relation was first introduced by Djoković (1973) and later defined in (Winkler, 1984) for bipartite graphs. Therefore, the relation Θ is also known as Djoković-Winkler relation. We say that two edges $e = xy$ and $f = uv$ of a graph G are in relation Θ, in symbols $e\Theta f$, if the following relation is verified:

$$d(x,u) + d(y,v) \neq d(x,v) + d(y,u). \tag{4.1}$$

For determining the edges in relation Θ, it is useful to know the following facts:

- Two adjacent edges of G are in relation Θ if and only if they belong to a common triangle.
- Two different edges of a shortest path are not in relation Θ.
- Let C be an even cycle. Then $e \in E(C)$ is in relation exactly to its antipodal edge.
- Edges from different components are never in relation Θ.

Figure 4.3 illustrates the Equation 4.1 and the previous facts. It shows a graph G with some edges that verify the relation Θ. The two edges $e_1 = v_1v_4$ and $f_1 = v_2v_3$ are in relation Θ because e_1 is the antipodal of f_1. The two adjacent edges $e_2 = v_4v_3$ and $f_2 = v_3v_5$ belong to a common triangle. Then, they are in relation Θ. On the other hand, the two edges $e_3 = v_1v_4$ and $f_3 = v_3v_5$ are not in relation Θ since the Equation 4.1 is not verified.

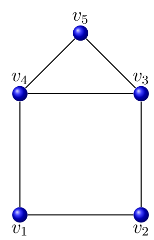


Figure 4.3: The verification of the relation Θ.

The relation Θ is always reflexive and symmetric, but not necessarily transitive. In the case of partial cubes this relation is transitive, and thus partitions the edge set of a partial cube into equivalent classes $E_1, ..., E_q$ called Θ-classes, Θ-partitions or cuts. The graph $G - E_i$ has exactly two connected components, and two edges e and f belong to the same class E_i if and only if $e\Theta f$. For the other classes of graphs, the smallest transitive relation containing Θ, also called the transitive closure, is denoted by Θ^*. Then, Θ^* is a transitive relation on $E(G)$ for any connected graph and partitions the edge set of G into equivalent classes $F_1, ..., F_k$ called Θ^*-classes, such that the graph $G - F_i$ contains two or more connected components. If the graph G is an odd cycle, then all the edges will be in the same Θ^*-class. In Figure 4.4, we show the classes of a graph G by using the relation Θ and the smallest transitive relation containing Θ. This graph has four Θ-classes E_1, E_2, E_3 and E_4. On the other hand, the same graph G has two Θ^*-classes F_1 and F_2.

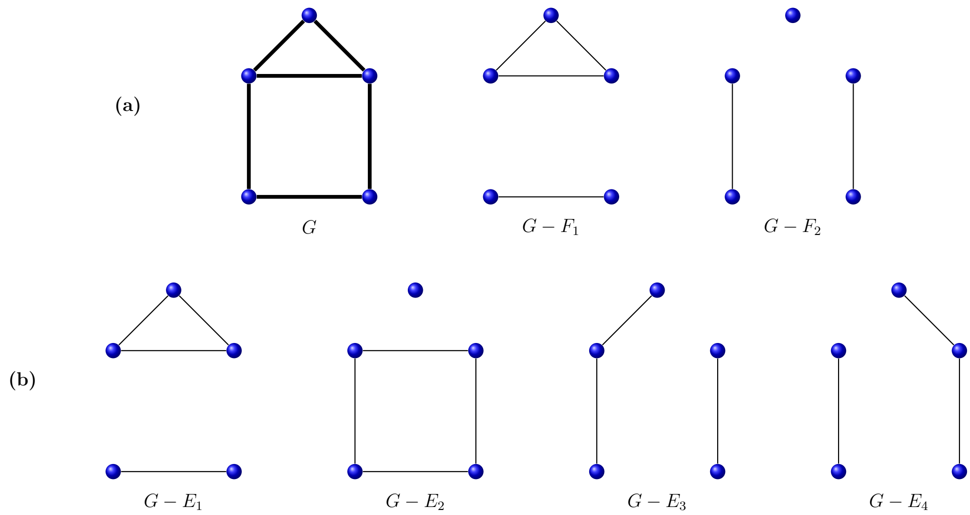


Figure 4.4: The equivalent classes of a graph G. (a): The graph G and its Θ^*-classes. (b): The Θ-classes of the graph G.

Finally, we have in (Hammack et al., 2011) the following theorem that shows the connection between the Djoković-Winkler relation and the characterization of partial cubes.

Theorem 4.1.1. *Let G be a connected graph, the following statements are equivalent:*

1. *G is a partial cube.*

2. *G is bipartite, and the components induced by $G - E_i$ are convex subgraphs of G for all the equivalent classes $\{E_i\}_{i=1}^q$.*

3. *G is bipartite and $\Theta = \Theta^*$.*

It should be noted that in this chapter we concentrate only on Θ-partitions. The application of the transitive closure Θ^* is investigated in chapter 5.

4.1.2 Standard Cut Method

The standard cut method was firstly investigated in (Klavzar et al., 1995) to compute the Wiener index of benzenoid graphs. Broadly speaking, this method is only used for computing distance-based topological indices of partial cubes. We noted in the previous subsection that the relation Θ partitions the edge set of a partial cube into equivalent Θ-classes $E_1, ..., E_q$, where two edges e and f belong to the same class E_i if and only if $e\Theta f$. Furthermore, the graph $G - E_i$ has exactly two connected components, for $1 \leq i \leq q$. By using this partition, we reveal the first application of the standard cut method.

Theorem 4.1.2. *(Klavzar et al., 1995) Let G be a partial cube and let $E_1, ..., E_q$ be its Θ-partitions. Let $n_1(E_i)$ and $n_2(E_i)$ be the number of vertices in the two connected components of $G - E_i$. Then,*

$$W(G) = \sum_{i=1}^{q} n_1(E_i)\, n_2(E_i). \tag{4.2}$$

The above theorem was applied in many articles to obtain exact expressions for the Wiener index of families of graphs; for example see (Klavžar et al., 1997) and (Khalifeh et al., 2010). Later, several instances of the standard cut method were considered for computing some distance-based topological indices, such as the Szeged index, the Gutman index, the hyper-Wiener index and the generalized Terminal Wiener index; see (Klavzar and J Nadjafi-Arani, 2015) and (Klavzar, 2008). In our work, we concentrate only on the Terminal wiener index and its generalization. Then, the standard cut method for this measure reads as follows:

Theorem 4.1.3. *(Ilić and Ilić, 2013) Let G be a partial cube and $E_1, ..., E_q$ be its Θ-partitions. Let $n_1^K(E_i)$ and $n_2^K(E_i)$ be the number of vertices of degree $K \geq 1$ in the two connected components of $G - E_i$. Then,*

$$TW_K(G) = \sum_{i=1}^{q} n_1^K(E_i) n_2^K(E_i). \tag{4.3}$$

Clearly in the case of the Terminal Wiener index, we set the degree $K = 1$.

In order to illustrate and demonstrate the efficiency of these two theorems of the standard cut method, we apply them on an interesting example of partial cubes called benzenoid systems (Gutman and Cyvin, 2012).

4.1.2.1 Illustration On a Benzenoid System

Benzenoid systems or hexagonal systems are connected graphs constructed in the following manner. Let $\mathcal{H}$ be the hexagonal lattice and let Z be a circuit on it. Then, a benzenoid system is formed by the vertices and edges of $\mathcal{H}$, lying on Z and in its interior. Figure 4.5 shows the construction method of a benzenoid system.

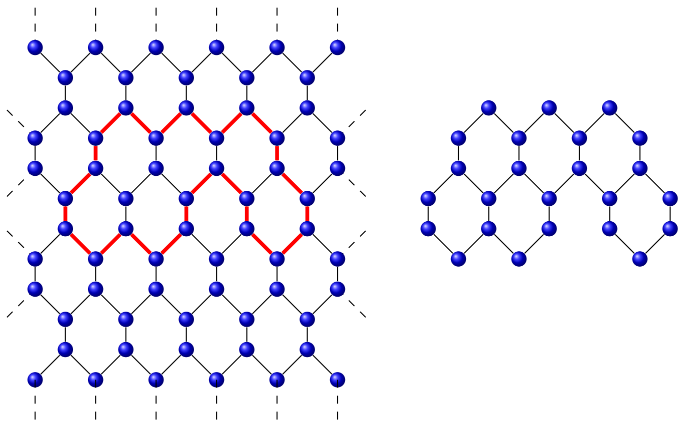


Figure 4.5: (Left) The hexagonal lattice $\mathcal{H}$ and the circuit Z. (Right) A benzenoid system G.

As an example, Let consider the benzenoid graph G depicted in Figure 4.5. The Θ-classes of G are exactly their elementary cuts, such that an elementary cut is a line segment that starts at the middle of a boundary edge of G and goes orthogonal to it until the first next boundary edge. Figure 4.6 shows the elementary cuts of the benzenoid system G. The graph G has 3 horizontal cuts X_i, 4 diagonal cuts Y_i along the North-East and the South-West directions, and 4 diagonal cuts Z_i along the North-West and the South-East directions. Thus, the relation Θ partitions the edge set of G into 11 equivalent classes. By using Theorem 4.1.2, we get:

$$W(G) = \sum_{i=1}^{3} n_1(X_i)n_2(X_i) + \sum_{i=1}^{4} n_1(Y_i)n_2(Y_i) + \sum_{i=1}^{4} n_1(Z_i)n_2(Z_i)$$

$$= (7*17+19*5+3*21)+(3*21+8*16+13*11+19*5)+$$
$$(5*19+11*13+17*7+21*3)$$
$$= 1126.$$

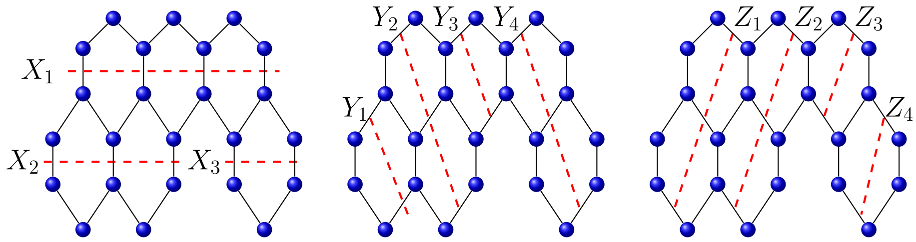


Figure 4.6: The cuts of the benzenoid graph G.

For proving the significance of the Theorem 4.1.3 and how the method can be easily applied, we calculate the generalized Terminal Wiener index of the benzenoid graph G in the case of $K = 3$. Hence, we get:

$$TW_3(G) = \sum_{i=1}^{3} n_1^3(X_i)n_2^3(X_i) + \sum_{i=1}^{4} n_1^3(Y_i)n_2^3(Y_i) + \sum_{i=1}^{4} n_1^3(Z_i)n_2^3(Z_i)$$
$$= (2*8+9*1)+(3*7+6*4+9*1)+(1*9+5*5+8*2)$$
$$= 129.$$

The above example demonstrates that the standard cut method is basic to apply for calculating distance-based topological indices. Furthermore, the studied example proves that this technique is useful for deriving analytical expressions of benzenoid graph families and partial cubes in general.

4.1.3 An Extended Cut Method

Some extensions of the standard cut method were proposed in the literature. Here, we discuss the first extension of the standard cut method, which was proposed by Chepoi et al. (1997). They generalized the edge-cut method for computing the Wiener index of a class of graphs larger than partial cubes named l_1-graphs.

For defining the class of l_1-graphs, we need to introduce the concept of scale embedding. Let G and H be two connected graphs, we say that H is a scale embedding into G if there exist a mapping $\beta : V(H) \to V(G)$, such that for all $u, v \in V(H)$ and $\lambda \in \mathbf{N}$,

$$d_G(\beta(u), \beta(v)) = \lambda d_H(u, v).$$

Figure 4.7 represents two graphs H_1 and H_2 that are embeddable with scale $\lambda_1 = 1$ and $\lambda_2 = 2$ in G, respectively.

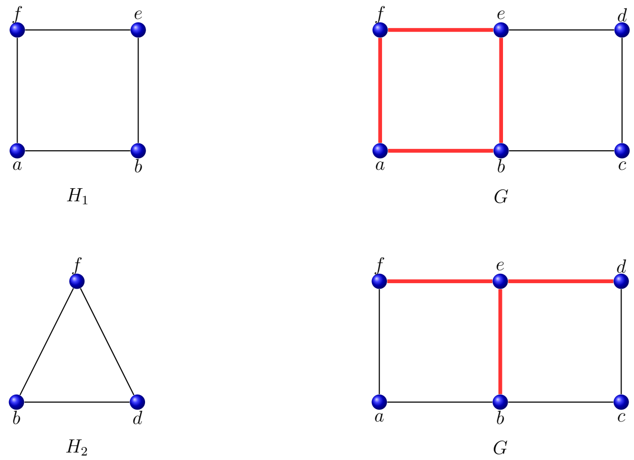


Figure 4.7: Two graphs H_1 and H_2 scale embeddable in G, with $\lambda_1 = 1$ and $\lambda_2 = 2$, respectively.

Assouad and Deza (1980) proved that a graph G is an l_1-graph if it admits a scale embedding into a hypercube for some $\lambda \geq 1$. Obviously, partial cubes are exactly the l_1-graphs embeddable with scale 1. We can also say, a scale embedding with $\lambda = 1$ is an isometric embedding. More broadly, in the bipartite case, l_1-graphs coincide with partial cubes, otherwise l_1-graphs include non-bipartite graphs, such as complete graphs and Petersen graphs. Figure 4.8 shows the mentioned non-bipartite l_1-graphs. Later, Deza and Tuma (1996) proved that a graph G is an l_1-graph if and only if it admits a partition of $E^\lambda(G)$ into convex cuts, where $E^\lambda(G)$ denotes a collection of edges of the graph G with each edge in G being repeated $\lambda > 1$ times. In other words, a graph G is an l_1-graph if and only if it admits a collection $\{E_i\}_{i=1}^q$ of convex edge-cuts of G such that each edge of

G is cut by exactly λ cuts. We note that an edge-cut E_i of G is a convex cut if the two components of $G - E_i$ are the convex subgraphs of G. Figure 4.9 represents the convex cut of a graph G and we can see that each edge of G is cut by precisely $\lambda = 2$ cuts. According to this characterization of l_1-graphs, Chepoi et al. (1997) proved the following first extension of the standard cut method.

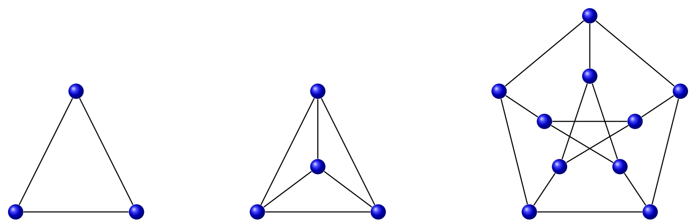


Figure 4.8: Some examples of non-bipartite l_1-graphs.

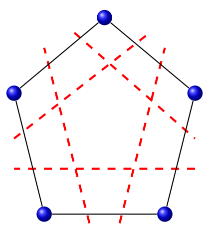


Figure 4.9: The convex edge-cuts of the l_1-graph G.

Theorem 4.1.4. *Let G be a scale λ-embeddable into a hypercube and let $\{E_i\}_{i=1}^{q}$ be the convex cuts of $E^{\lambda}(G)$. Then,*

$$W(G) = \frac{1}{\lambda} \sum_{i=1}^{q} n_1(E_i)\, n_2(E_i). \tag{4.4}$$

In the following Theorem, we present our work concerning the generalization of the edge-cut method for computing the generalized Terminal Wiener index of l_1-graphs.

Theorem 4.1.5. *Let G be a connected graph that admits a collection $\{E_i\}_{i=1}^{q}$ of convex cuts of $E^{\lambda}(G)$. Then,*

$$TW_K(G) = \frac{1}{\lambda} \sum_{i=1}^{q} n_1^K(E_i)\, n_2^K(E_i) \tag{4.5}$$

Proof. Let $P_{u,v}$ be the shortest path between the two vertices u and v of $V(G)$ and $deg(u) = deg(v) = K$. We can verify that : $\mid E(P_{u,v}) \cap E_i \mid = 0$ if u, v belong to the same component $G - E_i$, otherwise $\mid E(P_{u,v}) \cap E_i \mid = 1$.
Every edge $e \in E^{\lambda}(G)$ is cut by precisely λ convex cuts from $\{E_i\}_{i=1}^{q}$. Then :

$$\begin{aligned} TW_K(G) &= \sum_{\substack{\{u,v\} \subseteq V(G) \\ deg(u)=deg(v)=K}} d(u,v) \\ &= \sum_{\substack{\{u,v\} \subseteq V(G) \\ deg(u)=deg(v)=K}} \mid E(P_{u,v}) \mid \\ &= \frac{1}{\lambda} \sum_{i=1}^{q} \sum_{\substack{\{u,v\} \subseteq V(G) \\ deg(u)=deg(v)=K}} \mid E(P_{u,v}) \cap E_i \mid . \end{aligned}$$

Which gives the result. □

In the next section, we show the efficiency of the above theorems concerning the extended cut method by using them to quantify the topological structure of some complicated networks.

4.2 Application Of The Edge-Cut Method For Quantifying The Structure Of Some Large Networks

The aim of this section is to apply the extended cut method in order to analyze the topological structure of a kind of large networks named Equilateral Triangular Tetra Sheet network $ETTS_n$ and Hex Derived network HD_n. At first, We describe the construction method of these networks. Then, we give the exact analytical expressions for the Wiener index and the generalized Terminal Wiener index of these structures. Also, we compare the behavior of these measures by using graphical representations. The mentioned networks have been used as good models for communication networks to solve the problem of metric dimension and embedding problem (Raj and George, 2015). Moreover, they have been investigated as molecular structures in chemical and biological applications (Imran et al., 2016).

4.2.1 Quantifying the Topological Structure of Equilateral Triangular Tetra Sheet Network

Let $ETTS_n$ be an Equilateral Triangular Tetra Sheet of dimension n, where n denotes the number of vertices in a side of this network. The $ETTS_n$ can be built in the following iterative way. Initially, $ETTS_2$ is composed of one tetrahedron K_4, also known as a triangular pyramid. $ETTS_3$ is obtained from $ETTS_2$ by adding a layer of tetrahedrons around the bottom side of $ETTS_2$. Similarly, $ETTS_n$ is obtained from $ETTS_{n-1}$ by adding a layer of tetrahedrons around the bottom side of $ETSS_{n-1}$. Figure 4.10 illustrates the construction of some iterations of the Equilateral Triangular Tetra Sheet Network $ETTS_n$.

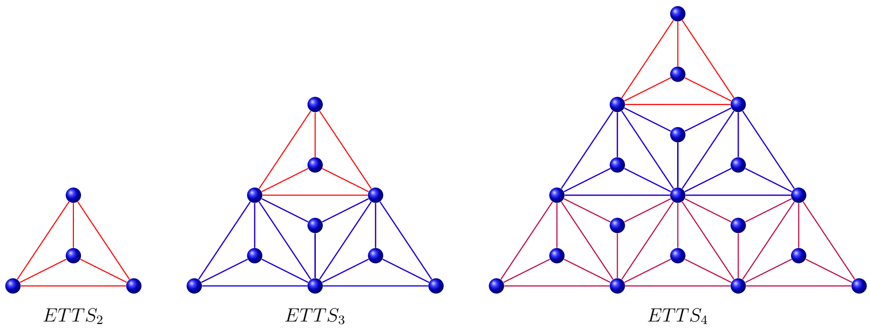


Figure 4.10: Equilateral Triangular Tetra Sheet Networks, with $n = \{2, 3, 4\}$.

From the construction method of the Equilateral Triangular Tetra sheet network, we can extract the following structural properties :

- The number of vertices of $ETTS_n$ is:

$$N = \frac{3n^2 - 3n + 2}{2}. \tag{4.6}$$

- The number of edges of $ETTS_n$ is:

$$M = \frac{9n^2 - 15n + 6}{2}. \tag{4.7}$$

- The diameter of $ETTS_n$ is :

$$D = n - 1. \tag{4.8}$$

The network $ETTS_n$ admits a collection of convex cuts as illustrated in Figure 4.11. We can observe that each edge of $ETTS_n$ is cut by precisely $\lambda = 2$ cuts. Thus, this structure belongs to the class of l_1-graphs and absolutely in this case the extended cut method is applicable. Now, we apply this approach on the $ETTS_n$ to get exact analytical expressions for the Wiener index W and the generalized Terminal Wiener index TW_K. We take into consideration the case of K equals to the maximum degree Δ and the minimum degree δ. From Figure 4.10, we can see that the maximum degree of the network $ETTS_n$ is $\Delta = 12$ and the minimum degree is equal to $\delta = 3$. The following theorem summarize the obtained results.

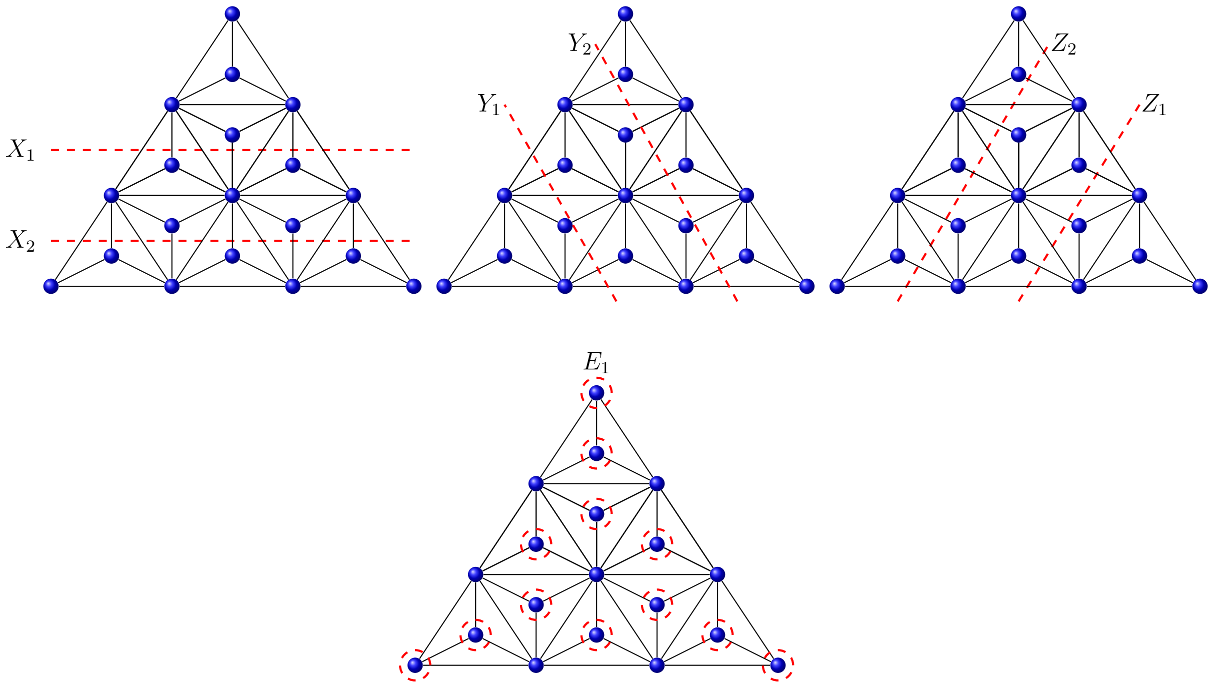


Figure 4.11: The convex edge-cuts of Equilateral Triangular Tetra Sheet Network $ETTS_4$.

Theorem 4.2.1. *Let $ETTS_n$ be an Equilateral Triangular Tetra Sheet network of dimension n, with $n \geq 2$. Then,*

$$W(ETTS_n) = \frac{9}{20}n^5 - \frac{3}{8}n^4 - \frac{1}{2}n^3 + \frac{3}{8}n^2 + \frac{1}{20}n. \tag{4.9}$$

$$TW_\delta(ETTS_n) = \frac{1}{5}n^5 - \frac{1}{2}n^4 + 2n^3 - 4n^2 + \frac{53}{10}n - 3. \tag{4.10}$$

$$TW_{\Delta}(ETTS_n) = \frac{1}{20}n^5 - \frac{5}{8}n^4 + 3n^3 - \frac{55}{8}n^2 + \frac{149}{20}n - 3. \tag{4.11}$$

Proof. Let consider the convex edge-cuts shown in Figure 4.11, such that we have three types of cuts classified according the directions: horizontal, diagonal and encircling cuts. Let X_i, $1 \leq i \leq n-2$, denote the horizontal edge-cuts of $ETTS_n$. For $1 \leq i \leq n-2$, let Y_i denote the diagonal edge-cuts along the North-West and South-East directions and Z_i denote the diagonal edge-cuts along the South-West and North-East directions. Let E_i, $1 \leq i \leq 3+(n-1)^2$, denote the encircling edge-cuts of $ETTS_n$.
We have the number of vertices of degree $\delta = 3$ is equal to : $N_\delta = 3+(n-1)^2$, and the number of vertices of degree $\Delta = 12$ is equal to : $N_\Delta = \frac{n^2-5n+6}{2}$.
Then, for $1 \leq i \leq 3+(n-1)^2$:

$$\begin{aligned} n_1(E_i) &= 1, \qquad n_2(E_i) = N-1. \\ n_1^\delta(E_i) &= 1, \qquad n_2^\delta(E_i) = N_\delta - 1. \end{aligned}$$

And, for $1 \leq i \leq n-2$:

$$\begin{array}{ll} n_1(X_1) = 5, & n_2(X_1) = N-5. \\ n_1(X_i) = \sum_{j=2}^{i} 3(j+1) - 2i + 7, & n_2(X_i) = N - n_1(X_i). \\ n_1^\delta(X_i) = i^2 + i + 1, & n_2^\delta(X_i) = N_\delta - n_1^\delta(X_i). \\ n_1^\Delta(X_1) = 0. & \\ n_1^\Delta(X_i) = \sum_{j=2}^{i}(j-1), & n_2^\Delta(X_i) = N_\Delta - n_1^\Delta(X_i). \end{array}$$

Due to the symmetry of $ETTS_n$ network, we have $n_1(X_i) = n_1(Y_i) = n_1(Z_i)$ and $n_1^K(X_i) = n_1^K(Y_i) = n_1^K(Z_i)$, for $1 \leq i \leq n-2$.
Now, we apply the two Theorems 4.1.4 and 4.1.5 of the extended cut method, such that $\lambda = 2$. Then,

$$W(ETTS_n) = \frac{1}{2}\Big[\sum_{i=1}^{n-2} n_1(X_i)n_2(X_i) + \sum_{i=1}^{n-2} n_1(Y_i)n_2(Y_i) + \sum_{i=1}^{n-2} n_1(Z_i)n_2(Z_i) +$$

$$\sum_{i=1} n_1(E_i)n_2(E_i)\Bigg]$$
$$= \frac{1}{2}\Bigg[3\sum_{i=1}^{n-2} n_1(X_i)n_2(X_i) + (N-1)(3+(n-1)^2)\Bigg].$$

$$TW_\delta(ETTS_n) = \frac{1}{2}\Bigg[\sum_{i=1}^{n-2} n_1^\delta(X_i)n_2^\delta(X_i) + \sum_{i=1}^{n-2} n_1^\delta(Y_i)n_2^\delta(Y_i) + \sum_{i=1}^{n-2} n_1^\delta(Z_i)n_2^\delta(Z_i) +$$
$$\sum_{i=1} n_1^\delta(E_i)n_2^\delta(E_i)\Bigg]$$
$$= \frac{1}{2}\Bigg[3\sum_{i=1}^{n-2} n_1^\delta(X_i)n_2^\delta(X_i) + (3+(n-1)^2)(2+(n-1)^2)\Bigg].$$

$$TW_\Delta(ETTS_n) = \frac{1}{2}\Bigg[\sum_{i=2}^{n-2} n_1^\Delta(X_i)n_2^\Delta(X_i) + \sum_{i=2}^{n-2} n_1^\Delta(Y_i)n_2^\Delta(Y_i) + \sum_{i=2}^{n-2} n_1^\Delta(Z_i)n_2^\Delta(Z_i)\Bigg]$$
$$= \frac{1}{2}\Bigg[3\sum_{i=2}^{n-2} n_1^\Delta(X_i)n_2^\Delta(X_i)\Bigg].$$

Which give the Equations 4.9, 4.10 and 4.11. □

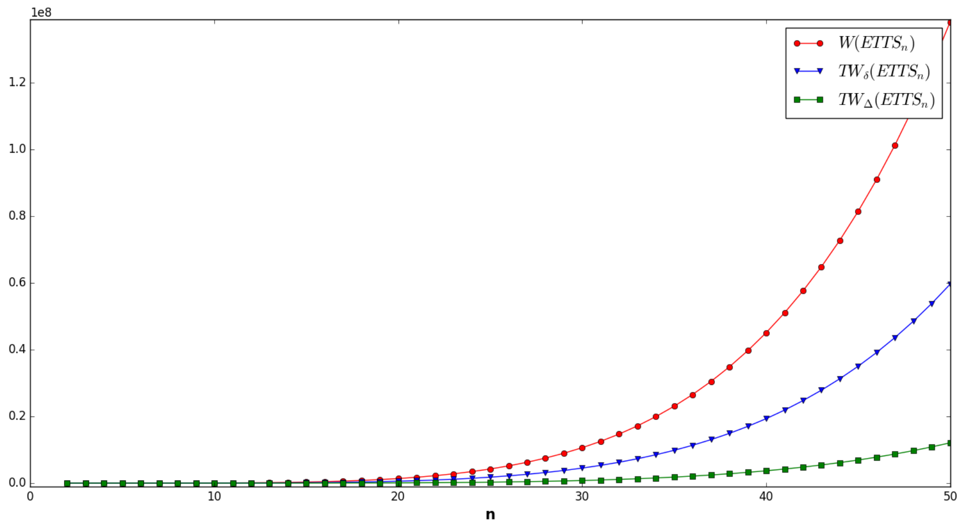


Figure 4.12: Comparison between the calculated measures of the network $ETTS_n$.

A comparison of the calculated topological indices of the Equilateral Triangular Tetra Sheet network is given in Figure 4.12. Along the horizontal line and the vertical line, we have the dimension n and the values of the three measures, respectively. We can see that all the topological indices are monotonically increasing and they change the monotony with the high increase of the dimension n. In general, the Wiener index of $ETTS_n$ shows a dominant change with the increasing values of the parameter n.

4.2.2 Quantifying the Topological Structure of Hex Derived Network

Let HDN_n be a Hex Derived Network of dimension $n \geq 2$, such that n denotes the number of vertices in a side of this structure. The construction of HDN_n is presented in the following iterative way. At first, HDN_2 is a 2-dimensional hexagonal mesh composed of six tetrahedrons K_4. Then, HDN_3 is obtained from HDN_2 by adding a layer of tetrahedrons K_4 around the boundary of HDN_2. The growth process to the next generations continues in a similar way, such that HDN_n is obtained from the previous iteration HDN_{n-1} by adding a layer of tetrahedrons K_4 around the boundary of HDN_{n-1}. Figure 4.13, illustrates two generations of the Hex Derived Network HDN_n.

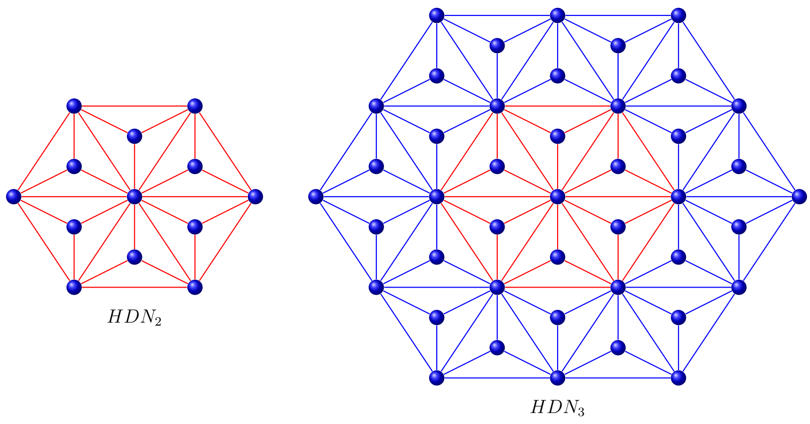


Figure 4.13: Hex Derived Networks HDN_n, with $n = \{2, 3\}$.

We extracted some structural properties of the Hex derived Network HDN_n, which are defined as follows :

- The number of vertices of the network HDN_n is:

$$N = 9n^2 - 15n + 7. \tag{4.12}$$

- The number of edges of the network HDN_n is:

$$M = 27n^2 - 51n + 24. \tag{4.13}$$

- The diameter of the network HDN_n is:

$$D = 2n - 2. \tag{4.14}$$

As we did for the previous structure, The network HDN_n admits a collection of convex cuts, which are illustrated in Figure 4.14. Clearly, each edge of HDN_n is cut by precisely $\lambda = 2$ cuts. Thus, this structure belongs to the class of l_1-graphs and the extended cut method is applicable. Now, we apply this approach on the HDN_n to get exact analytical expressions for the Wiener index W and the generalized Terminal Wiener index TW_K in the case of K equals to the maximum degree Δ and the minimum degree δ. From Figure 4.13, the maximum degree of the network HDN_n is $\Delta = 12$ and the minimum degree is equal to $\delta = 3$. The following theorem summarize the obtained results.

Theorem 4.2.2. *Let HDN_n be a Hex Derived Network of dimension n, with $n \geq 2$. Then,*

$$W(HDN_n) = \frac{369}{10}n^5 - \frac{507}{4}n^4 + 164n^3 - \frac{381}{4}n^2 + \frac{211}{10}n. \tag{4.15}$$

$$TW_\delta(HDN_n) = \frac{82}{5}n^5 - 64n^4 + 91n^3 - 56n^2 + \frac{63}{5}n. \tag{4.16}$$

$$TW_\Delta(HDN_n) = \frac{41}{10}n^5 - \frac{123}{4}n^4 + 92n^3 - \frac{549}{4}n^2 + \frac{1019}{10}n - 30. \tag{4.17}$$

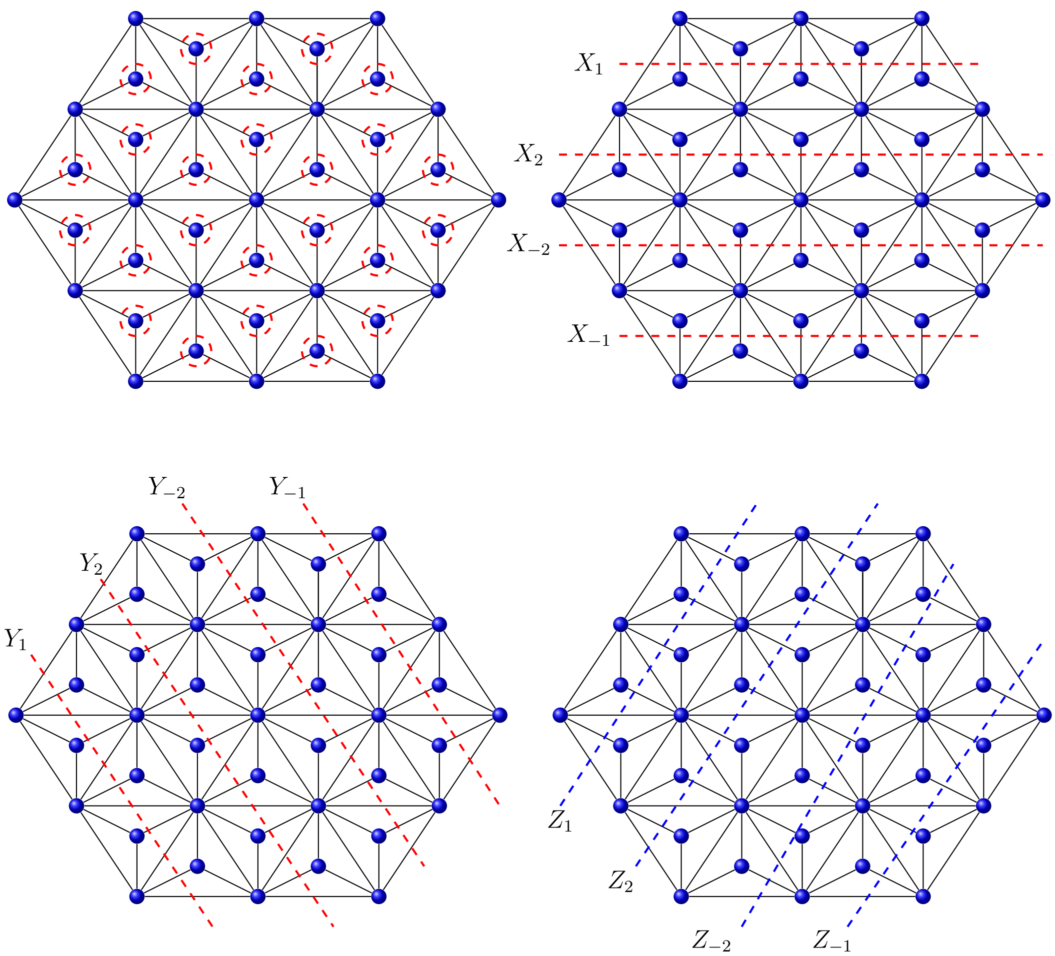


Figure 4.14: Edge-cuts of the Hex Derived Network HDN_3.

Proof. The proof is similar to the case of the network $ETTS_n$. Let consider the convex edge-cuts shown in Figure 4.14, such that we have three types of cuts: horizontal, diagonal and encircling cuts.

Let X_i and X_{-i}, for $1 \leq i \leq n-1$, denote the horizontal edge-cuts of HDN_n. Similarly, for $1 \leq i \leq n-1$, let Y_i, Y_{-i} denote the diagonal edge-cuts along the North-West and South-East directions and Z_i, Z_{-i} denote the diagonal edge-cuts along the South-West and North-East directions. Let E_i, $1 \leq i \leq 6(n-1)^2$, denote the encircling edge-cuts of HDN_n.

We have the number of vertices of degree $\delta = 3$ is equal to: $N_\delta = 6(n-1)^2$, and the number of vertices of degree $\Delta = 12$ is: $N_\Delta = 3n^2 - 9n + 7$.

Then, for $1 \leq i \leq 6(n-1)^2$:

$$n_1(E_i) = 1, \qquad n_2(E_i) = N - 1.$$
$$n_1^\delta(E_i) = 1, \qquad n_2^\delta(E_i) = N_\delta - 1.$$

and, for $1 \leq i \leq n-1$:

$$n_1(X_1) = 2n-1, \qquad n_2(X_1) = N - 2n + 1.$$
$$n_1(X_i) = 3\sum_{j=1}^{i-1}(n+j-1) + 2n + i - 2, \qquad n_2(X_i) = N - n_1(X_i).$$

$$n_1^\delta(X_1) = n-1, \qquad n_2^\delta(X_1) = N_\delta - n + 1.$$
$$n_1^\delta(X_i) = (n-1) + 2\sum_{j=1}^{i-1}(n+j-1), \qquad n_2^\delta(X_i) = N_\delta - n_1^\delta(X_i).$$

$$n_1^\Delta(X_1) = 0.$$
$$n_1^\Delta(X_i) = \sum_{j=1}^{i-1}(n+j-2), \qquad n_2^\Delta(X_i) = N_\Delta - n_1^\Delta(X_i).$$

Using the symmetry, we have for $1 \leq i \leq n-1$:

$n_1(X_i) = n_1(Y_i) = n_1(Z_i) = n_1(X_{-i}) = n_1(Y_{-i}) = n_1(Z_{-i})$.

$n_1^K(X_i) = n_1^K(Y_i) = n_1^K(Z_i) = n_1^K(X_{-i}) = n_1^K(Y_{-i}) = n_1^K(Z_{-i})$.

Now, we apply the two Theorems 4.1.4 and 4.1.5 of the extended cut method, with $\lambda = 2$.

$$W(HDN_n) = \frac{1}{2}\Bigg[\sum_{i=1}^{n-1} n_1(X_i)n_2(X_i) + \sum_{i=1}^{n-1} n_1(Y_i)n_2(Y_i) + \sum_{i=1}^{n-1} n_1(Z_i)n_2(Z_i) +$$

$$\sum_{i=1}^{n-1} n_1(X_{-i})n_2(X_{-i}) + \sum_{i=1}^{n-1} n_1(Y_{-i})n_2(Y_{-i}) + \sum_{i=1}^{n-1} n_1(Z_{-i})n_2(Z_{-i}) + \sum_{i=1} n_1(E_i)n_2(E_i)\Bigg]$$
$$= \frac{1}{2}\Bigg[6\sum_{i=1}^{n-1} n_1(X_i)n_2(X_i) + (N-1)6(n-1)^2\Bigg].$$

$$TW_\delta(HDN_n) = \frac{1}{2}\Bigg[\sum_{i=1}^{n-1} n_1^k(X_i)n_2^k(X_i) + \sum_{i=1}^{n-1} n_1^k(Y_i)n_2^k(Y_i) + \sum_{i=1}^{n-1} n_1^k(Z_i)n_2^k(Z_i) +$$
$$\sum_{i=1}^{n-1} n_1^k(X_{-i})n_2^k(X_{-i}) + \sum_{i=1}^{n-1} n_1^k(Y_{-i})n_2^k(Y_{-i}) + \sum_{i=1}^{n-1} n_1^k(Z_{-i})n_2^k(Z_{-i}) +$$
$$\sum_{i=1} n_1^k(E_i)n_2^k(E_i)\Bigg]$$
$$= \frac{1}{2}\Bigg[6\sum_{i=1}^{n-1} n_1^k(X_i)n_2^k(X_i) + (6(n-1)^2-1)6(n-1)^2\Bigg].$$

$$TW_\Delta(HDN_n) = \frac{1}{2}\Bigg[\sum_{i=1}^{n-1} n_1^k(X_i)n_2^k(X_i) + \sum_{i=1}^{n-1} n_1^k(Y_i)n_2^k(Y_i) + \sum_{i=1}^{n-1} n_1^k(Z_i)n_2^k(Z_i) +$$
$$\sum_{i=1}^{n-1} n_1^k(X_{-i})n_2^k(X_{-i}) + \sum_{i=1}^{n-1} n_1^k(Y_{-i})n_2^k(Y_{-i}) + \sum_{i=1}^{n-1} n_1^k(Z_{-i})n_2^k(Z_{-i})$$
$$= 3\sum_{i=1}^{n-1} n_1^k(X_i)n_2^k(X_i)$$

Which give the Equations 4.15, 4.16 and 4.17. □

Figure 4.15 shows the behavior of the calculated topological indices of the Hex Derived network. Along the horizontal line we have the dimension n and along the vertical line the values of the three measures are determined. Similarly, all the topological indices are increasing and change their monotony in some values of dimension n. Broadly speaking, The Wiener index of HDN_n shows a dominant change with the increasing values of the parameter n.

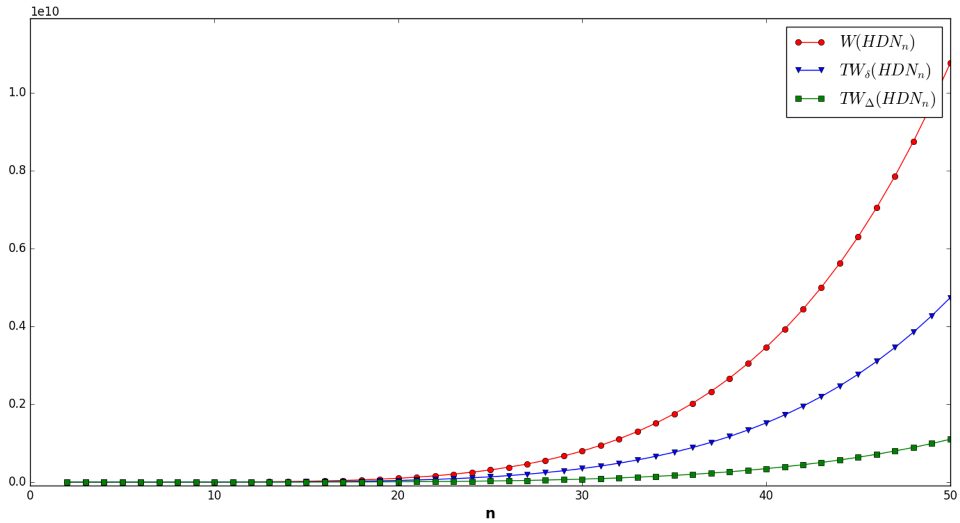


Figure 4.15: Comparison between the calculated measures of the network HDN_n.

4.3 Vertex-Cut Method For Computing The Generalized Terminal Wiener Index

In the previous section, we saw that the standard edge-cut method and its first extension are considered as powerful approaches to calculate the distance-based topological indices of partial cubes and l_1-graphs. In the case of some other classes of graphs the applicability of the edge-cut method is not possible. This limitation motivated the researchers to develop a new technique called the vertex-cut method in order to compute the desired measures successfully. In this section, we discuss the first apparition of the vertex-cut method. Then, we extend this technique to derive a new formula for computing the generalized Terminal Wiener index.

In Wiener's first paper (Wiener, 1947), it was shown that the Wiener index of a tree can be decomposed into easily calculable edge-contributions; see Theorem 2.2.1. After many years, a similar formula of Theorem 2.2.1 was proposed for computing the Wiener index of trees by using the concept of vertex-contributions (Škrekovski and Gutman, 2014). Then, in the same paper, the authors studied the case of general graphs and deduced an expression that relates the vertex decomposition of the Wiener index with the concept of betweenness centrality.

We previously defined that the betweenness centrality of a vertex $x \in V(G)$ can be computed as follows.

$$B(x) = \sum_{\substack{u,v \in V(G) \\ u \neq v \neq x}} \frac{\sigma_{u,v}(x)}{\sigma_{u,v}},$$

where $\sigma_{u,v}$ denotes the total number of shortest $u-v$ paths in G and $\sigma_{u,v}(x)$ represents the number of shortest $u-v$ paths passing through the vertex x.
The following theorem represents the vertex version of the Wiener index, which was proved in (Škrekovski and Gutman, 2014).

Theorem 4.3.1. *(Škrekovski and Gutman, 2014) Let G be a graph of order N. Then,*

$$W(G) = \sum_{v \in V(G)} B(v) + \binom{N}{2}. \tag{4.18}$$

In the recent paper (Arockiaraj et al., 2016), the authors analyzed the Theorem 4.3.1 to make its application uncomplicated. Therefore, they arrived to develop a new technique called the vertex-cut method for computing degree and distance-based topological indices as the sum of the vertex contributions. In order to show the obtained result about the vertex-cut method, we need to introduce the following basic notations.

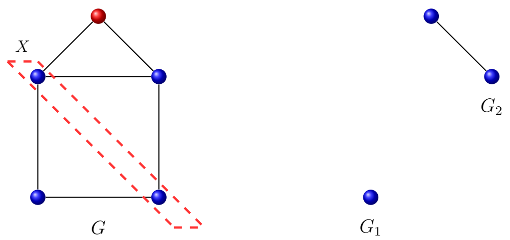


Figure 4.16: An example of a convex vertex-cut X and a corner vertex.

Let X be a vertex cut, we say that X is a convex vertex-cut if the two components G_1 and G_2 of $G - X$ are convex subgraphs of G. A vertex v of G is said to be a corner vertex if v doesn't belong to the shortest paths between any pair of vertices in G and L_G denotes the set of all corner vertices in G. An independent set S of $V(G)$ is defined as a set of vertices in which no two vertices are adjacent. Figure 4.16 represents an example of a convex vertex-cut X of a graph G, G_1 and G_2 are the resulting convex subgraphs and the red vertex of G is a corner vertex.

Thus, by using these notations, Arockiaraj et al. (2016) arrived to get the following first instance of the vertex-cut method.

Theorem 4.3.2. *(Arockiaraj et al., 2016) Let G be a connected graph and let $\{X_i\}_{i=0}^{k} \cup L_G$ be its vertex partition such that each X_i is a convex vertex cut and an independent set. Let $n_1(X_i)$ and $n_2(X_i)$ be the number of vertices in the two connected components of $G - X_i$. Then,*

$$W(G) = \sum_{i=1}^{k} n_1(X_i) n_2(X_i) + \binom{|V(G)|}{2}. \tag{4.19}$$

After the first investigation of the vertex-cut method, we developed a new formula based on this interesting approach for calculating the generalized Terminal Wiener index of certain graphs when the regular edge-cut method is useless. For proving our main result, we need to introduce a new measure based on the concept of the betweenness centrality.

Let $B_K(x)$ be an extension of the betweenness centrality, which quantifies the number of times a node x is connecting two vertices u and v of degree K. In other words, $B_K(x)$ is the sum of the fraction of all pairs of shortest paths between vertices of degree K that pass through x:

$$B_K(x) = \sum_{\substack{u,v \in V_K(G) \\ x \in V(G) \\ u \neq v \neq x}} \frac{\sigma_{u,v}(x)}{\sigma_{u,v}}, \tag{4.20}$$

where $\sigma_{u,v}$ denotes the total number of shortest $u - v$ paths in G, such that $deg(u) = deg(v) = K$, and $\sigma_{u,v}(x)$ represents the number of shortest $u - v$ paths with $deg(u) = deg(v) = K$ passing through the vertex x.

In the following theorem, we arrived to represent the relation between the generalized Terminal Wiener index and the new extension of the betweenness centrality .

Theorem 4.3.3. *Let G be a graph of order N and let N_K be the number of vertices of degree K. Then,*

$$TW_K(G) = \sum_{v \in V(G)} B_K(v) + \binom{N_K}{2}. \tag{4.21}$$

Proof. We can verify that the term $\sum_{x\in V(G)} \frac{\sigma_{u,v}(x)}{\sigma_{u,v}}$, which is the sum of the proposed betweenness centrality of all vertices $x \in V(G)$, is equal to $d(u,v)-1$. Then, by using the definition of the modified betweenness centrality $B_K(x)$, we have

$$\begin{aligned} B_K(x) &= \sum_{\substack{u,v\in V_K(G)\\ x\in V(G)\\ u\neq v\neq x}} \frac{\sigma_{u,v}(x)}{\sigma_{u,v}} \\ &= \sum_{\substack{u,v\in V_K(G)|x\\ u\neq v}} \sum_{x\in V(G)} \frac{\sigma_{u,v}(x)}{\sigma_{u,v}} \\ &= \sum_{\substack{u,v\in V_K(G)\\ u\neq v}} \Big[d(u,v)-1 \Big] \\ &= TW_K(G) - \binom{N_K}{2}. \end{aligned}$$

The proof is complete. □

Finally, using the above notations about the vertex partitions and Theorem 4.3.3, we obtained the new expression to compute the generalized Terminal Wiener index.

Theorem 4.3.4. *Let G be a connected graph and let $\{X_i\}_{i=0}^{k} \cup L_G$ be its vertex partition such that each X_i is a convex vertex cut and an independent set. Let $n_1^K(X_i)$ and $n_2^K(X_i)$ be the number of vertices of degree K in the two connected components of $G - X_i$. Then,*

$$TW_K(G) = \sum_{i=1}^{k} n_1^K(X_i) n_2^K(X_i) + \binom{N_K}{2}. \tag{4.22}$$

Proof. We have the following notation, for any set $X \subseteq V(G)$

$$B_K(X) = \sum_{v\in X} B_K(v).$$

Thus, using Theorem 4.3.3, we get

$$TW_K(G) = \sum_{v\in V(G)} B_K(v) + \binom{N_K}{2}$$

$$= \sum_{i=1}^{k} B_K(X_i) + B_K(L_G) + \binom{N_K}{2}.$$

It is clear that the modified betweenness centrality of a corner vertex is equal to 0. Hence, $B_K(L_G) = 0$.

From the definition of the extension of the betweenness centrality, the notion of $B_K(X_i)$ denotes how many times the elements of the vertex-cut X_i lie on the shortest paths between the vertices u and v of degree K, such that $u \in V_K(G_{i1})$ and $v \in V_K(G_{i2})$. G_{i1} and G_{i2} are the connected components of $G - X_i$. Therefore,

$$B_K(X_i) = n_1^K(X_i)n_2^K(X_i).$$

which gives the equation 4.22. □

4.4 Application Of The Vertex-Cut Method For Quantifying The Structure Of Silicate Networks

In this section, we demonstrate the efficiency and the significance of the vertex-cut method by computing the generalized Terminal Wiener index of an amazing structures called silicate networks. The silicates are the largest, the most interesting and the most complicated class of minerals by far. About 25% of all minerals are silicates and approximately 90% of the earth's crust is made of silicates. We can also find these structures in other variety of aspects. They have been used in chemistry to synthesis some inorganic networks (Arockiaraj et al., 2016) as well as in the field of interconnection networks to solve the minimum metric dimension problem (Manuel and Rajasingh, 2011). The structure of silicates is based on a fundamental chemical unit that is SiO_4 tetrahedron, also denoted by K_4, which is shown in Figure 4.17. The corner vertices of the tetrahedron SiO_4 represent the oxygen ions and the center vertex represents the silicon ion. These tetrahedrons can be arranged in a variety of ways, which lead to the different types of silicate networks such as the linear silicates, cyclic silicates, double chain silicates, sheet silicates, etc. Here, we particularly concentrate on two types of silicates networks, which are the Double Chain

Silicates network DS_n and the Silicates Sheet network SL_n. At first, we describe the construction method of these structures. Then, we apply the vertex-cut method to give there analytical expressions for the generalized Terminal Wiener index in the case of $K = \Delta$.

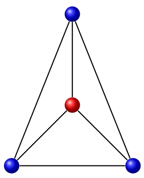


Figure 4.17: SiO_4 Tetrahedron.

4.4.1 Double Chain Silicates Network

Let CS_n be a Chain Silicates network of dimension n that is obtained by arranging n tetrahedron linearly. Thus, the Double Chain Silicates network DS_n of dimension n is obtained by joining two Chain Silicates structure of dimension $2n+1$. For illustration see Figure 4.18.

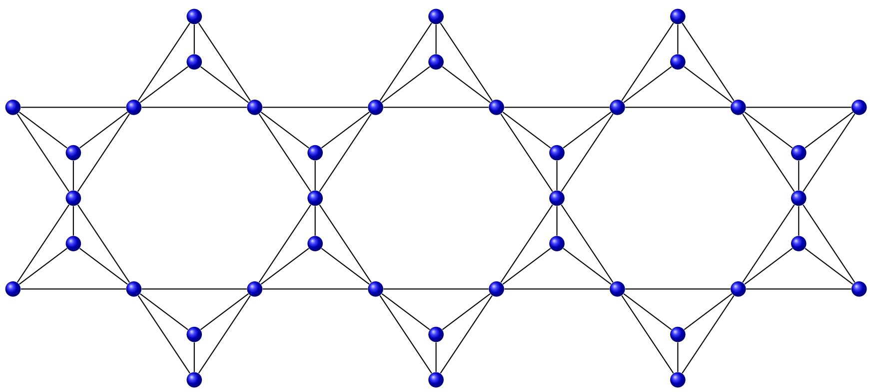


Figure 4.18: An example of a Double Chain Silicates network DS_3 of dimension $n = 3$.

From the construction method, we get the following structural properties of the Double Chain Silicates network DS_n:

- The number of vertices of DS_n is:

$$N = 11n + 7.$$

- The number of edges of DS_n is:

$$M = 24n + 12.$$

- The Diameter of DS_n is:

$$D = 2n + 2.$$

Now, we apply the vertex-cut method on the DS_n network. We start by identifying the vertex cuts and the set of corner vertices of this network as illustrated in Figure 4.19. In the following Theorem, we obtained the exact analytical expression for the generalized Terminal Wiener index TW_K in the case of K equals to the maximum degree $\Delta = 6$.

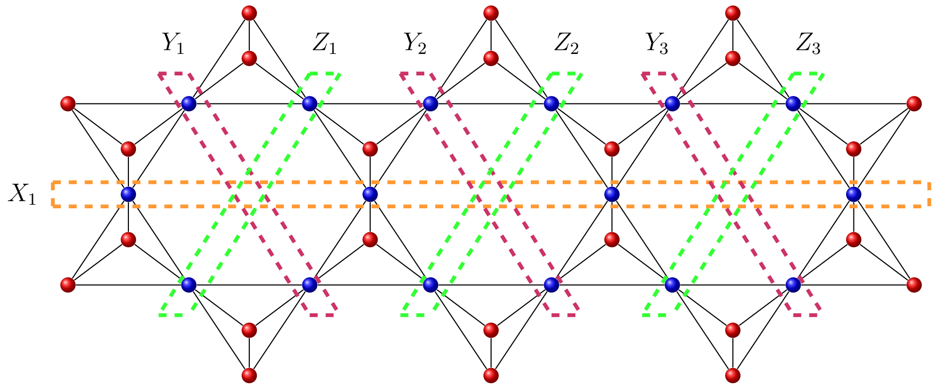


Figure 4.19: Vertex partitions of Double Chain Silicates network DS_3.

Theorem 4.4.1. *Let DS_n be a Double Chain Silicates network of dimension n. Then,*

$$TW_{\Delta}(DS_n) = \frac{25}{3}n^3 + \frac{23}{2}n^2 + \frac{43}{6}n. \tag{4.23}$$

Proof. Let consider the vertex cuts shown in Figure 4.19, such that we have two types of cuts: horizontal and diagonal cuts.

Let X_1 be the horizontal vertex cut of DS_n. For $1 \leq i \leq n$, let Y_i denote the diagonal vertex cuts along the North-West and South-East directions, and Z_i denote the diagonal vertex cuts along the South-West and North-East directions. Obviously, the set of corner vertices, which are represented with red color in Figure 4.19, do not belong on the defined vertex cuts. We have the number of vertices of degree $\Delta = 6$ is equal to: $N_{\Delta} = 5n + 1$.

Then,

$$n_1^{\Delta}(X_1) = 2n, \qquad n_2^{\Delta}(X_1) = N_{\Delta} - (n_1^{\Delta}(X_1) + |X_1|).$$

And, for $1 \leq i \leq n$:

$$n_1^{\Delta}(Y_i) = 5i - 3, \qquad n_2^{\Delta}(Y_i) = N_{\Delta} - (n_1^{\Delta}(Y_i) + |Y_i|).$$

Due to the two-fold symmetry of DS_n network, we have $n_1^{\Delta}(Y_i) = n_1^{\Delta}(Z_i)$ and $n_2^{\Delta}(Y_i) = n_2^{\Delta}(Z_i)$. Thus, by applying the vertex cut Theorem 4.3.4, we have

$$\begin{aligned} TW_{\Delta}(DS_n) &= n_1^{\Delta}(X_1)n_2^{\Delta}(X_1) + \sum_{i=1}^{n} n_1^{\Delta}(Y_i)n_2^{\Delta}(Y_i) + \sum_{i=1}^{n} n_1^{\Delta}(Z_i)n_2^{\Delta}(Z_i) + \binom{5n+1}{2} \\ &= 4n^2 + 2\sum_{i=1}^{n} n_1^{\Delta}(Y_i)n_2^{\Delta}(Y_i) + \frac{5n(5n+1)}{2}. \end{aligned}$$

Which gives the Equation 4.23. □

Figure 4.20 represents the behavior of the generalized Terminal Wiener index of the DS_n network. We can observe that this measure increases by changing the value of the dimension n. Moreover, the value of the generalized Terminal Wiener index approximates to $\frac{25}{3}n^3$ when the parameter n get larger enough.

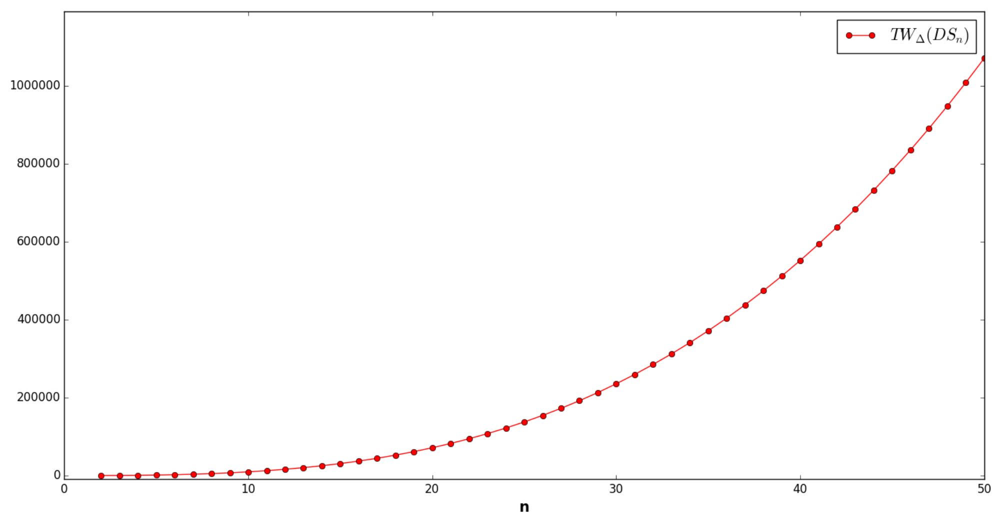


Figure 4.20: The graphical representation of $TW_{\Delta}(DS_n)$.

4.4.2 Silicates Sheet Network

Let SL_n be a Silicates Sheet network of dimension n, where n is the number of hexagons between the center and boundary of SL_n.

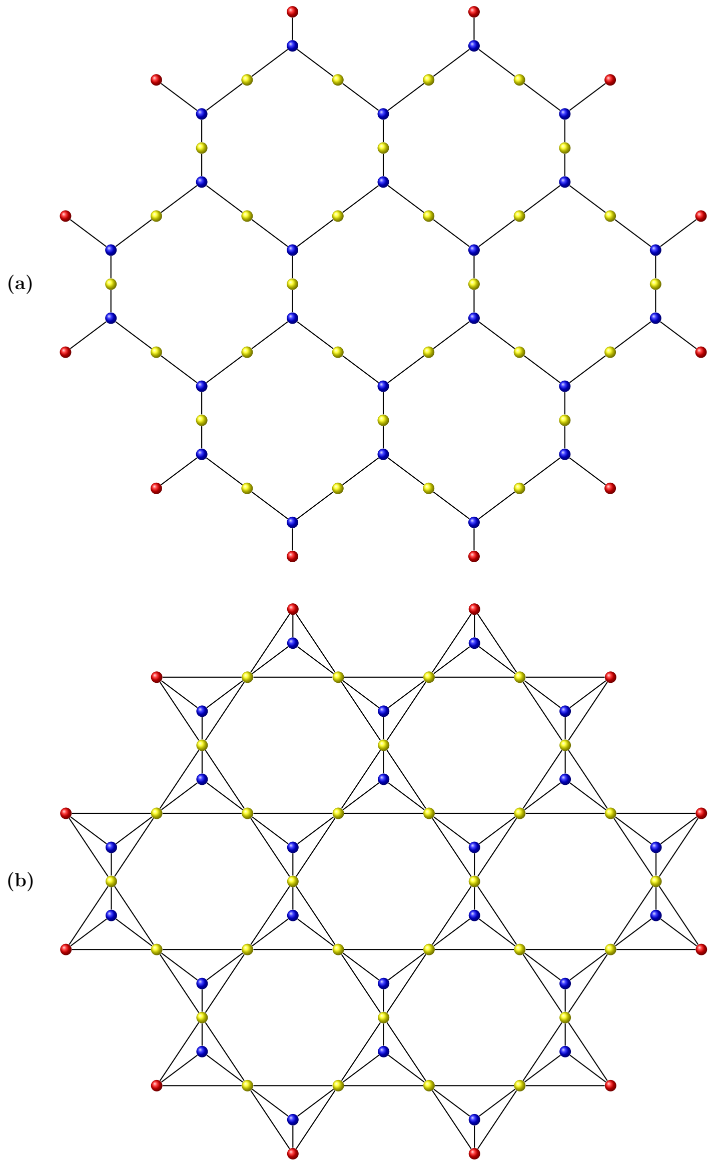


Figure 4.21: Silicates Sheet network construction. (a): A Honeycomb of dimension $n = 2$ with its subdivision. (b): The obtained Silicates Sheet network of dimension $n = 2$.

The construction method of a Silicates Sheet network is obtained from a Honeycomb network. At first, we consider a Honeycomb network of dimension n. Then, we add $6n$ new pendent edges on each 2-degree vertex of the Honeycomb network and we subdivide each edge of this structure except the pendent edges by adding a new vertex. Finally, we associate each adjacent new vertices in order to form a tetrahedron SiO_4 as in Figure 4.21 (b). From the construction method of the Silicates Sheet network SL_n, we have the following structural properties.

- The number of vertices of SL_n is:

$$N = 15n^2 + 3n.$$

- The number of edges of SL_n is:

$$M = 36n^2.$$

- The Diameter of SL_n is:

$$D = 4n.$$

Now, we apply the vertex-cut method on the SL_n network. We start by the classification of the vertex set as shown in Figure 4.22. The following theorem shows the exact analytical expression for the generalized Terminal Wiener index TW_K, with $K = \Delta$.

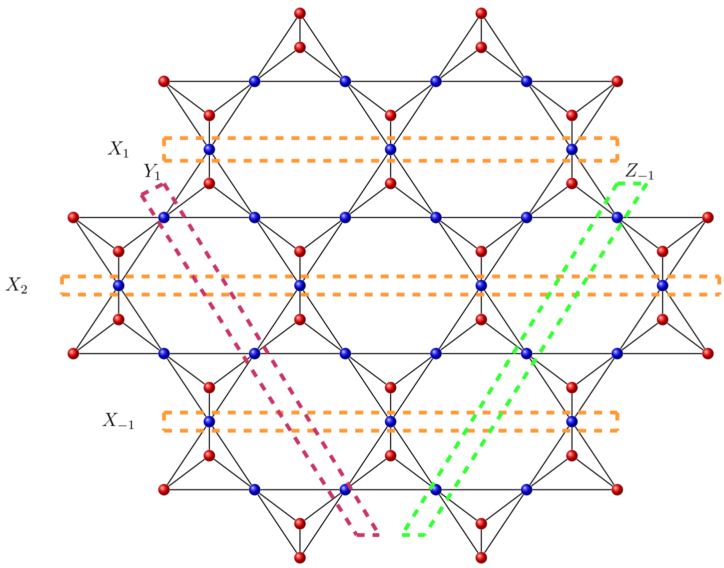


Figure 4.22: Some vertex cuts of Silicates Sheet network SL_2.

Theorem 4.4.2. *Let SL_n be a Silicates Sheet network of dimension n. Then,*

$$TW_\Delta(SL_n) = \frac{369}{5}n^5 - \frac{123}{2}n^4 + 15n^3 - \frac{3}{2}n^2 + \frac{6}{5}n. \tag{4.24}$$

Proof. Let consider the vertex cuts shown in Figure 4.22, such that we have two types of cuts: horizontal and diagonal cuts.

Let X_i, for $1 \leq i \leq n$, and X_{-i}, for $1 \leq i \leq n-1$, be the horizontal vertex cuts of SL_n. Similarly, let Y_i, Y_{-i} denote the diagonal vertex cuts along the North-West and south-East directions and Z_i, Z_{-i} be the diagonal edge cuts along the South-West and North-East directions, for $1 \leq i \leq n$ and $-1 \leq -i \leq -(n-1)$. Obviously, The set of vertices which do not lie on the defined vertex cuts are the corner vertices. We have the number of vertices of degree $\Delta = 6$ is equal to $N_\Delta = 9n^2 - 3n$.

Then, for $1 \leq i \leq n$:

$$\begin{aligned}
&n_1^\Delta(X_1) = 2n.\\
&n_2^\Delta(X_1) = N_\Delta - (n_1^\Delta(X_1) + |X_1|).\\
&n_1^\Delta(X_i) = 2n + \textstyle\sum_{j=1}^{i-1} \Big[(2n+2j) + (n+j)\Big].\\
&n_2^\Delta(X_i) = N_\Delta - (n_1^\Delta(X_i) + |X_i|).
\end{aligned}$$

Using the symmetry, we have $n_1^\Delta(X_i) = n_1^\Delta(Y_i) = n_1^\Delta(Z_i) = n_1^\Delta(X_{-i}) = n_1^\Delta(Y_{-i}) = n_1^\Delta(Z_{-i})$, for $1 \leq i \leq n$ and $-1 \leq -i \leq -(n-1)$.

From the application of the vertex-cut Theorem 4.3.4, we get

$$\begin{aligned}
TW_\Delta(SL_n) = &\sum_{i=1}^{n} n_1^\Delta(X_1) n_2^\Delta(X_1) + \sum_{i=1}^{n-1} n_1^\Delta(X_{-1}) n_2^\Delta(X_{-1}) + \sum_{i=1}^{n} n_1^\Delta(Y_i) n_2^\Delta(Y_i) +\\
&\sum_{i=1}^{n-1} n_1^\Delta(Y_{-i}) n_2^\Delta(Y_{-i}) + \sum_{i=1}^{n} n_1^\Delta(Z_i) n_2^\Delta(Z_i) + \sum_{i=1}^{n-1} n_1^\Delta(Z_{-i}) n_2^\Delta(Z_{-i}) +\\
&\binom{9n^2-3n}{2}\\
= &\,3\sum_{i=1}^{n} n_1^\Delta(X_1) n_2^\Delta(X_1) + \binom{9n^2-3n}{2} + 3\sum_{i=1}^{n-1} n_1^\Delta(X_{-1}) n_2^\Delta(X_{-1}).
\end{aligned}$$

Which gives the Equation 4.24. □

Figure 4.23 shows the behavior of the generalized Terminal Wiener index of the SL_n network. We can observe that this measure increases by changing the value of the dimension n and approximates to $\frac{369}{5}n^5$ when n get larger enough.

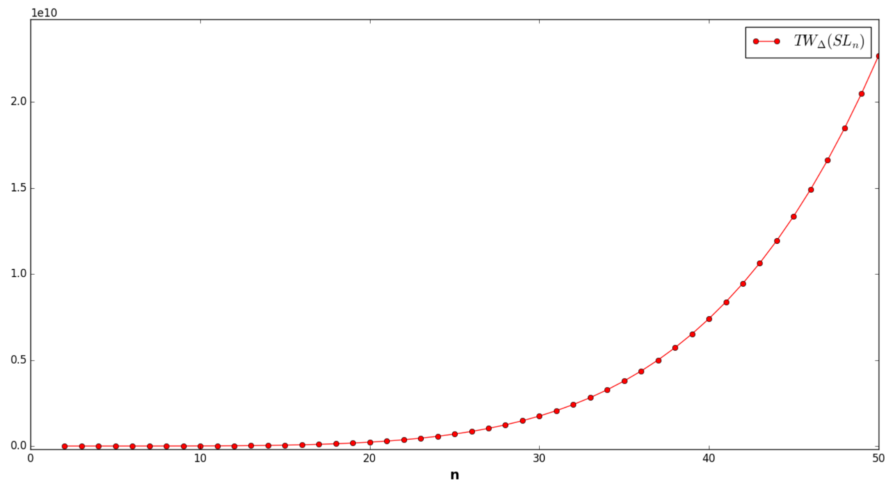


Figure 4.23: The graphical representation of $TW_\Delta(SL_n)$.

4.5 Summary

In order to analyze and quantify the topological structure of large and complex networks, we have investigated a powerful approach called the cut method. The power of this method is that it allows the computation of distance-based topological indices without using the concept of distance. In this chapter, we discussed two versions of the cut method for calculating the generalized Terminal Wiener index avoiding the computation of shortest paths. The first version is the edge-cut method and its objective is to partition the edge set of a graph G into classes. Then, by assembling the indices of the resulting components we can generate the property of the whole graph G. Before discussing the edge-cut method and its kinds, we introduced some basic concepts that are required to understand this technique, such as the class of partial cubes and the relation Θ. Then, we introduced the first type of the edge-cut method called the standard cut method, which is applicable only for partial cubes. We showed the first instance of this method for computing the Wiener index of partial cubes as well as its application for computing the generalized Terminal Wiener index. In order to prove the efficiency of the standard cut method, we presented an illustration for calculating the Wiener index and the generalized Terminal Wiener index of a benzenoid system. After that, we studied the first extension of the standard cut method for computing the Wiener index of a class of graphs larger than partial cubes called l_1-graphs. Moreover, we presented our work concerning the generalization of the edge-cut method for computing the generalized Terminal Wiener index of this class of graphs and we showed the efficiency of this method by analyzing the topological structure of a kind of large networks named Equilateral Triangular Tetra Sheet network $ETTS_n$ and Hex Derived network HD_n. The second version of the cut method is based on vertex-partitions and called the vertex-cut method. We discussed the first apparition of this method for computing the Wiener index. Then, we developed this method to derive a new formula for computing the generalized Terminal Wiener index. Finally, we used the proposed approach to obtain exact analytical expressions of the generalized Terminal Wiener index of some Silicate networks as well as to demonstrate its significance and effectiveness.

In the next chapter, we continue our work on the cut method but this time we investigate another extension of the edge-cut method, which can be a useful tool to design a fast algorithm for computing the generalized Terminal Wiener index. Moreover, the new algorithm can be implemented to run in linear time for some classes of graphs.

Chapter

5 Reducing the Computational Complexity of the Generalized Terminal Wiener Index

Contents

Is there a fast algorithm for general graphs that would calculate distance-based topological indices without calculating the distance matrix? This question is an open problem that was posed by Mohar and Pisanski (1988). In their paper, they introduced an algorithm to find the Wiener index of acyclic graphs in linear time and motivated the researchers to find new methods and develop algorithms that would compute the Wiener index without using the distance matrix. As a consequence, a lot of works have been done in this direction for some kind of graphs. For instance, Klavzar et al. (1995) introduced the standard cut method to compute the Wiener index of partial cubes and Chepoi et al. (1997) discussed the first extension of the standard cut method. These two last methods were investigated in details in the last chapter. In (Chepoi and Klavžar, 1997), it was proved that the Wiener index of a hexagonal graph G can be expressed as the sum of the

Wiener indices of three weighted quotient trees, which are obtained from the canonical metric representation of G. Moreover, this observation leads to a linear algorithm for computing the Wiener index of hexagonal graphs. Later, similar linear algorithms were developed for calculating some other distance-based topological indices; see (Kelenc et al., 2015), (Tratnik, 2016) and (Črepnjak and Tratnik, 2017). Furthermore, the relation between the Wiener index of a graph and its canonical metric representation was extended to general graphs by Klavžar (2006).

Our aim in this chapter is to extend the mentioned works done on reducing the computation of the Wiener index to the case of the generalized Terminal Wiener index. At first, we introduce the concept of the canonical metric representation of a graph G and we use this concept to reduce the computation of the generalized Terminal Wiener index to the calculation of the Wiener index of the appropriately quotient graphs. Then, we apply the main result on an hierarchical network to demonstrate the efficiency of the method as well as to compare this technique with some other algorithms based on the calculation of the distance matrix. After reducing the computation of the generalized Terminal Wiener index for general graphs, we move to the case of partial cubes and we prove that this index of such graphs can be expressed as the sum of the Wiener indices of a set of weighted quotient graphs with respect to a combination of partition. As a result, we present a linear algorithm for a system that full under partial cubes called the graphenylene system and we prove its correctness. Finally, we show the ease of the discussed method to quantify the topological structure of some complex partial cubes, such as the graphenylene chain network GC_n and the graphenylene sheet network GS_n. We note that the first part of this chapter, which treats the generalized Terminal Wiener index for general graphs, was presented in an international conference (Zeryouh et al., 2018a) and the rest of the chapter was published in an international journal (Zeryouh et al., 2019).

5.1 Computing The Generalized Terminal Wiener Index Via Canonical Metric Representation

In the last chapter, we discussed the standard cut method and its first extension applied on l_1-graphs. Here, we introduce an other extension that generalizes the standard cut method to general graphs. The main purpose of this method is to transform an original graph G into smaller graphs called quotient graphs using a compulsory concept called the canonical metric representation. Then, the given topological index of the original graph G is obtained from the quotient graphs. The aim of this section is to extract the connection between the generalized Terminal Wiener index of a graph G and its canonical metric representation to reduce its computation. For describing and proving our main result, we need to recall and illustrate some concepts that are presented in the following subsection.

5.1.1 Some Basic Concepts and the Canonical Metric Representation of Graphs

Let recall from the subsection 4.1.1 some important definitions and facts. The relation Θ partitions the edge set of a partial cube into equivalent classes $E_1, ..., E_q$ called Θ-classes. The transitive closure Θ^* is the smallest transitive relation containing Θ and partitions the edge set of any connected graph G into equivalent classes $F_1, ..., F_k$ called Θ^*-classes, such that the graph $G - F_i$ contains two or more connected components. The quotient graph G/F_i, for $i = 1, ..., k$, is a graph in which the vertices are the connected components of $G - F_i$. Two vertices C_1 and C_2 in G/F_i being adjacent if there exist vertices $x \in C_1$ and $y \in C_2$ such that $xy \in F_i$.

Let define the mapping $\alpha : G \to \Box_{i=1}^{k} G/F_i$ with

$$\alpha : u \to (\alpha_1(u), ..., \alpha_k(u)),$$

where $\alpha_i(u)$ is the connected component of $G - F_i$ that contains the vertex u. Then, α is the canonical metric representation of the graph G, also called the canonical embedding of G. This representation was proposed by Graham and Winkler (1985) and has many interesting properties. For instance, α is an irredundant isometric embedding, where irredundant means that every factor graph G/F_i has at least two vertices and each vertex of G/F_i appears as a coordinate of some vertex $\alpha(u)$. For more results on the canonical metric representation see the original paper (Graham and Winkler, 1985). Figure 5.1 shows an example of a graph G and its canonical metric representation. In this example, the graph G has two Θ^*-classes, which are $G - F_1$ and $G - F_2$. From the connected components of $G - F_1$ and $G - F_2$, we obtained the quotient graphs G/F_1 and G/F_2. Thus, $\alpha : G \rightarrow G/F_1 \,\square\, G/F_2$ is the canonical metric representation of G.

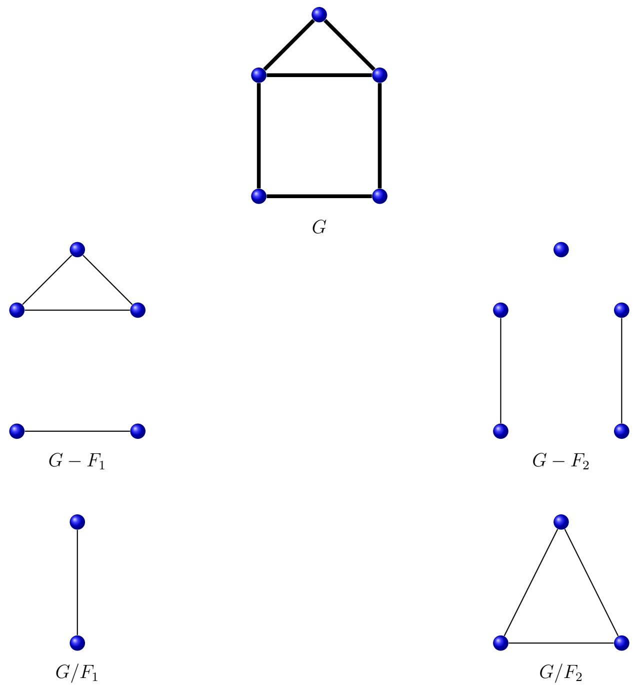


Figure 5.1: The canonical metric representation of the graph G.

Now, we define the weighted version of the Wiener index. Let (G, ω) be a weighted graph, such that the weight function $\omega : V(G) \rightarrow \mathbf{R}$ assigns positive real numbers to the vertices of G. Then, the weighted Wiener index is defined in (Klavžar et al., 1997) as follows:

$$W(G, \omega) = \sum_{\{u,v\} \subseteq V(G)} \omega(u)\omega(v)d_G(u, v). \tag{5.1}$$

Let $(G/F_i, \omega_i)$ be a weighted quotient graph, such that the weight function $\omega_i : V(G/F_i) \rightarrow \mathbf{R}$ assigns to a vertex of G/F_i the number of vertices in the corresponding connected component of $G - F_i$. Figure 5.2 illustrates the weighted quotient graphs obtained from the previous example depicted in Figure 5.1.

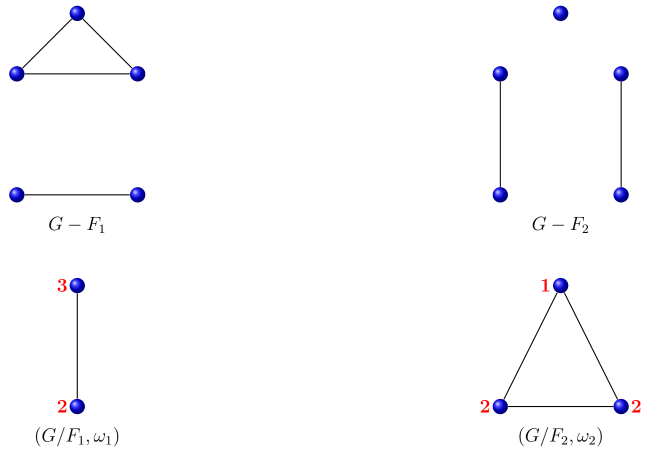


Figure 5.2: The construction of weighted quotient graphs.

Finally, by using the defined notions, Klavžar (2006) arrived to reduce the computation of the Wiener index by establishing the relation between the Wiener index of a graph G and its canonical metric representation. In the following subsection, the obtained result by Klavžar (2006) is extended to the case of the generalized Terminal Wiener index.

5.1.2 Reducing the Computation of the Generalized Terminal Wiener Index

In this part, we prove that the computation of the generalized Terminal Wiener index can be reduced to the computation of the Wiener index of the appropriately quotient graphs. Let define the canonical metric representation of a graphs G as follows:

$$\alpha : G \to \square_{i=1}^{k} G/F_i$$
$$u \to (\alpha_1(u), ..., \alpha_k(u)),$$

where $\alpha_i(u)$ is the connected components of $G - F_i$ that contains **the vertex u of degree K**. Then, we define the weighted quotient graphs $(G/F_i, \omega_i)$, such that the weight of a vertex of G/F_i is **the number of vertices of degree K** in the corresponding connected component of $G - F_i$. It should be noted that we must consider only the vertices of degree K that already exist in the original graph G. For instance, in Figure 5.3, we constructed the quotient graphs of the connected components $G - F_1$ and $G - F_2$ in the case of $K = 2$. Finally, the computational complexity of the generalized Terminal Wiener index of a graph G can be reduced as follows:

Theorem 5.1.1. *For any connected graph G with N_K vertices of degree K, we have:*

$$TW_K(G) = \sum_{1 \leq i \leq k} W(G/F_i, \omega_i). \tag{5.2}$$

Proof. Let $C_1^{(i)}, ..., C_{r_i}^{(i)}$ be the connected components of $G - F_i$, with $1 \leq i \leq k$. We denote by $|V_K(C_j^{(i)})|$ the number of vertices of degree K in the component $C_j^{(i)}$.
By using the defined canonical metric representation α, we can see that:

$$\begin{aligned} TW_K(G) &= \sum_{\{u,v\} \subseteq V_K(G)} d_G(u, v) \\ &= \sum_{\{u,v\} \subseteq V_K(G)} d_G(\alpha(u), \alpha(v)) \\ &= \sum_{\{u,v\} \subseteq V_K(G)} \sum_{i=1}^{k} d_{G/F_i}(\alpha_i(u), \alpha_i(v)) \\ &= \sum_{i=1}^{k} \sum_{1 \leq j \leq j' \leq r_i} d_{G/F_i}\left(C_j^{(i)}, C_{j'}^{(i)}\right) |V_K(C_j^{(i)})||V_K(C_{j'}^{(i)})| \\ &= \sum_{i=1}^{k} W(G/F_i, \omega_i). \end{aligned}$$

$\square$

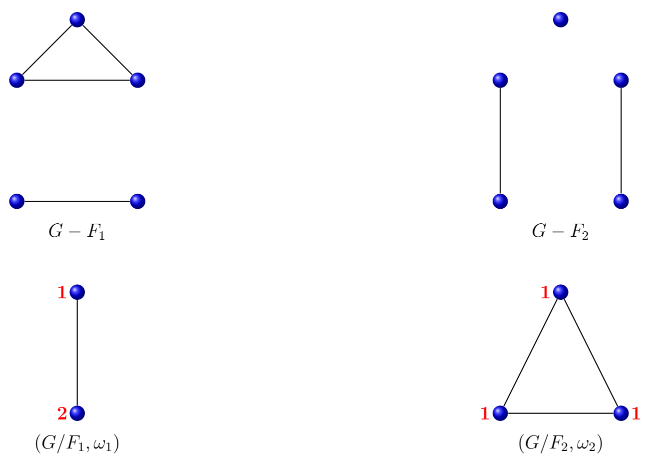


Figure 5.3: The obtained weighted quotient graphs, for $K = 2$.

5.2 Application Of The Canonical Metric For Quantifying The Topological Structure Of An Hierarchical Network

In this section, we apply the proposed method for computing the generalized Terminal Wiener index of an hierarchical network called the Dendrimer graph. Our main focus is to prove that this technique is an efficient tool for a hand manipulation and is useful to design a new algorithm that reduces the computational complexity of the generalized Terminal Wiener index. At first, we show the construction method of the Dendrimer graph $\mathcal{D}_{d,h}$ and we analyze some of its structural properties. Then, we give the analytical expression of the generalized Terminal Wiener index of this structure for $K = 1$, which clearly refers to the case of the Terminal Wiener index. After that, we develop an algorithm in order to compare the performance of the new technique with the other algorithms that already exist in the literature.

5.2.1 The Generalized Terminal Wiener Index of the Dendrimer Graph

The Dendrimer graph is an interesting topology with a wide range of applications as the Dendrimer tree that was discussed in subsection 3.3.1. The unique difference between these two topologies is in the contents of the core level.

Let $\mathcal{D}_{d,h}(d \geq 2, h \geq 0)$ be a Dendrimer graph, where h denotes the levels or iterations and d the number of vertices added to every vertex in each iteration. The $\mathcal{D}_{d,h}$ can be built in the following iterative way. Initially, the Dendrimer graph is composed of a core $\mathcal{D}_0$ that contains a cycle of order n' and $d-1$ vertices attached to each vertex of the cycle, which gives p_0 pendent vertices. $\mathcal{D}_{d,1}$ is obtained from $\mathcal{D}_0$ by adding d vertices to each pendent vertex of $\mathcal{D}_0$. Similarly, we obtain $\mathcal{D}_{d,h}$ from $\mathcal{D}_{d,h-1}$ by attaching d vertices to every pendent vertex of $\mathcal{D}_{d,h-1}$. Figure 5.4 illustrates some iterations of the Dendrimer graph $\mathcal{D}_{d,h}$. The internal vertices of the network $\mathcal{D}_{d,h}$ have degree equal to $d+1$, and the level h denotes the distance between all pendent vertices and terminal vertices of the core $\mathcal{D}_0$.

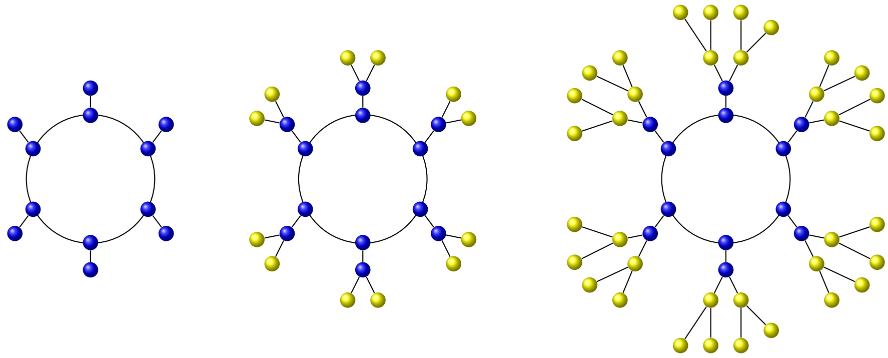


Figure 5.4: (Left) The core $\mathcal{D}_0$ of the Dendrimer graph, with $n' = 6$ and $p_0 = 6$. (Middle and Right) Dendrimer graphs $\mathcal{D}_{d,h}$, with $d = 2$ and $h = \{1, 2\}$.

From the construction method of the Dendrimer graph $\mathcal{D}_{d,h}$, we can derive the subsequent structural properties:

- The number of pendent vertices of $\mathcal{D}_{d,h}$ is:

$$p_h = d^h(d-1)n'. \tag{5.3}$$

- The number of vertices of $\mathcal{D}_{d,h}$ is equal to:

$$N_h = n' + n'(d^{h+1} - 1). \tag{5.4}$$

- The diameter of $\mathcal{D}_{d,h}$ is equal to :

$$D_h = \lfloor \frac{n'}{2} \rfloor + 2(h+1). \tag{5.5}$$

Now, we apply the proposed method to compute the Terminal Wiener index of the Dendrimer graph $\mathcal{D}_{d,h}$. Before of that, we express the Wiener index of a weighted cycle, which is considered as a crucial Lemma for proving our main result.

Lemma 5.2.1. *Let $(\mathcal{C}_{n'}, \omega)$ be a weighted cycle of order n', such that all the vertices have* *the same weight ω. Then,*

$$W(\mathcal{C}_{n'},\omega)=\begin{cases}\frac{1}{8}n'^3\omega^2 & \text{if } n' \text{ is even,}\\ \\ \frac{1}{8}(n'^3-n')\omega^2 & \text{if } n' \text{ is odd.}\end{cases} \tag{5.6}$$

Proof. Simply, by using the definition of the weighted Wiener index with identical weights $\omega(u)=\omega(v)$. □

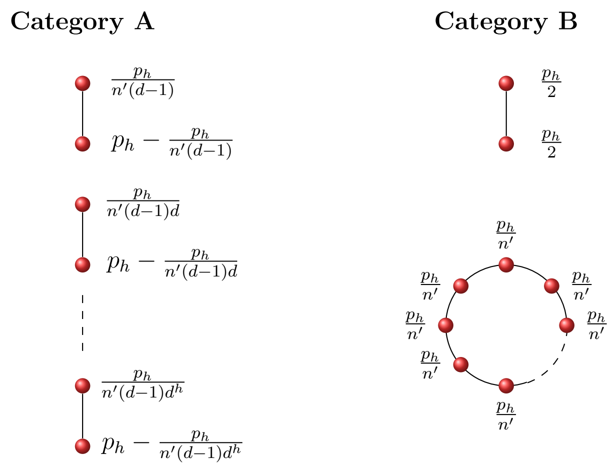


Figure 5.5: The weighted quotient graphs of the Dendrimer graph $\mathcal{D}_{d,h}$.

Theorem 5.2.1. *Let $\mathcal{D}_{d,h}$ be a Dendrimer graph, with $d \geq 2$ and $h \geq 0$. Then,*

- *If n' is even:*

$$TW(\mathcal{D}_{d,h})=n'd^{2h}\left[(d-1)^2\left(\frac{1}{8}n'^2+n'(h+1)\right)-d\right]+n'd^h. \tag{5.7}$$

- *If n' is odd:*

$$TW(\mathcal{D}_{d,h})=n'd^{2h}\left[(d-1)^2\left(\frac{1}{8}(n'^2-1)+n'(h+1)\right)-d\right]+n'd^h. \tag{5.8}$$

Proof. At first, we determine the classes of the Dendrimer graph $\mathcal{D}_{d,h}$. In the case of an even n', it is easy to observe that an edge e of the cycle $\mathcal{C}$ of the graph $\mathcal{D}_{d,h}$ is in relation θ with its antipodal edge on $\mathcal{C}$. Otherwise, all the edges of the cycle $\mathcal{C}$ will be in the same θ^*-class.

Then, we extract the corresponding weighted quotient graphs of the Dendrimer graph $\mathcal{D}_{d,h}$ as illustrated in Figure 5.5. In category A, we represent the weighted quotient graphs obtained by removing an edge that doesn't belong to the cycle of the core $\mathcal{D}_0$, and category B contains the weighted quotient graphs obtained by removing an edge or all the edges of the cycle of the core $\mathcal{D}_0$.
Each weighted quotient graph from category A is repeated $d^i(d-1)n'$ times, with $0 \leq i \leq h$. In the case of an even cycle, the weighted quotient graph from category B; which is depicted in the top of this category, is repeated $\frac{n'}{2}$ times.
Finally, we apply the reduced Theorem 5.1.1 with the above observations, Lemma 5.2.1 and the structural properties of $\mathcal{D}_{d,h}$.

- if n' is even

$$TW(\mathcal{D}_{d,h}) = \frac{n'}{2}\left[\frac{p_h}{2}\frac{p_h}{2}\right] + \sum_{i=0}^{h} d^i(d-1)n'\left[\frac{p_h}{n'(d-1)d^i} * \left(p_h - \frac{p_h}{n'(d-1)d^i}\right)\right].$$

- if n' is odd

$$TW(\mathcal{D}_{d,h}) = \left(\frac{p_h}{n'}\right)^2\left(\frac{1}{8}n'^3 - \frac{1}{8}n'\right) + \sum_{i=0}^{h} d^i(d-1)n'\left[\frac{p_h}{n'(d-1)d^i}\left(p_h - \frac{p_h}{n'(d-1)d^i}\right)\right].$$

Which yield the Equations 5.7, 5.8, respectively. □

From the proof of Theorem 5.2.1, we can deduce that this technique is an easy tool for a hand manipulation. Broadly speaking, by using the proposed method, we focus only on counting the number of pendent vertices in the corresponding connected components and constructing the appropriate quotient graphs. In Figure 5.6, we show the behavior of the Terminal Wiener index of the Dendrimer graph $\mathcal{D}_{d,h}$, where the considered order of the cycle is $n' = 6$. Along the X-axis and Y-axis, we have the number of pendent vertices d that are added in each iteration and the number of iterations h, respectively. In the Z-axis the values of the Terminal Wiener index of $\mathcal{D}_{d,h}$ are determined. We can observe that the Terminal Wiener index of this structure is increasing by changing the values of the parameters d and h. The large changes in the perceptual appearance of the colors clarify the behavior of the Terminal Wiener index for this structure.

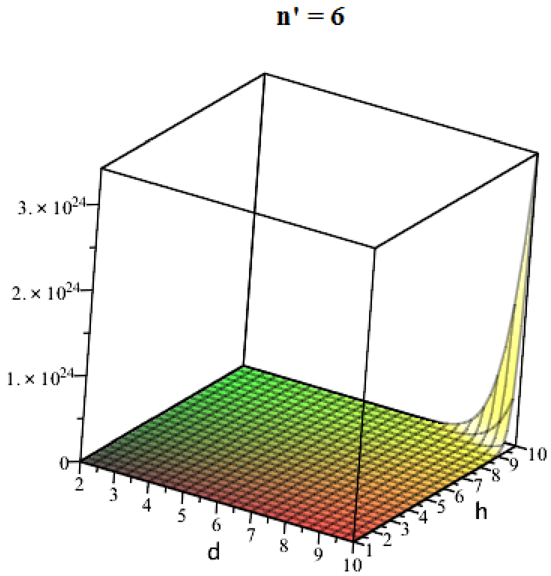


Figure 5.6: The graphical representation of the Terminal Wiener index of the Dendrimer graph $\mathcal{D}_{d,h}$.

5.2.2 An Algorithm for Computing the Generalized Terminal Wiener Index of the Dendrimer Graph

In this subsection, we prove the usefulness of the method based on the canonical metric for developing a new algorithm that reduces the computation of the generalized Terminal Wiener index of the Dendrimer graph $\mathcal{D}_{d,h}$. Then, we compare the proposed algorithm with some other algorithms that exist in the literature.

Here, we consider the case of $K \geq 1$. Therefore, the generalized Terminal Wiener index of the Dendrimer graph $\mathcal{D}_{d,h}$ via canonical metric representation contains three main steps, which can be arranged to design an algorithm as follows:

[1]A linear algorithm proposed by Mohar and Pisanski (1988) and then by Fahd and Jamil (2018) for computing the Wiener index of trees.

Algorithm 1: Generalized Terminal Wiener index of $\mathcal{D}_{d,h}$

Input: adjList: The adjacency list of $\mathcal{D}_{d,h}$; n': the order of the cycle; K: the considered degree of vertices; N_K: the number of vertices with degree K.

Output: $TW_K(\mathcal{D}_{d,h})$

1 Compute the Wiener index W_A of the quotient graphs in category A by using the LTA method[1] :

```
for all u_i in D_{d,h} do
    Compute deg(u_i) the degree of vertex u_i.
    if u_i is a leaf then
        Push u_i into the stack.
    if deg(u_i) = K then
        Set the weight ω(u_i) equal to 1.
    else
        Set the weight ω(u_i) equal to 0.
while Stack is nonempty do
    Remove the last pushed leaf u_i from the stack.
    W_A = W_A + ω(u_i) * (N_K − ω(u_i)).
    Set deg(u_i) = 0.
    for each neighbor of u_i do
        if deg(neighbor) > 0 then
            Set ω(neighbor) = ω(neighbor) + ω(u_i).
            Set deg(neighbor) = deg(neighbor) − 1.
            if neighbor is a leaf then
                Push neighbor into the stack.
```

2 Compute the Wiener index W_B of the quotient graphs in category B as follows:

```
for all u_i in the cycle of D_0 do
    Set the weight ω(u_i) equal to the number of vertices with degree K in the
     tree attached to u_i.
Set SumWeight = 0.
if n' is even then
    for i = 1 to n'/2 do
        SumWeight = SumWeight + ω(u_i).
    W_B = n'/2 * [SumWeight * (N_K − SumWeight)].
if n' is odd then
    for i = 1 to ⌊n'/2⌋ do
        SumWeight = SumWeight + ω(u_i).
    W_B = n' * [SumWeight * (N_K − SumWeight)] * 1/2.
```

3 Apply Theorem 5.2.1 such that:

$TW_K(\mathcal{D}_{d,h}) = W_A + W_B$.

5.2.2.1 Evaluation and Comparison with some Algorithms

Here, we evaluate the performance of our algorithm by comparing the runtime results of the proposed algorithm (PA) with the execution times of the Breadth First Search (BFS) and Floyd-Warshall (FW) algorithms. For further details on the BFS and FW algorithms, we refer to see (Mohar and Pisanski, 1988) and the book (Hammack et al., 2011).

We involved python 2.7.12 to implement all the algorithms and we considered the case of $K = 1$. For constructing a Dendrimer graph $\mathcal{D}_{d,h}$, we used the NetworkX library, which is a Python package dedicated to create, manipulate and analyze complex networks. Concerning the running times of the experiments are obtained by using the Jupyter notebook environment and particularly through the magic function ”timeit”, a module used to calculate how long a script takes to run averaged over multiple runs. It should be noted that the reported computational results are obtained on a computer with an Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz OS (64-bits) and 4GB RAM memory. The resulting running times are presented in Figure 5.7 and Table 5.1.

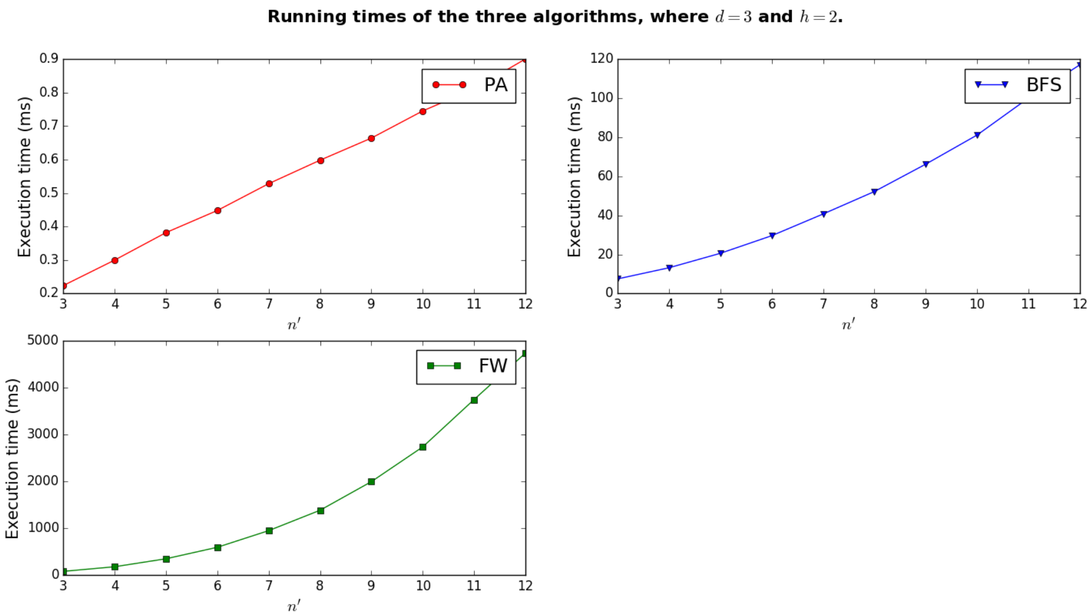


Figure 5.7: The graphical representation of the time executions for a set of Dendrimer graphs $\mathcal{D}_{d,h}$, where $d = 3$ and $h = 2$.

Table A: Dendrimer graphs $\mathcal{D}_{d,h}$ with an even cycle of order $n' = 8$.

	$n' = 8$											
	d = 2				d = 3				d = 4			
h	N_h	PA	BFS	FW	N_h	PA	BFS	FW	N_h	PA	BFS	FW
1	32	0.085	0.910	4.81	72	0.197	5.98	55	128	0.351	21.4	311
2	64	0.177	3.53	35.3	216	0.605	52.9	1400	512	1.44	342	19900
3	128	0.350	14	268	648	1.82	487	39000	2048	5.94	5870	2613000
4	256	0.719	55.6	2100	1944	5.75	4570	2104000	8192	24.4	152000	–
5	512	1.46	225	17200	5832	17.7	4770	–	32768	101	3718000	–

Table B: Dendrimer graphs $\mathcal{D}_{d,h}$ with an odd cycle of order $n' = 7$.

	$n' = 7$											
	d = 2				d = 3				d = 4			
h	N_h	PA	BFS	FW	N_h	PA	BFS	FW	N_h	PA	BFS	FW
1	28	0.0784	0.692	3.34	63	0.175	4.57	37.3	112	0.312	16.1	207
2	56	0.158	2.7	23.8	189	0.528	40.4	936	448	1.27	260	13300
3	112	0.316	10.6	180	567	1.62	372	25800	1792	5.17	4480	–
4	224	0.630	42.3	1410	1701	5.08	3520	1401000	7168	21.2	86000	–
5	448	1.27	170	11300	5103	15.2	35200	–	28672	88.7	2827000	–

Table 5.1: The resulting running times (ms) for a set of Dendrimer graphs $\mathcal{D}_{d,h}$. Note that N_h is the order of the corresponding $\mathcal{D}_{d,h}$.

Table 5.1 gives the runtime results of the generalized Terminal Wiener index ($K = 1$) of a set of Dendrimer graphs by using the three algorithms: Breadth First Search (BFS) algorithm, Floyd-Warshall (FW) algorithm and the Proposed Algorithm (PA). The parameters d and h denote the number of new vertices added in each iteration and the number of iterations, respectively. For each d and h, we calculated the order N_h of the Dendrimer graph $\mathcal{D}_{d,h}$. In the above table (Table A), we considered the case of Dendrimer graphs with an even cycle of order $n' = 8$. Otherwise, Table B represents the case of Dendrimer graphs with an odd cycle of order $n' = 7$. The execution times of the algorithms are given in milliseconds. Clearly, we can observe that the PA has the best performance over all the experiments. The BFS and the FW algorithms have the worst performance and especially when the order N_h is increasing.

In Figure 5.7, we show the behavior of the three algorithms to calculate the generalized Terminal Wiener index of Dendrimer graphs in the case of $K = 1$, $d = 3$ and $h = 2$. Along the horizontal line we have the order n' of the cycle and along the vertical line the execution times in milliseconds are obtained. We can see that the two algorithms BFS and FW show a polynomial grow when n' increases. Clearly, this polynomial representation proves the time complexity of these two algorithms that was demonstrated in (Mohar and Pisanski, 1988) to be equal $O(mn)$ and $O(n^3)$, respectively, where n is the order of a graph and m the number of its edges. In contrast, the runtime of PA grows approximately linearly as the order n' of the cycle increases, which proves the good performance of this method.

In the next section, we use a similar method to the PA and we demonstrate that this approach has a linear time complexity for some classes of graphs.

5.3 A Linear Method For Computing The Generalized Terminal Wiener Index Of Some Systems That Full Under Partial Cubes

In this section, we continue on our study of reducing the computation of the generalized Terminal Wiener index via the concept of the canonical metric representation, but this time we focus our attention only on the class of partial cubes. At first, we prove that the problem of computing the generalized Terminal Wiener index of partial cubes can be reduced to the problem of calculating the weighted Wiener index of quotient graphs using a new kind of partitions. Then, in order to prove that the method can be implemented to run in a linear time complexity, we apply the obtained main result on a system that full under partial cubes called the graphenylene system.

5.3.1 The Generalized Terminal Wiener Index of Partial Cubes

We previously defined the class of partial cubes in the subsection 4.1.1. So, let G be a partial cube and $\varepsilon = \{E_1, ..., E_q\}$ be its θ-classes. In the case of a huge partial cube G, the number of θ-classes can be considerable. Wherefore, we use in the remaining of this study a new partition of the set $E(G)$, denoted by $\mathcal{F}_1, ..., \mathcal{F}_r$, which is coarser than ε. We say that the partition $\mathcal{F}_1, ..., \mathcal{F}_r$ is coarser than ε if and only if $\mathcal{F}_i$ is the union of one or more θ-classes of G, for $1 \leq i \leq r$.

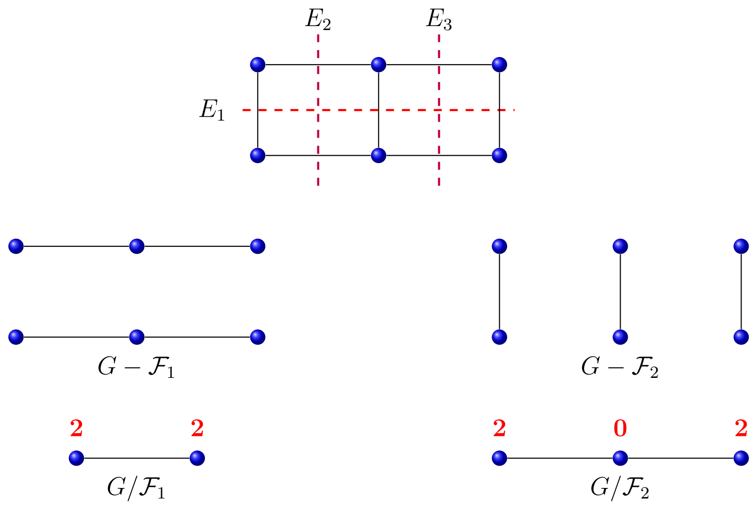


Figure 5.8: The construction of quotient graphs from a coarser partition $\{\mathcal{F}_1, \mathcal{F}_2\}$.

Now, we describe the construction process of weighted quotient graphs using the new partition $\mathcal{F}_1, ..., \mathcal{F}_r$. Let $G/\mathcal{F}_i$ be the quotient graph obtained from the connected components of $G - \mathcal{F}_i$. Two vertices C and C' in $G/\mathcal{F}_i$ being adjacent if there exist vertices $x \in C$ and $y \in C'$ such that $xy \in \mathcal{F}_i$. Let $(G/\mathcal{F}_i, \omega_i)$ be a weighted quotient graph, such that $\omega_i : V(G/\mathcal{F}_i) \to \mathbf{N}$ is the weight function that assigns to a vertex of $G/\mathcal{F}_i$ the number

of vertices of degree K in the corresponding connected components of $G - \mathcal{F}_i$. We note that we must consider only the vertices of degree K that exist in the partial cube G. For instance, the graph G in Figure 5.8 is a partial cube and $\varepsilon = \{E_1, E_2, E_3\}$ is its Θ-classes. The partition $\mathcal{F}_1, \mathcal{F}_2$ is coarser than ε since $\mathcal{F}_1 = E_1$ and $\mathcal{F}_2 = E_2 \cup E_3$. We take as an example $K = 2$. Then, the weighted quotient graphs is constructed from the connected components $G - \mathcal{F}_i$, for $i = \{1, 2\}$, as illustrated in Figure 5.8.
Hence, the computational complexity of the generalized Terminal Wiener index of a partial cube G can be reduced as follows:

Theorem 5.3.1. *Let G be a partial cube with N_K vertices of degree K. Then*

$$TW_K(G) = \sum_{i=1}^{r} W(G/\mathcal{F}_i, \omega_i). \tag{5.9}$$

Proof. Let α be the canonical metric representation of a partial cube G, such that α is defined as:

$$\begin{aligned} \alpha :& G \rightarrow \square_{i=1}^{r} G/\mathcal{F}_i \\ & v \rightarrow (\alpha_1(v), ..., \alpha_r(v)), \end{aligned}$$

where $\alpha_i(v)$ is the connected component of $G - \mathcal{F}_i$ that contains the vertex v of degree K.
Let $C_1^{(i)}, ..., C_{s_i}^{(i)}$ be the connected components of $G - \mathcal{F}_i$ and $\mid V_K(C_j^{(i)}) \mid$ be the number of vertices with degree K in the component $C_j^{(i)}$. It should be noted that $\sum_{j=1}^{s_i} \mid V_K(C_j^{(i)}) \mid = N_K$.
Let $P_{u,v}$ be the shortest path between $u, v \in V_K(G)$. We can verify that $\mid E(P_{u,v}) \cap \mathcal{F}_i \mid = 0$ if u, v belong to the same component $C_j^{(i)}$, otherwise $\mid E(P_{u,v}) \cap \mathcal{F}_i \mid \neq 0$.
Let $C_j^{(i)}$ and $C_{j'}^{(i)}$ be two connected components of $G - \mathcal{F}_i$. To be specific, $C_j^{(i)}$ and $C_{j'}^{(i)}$ are two vertices of $G/\mathcal{F}_i$. It was proved in (Klavžar and Nadjafi-Arani, 2014) that:

$$\mid E(P_{u,v}) \cap \mathcal{F}_i \mid = \mid E(P_{u',v'}) \cap \mathcal{F}_i \mid,$$

where $u, u' \in V(C_j^{(i)})$ and $v, v' \in V(C_{j'}^{(i)})$. This observation yields to:

$$d_{G/\mathcal{F}_i}(C_j^{(i)}, C_{j'}^{(i)}) = \mid E(P_{u,v}) \cap \mathcal{F}_i \mid .$$

Now, we define a function $\delta : V(G) \to \{0, 1\}$ as follows:

$$\delta(P_{u,v}) = \begin{cases} 0; & \text{if } u, v \in C_p^{(i)} \\ 1; & \text{if } u \in C_p^{(i)} \text{and } v \in C_{p'}^{(i)} \end{cases}$$

The summation $\sum_{u,v \in V_K(G)} \delta(P_{u,v})$ is equal to the number of times that we pass through the edges of $\mathcal{F}_i$.

Thus, $\sum_{u,v \in V_K(G)} \delta(P_{u,v}) = \sum_{1 \leq j < j' \leq s_i} \mid V_K(C_j^{(i)}) \mid\mid V_K(C_{j'}^{(i)}) \mid$.

From the above notations and the canonical metric representation α, we can see that:

$$\begin{aligned} TW_K(G) &= \sum_{\{u,v\} \in V_K(G)} d_G(u, v) \\ &= \sum_{\{u,v\} \in V_K(G)} d_G(\alpha(u), \alpha(v)) \\ &= \sum_{\{u,v\} \in V_K(G)} \sum_{i=1}^{r} d_{G/\mathcal{F}_i}(\alpha_i(u), \alpha_i(v)) \\ &= \sum_{\{u,v\} \in V_K(G)} \sum_{i=1}^{r} \mid E(P_{u,v}) \cap \mathcal{F}_i \mid \\ &= \sum_{i=1}^{r} \sum_{1 \leq j < j' \leq s_i} \mid V_K(C_j^{(i)}) \mid\mid V_K(C_{j'}^{(i)}) \mid d_{G/\mathcal{F}_i}(C_j^{(i)}, C_{j'}^{(i)}) \\ &= \sum_{i=1}^{r} W(G/\mathcal{F}_i, \omega_i). \end{aligned}$$

The proof is complete. □

5.3.2 A Linear Time Algorithm for Computing the Generalized Terminal Wiener Index of Graphenylene Systems

In this part, we apply the main result obtained in the previous subsection on a kind of systems that full under partial cubes named the graphenylene system. Our aim is to show that the discussed method is useful to develop an algorithm that compute the generalized Terminal Wiener index in a linear time complexity.

The graphenylene system is a two dimensional structure with an interesting topology, which was proposed by Balaban and Vollhardt (2011). This kind of systems is a connected graph constructed in the following manner. Let $\mathcal{H}$ be a semiregular (4.6.12)-tiling lattice, such that there are one square C_4, one hexagon C_6 and one dodecagon C_{12} on each vertex. Let Z be a circuit on $\mathcal{H}$. Then, a graphenylene system is formed by the vertices and edges of $\mathcal{H}$ lying on Z and in its interior. Fig. 5.9 illustrates the construction method of a graphenylene system. It was proved in (Pan et al., 2018) that all connected subgraphs of (4.6.12)-tiling lattice are partial cubes. Thus, each graphenylene system is a partial cube.

Let G be a graphenylene system and $\{E_1, ..., E_q\}$ be its θ-classes. From the construction method of a graphenylene system, we can see that G contains a dodecagon C_{12}, which holds six different directions of edges, see Figure 5.10. Therefore, let $\{\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_6\}$ be a partition coarser than θ-classes, such that each $\mathcal{F}_i$ is the union of all θ-partitions with the same direction. Let $G/\mathcal{F}_i$ be the quotient graph whose vertices are the connected components of $G - \mathcal{F}_i, i \in \{1, 2, ..., 6\}$. We extend the quotient graphs $G/\mathcal{F}_i$ to weighted graphs $(G/\mathcal{F}_i, \omega_i)$ as illustrated in the previous subsection. The following fundamental result is a special case of the Theorem 5.3.1.

Corollary 5.3.1. *Let G be a graphenylene system and $\{\mathcal{F}_i\}_{i=1}^{6}$ be a partition coarser than θ-classes. Then*

$$TW_K(G) = \sum_{i=1}^{6} W(G/\mathcal{F}_i, \omega_i). \tag{5.10}$$

To demonstrate that the generalized Terminal Wiener index of a graphenylene system G can be computed in a linear time, first of all we have to prove that each quotient graph $G/\mathcal{F}_i$ is a tree, for $i \in \{1, 2, ..., 6\}$.

Lemma 5.3.1. *Let G be a graphenylene system and $\{\mathcal{F}_i\}_{i=1}^{6}$ be a partition coarser than θ-classes. Then, each quotient graph $G/\mathcal{F}_i$ is a tree, for $i \in \{1, 2, ..., 6\}$.*

Proof. Let suppose that $G/\mathcal{F}_i$ is not a tree and obviously contains a cycle. This implies that G must contains an interior face different than a square C_4, a hexagon C_6 and a

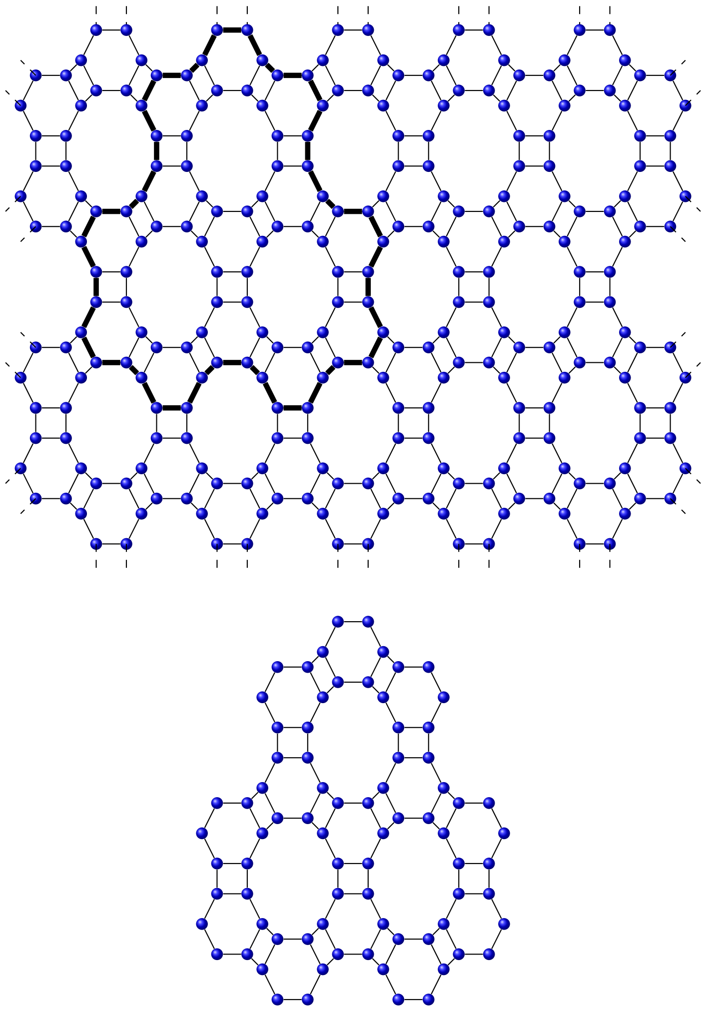


Figure 5.9: (Top) The (4.6.12)-tiling lattice $\mathcal{H}$ and the circuit Z. (Bottom) A graphenylene system G.

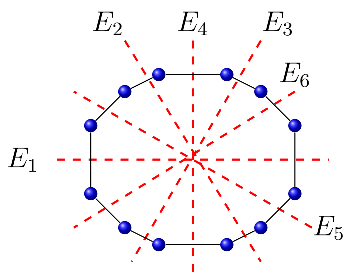


Figure 5.10: The six different directions of the elementary cuts E_i, $i \in \{1, 2, ..., 6\}$.

dodecagon C_{12}. From this contradiction, there is no cycle in $G/\mathcal{F}_i$ and therefore, $G/\mathcal{F}_i$ is a tree. □

Now, we are ready to describe the algorithm that computes the generalized Terminal Wiener index of graphenylene systems and some other systems that full under partial cubes. Let G be a graphenylene system of order N and with N_K vertices of degree K. At first, we compute the quotient trees $G/\mathcal{F}_i$ using a procedure called **QuotientTrees**. Then, for each $G/\mathcal{F}_i$ we compute the vertex weights ω_i using the procedure **CalculateWeights**. The weighted Wiener index of each quotient tree $G/\mathcal{F}_i$ is computed into a variable W_i, and finally the Corollary 5.3.1 is applied to get the result. Therefore, the algorithm reads as follows:

Algorithm 2: Generalized Terminal Wiener index of graphenylene systems

Input: A graphenylene system G of order N
Output: $TW_K(G)$
$(G/\mathcal{F}_1, G/\mathcal{F}_2, ..., G/\mathcal{F}_6) \leftarrow$ **QuotientTrees(**G**)**
for $i = 1$ *to* 6 **do**
  $\omega_i \leftarrow$ **CalculateWeights(**$G/\mathcal{F}_i, G$**)**
for $i = 1$ *to* 6 **do**
  $W_i \leftarrow W(G/\mathcal{F}_i, \omega_i)$
$TW_K(G) = \sum_{i=1}^{6} W_i$

The correctness of the algorithm follows from the Corollary 5.3.1. We now check the time complexity of the algorithm and prove that this index can be computed in linear time.

We recall from (Chepoi, 1996) that the weighted quotient trees $(G/\mathcal{F}_i, \omega_i)$ can be obtained in linear time. In other words, the two procedures **QuotientTrees** and **CalculateWeights** can be done in total $O(N)$ time. Concerning the weighted Wiener index of a tree, it was mentioned in (Chepoi and Klavžar, 1997) that this index can be efficiently computed by using the following formula:

$$W(T, \omega) = \sum_{e \in E(T)} n_1(e) n_2(e), \tag{5.11}$$

where $n_i(e) = \sum_{u \in T-e} \omega(u)$, for $i = 1, 2$.

The Equation 5.11 can be implemented using a method parallel to the linear algorithm from (Mohar and Pisanski, 1988) and (Fahd and Jamil, 2018). We have already described this linear algorithm in the first part of the Algorithm 1, which is illustrated in subsection 5.2.2. Thus, through a simple modification of this linear algorithm and particularly by taking into consideration the vertex weights of quotient trees we can calculate the variable W_i, for $i = 1, 2, ..., 6$, in linear time. As a result, we have the following theorem.

Theorem 5.3.2. *If G is a graphenylene system of order N, then the generalized Terminal Wiener index can be computed in $O(N)$ time.*

The computation in linear time is also possible for some other systems that full under partial cubes, such as square systems, hexagonal systems and C_4C_8 systems. For instance, Chepoi and Klavžar (1997) proved that the Wiener index and the Szeged index of hexagonal systems can be computed in linear time and in (Črepnjak and Tratnik, 2017) it was proved that these two indices of C_4C_8 systems can be also computed in linear time. As a consequence, we state the following generalization.

Corollary 5.3.2. *If G is a system with a fixed number of θ-classes that have the same direction, and all the corresponding quotient graphs are trees. Then, the generalized Terminal Wiener index can be computed in linear time.*

5.4 Application Of The Linear Method For Quantifying The Topological Structure Of Graphenylene Networks

Previously, we proved that the proposed linear method computes the generalized Terminal Wiener index in a linear time complexity. Here, our purpose is to show the ease of this technique to analyze complex partial cubes. Therefore, we apply the proposed method to quantify the topological structure of two graphenylene networks, which are the graphenylene chain network GC_n and the graphenylene sheet network GS_n. At first, we describe the construction method of these structures. Then, we extract the exact analytical expression for the generalized Terminal Wiener index of these two graphenylene networks.

5.4.1 Graphenylene Chain Networks

Let HS_n be a hexagonal-square chain of dimension n, which is obtained by alternating C_6 and C_4. The graph HS_n contains $2n+1$ hexagon C_6 and $2n$ square C_4. Thus, the graphenylene chain network GC_n of dimension n is obtained by joining two hexagonal-square chains of dimension n. The construction method of a graphenylene chain network is illustrated in Figure 5.11. The number of vertices of GC_n is equal to $24n+12$.

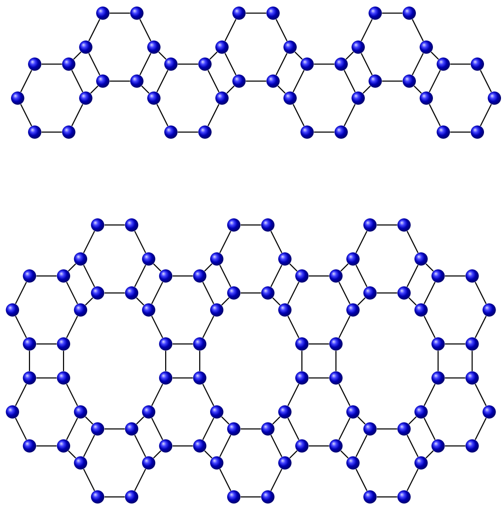


Figure 5.11: The construction method of a graphenylene chain network of dimension 3. (Top) A hexagonal-square chain HS_3. (Bottom) A graphenylene chain network GC_3.

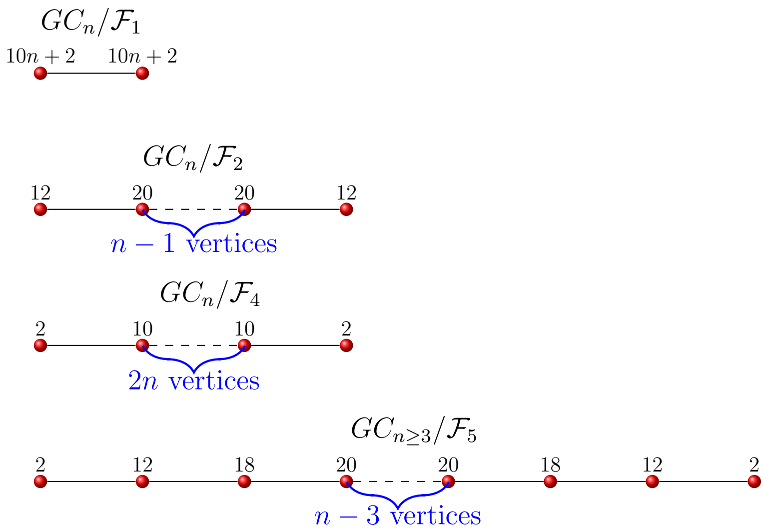
Now, we apply the linear method to obtain exact formula for the generalized Terminal Wiener index of the graphenylene chain network GC_n. We consider the case of K equals to the maximum degree Δ. From Figure 5.11, we can observe that $\Delta = 3$ and the number of vertices of degree 3 is equal to: $N_\Delta = 20n + 4$.

Theorem 5.4.1. *Let GC_n be a graphenylene chain network of dimention $n \geq 2$. Then,*

$$TW_\Delta(GC_n) = 400n^3 + 340n^2 + 524n - 152. \tag{5.12}$$

Proof. The graphenylene chain network GC_n is a partial cube and holds six different directions of edges, as shown in Figure 5.10. First, for $i = 1, 2, ..., 6$, we determine the components $GC_n - \mathcal{F}_i$, where $\mathcal{F}_i$ is the union of the elementary cuts with the same direction i. Figure 5.12 illustrates an example for $n = 3$ (the red vertices represent the vertices with degree $K = 3$ in GC_n).

Then, we get the following weighted quotient trees $(GC_n/\mathcal{F}_i, \omega_i)$, for $i = 1, 2, ..., 6$ and $n \geq 2$.



Due to the symmetry in GC_n network, the weighted quotient tree $GC_n/\mathcal{F}_3$ is isomorphic to $GC_n/\mathcal{F}_2$ and $GS_n/\mathcal{F}_6$ is isomorphic to $GS_n/\mathcal{F}_5$.

Next, we calculate the weighted Wiener index of quotient trees $GC_n/\mathcal{F}_i$, for $i = 1, 2, ..., 6$, using the Equation 5.11.

$$W(GC_n/\mathcal{F}_1, \omega_1) = 100n^2 + 40n + 4.$$

$$\begin{aligned} W(GC_n/\mathcal{F}_2, \omega_2) &= \sum_{i=0}^{n-1}(12 + 20i) * \Big[N_\Delta - (12 + 20i)\Big] \\ &= \frac{200}{3}n^3 + 40n^2 + \frac{112}{3}n. \\ W(GC_n/\mathcal{F}_4, \omega_4) &= \sum_{i=0}^{2n}(2 + 10i) * \Big[N_\Delta - (2 + 10i)\Big] \\ &= \frac{400}{3}n^3 + 80n^2 + \frac{44}{3}n + 4. \\ W(GC_n/\mathcal{F}_5, \omega_5) &= 2 * \Big[2 * (N_\Delta - 2) + 14 * (N_\Delta - 14)\Big] + \\ &\quad \sum_{i=0}^{n-3}(32 + 20i) * \Big[N_\Delta - (32 + 20i)\Big] \end{aligned}$$

$$= \frac{200}{3}n^3 + 40n^2 + \frac{592}{3}n - 80.$$

Clearly, we have $W(GC_n/\mathcal{F}_2, \omega_2) = W(GC_n/\mathcal{F}_3, \omega_3)$ and $W(GC_n/\mathcal{F}_5, \omega_5) = W(GC_n/\mathcal{F}_6, \omega_6)$. Therefore, by applying the Corollary 5.3.1, we get the result. $\square$

$GC_3 - \mathcal{F}_1$

$GC_3 - \mathcal{F}_2$

$GC_3 - \mathcal{F}_4$

$GC_3 - \mathcal{F}_5$

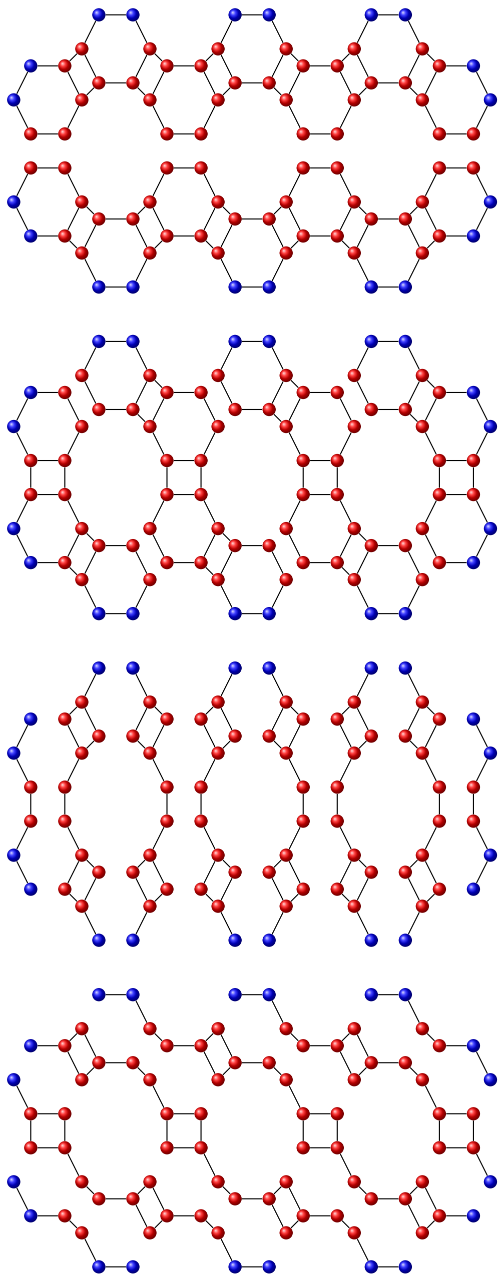


Figure 5.12: The components of $GC_3 - \mathcal{F}_i$, for $i = 1, 2, 4, 5$.

We show in Figure 5.13 the behavior of the generalized Terminal Wiener index of the Graphenylene chain network GC_n. Along the horizontal axis, we have the dimension n. In the vertical axis the values of the generalized Terminal Wiener index of GC_n are determined. From this graphical representation, we can see that the value of the generalized Terminal Wiener index increases with the growth of the dimension n and approximates to $400n^3$ if the dimension n gets large enough.

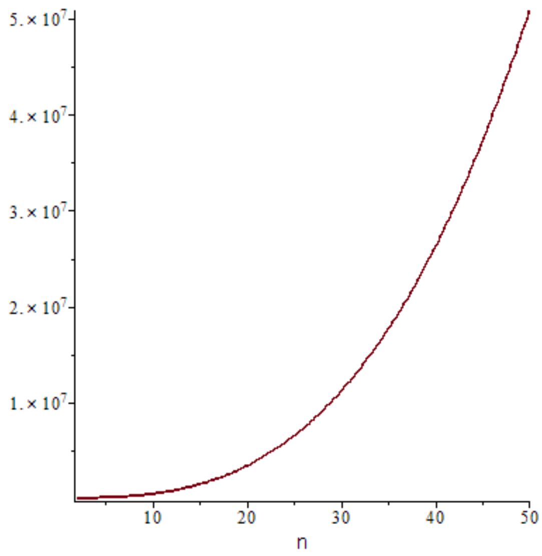


Figure 5.13: The graphical behavior of the generalized Terminal Wiener index of the graphenylene chain network GC_n.

5.4.2 Graphenylene Sheet Networks

Let GS_n be a graphenylene sheet network of dimension n, where n denotes the number of dodecagon C_{12} in each side of this network. The GS_n can be obtained by joining n graphenylene chain network of dimension n. An example of a graphenylene sheet network is shown in Figure 5.14. The number of vertices of GS_n is $12n^2 + 24n$.

Now, we calculate the generalized Terminal Wiener index of the graphenylene sheet network GS_n in the case of $K = \Delta$. From Figure 5.14, we can see that the number of vertices of degree Δ is equal to: $N_\Delta = 12n^2 + 16n - 4$.

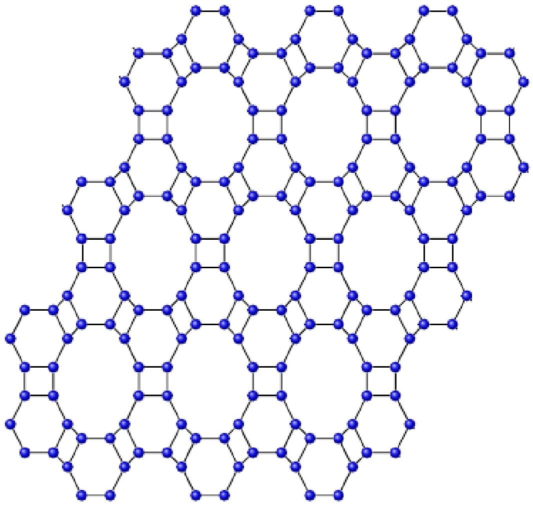
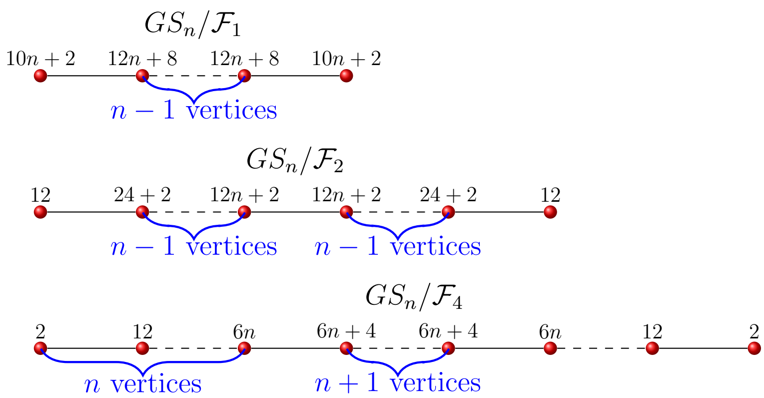


Figure 5.14: A graphenylene sheet network GS_3 of dimension $n = 3$.

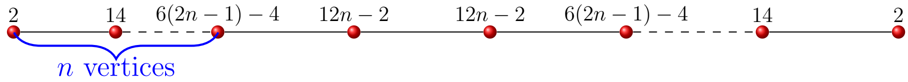
Theorem 5.4.2. *Let GS_n be a graphenylene sheet network of dimension $n \geq 2$. Then,*

$$TW_{\Delta}(GS_n) = 222n^5 + 740n^4 + \frac{1102}{3}n^3 - 252n^2 + \frac{152}{3}n. \tag{5.13}$$

Proof. The graphenylene sheet network GS_n is a partial cube and holds six different directions of edges, as illustrated in Figure 5.10. The calculation process of the generalized Terminal Wiener index TW_{Δ} of GS_n is similar to that for GC_n. At first, we determine the components $GS_n - \mathcal{F}_i$, where $\mathcal{F}_i$ is the union of the elementary cuts with the same direction i. Then, we construct the weighted quotient trees $(GS_n/\mathcal{F}_i, \omega_i)$, for $i = 1, 2, ..., 6$ and $n \geq 2$, which are defined as follows:



$GS_n/\mathcal{F}_6$



Due to the symmetry of the GS_n network, the weighted quotient tree $GS_n/\mathcal{F}_3$ is isomorphic to $GS_n/\mathcal{F}_1$ and $GS_n/\mathcal{F}_5$ is isomorphic to $GS_n/\mathcal{F}_4$.
Now, we calculate the Weighted Wiener index of quotient trees $GS_n/\mathcal{F}_i$, for $i = 1, 2, ..., 6$, using the Equation 5.11.

$$
\begin{aligned}
W(GS_n/\mathcal{F}_1, \omega_1) &= \sum_{i=0}^{n-1} \Big[(10n+2) + (12n+8)i\Big]\Big[N_\Delta - \big[(10n+2) + (12n+8)i\big]\Big].\\
W(GS_n/\mathcal{F}_2, \omega_2) &= 2\Bigg[\sum_{i=2}^{n-1}\Big[12 + \sum_{j=2}^{i}(12j+2)\Big]\Big[N_\Delta - [12 + \sum_{j=2}^{i}(12j+2)]\Big]\Bigg] + \\
&\left(\frac{N_\Delta}{2}\right)^2 + 24\Big[N_\Delta - 12\Big].\\
W(GS_n/\mathcal{F}_4, \omega_4) &= 4[N_\Delta - 2] + 2\Bigg[\sum_{i=2}^{n}\Big[2 + \sum_{j=2}^{i} 6i\Big]\Big[N_\Delta - [2 + \sum_{j=2}^{i} 6i]\Big]\Bigg] + \\
&\sum_{i=1}^{n}\Big[2 + \sum_{i=2}^{n} 6i + (6n+4)i\Big].
\end{aligned}
$$

$$
\begin{aligned}
W(GS_n/\mathcal{F}_6, \omega_6) &= 2\Bigg[\sum_{i=0}^{n-1}\Big[\sum_{j=0}^{i}(6(2i+1)-4)\Big]\Big[N_\Delta - \Big[\sum_{j=0}^{i}(6(2i+1)-4)\Big]\Big]\Bigg]\\
&+ \left(\frac{N_\Delta}{2}\right)^2.
\end{aligned}
$$

Obviously, we have $W(GS_n/\mathcal{F}_1, \omega_1) = W(GS_n/\mathcal{F}_3, \omega_3)$ and $W(GS_n/\mathcal{F}_4, \omega_4) = W(GS_n/\mathcal{F}_5, \omega_5)$. Finally, we apply the Corollary 5.3.1 to get the result. □

We show in Figure 5.15 the behavior of the generalized Terminal Wiener index of the Graphenylene sheet network GS_n. Along the horizontal axis, we have the dimension n and in the vertical axis the values of the generalized Terminal Wiener index of GS_n are determined. From this graphical representation, we observe that the value of the generalized Terminal Wiener index shows a dominant change with the increase of the dimension n and approximates to $222n^5$ if the dimension n gets large enough.

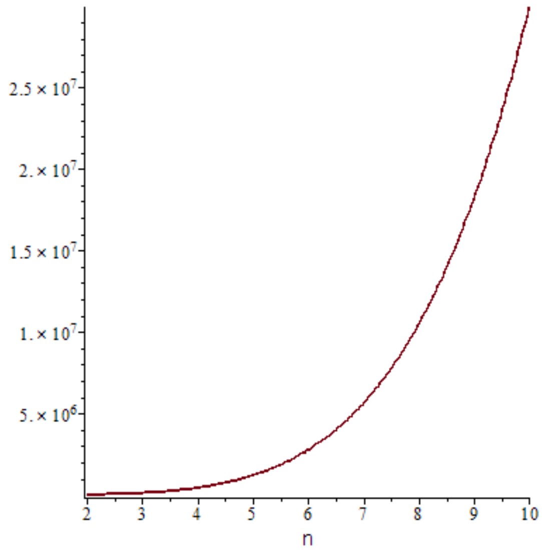


Figure 5.15: The graphical behavior of the generalized Terminal Wiener index of the graphenylene sheet network GS_n.

5.5 Summary

In this chapter, we reduced the computation of the generalized Terminal Wiener index for general graphs and partial cubes. For the case of general graphs, we started by introducing the concept of the canonical metric representation of a graph G, such that we explained the construction process of quotient graphs. Then, we used this concept to reduce the computation of the generalized Terminal wiener index by establishing the relation between the generalized Terminal wiener index of a graph G and the Wiener indices of the appropriately quotient graphs. After that, we applied the proposed method to calculate the generalized Terminal Wiener index of the Dendrimer graph as well as to prove its efficiency for a hand manipulation. Moreover, we developed an algorithm to compute the generalized Terminal wiener index of this structure and we showed its performance by proceeding a comparison between the proposed algorithm and two algorithms from the literature, which are the BFS algorithm and the FW algorithm. For the second case of partial cubes, we demonstrated that the problem of computing the generalized Terminal Wiener index can be reduced to the problem of computing the weighted Wiener indices of quotient graphs with respect to a combination of partitions. Then, we investigated the graphenylene system to show that the discussed method for partial cubes is useful to develop a linear algorithm. Also, we proved the correctness of this algorithm for some other systems that full under partial cubes, such as the hexagonal systems and the C_4C_8 systems. In the end of this chapter, we showed the ease of the discussed method to quantify the topological structure of graphenylene chain network GC_n and graphenylene sheet network GS_n.

CONCLUSIONS AND PERSPECTIVES

In this concluding chapter, we summarize the main results of our research performed on the computation of the Terminal Wiener index and the generalized Terminal Wiener index. Furthermore, we propose and discuss horizons for future research.

During this thesis, we investigated two distance-based topological indices that are the Terminal Wiener index and the generalized Terminal Wiener index. In one hand, we concentrated on the extremal values that the Terminal Wiener index assumes in some classes of graphs. On the other hand, we focused on developing some new methods that ameliorate the calculation of the Terminal Wiener index and its generalization, as well as to reduce the computational complexity of these indices in the case of some large and complex networks.

We started this document with an overview of some basic concepts of graph theory. In particularly, we presented the concept of distance, some of the well-known families of graphs, the widely used matrices to represent a graph and some graph operations to generate new graphs from initial ones. Then, we moved to **chapter 2**, which is considered as the starting point of our research. In this chapter, we introduced a new branch of graph theory and network science called the Quantitative Graph Theory. Also, we defined the concept of topological indices, their existing classes and their applications in different fields. Then, we focused on distance-based topological indices, such as the Wiener index, the Terminal Wiener index and the generalized Terminal Wiener index. Our aim was to provide a general overview about the main mathematical properties of these indices.

Thus, we reported some extremal values that these topological indices assume in various classes of graphs and especially in the case of trees. Furthermore, we showed that the computation of these indices in the case of trees is much easier than that an arbitrary network. Our main result in this chapter was our first contribution concerning the structure of trees that have the second maximal Terminal Wiener index and a transformation that increases the value of this index.

Since the emergence of distance-based topological indices, all the research have shown that the computation of these indices using the basic definition is a difficult task and especially when dealing with networks having a large number of vertices. In this context, we proposed three methods in **chapter 3** in order to overcome the problem of calculating the Terminal Wiener index. The first approach was a decomposition method dedicated to some networks constructed from several copies of components, such as Star-Trees and Path-Trees. The second method was based on an important concept called Thorny graph and the last method was a rewrite of the basic definition of the Terminal Wiener index. We discussed each method separately and we used the obtained results to analyze some important structures having a large number of vertices. In the case of some large and complex structures, the application of the three proposed methods is dominated by computing the distances between the corresponding vertices. Based on this limitation, we investigated in **chapter 4** a new powerful approach called the cut method, which allows the computation of distance-based topological indices without using the concept of distance. There exist two versions of the cut method: a method based on edge-partitions and a method based on vertex-partitions. As a result, we arrived to ameliorate each version of the cut method in order to calculate efficiently the Terminal Wiener index and its generalization. Moreover, we proved the effectiveness of our findings by quantifying the topological structure of some large and complex networks, such as Equilateral Triangular Tetra Sheet network, Hex Derived network and Silicate networks. An open problem was thrown in the literature concerning fast algorithms that would calculate distance-based topological indices without using the distance matrix. According to this problematic, we investigated in **chapter 5** the computational complexity of the Terminal Wiener index and its generalization using another extension of the edge-cut method that is based on an important concept called the canonical metric representation. At first, we used this concept to reduce the computation complexity of the generalized Terminal Wiener index of general graphs. Also, we applied the obtained main result on an hierarchical network, called the Dendrimer graph, to demonstrate the efficiency of this method, as well as to develop a fast algorithm for this kind of graphs. As a last contribution, we used the concept of the canonical metric representation to develop an algorithm that computes the Terminal Wiener index and its generalization of some complex systems in linear time complexity. We note that these systems full under a special class of networks called partial cubes. Finally, we showed the ease of the discussed method to quantify the topological structure of some complicated partial cubes, such as the Graphenylene chain network and the Graphenylene sheet network.

Regarding the further studies, we would like to investigate some newly distance-based topological indices and reduce their computational complexity using the proposed techniques. Also, we hope to generalize our finding to the case of weighted and oriented networks. As a result, we will develop some fast algorithms to compute the topological indices of such networks. In some papers, it was proved that the Wiener index and some other distance-based topological indices of some systems can be obtained in sub-linear time with respect to the number of vertices. In other words, they can be computed depending on the length of the boundary cycle of the corresponding system. Thus, in this context, we hope to extend this result to the case of the generalized Terminal Wiener index. From a practical point of view, we would like to apply the proposed techniques to quantify the topological structure of some interesting complex networks with small world or scale-free properties. In general, we would like to have the opportunity to ameliorate the proposed techniques and apply them to analyze some real situations.

Bibliography

Abramov, O. and Lokot, T. (2011). Typology by means of language networks: Applying information theoretic measures to morphological derivation networks. In *Towards an Information Theory of Complex Networks*, pages 321–346. Springer.

Al Hagri, G., El Marraki, M., and Essalih, M. (2011). The wiener index of some particular graph. *International Journal of Computer Science and Network Security (IJCSNS)*, 11(5):64–69.

Alizadeh, Y., Andova, V., Klavzar, S., and Škrekovski, R. (2014). Wiener dimension: Fundamental properties and (5, 0)-nanotubical fullerenes. *MATCH Commun. Math. Comput. Chem*, 72:279–294.

Alsuwaiyel, M. H. (2010). On the average distance of the hypercube tree. *International Journal of Computer Mathematics*, 87(6):1208–1216.

Appel, K. and Haken, W. (1977). The solution of the four-color-map problem. *Scientific American*, 237(4):108–121.

Arockiaraj, M., Kavitha, S. R. J., and Balasubramanian, K. (2016). Vertex cut method for degree and distance-based topological indices and its applications to silicate networks. *Journal of Mathematical Chemistry*, 54(8):1728–1747.

Ashrafi, A. R. and Diudea, M. V. (2016). *Distance, symmetry, and topology in carbon nanomaterials*, volume 9. Springer.

Assouad, P. and Deza, M. (1980). Espaces métriques plongeables dans un hypercube: aspects combinatoires. In *Annals of Discrete Mathematics*, volume 8, pages 197–210. Elsevier.

Balaban, A. T. and Vollhardt, K. P. C. (2011). Heliphenes and related structures. *The Open Organic Chemistry Journal*, 5(1).

Balakrishnan, R. and Ranganathan, K. (2012). *A textbook of graph theory*. Springer Science & Business Media.

Balakrishnan, R., Sridharan, N., and Iyer, K. V. (2008). Wiener index of graphs with more than one cut-vertex. *Applied Mathematics Letters*, 21(9):922–927.

Benjamin, A., Chartrand, G., and Zhang, P. (2015). *The fascinating world of graph theory*. Princeton University Press.

Biggs, N., Lloyd, E. K., and Wilson, R. J. (1976). *Graph Theory, 1736-1936*. Oxford University Press.

Bonchev, D. (2002). The wiener number–some applications and new developments. In *Topology in Chemistry*, pages 58–88. Elsevier.

Bonchev, D. and Klein, D. J. (2002). On the wiener number of thorn trees, stars, rings, and rods. *Croatica chemica ACTA*, 75(2):613–620.

Brückler, F. M., Došlić, T., Graovac, A., and Gutman, I. (2011). On a class of distance-based molecular structure descriptors. *Chemical physics letters*, 503(4-6):336–338.

Bryant, D. and Moulton, V. (2004). Neighbor-net: an agglomerative method for the construction of phylogenetic networks. *Molecular biology and evolution*, 21(2):255–265.

Cayley, A. (1857). On the theory of the analytical forms called trees. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 13(85):172–176.

Cayley, A. (1874). On the mathematical theory of isomers. *Philosophical Magazine, Series 5*, 47:444–446.

Chen, D.-f., Wei, F.-y., Zhu, H.-y., Wu, Y., and Nong, S.-z. (2016). Extremal trees of terminal wiener index. In *Fuzzy Systems & Operations Research and Management*, pages 351–360. Springer.

Chen, Y.-H. and Zhang, X.-D. (2013). The wiener and terminal wiener indices of trees. *arXiv preprint arXiv:1305.6196*.

Chen, Y.-H. and Zhang, X.-D. (2015). The terminal wiener index of trees with diameter or maximum degree. *arXiv preprint arXiv:1510.03126*.

Chepoi, V. (1996). On distances in benzenoid systems. *Journal of chemical information and computer sciences*, 36(6):1169–1172.

Chepoi, V., Deza, M., and Grishukhin, V. (1997). Clin d'oeil on l1-embeddable planar graphs. *Discrete Applied Mathematics*, 80(1):3–19.

Chepoi, V. and Klavžar, S. (1997). The wiener index and the szeged index of benzenoid systems in linear time. *Journal of chemical information and computer sciences*, 37(4):752–755.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to algorithms*. MIT press.

Črepnjak, M. and Tratnik, N. (2017). The szeged index and the wiener index of partial cubes with applications to chemical graphs. *Applied Mathematics and Computation*, 309:324–333.

Dehmer, M., Kraus, V., Emmert-Streib, F., and Pickl, S. (2014). *Quantitative graph theory*. CRC press.

Dehmer, M. and Mowshowitz, A. (2011). A history of graph entropy measures. *Information Sciences*, 181(1):57–78.

Dehmer, M., Varmuza, K., Bonchev, D., and Emmert-Streib, F. (2012). *Statistical modelling of molecular descriptors in QSAR/QSPR*. Wiley-VCH Weinheim, Germany.

Devillers, J. and Balaban, A. T. (2000). *Topological indices and related descriptors in QSAR and QSPR*. CRC Press.

Deza, M. and Tuma, J. (1996). A note on l1-rigid planar graphs. *European Journal of Combinatorics*, 17(2-3):157–160.

Diudea, M. V., Ivanciuc, O., Nikolic, S., and Trinajstic, N. (1997). Matrices of reciprocal distance, polynomials and derived numbers. *MATCH Commun. Math. Comput. Chem*, 35:41–64.

Djoković, D. Ž. (1973). Distance-preserving subgraphs of hypercubes. *Journal of Combinatorial Theory, Series B*, 14(3):263–267.

Dobrynin, A. A., Gutman, I., Klavžar, S., and Žigert, P. (2002). Wiener index of hexagonal systems. *Acta Applicandae Mathematica*, 72(3):247–294.

Dobrynin, A. A. and Kochetova, A. A. (1994). Degree distance of a graph: A degree analog of the wiener index. *Journal of Chemical Information and Computer Sciences*, 34(5):1082–1086.

El Marraki, M. and Al Hagri, G. (2010). Calculation of the wiener index for some particular trees. *Journal of Theoretical and Applied Information Technology (JATIT)*, 22(2):77–83.

Entringer, R. C., Jackson, D. E., and Snyder, D. (1976). Distance in graphs. *Czechoslovak Mathematical Journal*, 26(2):283–296.

Essalih, M. (2013). L'étude des indices topologiques, leurs applications en qsar/qspr et leurs corrélations aux représentations moléculaires «plerograph» et «kenograph».

Essalih, M., El Marraki, M., and El Hagri, G. (2011). Calculation of some topological indices of graphs. *Journal of Theoretical & Applied Information Technology*, 30(2).

Estrada, E. (2002). Characterization of the folding degree of proteins. *Bioinformatics*, 18(5):697–704.

Estrada, E. (2012). *The structure of complex networks: theory and applications*. Oxford University Press.

Euler, L. (1736). Solutio problematis ad geometriam situs pertinensis. *Comm. Acad. Sci. Imper. Petropol.*, 8:128–140.

Fahd, K. M. and Jamil, M. K. (2018). Algorithms for computing wiener indices of acyclic and unicyclic graphs. *arXiv preprint arXiv:1801.05146*.

Fath-Tabar, G., Furtula, B., and Gutman, I. (2010). A new geometric–arithmetic index. *Journal of mathematical chemistry*, 47(1):477.

Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41.

Graham, R. L. and Pollak, H. O. (1971). On the addressing problem for loop switching. *Bell System Technical Journal*, 50(8):2495–2519.

Graham, R. L. and Winkler, P. M. (1985). On isometric embeddings of graphs. *Transactions of the American mathematical Society*, 288(2):527–536.

Gutman, I. (1997). A property of the wiener number and its modifications.

Gutman, I. (1998). Distance of thorny graphs. *Publ. Inst. Math.(Beograd)*, 63(31-36):73–74.

Gutman, I. and Cyvin, S. J. (2012). *Introduction to the theory of benzenoid hydrocarbons*. Springer Science & Business Media.

Gutman, I., Dobrynin, A. A., Klavzar, S., and Pavlovic, L. (2004). Wiener-type invariants of trees and their relation. *Bull. Inst. Combin. Appl*, 40(27):23–30.

Gutman, I., Essalih, M., El Marraki, M., and Furtula, B. (2013a). Why plerograms are not used in chemical graph theory? the case of terminal-wiener index. *Chemical Physics Letters*, 568:195–197.

Gutman, I., Furtula, B., and Petrović, M. (2009a). Terminal wiener index. *Journal of mathematical chemistry*, 46(2):522–531.

Gutman, I., Furtula, B., Tošović, J., Essalih, M., and MARRAKI, M. (2013b). On terminal wiener indices of kenograms and plerograms. *Iranian Journal of Mathematical Chemistry*, 4(1):77–89.

Gutman, I., Li, X., and Zhang, J. (2009b). Graph energy. *Analysis of Complex Networks. From Biology to Linguistics, Wiley–VCH, Weinheim.*

Gutman, I. and Trinajstić, N. (1972). Graph theory and molecular orbitals. total φ-electron energy of alternant hydrocarbons. *Chemical Physics Letters*, 17(4):535–538.

Hammack, R., Imrich, W., and Klavžar, S. (2011). *Handbook of product graphs.* CRC press.

Harary, F. (1959). Status and contrastatus. *Sociometry*, 22(1):23–43.

Harris, J. M., Hirst, J. L., and Mossinghoff, M. J. (2008). *Combinatorics and graph theory*, volume 2. Springer.

Heydari, A. and Gutman, I. (2010). On the terminal wiener index of thorn graphs. *Kragujevac J. Sci*, 32(57-64):47.

Horvat, B., Pisanski, T., and Randic, M. (2008). Terminal polynomials and star-like graphs. *MATCH Commun. Math. Comput. Chem*, 60(2):493–512.

Hosoya, H. (1971). Topological index. a newly proposed quantity characterizing the topological nature of structural isomers of saturated hydrocarbons. *Bulletin of the Chemical Society of Japan*, 44(9):2332–2339.

Ilić, A. and Ilić, M. (2013). Generalizations of wiener polarity index and terminal wiener index. *Graphs and Combinatorics*, 29(5):1403–1416.

Imran, M., Baig, A. Q., and Ali, H. (2016). On molecular topological properties of hex-derived networks. *Journal of Chemometrics*, 30(3):121–129.

Junker, B. H. and Schreiber, F. (2011). *Analysis of biological networks*, volume 2. John Wiley & Sons.

Kelenc, A., Klavzar, S., and Tratnik, N. (2015). The edge-wiener index of benzenoid systems in linear time. *MATCH Commun. Math. Comput. Chem*, 74(3):521–532.

Khalifeh, M., Yousefi-Azari, H., and Ashrafi, A. (2010). Another aspect of graph invariants depending on the path metric and an application in nanoscience. *Computers & Mathematics with Applications*, 60(8):2460–2468.

Kirchhoff, G. (1847). Uber die au ösung der gleichungen, auf welche man bei der untersuchung der linearen verteilung galvanischer ströme gefuhrt wird. *Annalen der Physik*, 72:497–508.

Klajnert, B. and Bryszewska, M. (2001). Dendrimers: properties and applications.

Klavzar, S. (1998). Applications of isometric embeddings to chemical graphs. In *Discrete Mathematical Chemistry*, pages 249–260. Citeseer.

Klavžar, S. (2006). On the canonical metric representation, average distance, and partial hamming graphs. *European Journal of Combinatorics*, 27(1):68–73.

Klavzar, S. (2008). A bird's eye view of the cut method and a survey of its applications in chemical graph theory. *MATCH Commun. Math. Comput. Chem*, 60(2):255–274.

Klavzar, S. and Gutman, I. (1997). Wiener number of vertex-weighted graphs and a chemical application. *Discrete Applied Mathematics*, 80(1):73–81.

Klavzar, S., Gutman, I., and Mohar, B. (1995). Labeling of benzenoid systems which reflects the vertex-distance relations. *Journal of chemical information and computer sciences*, 35(3):590–593.

Klavžar, S., Gutman, I., and Rajapakse, A. (1997). Wiener numbers of pericondensed benzenoid hydrocarbons. *Croatica chemica acta*, 70(4):979–999.

Klavzar, S. and J Nadjafi-Arani, M. (2015). Cut method: update on recent developments and equivalence of independent approaches. *Current Organic Chemistry*, 19(4):348–358.

Klavžar, S. and Nadjafi-Arani, M. J. (2014). Wiener index in weighted graphs via unification of θ-classes. *European Journal of Combinatorics*, 36:71–76.

Klein, D. J., Lukovits, I., and Gutman, I. (1995). On the definition of the hyper-wiener index for cycle-containing structures. *Journal of chemical information and computer sciences*, 35(1):50–52.

Liu, M. and Liu, B. (2013). A survey on recent results of variable wiener index. *MATCH Commun. Math. Comput. Chem*, 69(3):491–520.

Malkoch, M., Malmström Jonsson, E., and Nyström, A. (2012). Dendrimers: properties and applications.

Mansour, T. and Schork, M. (2009). The vertex pi index and szeged index of bridge graphs. *Discrete Applied Mathematics*, 157(7):1600–1606.

Mansour, T. and Schork, M. (2010). Wiener, hyper-wiener, detour and hyper-detour indices of bridge and chain graphs. *Journal of mathematical chemistry*, 47(1):72.

Manuel, P. D. and Rajasingh, I. (2011). Minimum metric dimension of silicate networks. *Ars Comb.*, 98:501–510.

Mohar, B. and Pisanski, T. (1988). How to compute the wiener index of a graph. *Journal of Mathematical Chemistry*, 2(3):267–277.

Nantasenamat, C., Isarankura-Na-Ayudhya, C., Naenna, T., and Prachayasittikul, V. (2009). A practical overview of quantitative structure-activity relationship.

Pan, Y.-j., Xie, M.-f., and Zhang, F.-j. (2018). Partial cubes and archimedean tilings. *Acta Mathematicae Applicatae Sinica, English Series*, 34(4):782–791.

Pavlović, L. and Gutman, I. (1997). Wiener numbers of phenylenes: an exact result. *Journal of chemical information and computer sciences*, 37(2):355–358.

Plavšić, D., Nikolić, S., Trinajstić, N., and Mihalić, Z. (1993). On the harary index for the characterization of chemical graphs. *Journal of Mathematical Chemistry*, 12(1):235–250.

Quadras, J., Balasubramanian, K., and Christy, K. A. (2016). Analytical expressions for wiener indices of n-circumscribed peri-condensed benzenoid graphs. *Journal of Mathematical Chemistry*, 54(3):823–843.

Rahman, M. S. (2017). *Basic Graph Theory*. Springer.

Raj, F. S. and George, A. (2015). On the metric dimension of few network sheets. In *Robotics, Automation, Control and Embedded Systems (RACE), 2015 International Conference on*, pages 1–6. IEEE.

Randic, M. (1975). Characterization of molecular branching. *Journal of the American Chemical Society*, 97(23):6609–6615.

Randić, M. (1993). Novel molecular descriptor for structure property studies. *Chemical Physics Letters*, 211(4-5):478–483.

Randić, M., Zupan, J., and Vikić-Topić, D. (2007). On representation of proteins by star-like graphs. *Journal of Molecular Graphics and Modelling*, 26(1):290–305.

Schmuck, N. S. (2010). *The Wiener index of a graph*. na.

Škrekovski, R. and Gutman, I. (2014). Vertex version of the wiener theorem. *MATCH Commun. Math. Comput. Chem*, 72(1):295–300.

Smolenskii, E., Shuvalova, E., Maslova, L., Chuvaeva, I., and Molchanova, M. (2009). Reduced matrix of topological distances with a minimum number of independent parameters: distance vectors and molecular codes. *Journal of mathematical chemistry*, 45(4):1004–1020.

Šoltés, L. (1991). Transmission in graphs: a bound and vertex removing. *Mathematica Slovaca*, 41(1):11–16.

Székely, L. A., Wang, H., and Wu, T. (2011). The sum of the distances between the leaves of a tree and the 'semi-regular' property. *Discrete mathematics*, 311(13):1197–1203.

Todeschini, R. and Consonni, V. (2008). *Handbook of molecular descriptors*, volume 11. John Wiley & Sons.

Tratnik, N. (2016). The edge-szeged index and the pi index of benzenoid systems in linear time. *arXiv preprint arXiv:1606.07236*.

Von Collatz, L. and Sinogowitz, U. (1957). Spektren endlicher grafen. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 21, pages 63–77. Springer.

Vukičević, D. and Graovac, A. (2004). On modified wiener indices of thorn graphs. *MATCH Communications in mathematical and in computer chemistry*, 50(50):93–108.

Walikar, H. B., Ramane, H. S., Sindagi, L., Shirakol, S. S., and Gutman, I. (2006). Hosoya polynomial of thorn trees, rods, rings, and stars. *Kragujevac Journal of Science*, 28:47–56.

Wasserman, S. and Faust, K. (1994). *Social network analysis: Methods and applications*, volume 8. Cambridge university press.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world'networks. *nature*, 393(6684):440.

Welk, M., Dehmer, M., and Emmert-Streib, F. (2014). Discrimination of image textures using graph indices. *Quantitative Graph Theory: Mathematical Foundations and Applications*, pages 355–386.

West, D. B. et al. (2001). *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River.

Wiener, H. (1947). Structural determination of paraffin boiling points. *Journal of the American Chemical Society*, 69(1):17–20.

Winkler, P. M. (1984). Isometric embedding in products of complete graphs. *Discrete Applied Mathematics*, 7(2):221–225.

Xiao, W. and Parhami, B. (2006). Cayley graphs as models of deterministic small-world networks. *Information Processing Letters*, 97(3):115–117.

Xu, K., Das, K. C., and Trinajstić, N. (2015). *The Harary index of a graph*. Springer.

Xu, K., Liu, M., Das, K. C., Gutman, I., and Furtula, B. (2014). A survey on graphs extremal with respect to distance-based topological indices. *MATCH Commun. Math. Comput. Chem*, 71(3):461–508.

Yeh, Y.-N. and Gutman, I. (1994). On the sum of all distances in composite graphs. *Discrete Mathematics*, 135(1-3):359–365.

Zaretskii, K. (1965). Constructing a tree on the basis of a set of distances between the hanging vertices. *Uspekhi Matematicheskikh Nauk*, 20(6):90–92.

Zeryouh, M., El Marraki, M., and Essalih, M. (2015a). Some tools of qsar/qspr and drug development: Wiener and terminal wiener indices. In *2015 International Conference on Cloud Technologies and Applications (CloudTech)*, pages 1–4. IEEE.

Zeryouh, M., El Marraki, M., and Essalih, M. (2015b). Terminal wiener index of star-tree and path-tree. *Applied Mathematical Sciences*, 9(39):1919–1929.

Zeryouh, M., El Marraki, M., and Essalih, M. (2016). On the terminal wiener index of networks. In *Multimedia Computing and Systems (ICMCS), 2016 5th International Conference on*, pages 533–536. IEEE.

Zeryouh, M., El Marraki, M., and Essalih, M. (2017). Studying the structure of some networks using certain topological indices. In *International Workshop on Complex Networks and their Applications*, pages 543–554. Springer.

Zeryouh, M., El Marraki, M., and Essalih, M. (2018a). A measure for quantifying the topological structure of some networks. In *International Conference on Networked Systems*, pages 371–381. Springer.

Zeryouh, M., El Marraki, M., and Essalih, M. (2018b). Quantifying the structure of some networks by using a vertex-cut method. In *International Conference on Modern Intelligent Systems Concepts*.

Zeryouh, M., El Marraki, M., and Essalih, M. (2019). An efficient computational method for a distance-based measure of graphenylene networks. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(3):778–785.

Zhu, H.-Y., Klein, D. J., and Lukovits, I. (1996). Extensions of the wiener number. *Journal of Chemical Information and Computer Sciences*, 36(3):420–428.

Résumé

De nombreux systèmes complexes et phénomènes du monde réel peuvent être modélisés sous forme de réseaux et analysés à l'aide de certaines techniques dérivées de la théorie des graphes. L'une des approches utilisées pour la caractérisation des informations structurelles des réseaux est une sorte de mesures quantitatives appelées les indices topologiques. La classe la plus considérée de ces mesures est celle des indices topologiques basés sur la distance, qui inclut l'une des plus importantes et anciennes mesures appelé l'indice de Wiener. Dans cette thèse, nous allons nous concentrer sur deux extensions récentes de l'indice de Wiener notamment l'indice de Terminal de Wiener et l'indice de Terminal de Wiener généralisé.
L'objectif de ce travail est d'améliorer et de développer de nouvelles méthodes permettant le calcul efficace de l'indice de Terminal de Wiener et sa généralisation. Tout d'abord, nous étudions quelques propriétés fondamentales de ces indices, en particulier les limites maximales de l'indice de Terminal de Wiener pour le cas des arbres. Ensuite, nous proposons trois méthodes pour résoudre le problème de calcul de l'indice de Terminal de Wiener pour certains types de réseaux. Cependant, l'utilisation de ces méthodes dans le cas des réseaux larges et complexes serait une tâche difficile. Pour cette raison, nous étudions une nouvelle approche appelée la méthode de coupe "cut method" et nous utilisons cette technique pour dériver de nouvelles formules efficaces pour le calcul de l'indice de Terminal de Wiener et sa généralisation. De plus, nous nous concentrons sur une extension de la méthode de coupe basée sur un concept appelé la représentation canonique d'un réseau afin de réduire la complexité de calcul de ces indices et de développer un algorithme linéaire pour certains systèmes complexes.

Mots-clefs (5) : Réseaux, Théorie des graphes, Indices topologiques, L'indice de Wiener, L'indice de Terminal de Wiener, L'indice de Terminal de Wiener généralisé, Méthode de coupe, Complexité, Représentation canonique.

Abstract

Many complex systems and real world phenomena can be modeled as networks and analyzed using techniques derived from graph theory. One of the used approaches to characterize the structural information of networks is a kind of quantitative measures called topological indices. The most considerable class of these measures is distance-based topological indices, which includes the oldest and the well-known measure called the Wiener index. In this thesis, we focus on two recent extensions of the Wiener index that are the Terminal Wiener index and the generalized Terminal Wiener index.
The ultimate objective of this work is to improve and develop some new methods that would compute efficiently the Terminal Wiener index and its generalization. First of all, we study some fundamental properties of these indices, and especially the maximal bounds that the Terminal Wiener index assumes for the class of trees. After that, we propose three methods to overcome the problem of calculating the Terminal Wiener index of some kinds of networks. However, the use of these methods in the case of some large and complex networks would be a difficult task. For this reason, we study a new powerful approach called the cut method and we use this technique to derive some new efficient formulas for computing the Terminal Wiener index and its generalization. Furthermore, we focus on an extension of the cut method that is based on an important concept called the canonical metric representation in order to reduce the computational complexity of these indices, as well as to develop a linear algorithm for some complex systems.

Key Words (5): Networks, Graph Theory, Topological indices, Wiener index, Terminal Wiener index, Generalized Terminal Wiener index, Cut method, Complexity, Canonical metric representation.