Resonant Sets in Benzenoid Systems

by

Xi Chen

A Dissertation Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy in Computational Sciences

Middle Tennessee State University

December 2019

Dissertation Committee:

Dr. Dong Ye, Chair

Dr. Xiaoya Zha

Dr. Chris Stephens

Dr. Jing Kong

Dr. John Wallin

To my husband Jingsai Liang and my son William.

ACKNOWLEDGMENTS

I have received lots of help and support during the writing of this dissertation. First, I would like to express my great thanks to my advisor, Dr. Dong Ye. This could not be possible without his support and guidance. He led me into the magical world of graph theory and showed me how to be a good researcher.

I also wish to thank the director of computational science program, Dr. John Wallin. This work could be done without his finical support. I would also like to thank Dr.Xiaoya Zha for his guidance to this work. I am also grateful to my committee members, Dr. Chris Stephens, Dr. Jing Kong, for their valuable suggestions and opinions to this dissertation.

Most importantly, I would like to express my thanks to my family. I would like to thank my husband, Jingsai, for his encouragement and support. I would like to thank my mom and my dad for their trust in me. I would like to thank my son, William. This dissertation would be impossible without their endless love.

ABSTRACT

A benzenoid system H is a finite 2-connected plane bipartite graph in which every interior face is bounded by a regular hexagon. A benzenoid system is called as cata-condensed if it is outer planar. A perfect matching is a set of independent edges which cover every vertex exactly once. A set of disjoint hexagons S of a benzenoid system H is a resonant set if the subgraph obtained from H by deleting all vertices of hexagons in S has a perfect matching. The resonant set is forcing if the subgraph has a unique perfect matching. In chapter 2, we define a forcing resonance polynomial of H as $f(x) = \sum_{i=1}^{cl(H)} a_i x^i$ where a_i is the number of distinct forcing resonant set of size i and $cl(H)$ is the Clar number of H. We put all coefficients of this polynomial in a vector called as coefficient vector. We design a recursive algorithm to compute the forcing resonance polynomial of cata-condensed benzenoid systems with n hexagons. The forcing resonance polynomial of H can be used to enumerate the number of forcing resonant sets and its coefficient vector can be applied to predict the stability of benzenoid system more accurately than Clar number and Kekulé count, which are all traditional stability indicators of molecules. The coefficient vector is also better than HOMO-LUMO gap in terms of describing the structural characteristics of molecules. In chapter 3, we also design an algorithm to reconstruct the cata-condensed benzenoid systems in a specific case.

Forcing set is concept originated from the research on the application of Kekulé structure in the resonance theory in chemistry. This concept has been generalized to any graph G. For example, let G be a graph with m edges and n vertices. A face of G is forcing face if the subgraph of G obtained by deleting this face and all edges incident to this face has a unique perfect matching. In chapter 4, we give a forcing face detection algorithm based on a well-known unique perfect matching algorithm in $O(m^2\mathrm{log}n^4)$ time. We also give an algorithm to construct graphs with unique perfect matching through odd bridges, inspired by reversely thinking this unique perfect matching algorithm. We present a forcing face construction algorithm based on the proposed unique perfect matching construction algorithm.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In organic chemistry, *aromaticity* is a property to describe a conjugated system, which is usually made by a planar ring with resonance bonds alternating single and double bonds, that increases the stability of the molecules than other structures. The most common aromatic compounds are benzene C_6H_6 and its derivatives. *Polycyclic aromatic hydrocarbons* (PAHs) are actually hydrocarbons composed by multiple such aromatic rings. The simplest PAHs Naphthalene is combined by only two benzene rings which chemical structure $C_{10}H_8$ is depicted in Figure 1.

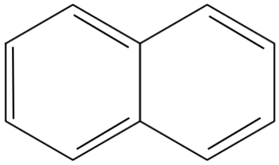


Figure 1: Chemical structure description of Naphthalene.

Chemists use *HOMO-LUMO gap* to predict the stability of chemical molecule [40, 54, 2, 27]. HOMO denotes the highest occupied molecular orbital and LUMO denotes the lowest unoccupied molecular orbital. In other words, the HOMO has the highest energy among orbitals that have electrons and the LUMO has the lowest energy among orbitals that do not have electrons. So, a more closer gap between HOMO and LUMO will more likely excite the electron movement, which makes the molecules not stable.

In general, Chemical molecules with a larger HOMO-LUMO gap has better stability [27, 75]. For example, in paper [19], scientists use HOMO-LUMO gap as the measurement to determine that Si_{22} is the most chemical stable cluster among medium-sized neutral and charged silicon clusters.

The first well-known specifical explanation of the stability for benzene is proposed by Hückel [35, 34, 36] in 1931, that is a planar ring structure would have aromatic property if there are $4n+2$ electrons in a conjugated system of p orbitals in the ring, where n is a non-negative integer. This *Hückel rule* can be easily verified in a *benzene* ring, in which six π-electrons are delocalized over all six carbon atoms.

Later on, *Kekulé structure* is applied in chemical field. The number of Kekulé structure is called as *Kekulé counts* which plays an important role to measure the stability of benzenoid hydrocarbons [17, 16, 14]. Kekulé structure is actually same as the concept of perfect matching of a graph G, that is a set of disjoint edges covering all vertices of G. Researchers have published many papers talking the perfect matching of benzenoid systems [55, 69, 21, 60].

In 1972, E. Clar [15] formulated a famous *Clar's π-sextet rule* to describe the characterizations such as aromaticity of PAHs. In Clar's aromatic sextet theory, Clar proposed that the Kekulé structure [37] with the largest number of disjoint aromatic π-sextets plays a key role to measure the aromaticity of benzenoid hydrocarbons. This simple rule was summarized based on experimental observation [71], but has been validated both theoretically and experimentally in the past 40 years [58]. The quantification of Clary theory and its derivatives have been successfully tested its correctness on a series of numerical data like resonance energies, bond lengths, etc [44].

The Clar theory not only can be applied in PAHs like benzenoid system [43], but also has been applied to various novel conjugated nanostructures material [44] like *graphene*. In 2004, the characterization of graphene was firstly experimental realized [46] and later on became a rapidly rising star in the modern material science and technology [25]. Scientists in [63] showed that Clar's theory of the aromatic sextet is a simple but very powerful tool to predict the stability and other properties of graphene nanoribbons with different type of edges. Many other scientists have been working on this field to explore the Kekulé structure [26], aromaticity [50, 20, 67], and Clar's sextet rule [65] of graphene and graphene nanoribbons. Since the edges of graphene plays a crucial role in determining the chemical, physical, and other properties [81], Clar sextet and aromaticity analysis are already developed in triangular, rectangular, honeycomb [48], square [49], and other irregular-shaped [45] graphene antidot lattices.

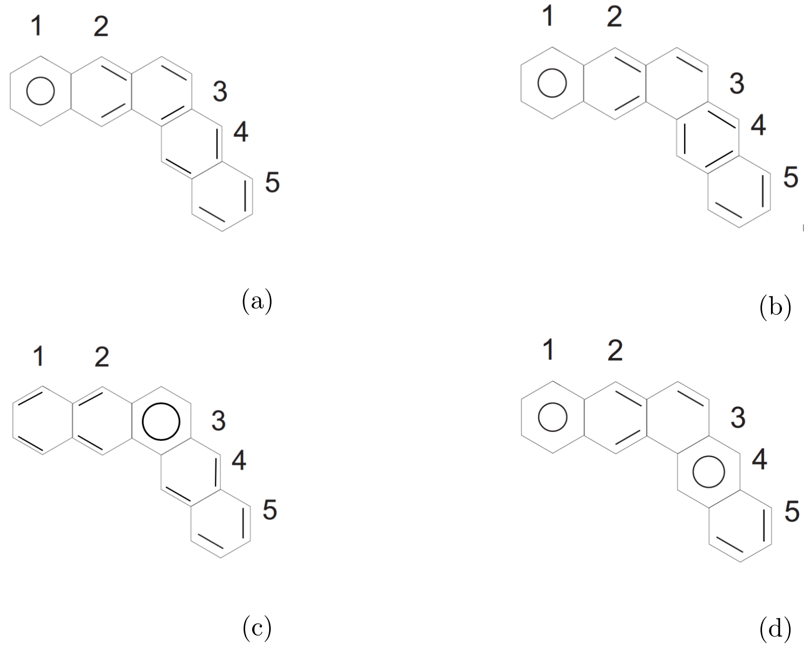
A *hexagonal system* is obtained when all carbon-hydrogen bonds are deleted from a benzenoid system [61]. Every interior face in a hexagonal system is bounded by a regular hexagon. A benzenoid system is *cata-condensed* if all vertices appear on its boundary. Gutman [28] declared three requirements to construct *Clar formula* (*Clar structure*) through drawing circles in some hexagons in a hexagonal system. Three constraints of drawing circles are as follows:

1. Circles must be drew in non-adjacent hexagons.
2. All circles must be arranged in a way such that the remainder of the hexagonal system after deleting all hexagons with circles inside and all edges incident to these hexagons is either empty or has a Kekulé structure.
3. As many circles as possible are drew constrained by (1) and (2).

If we draw circles constrained by all three conditions, the maximum number of circles we can draw in a hexagonal system is *Clar number* of this hexagonal system. For example, Figure 2 shows a cata-condensed benzenoid system with five hexagons. The Clar number is 2 because we at most can draw circles in two hexagons like $\{1, 4\}$ in (d).

If we only draw circles constrained by (1) and (2), we do not need to draw as many circles as possible. The set of these disjoint hexagons with circles inside is called as a *sextet pattern*, *resonant set*, or *generalized Clar formula.*

For example, in Figure 2, $\{1\}$ in (a), $\{3\}$ in (c), and $\{1, 4\}$ in (d) are all resonant sets. The size of the maximum resonant set is the Clar number.



(a)

(b)

(c)

(d)

Figure 2: Resonant set and forcing resonant set

Clar noticed that for isomeric benzenoid hydrocarbons, the one has better stability if it has larger Clar number [15]. And Kekulé counts is not as useful in predicting the stability of fullerene as in benzenoid hydrocarbons, where *fullerenes* are carbon-cage molecules consisting of only carbon atoms which are arranged on a sphere with 12 pentagons faces and other hexagons faces [76]. The most stable fullerene molecule icosahedral C_{60} (Buckminsterfullerene) does not have the highest Kekulé number among all its isomers

[4]. But Clar number still is a great indicator of relative stability of fullerene isomers of C_{60}. Computations show that the icosahedral C_{60} has the largest Kekulé count among isomers achieving the maximum Clar number [77]. The upper bound for the Clar number of fullerene graphs and extremal fullerene graphs with the maximum Clar number have been investigated in paper [76, 66, 24].

Thus, calculating Clar number of benzenoid systems is a very meaningful and helpful topic in analyzing the aromaticity of isomers of benzenoid systems. Klavžar, Žigert and Gutman [38] proposed a method to determine the Clar number of a cata-condensed benzenoid hydrocarbon. This remarkably simple method states that the Clar number is equal to the minimum number of straight lines required to intersect all hexagons. Hansen and Zheng [30] proposed a mixed-integer linear program to compute the Clar number of benzenoid system. Abeleo and Atkinson[1] proved that computing the Clar number of benzenoid system is linear and can be computed in polynomial time. However, calculating the Clar number in general 2-connected plane graphs has been proved as NP-hard [6, 51].

The upper bound of Clar number of cata-condensed benzenoid system with h hexagons is $\lfloor (2h+1)/3 \rfloor$ [29, 5], which is a same conclusion proved in paper [68]. But in the paper [5], authors showed that there exists a cata-condensed benzenoid system with h hexagons which can attain the upper bound $\lfloor (2h+1)/3 \rfloor$. In other words, the maximum Clar number of cata-condensed benzenoid systems with h hexagons is $\lfloor (2h+1)/3 \rfloor$.

Forcing set [10] including *forcing edge* [42, 82], *forcing number* [84] and *forcing hexagon* [9] is another very important theoretical topics over the years. Giving a graph G, a *forcing edge* is that the remaining graph after deleting this edge and all incident edges to this edge from graph G has a unique perfect matching. Analogously, a *forcing face* is that the remaining graph after deleting the face and all incident edges to this face from graph G has a unique perfect matching. This "forcing" concept is originated from the research on the application of Kekulé structure in the resonance theory in chemistry [10], but this concept has been extended on general graphs [18], such as forcing face in plane bipartite graphs [11]. The forcing related calculation on fullerene [57] and cata-condensed benzenoid system [84] have been studied.

Based these theoretical results, many types of polynomials are developed [71] to systematically count the Kekulé count, forcing set, resonant set and its generalized derivatives, such as *sextet polynomial* [33], *Clar covering polynomial* (*Zhang-Zhang polynomial*) [12, 64, 13, 74, 78, 79], *Clar polynomial* [52, 62], and *forcing polynomial* [80].

In order to count sextet pattern (resonant set), Hosoya and Yamaguchi

[33] defined the *sextet polynomial* $R_G(x)$ given a hexagonal system G as follows:

$$R_G(x) = \sum_{i=0}^{\mathrm{cl}(G)} r_i x^i \tag{1}$$

where $\mathrm{cl}(G)$ is the Clar number of G and r_i represents the number of sextet pattern with i hexagons in G. It has been proved [47] that the number of Kekulé structure in G equals to $R_G(1)$ when G is cata-condensed. For example, the sextet polynomial of the hexagonal system in Figure 2 is $4x^2 + 5x + 1$ because it has sextet pattern $\{\emptyset\}$, $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, $\{5\}$, $\{1, 4\}$, $\{1, 5\}$, $\{2, 4\}$, $\{2, 5\}$. The number of perfect matching of this hexagonal system is 10.

As an extension of sextet polynomial, Randić, Shiu, *et al* [52, 62] alleviated the condition (3) in constructing the Clar formula by a new condition (4) that is drawing the circle one be one until no new circle can be drew constrained by (1) and (2). The set of hexagons obtained under conditions (1), (2), and (4) is called as *extended Clar structure*. For example, the number of circle in (c) of Figure 2 is not Clar number but there is no way to draw a second circle in (c) constrained by (1) and (2), thus $\{3\}$ in (c) is an extended Clar structure. The count polynomial of extended Clar structure of a hexagonal system G is called as *Clar polynomial* and is defined as follows:

$$C_G(x) = \sum_{i=0}^{\mathrm{cl}(G)} c_i x^i \tag{2}$$

where c_i represents the number of extended Clar structure with i hexagons in G. For example, the Clar polynomial of the hexagonal system in Figure 2 is $4x^2 + x$ because it has extended Clar structure $\{3\}$, $\{1,4\}$, $\{1,5\}$, $\{2,4\}$, $\{2,5\}$. Clar and sextet polynomial have been calculated on fullerene [56] and buckminsterfullerene [62].

1.1 Forcing Resonant Polynomial of cata-condensed Benzenoid Systems

Basically, there are three different ways to measure the stability of benzenoid systems - Kekulé count, Clar number, and HOMO-LOMO gap. Many different counting polynomials are also formulated based on these concepts. However, using Clar number itself to rank the stability of isomers may encounter a problem because there are so many isomers having the same Clar number. We also cannot reconstruct the most stable molecular structure solely based on HOMO-LUMO gap or Kekulé count.

A resonant set is a *forcing resonant set* if the remainder graph after deleting this resonant set has a unique perfect matching. For example, in Figure 2, only $\{3\}$ in (c) and $\{1,4\}$ in (d) are forcing resonant sets, because $\{1\}$ in

(a) has another perfect matching in (b).

In chapter 2, we define a *forcing resonant set polynomial* of a hexagonal system G that counts the number of forcing resonant sets as follows:

$$P_G(x) = \sum_{i=1}^{\mathrm{cl}(G)} a_i x^i \tag{3}$$

where a_i represents the number of forcing resonant set with i hexagons in G.

We define a new concept - coefficient vector of forcing resonant set polynomial - to not only predict the stability of benzenoid system more accurately, but also has the ability to describe the structural characteristics of molecules. We will show that our proposed forcing resonant polynomial is strongly related to the Clar polynomial when the hexagonal system is cata-condensed.

1.2 Construction of cata-condensed Benzenoid Systems

There are many algorithms to calculate the Clar number of hexagonal systems, but it is not an easy ”reverse engineering” job to reconstruct hexagonal systems given a Clar number.

In chapter 3, we propose an algorithm to enumerate all cata-condensed benzenoid systems given a Clar number in a specific case, based on an enumeration method of benzenoid systems proposed by Brinkmann, Caporossi, and Hansen [7] in 2002 and a character of cata-condensed hexagonal system with maximum Clar number proved by Zhai, Alrowaili, and Ye [68] in 2018.

1.3 Forcing Face and Unique Perfect Matching

In chapter 4, inspired by a well-known unique perfect matching algorithm proposed by Gabow, Kaplan, and Tarjan [22] in 2002, we propose an algorithm to check the existence of forcing face in a graph G for the first time. This work is based on a conclusion about the relationship between unique perfect matching and odd bridges [41] and a dynamical connectivity algorithm - top tree [32, 59, 3].

In chapter 4, by reversely thinking the well-known unique perfect matching algorithm, we also propose another algorithm to construct graphs with unique perfect matching through bridges. Based on this proposed algorithm, we finally present a forcing face construction algorithm.

We will prove the correctness of all proposed algorithms, calculate their time complexity, and finally show some applications.

CHAPTER 2

FORCING RESONANT POLYNOMIAL OF CATA-CONDENSED BENZENOID SYSTEM

2.1 Introduction

A *graph* is a set of vertices and edges. A *bipartite* graph is a graph whose vertices can be partitioned into two disjoint sets and there is no edge connected any pair of vertices in each set. A *planar* graph is a graph that can be embedded in a plane, which means this graph can be drawn without any crossing edges in a plane. A graph is *connected* if there is a path between any pair of vertices. A connected graph is *2-connected* if the graph is still connected after any single vertex is removed from this graph. A *perfect matching* of a graph, which is known as *Kekulé structure* in chemical field, is a set of disjoint and independent edges that covers all vertices of the graph.

A benzenoid system is *cata-condensed* if all vertices appear on its boundary. A *hexagonal system* is obtained when all all carbon-hydrogen bonds are deleted from a benzenoid system. A hexagonal system is also a finite 2-connected planar bipartite graph in which each interior face bounded by a regular hexagon of side length one. A set of disjoint hexagons $\mathcal{H}$ of a benzenoid system G is a *resonant set* if the subgraph G' obtained by deleting all vertices covered by $\mathcal{H}$ from G has a perfect matching. A resonant set is *forcing* if the subgraph obtained by deleting all vertices covered by this resonant set from G has a unique perfect matching. The maximum resonant set is called *Clar formula* of G. The *Clar number* $cl(G)$ of G is the size of Clar formula. The *spectrum* of forcing resonant set (FRS) is defined as:

$$\mathrm{spec}_{\mathrm{FRS}}(G) = \{|\mathcal{H}| : \mathcal{H} \text{ is a forcing resonant set of } G\}$$

This following theorem [83] proved that a maximum resonant set of a hexagonal system is a forcing resonant set.

Theorem 2.1. *Let G be a hexagonal system. The subgraph G' obtained by deleting all vertices of a maximum resonant set from G has a unique perfect matching.*

But not all forcing resonant sets are maximum resonant set. We define a counting polynomial of forcing resonant set on cata-condensed benzenoid systems (*forcing resonant polynomial*) as following:

Definition 2.2. *Let G be a benzenoid system. The forcing resonant polynomial $P_G(x)$ can be defined as*

$$P_G(x) = \sum_{i=1}^{cl(G)} a_i x^i \tag{4}$$

where a_i is the number of forcing resonant sets of size i.

The minimum size of forcing resonants is 1 since any non-empty hexagonal system itself has more than one unique perfect matching, so that the size of a forcing set cannot be 0. The maximum size of forcing resonant sets is Clar number because it is the maximum size of all resonant sets.

In section 2.2, we will provide a recurrence relation to calculate the forcing resonant polynomial of all cata-condensed hexagonal systems. In section 2.3, we will design a data structure weighted tree to implement this algorithm. We will also show one algorithm of calculating the number of perfect matching based on paper [53] in this section. In section 2.4, we will show the implementation results of our proposed algorithm. In section 2.5, we will draw conclusions.

2.2 Recurrence Relation of Forcing Resonant Polynomial

Let L_k be the *linear hexagonal chain* with length k. L_k is consisted of k hexagons $h_1, h_2, ..h_k$ where edge $h_{i-1} \cap h_i$ is in the opposite position of edge $h_{i+1} \cap h_i$. The graph of L_k is showed in Figure 3. We have this following theorem for linear hexagonal chain from paper [70].

Theorem 2.3. *A hexagonal system H has $cl(H) = 1$ if and only if H is a linear chain.*

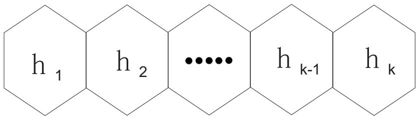


Figure 3: A linear hexagonal chain L_k with k hexagons

Theorem 2.4. *Let L_k be the linear hexagonal chain with k hexagons. Then*

$$spec_{FRS}(L_k) = \{1\} \tag{5}$$

Proof. Let $\mathcal{H}$ be a forcing resonant set of L_k. Suppose that $|\mathcal{H}| = m$.

We know that $m \leq 1$ according to Theorem 2.3. Now we need to prove $0 \notin spec_{FRS}(L_k)$.

Suppose $0 \in spec_{FRS}(L_k)$. That means $\mathcal{H}$ is an empty set. The rest of graph $L_{k'}$ by deleting all the vertices covered by $\mathcal{H}$ is still L_k. Now we proof L_k does not have a unique perfect matching.

1. when $k = 1$. L_k has two perfect matchings.

2. when $k > 1$. L_k can be decomposed into two graphs G_1 and G_2 which are showed in Figure 4. G_1 has two perfect matchings, denoted by $\mathcal{M}_1$ and $\mathcal{M}_2$. Now we use induction to prove G_2 has a unique perfect matchings. Let r be the number of hexagons in G_2.

(a) Let $r = 0$. There is no hexagon in G_2 which is showed in Figure 5. In this case, we rename G_2 as G_3 (See figure 5). Obviously, G_3 has a unique perfect matching, which means G_2 has a unique perfect matching if it contains 0 hexagon.

(b) Assume G_2 has a unique perfect matching for $r = n$. In this case we rename G_2 as G_4. So, G_4 has a unique perfect matching.

Let $r = n + 1$. The two suspended edges must must belong to the perfect matching of G_2 for $r = n + 1$. Delete the two suspended edges, The remaining graph is G_4. We can obtain a unique perfect matching of G_2 for $r = n + 1$ by adding the two suspended edges into the unique perfecting of G_4.

Thus, G_2 has a unique perfect matching, denoted by $\mathcal{M}_3$. We can obtain two perfect matchings of L_k. One of them is combination of $\mathcal{M}_3$ and $\mathcal{M}_1$. Another is combination of $\mathcal{M}_3$ and $\mathcal{M}_2$. So L_k does not have a unique perfect matching for $k > 1$.

Thus, L_k does not have a unique perfect matchings. which contradicts with the definition of forcing resonant set. Thus, $0 \notin spec_{FRS}(L_k)$. Thus, $spec_{FRS}(L_k) = \{1\}$ □

Theorem 2.5. *Let L_k be a linear hexagonal chain with k hexagons. The forcing resonant polynomial of L_k is kx. That is*

$$P_{L_k}(x) = kx \tag{6}$$

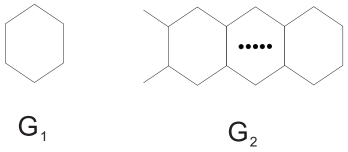


Figure 4: Decomposed linear hexagonal chain into two subgraphs

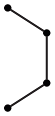


Figure 5: G_3: when the number of hexagons in G_2 is 0

Proof. Let h_i be the ith hexagon of L_k. By Theorem 2.4 we know the size of forcing resonant set is 1. So the forcing polynomial can be expressed as

$$P_{L_k}(x) = a_1 x$$

a_1 is the number of forcing resonant sets of size 1.

Let $K = \{h_i\}$. Let $L_{k'}$ be the subgraph of by deleting all the vertices covered by K from G. Then $L_{k'}$ has three possible cases.

- Case 1: $L_{k'}$ is the graph 1 in Figure 6. According to the proof in Theorem 2.4, $L_{k'}$ has a unique perfect matching.

- Case 2: $L_{k'}$ is the graph 2 in Figure 6. Similarly as the proof in Theorem 2.4, we can proof $L_{k'}$ has a unique perfect matching.

- Case 3: $L_{k'}$ contains two disjointed components graph 1 and graph 2 in Figure 6. Let $\mathcal{M}_1$ be the perfect matching of graph 1. Let $\mathcal{M}_2$ be the perfect matching of graph 2. Combine these two perfect matchings into one edge set $\mathcal{M}$. $\mathcal{M}$ is the unique perfect matching of $L_{k'}$.

So, $\{h_i\}$ is a forcing resonant set of L_k. Thus, $a_1 = k$. Then, we have

$$P_{L_k}(x) = kx$$

□

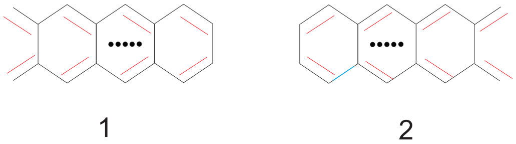


Figure 6: $L_{k'}$: subgraph after deleting one hexagon from a linear hexagonal chain

Here are some examples:

G_1 is a hexagonal linear chain with five hexagons and G_2 is a hexagonal linear chain with 3 hexagons. (G_1 and G_2 are showed in Figure 7)

So, the forcing resonant polynomial of G_1 is:

$$P_{G_1}(x) = 5x.$$

So, the forcing resonant polynomial of G_2 is:

$$P_{G_2}(x) = 3x.$$

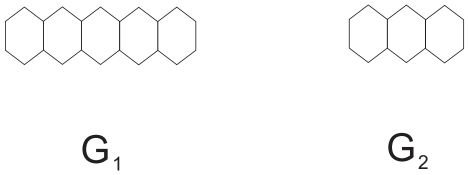


Figure 7: Examples of forcing resonant polynomial of linear chains

Proposition 2.6. *Let G be a graph with disjointed components $G_1, G_2, ...G_k$ which are cata-condensed benzenoid systems. Then*

$$P_G(x) = \prod_{i=1}^{k} P_{G_i}(x). \tag{7}$$

Proof. Clearly, G has a unique perfect matching if and only if every disjoined component G_i has a unique perfect matching.

Let $\mathcal{H}$ be the set of forcing resonant sets of G. Let $\mathcal{H}_i$ be the set of forcing resonant sets of G_i. Let $M = \{H_1, H_2, ..., H_k | H_i \in \mathcal{H}_i\}$. Then, M is a forcing resonant set of G. Let n denote the total number of M. $n = \prod_{i=1}^{k} |\mathcal{H}_i|$. $\mathcal{H}$ is one to one corresponding relationship with $\{M_1, M_2, ...M_n\}$.

Suppose $P_G(x) = \sum_{m=0}^{cl(G)} a_m x^m$. That mean the number of forcing resonant sets of G with length m is a_m.

Suppose $P_{G_i}(x) = \sum_{j_i=0}^{cl(G_i)} a_{j_i} x^{j_i}$, where $i = 1.2...k$. In other words, the number of forcing resonant sets with length j_i is a_{j_i} for anyone of the component $G_i, i = 1, 2, ...k$.

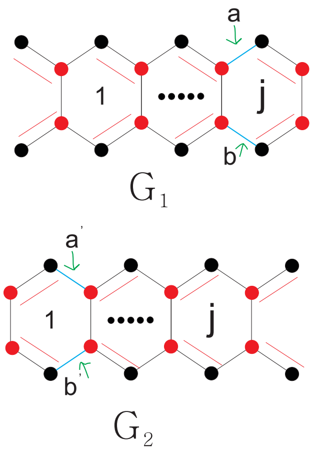


Figure 8: G_1 and G_2: two components of a hexagonal system

Thus,

$$a_m x^m = \sum_{j_i=0, j_1+j_2+\ldots j_k=m}^{cl(G_i)} \prod_{i=1}^{k} a_{j_i} x^{x^{j_i}}.$$

That is, $P_G(x) = \prod_{i=1}^{k} P_{G_i}(x)$. □

Proposition 2.7. *Graph G_1 and Graph G_2 are showed in Figure 8. The blue edge of G_1 does not belong to the unique perfect matching of G_1. The blue edge of G_2 does not to belong the unique perfect matching of G_2.*

Proof. The set of red edges in G_1 is the unique perfect matching $\mathcal{M}_1$ of G_1. Clearly, the blue edge does not belong to $\mathcal{M}_1$.

The set of red edges in G_2 is the unique perfect matching $\mathcal{M}_2$ of G_2. Clearly, the blue edge does not belong to $\mathcal{M}_2$. □

Let G be a cata-condensed benzenoid system. It is well known that every maximal linear hexagonal chain of G has at least two hexagons if G is not a linear hexagonal chain. We define a maximal linear hexagonal chain L as *pendant chain* of G if one of the end hexagons of L has only one adjacent hexagon in G.

Lemma 2.8. *Let G be a cata-condensed benzenoid system. Every forcing resonant set of G contains exactly one hexagon of L if L is not a non-pendant chain with two hexagons.*

Proof. If G is a linear hexagonal chain. According to theorem 2.5, every forcing resonant set of G must contain exactly one hexagon of L. Otherwise, we can proof Lemma 2.8 by classifying L into two different conditions.

Let $\mathcal{H}$ be the forcing resonant set of G. Assume that any of the hexagon which is belonged to $\mathcal{H}$ does not belong to L. We can construct a subgraph G' by deleteing all the vertices covered by $\mathcal{H}$ from G. Then G' consists of several disjoint components. Let G'_1 be the component which contains L.

1. L is a pendant chain except linear hexagonal chain.

   G'_1 has two possible structures which are showed in Figure 9. Similarly as the proof in Property 6, the three edges a, b and c do not belong to the unique perfect matching of the subgraph G'_1. Figure 9 can be disconnected by deleting edges a, b and c. One of the components of G'_1 is a linear hexagon chain. So this component does not have a unique perfect matching. Thus G'_1 does not have a unique perfect matching. That means G' does not have a unique perfect matching, which contradicts with the definition of the forcing resonant set. Thus, $\mathcal{H}$ contains at least one of the hexagons of L.

By Theorem 2.4 and 2.5, every forcing resonant set of G can not contain more than one hexagons of L. So, the forcing resonant set of G contains exactly one hexagon of L.

2. L is a non-pendant chain with at least three hexagons.

   G'_1 has three possible structures which are showed in Figure 10.

   Similarly as the proof in 1, we can proof $\mathcal{H}$ contains at least one of the hexagons of L.

   By Theorem 2.4 and 2.5, every forcing resonant set of G can not contain more than one hexagons of L. So, the forcing resonant set of G contains exactly one hexagon of L.

□

Corollary 2.9. *Every forcing resonant set $\mathcal{H}$ hits every maximal linear hexagonal chain.*

Proof. Let L be a maximal linear hexagonal chain of G if G is a cata-condensed benzenoid system. If L is not a non-pendant chain with two hexagons. Then every forcing resonant set $\mathcal{H}$ contains exactly one hexagon of L. In other words, Every forcing resonant set $\mathcal{H}$ hits every maximal linear hexagonal chain.

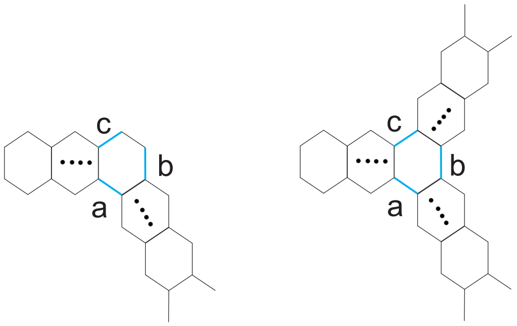


Figure 9: L is a pendant chain except linear hexagonal chain

Otherwise, both of the end hexagons of L with exactly two hexagons have more than one adjacent hexagons in G. We can divide $\mathcal{H}$ into two classes:

1. One of the end hexagons of L belongs to $\mathcal{H}$. Clearly, $\mathcal{H}$ hits every maximal linear hexagonal chain.

2. Both of the end hexagons do not belong to $\mathcal{H}$. By Lemma 2.8, one of the hexagon in the maximal linear hexagonal chains which share a common hexagon with L must belong to $\mathcal{H}$. Thus $\mathcal{H}$ hits the common hexagons. That means $\mathcal{H}$ hits L. Then $\mathcal{H}$ hits every maximal linear hexagonal chain.

□

Lemma 2.10. *Let G be a cata-condensed benzenoid system. Let A be a*

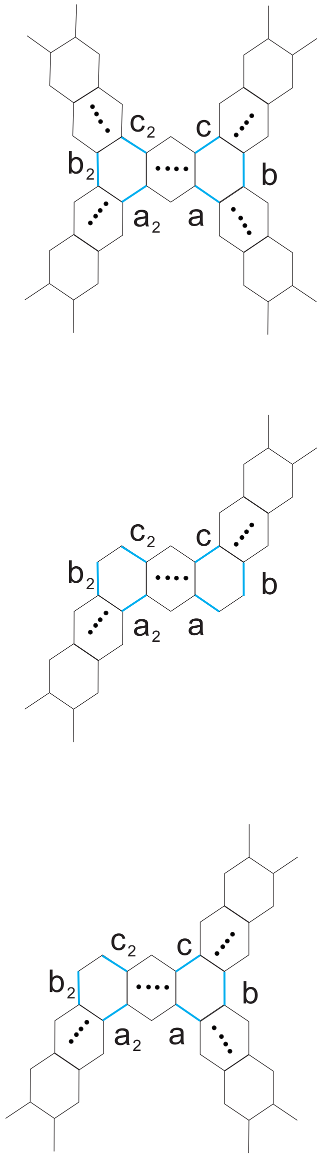


Figure 10: *L* is a non-pendant chain with at least three hexagons

hexagon of G. Then,

$$P_G(x) = P_G(x, A^C) + P_G(x, A)$$

$P_G(x, A^C)$ denotes the polynomial of forcing resonant sets which do not contain the hexagon A of G. $P_G(x, A)$ denotes the polynomial of forcing resonant sets which contain the hexagon A of G.

Proof. The forcing resonant sets can be classified into two cases. One is the sets do not contain A. The other one is the sets contain A.

Let,

$$P_G(x) = \sum_{i=0}^{cl(G)} a_i x^i$$

$$P_G(x, A^C) = \sum_{m=0}^{cl(G)} a_m x^m$$

$$P_G(x, A) = \sum_{n=0}^{cl(G)} a_n x^n$$

Thus,

$$P_G(x) = \sum_{i=0}^{cl(G)} a_i x^i = \sum_{m=0}^{cl(G)} a_m x^m + \sum_{n=0}^{cl(G)} a_n x^n = P_G(x, A^C) + P_G(x, A).$$

□

We propose a recursive formula which is the main idea of our method to computer the $P_G(x)$ in Theorem 2.11. Note that $P_G(x)$ equals 1 if G is an empty graph.

Theorem 2.11. *Let G be a cata-condensed benzenoid system and L be a pendant chain with r hexagons. Let H be the subgraph consisting of all hexagons of G except these in L, and H' be the subgraph of H consisting* *of all hexagons except these contained in the maximal linear chains of G with a common hexagon with L. Then,*

$$P_G(x) = (r-1)xP_H(x) + xP_{H'}(x) \tag{8}$$

Proof. Let A' and A be the pair of end hexagons of L. The forcing resonant sets of G can be classified into two classes. One is the sets that do not contain hexagon A and the other one is the sets that contain hexagon A.

- Case 1, the forcing resonant sets of G do not contain hexagon A.

  Let $\mathcal{H}$ be the forcing resonant set of G. Let $\mathcal{M}$ be the unique perfect matching of the subgraph which is constructed by deleting all vertices covered by $\mathcal{H}$ from G. Let G' be the subgraph consisting of all hexagons of L except A. Since A does not belong to $\mathcal{H}$, one of the hexagon of G' must belong to $\mathcal{H}$ according to Lemma 2.8.

  Let a, b, and c be the three edges of A that do not belong to subgraph G' and subgraph H.

  According to Property 2.7, edge a and edge b does not belong to the unique perfect matching $\mathcal{M}$.

  The subgraph H consists of at most two disjoint components, denoted by H_1 and H_2. Each of the components has even number of the vertices. Thus, edge c does not belong to the unique perfect matching $\mathcal{M}$.

Thus, The graph G can be decomposed into two disjoint parts, G' and H. According to Property 2.6, we can achieve the following formula,

$$P_G(x, A^C) = P_{G'}(x)P(H) = (r-1)xP(H)$$

$P_G(x, A^C)$ denotes the polynomial of forcing resonant sets which do not contain hexagon A of G.

- Case 2, the forcing resonant sets of G contain hexagon A.

  Hexagon A is contained by at most three different linear hexagonal chains. Let L_2 and L_3 be the other two maximal linear hexagonal chains which contain hexagon A. Let A and A_2 be the pair of end vertices of L_2. Let A and A_3 be the pair of end vertices of L_3. Let G'' be the graph consisting of all hexagons in L, L_2 and L_3 except hexagon A_2 and hexagon A_3.

  $\{A\}$ is the only forcing resonant set that contains A of G'' by Lemma 2.8.

  Similarly, the blue edge a_2, b_2, c_2, a_3, b_3 and c_3 do not belong to the unique perfect matching $\mathcal{M}$. Thus, graph G can be decomposed into two disjoint parts G'' and H'. Thus,

$$P_G(x, A) = P_{G''}(x, A)P(H') = xP(H').$$

  $P_G(x, A)$ denotes the polynomial of forcing resonant sets which contain hexagon A of G. $P_{G''}(x, A)$ denotes the polynomial of forcing resonant sets which contain hexagon A of G''.

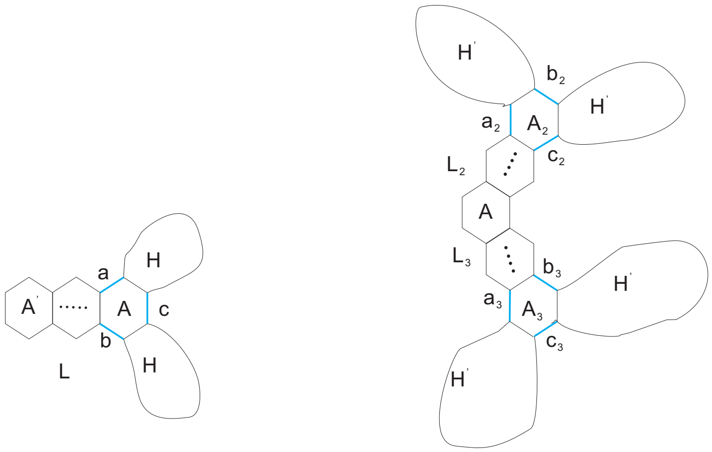


Figure 11: Decomposition of cata-condensed benzenoid system G

By Lemma 2.10, $P_G(x) = P_G(x, A^C) + P_G(x, A) = (r-1)xP_H(x) + xP_{H'}(x)$.

□

Here is an example:

We use Theorem 2.11 to compute the forcing resonant polynomial of G in Figure 12.

$$P_G(x) = 2xP_{G_1}(x)P_{G_2}(x) + x = 2x * x * 3x + x = 6x^3 + x$$

We list all forcing resonant sets of G:

$\{1,4,5\}, \{1,4,6\}, \{1,4,7\}, \{2,4,5\}, \{2,4,6\}, \{2,4,7\}, \{3\}$.

We can observe that there are six forcing resonant sets with size three

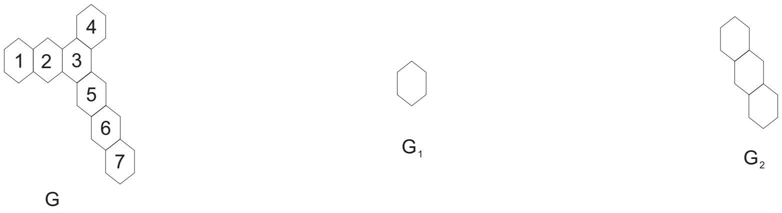


Figure 12: Example of a cata-condensed benzenoid system

and one forcing resonant set with size 1. This result matches the polynomial which is obtained based on our method.

2.3 Algorithm Implementation

2.3.1 Weighted Tree of Benzenoid Systems

In this section, we use a weighted tree (T, w) to store the information of a cata-condensed benzenoid system. The two basic rules are:

1. A vertex of T represents a maximal linear hexagonal chain.

2. The weight of the root of the tree is the number of hexagons in the corresponding maximal linear hexagonal chain and the weights of all other nodes are the 1 less than the number of hexagons in the corresponding maximal linear hexagonal chains.

We can construct (T, w) step by step:

1. Start with a pendant chain L. L corresponds with the first vertex in (T, w).

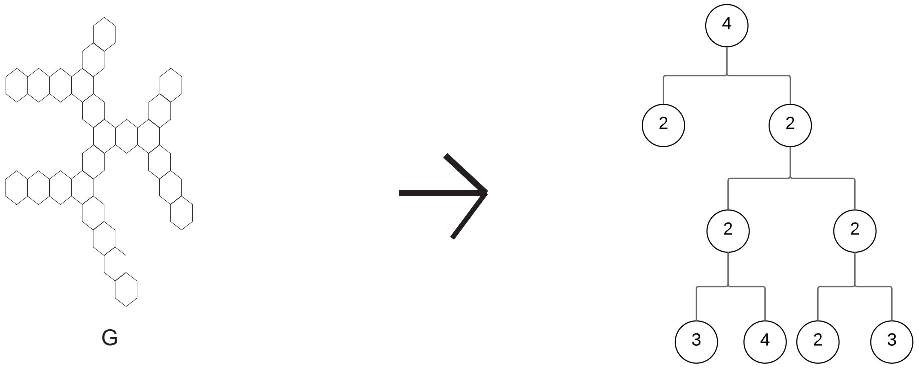


Figure 13: Weighted tree of a benzenoid system

2. The children of a vertex v in (T, w) are defined to be the maximal linear hexagonal chains which share a common hexagon with the corresponding chain of the vertex v.

3. Continue to do step 2 until the number of vertices equals the number of maximal linear hexagon chains.

Note that the vertex which corresponds with the initial pendant hexagonal chain L is the root of (T, w). Figure 13 shows a cata-condensed benzenoid system and its corresponding weighted tree. Let G be a cata-condensed benzenoid system with n maximal linear hexagonal chains. Let (T, w) be the corresponding weighted tree of G. Thus, the number of vertices in T is n.

We use pre-order traversal to save the tree. For example, the list of the tree in Figure 12 is [4, 2, -1, -1, 2, 2, 3, -1, -1, 4, -1, -1, 2, 2, -1, -1, 3, -1, -1], where -1 represents an empty child.

Let node.value be the weight of the node. Note that $P_T(x) = 1$ if (T, w) is an empty tree. We need to rebuild the tree when node.value is 1, since the minimum length of a chain is larger than 1 if there are more than one vertices in the tree. The idea to rebuild the tree is to reverse the relationship of parent and current as shown in Figure 14, where current is the left child of parent. After reversing, parent becomes the left child of current and parent's right child becomes the right child of current.

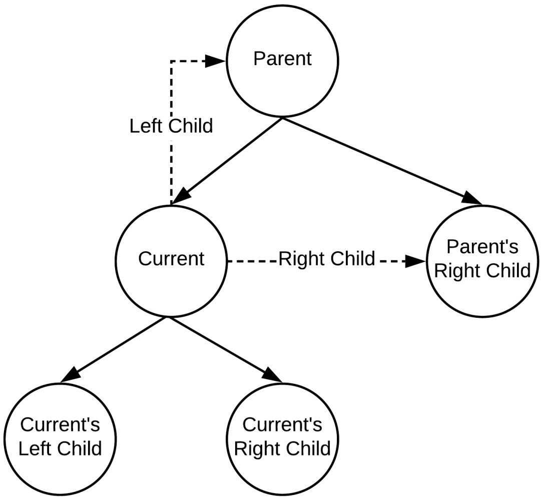


Figure 14: Rebuild the tree when the root value is 1

The recursive algorithm is described as below:

Algorithm 1: Calculation of Forcing Resonant Polynomial

```
Input: node − root of the weighted tree
Output: a polynomial
poly(node):
    if node.value = 1 then
        rebuild the tree from another pendant chain.
    end
    return (node.value − 1) ∗ x ∗
        poly(node.left) ∗ poly(node.right) +
        x ∗ poly(node.left.left) ∗ poly(node.left.right)
        ∗ poly(node.right.left) ∗ poly(node.right.right)

# remove the root 1
rebuild(current):
    parent = current
    current = parent.left
    set current.left = ∅
    set current.right = parent.right
    set parent = current
    set current = current.origin_left

    while current ≠ ∅ do
        if current.left = ∅ then
            swap the right child as left child
        end
        # reverse the parent−child(current) relationship
        # as illustrated in Figure 13
        set current.left = parent
        set current.right = parent.right
        set parent = current
        set current = current.origin_left
    end
    # increase the value on the last branch by 1
    parent.data = parent.data + 1
```

2.3.2 Count the Number of Perfect Matching

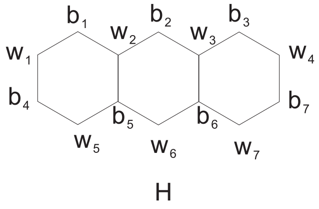


Figure 15: Count the number of perfect matching

This is a brief introduction of an algorithm of counting perfect matchings in a bipartite graph from paper [53]. B and W are two disjoint sets of vertices of bipartite hexagonal systems. One example of B and W are illustrated in Figure 15.

Active is a temporary set to hold active vertices of B or W, where a vertex in B or W is called active if this vertex is being used to add new vertices to the opposite set W or B. All the new vertices found in one loop is temporarily saved in a set New. For example, B and W are initialized as $\{b_1\}$ and $\emptyset$. Then the only one active vertex b_1 in B is used to add new vertices w_1, w_2 to W. So, in the next loop, b_1 is not active anymore, w_1, w_2 are active vertices instead to add b_2, b_4, b_5 to B. Then b_2, b_4, b_5 will be active in the next loop, and so on so forth, until all vertices are added to either B or W.

Algorithm 2: Calculation of Number of Perfect Mathcings

```
Input: adj_list − adjacency list of a graph G.
       matrix − biadjacency matrix initialized as
            a zero matrix.
Output: npm − the number of perfect matching in graph G.

B, W, Active= {0}, ∅, {0}
i = 0
while V(B) ∪ V(W) ⊂ V(G) do
    New = ∅
    for each vertex u ∈ Active do
        for each vertex v ∈ adj_list[u] do
            if v ∉ B and v ∉ W and v ∉ New then
                New = New ∪ {v}
            end
        end
    end

    // add New to B or W alternatively in each loop.
    if i is odd then
        B = B ∪ New
    else
        W = W ∪ New
    end

    // i changes to even/odd alternatively in each loop.
    i = i + 1
    // New becomes Active in next loop.
    Active = New
end

for each vertex b ∈ B do
    for each vertex w ∈ adj_list[b] do
        matrix[b, w] = 1
    end
end

npm = abs(det(matrix))
```

2.3.3 Calculate HOMO-LUMO Gap

Let G be a connected graph with n vertices, $v_1, v_2, ...v_n$. Let $S(G) = \{\lambda_1, \lambda_2, \lambda_3...\lambda_n\}$ be the *spectrum* of G where λ_i is the ith eigenvalue of the adjacent matrix of G and $\lambda_i \geq \lambda_j$ when $i < j$. Then, HOMO-LUMO gap equals $\lambda_H - \lambda_L$ with $H = \lfloor (n+1)/2 \rfloor$ and $L = \lceil (n+1)/2 \rceil$.

2.4 Algorithm Result

Let G be a cata-condensed benzenoid system. All cata-condensed benzenoid systems with n hexagons can be obtained as planar code format based on this work [7]. We converted planar code of graph to tree and then calculate the forcing resonant polynomial and other properties based on the in order traversal of the tree.

Figure 16 lists cata-condensed benzenoid systems with five hexagons and Figure 17 lists cata-condensed benzennoid systems with six hexagons.

The *coefficient vector* of cata-condensed benzenoid system defined as:

$$a = \begin{bmatrix} a_{cl(G)} \\ a_{cl(G)-1} \\ \vdots \\ a_1 \end{bmatrix}$$

where a_i is the coefficient of x^i in $P_G(x)$, then a is called the the *coefficient vector* of G.

Let's recall the definition of *lexicographic order* of a vector:

$$v_1 = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} > v_2 = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \iff \exists 1 \leq k \leq n, \quad \text{such that}$$

$x_1 = y_1, x_2 = y_2, ..., x_{k-1} = y_{k-1},$ but $x_k > y_k$.

We can construct graph based on the Clar number and we can try to reconstruct graph based on the coefficient vector. But currently it is impossible to reconstruct graph based on the HOMO-LUMO gap and based on the number of perfect matchings. The implementation results are showed in Table 1 (cata-condensed benzenoid systems with five hexagons), Table 2 (cata-condensed benzenoid systems with six hexagons), Table 3 (in appenix) (cata-condensed benzenoid systems with seven hexagons), Table 4 (in appenix) (cata-condensed bezenoid systems with eight hexagons).

There are 12 cata-condensed benzenoid systems with 5 hexagons, 36 cata-condensed benzenoid systems with 6 hexagons, 118 cata-condensed benzenoid systems with 7 hexagons and 411 cata-condensed benzenoid systems with 8 hexagons. Figure 18 show their linear square fitting lines between coefficient vector and HOMO-LUMO gap respectively with Table 1, Table 2, Table 3 and Table 4. From these four figures, we can see that the coefficient vector increases as HOMO-LUMO gap increases. Thus, we can use the coefficient vector to predict the stability of systems. In other words, we find a new stability predictor. We can also obtain the conclusion that the isomer with a larger coefficient vector is more stable. We can come out the following conclusion by observing the results in these tables that the coefficient vector is a refined stability predictor than Clar number, because the number of isomers with the same coefficient vector is smaller than the number of isomers with the same Clar number.

Table 1: All cata-condensed Benzenoid Systems with 5 hexagons

Graph index	Polynomial	Clar number	Coefficient vector	x coordinate	HOMO-LUMO gap	Number of perfect matching
1	$2x^3 + x^2$	3	[2, 1, 0]	8	1.0638	14
2	$2x^3 + x$	3	[2, 0, 1]	7	0.9982	13
3	$x^3 + 3x^2$	3	[1, 3, 0]	6	1.0997	13
4	$x^3 + 3x^2$	3	[1, 3, 0]	6	1.0709	13
5	$x^3 + 3x^2$	3	[1, 3, 0]	6	1.0038	13
6	$x^3 + 2x^2$	3	[1, 2, 0]	5	0.9835	12
7	$x^3 + 2x^2$	3	[1, 2, 0]	5	0.947	12
8	$5x^2$	2	[0, 5, 0]	4	0.8372	11
9	$5x^2$	2	[0, 5, 0]	4	0.8096	11
10	$4x^2 + x$	2	[0, 4, 1]	3	0.8743	10
11	$3x^2 + x$	2	[0, 3, 1]	2	0.6541	9
12	$5x$	1	[0, 0, 5]	1	0.4394	6

Table 2: All cata-condensed Benzenoid Systems with 6 hexagons

Graph index	Polynomial	Clar number	Coefficient vector	x coordinate	HOMO-LUMO gap	Number of perfect matching
1	$x^4 + 2x^3$	4	[1, 2, 0, 0]	16	1.0229	24
2	$x^4 + x^3 + 2x^2$	4	[1, 1, 2, 0]	15	1.0901	23
3	$x^4 + x^3 + 2x^2$	4	[1, 1, 2, 0]	15	1.0727	23

Table 2: Continued.

4	$x^4 + x^3 + x^2$	4	[1, 1, 1, 0]	14	1.0449	22
5	$5x^3$	3	[0, 5, 0, 0]	13	1.0929	22
6	$5x^3$	3	[0, 5, 0, 0]	13	1.0083	22
7	$5x^3$	3	[0, 5, 0, 0]	13	0.9411	22
8	$4x^3 + x^2$	3	[0, 4, 1, 0]	12	1.0785	21
9	$4x^3 + x^2$	3	[0, 4, 1, 0]	12	1.0133	21
10	$4x^3 + x^2$	3	[0, 4, 1, 0]	12	1.0044	21
11	$4x^3 + x^2$	3	[0, 4, 1, 0]	12	0.9969	21
12	$4x^3 + x^2$	3	[0, 4, 1, 0]	12	0.9428	21
13	$4x^3 + x^2$	3	[0, 4, 1, 0]	12	0.8933	20
14	$4x^3 + x^2$	3	[0, 4, 1, 0]	12	0.8902	20
15	$4x^3 + x$	3	[0, 4, 0, 1]	11	1.0115	19
16	$3x^3 + 2x^2$	3	[0, 3, 2, 0]	10	0.9013	19
17	$3x^3 + 2x^2$	3	[0, 3, 2, 0]	10	0.9011	19
18	$3x^3 + 2x^2$	3	[0, 3, 2, 0]	10	0.8755	19
19	$3x^3 + 2x^2$	3	[0, 3, 2, 0]	10	0.8571	19
20	$3x^3 + 2x^2$	3	[0, 3, 2, 0]	10	0.791	19
21	$3x^3 + x$	3	[0, 3, 0, 1]	9	0.7114	17
22	$2x^3 + 4x^2$	3	[0, 2, 4, 0]	8	0.8643	18
23	$2x^3 + 4x^2$	3	[0, 2, 4, 0]	8	0.84	18
24	$2x^3 + 4x^2$	3	[0, 2, 4, 0]	8	0.8387	18
25	$2x^3 + 4x^2$	3	[0, 2, 4, 0]	8	0.8258	18
26	$2x^3 + 3x^2$	3	[0, 2, 3, 0]	7	0.8969	17
27	$2x^3 + 3x^2$	3	[0, 2, 3, 0]	7	0.8575	17
28	$2x^3 + 2x^2$	3	[0, 2, 2, 0]	6	0.7213	16
29	$2x^3 + 2x^2$	3	[0, 2, 2, 0]	6	0.7168	16
30	$8x^2$	2	[0, 0, 8, 0]	5	0.7203	15
31	$8x^2$	2	[0, 0, 8, 0]	5	0.6965	15
32	$7x^2$	2	[0, 0, 7, 0]	4	0.6142	14
33	$7x^2$	2	[0, 0, 7, 0]	4	0.6066	14
34	$6x^2 + x$	2	[0, 0, 6, 1]	3	0.6715	13
35	$4x^2 + x$	2	[0, 0, 4, 1]	2	0.4872	11
36	$6x$	1	[0, 0, 0, 6]	1	0.3387	7

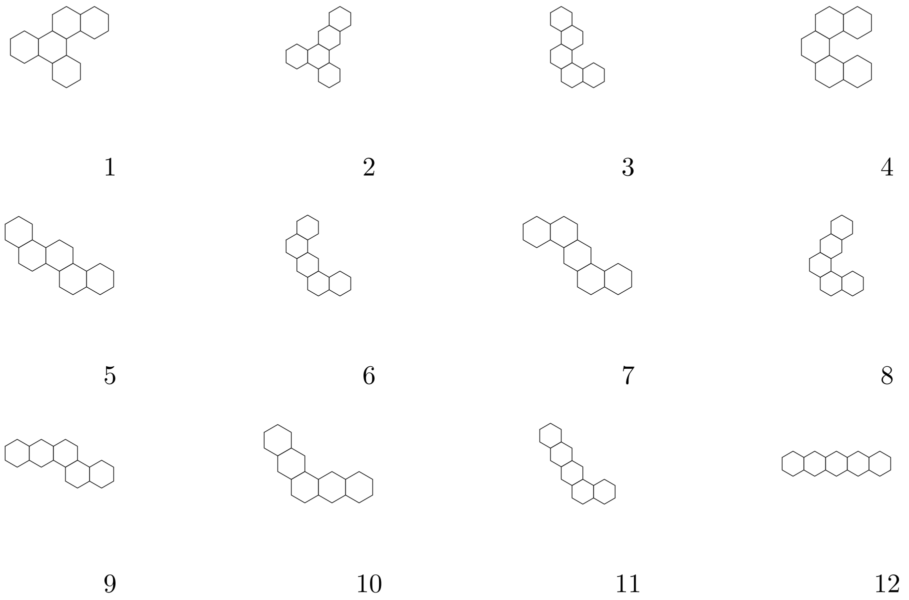


Figure 16: 12 cata-condensed benzenoid systems with 5 hexagons

2.5 Conclusion

As a summary, we obtain the following conclusions by comparing the implementation results:

1. The coefficient vector increases as the HOMO-LUMO gap increases.
2. The stability of G is relative with the coefficient vector. The one that has larger coefficient has better stability.
3. The coefficient vector is a more accurate indicator than Clar number to predict the stability of G.

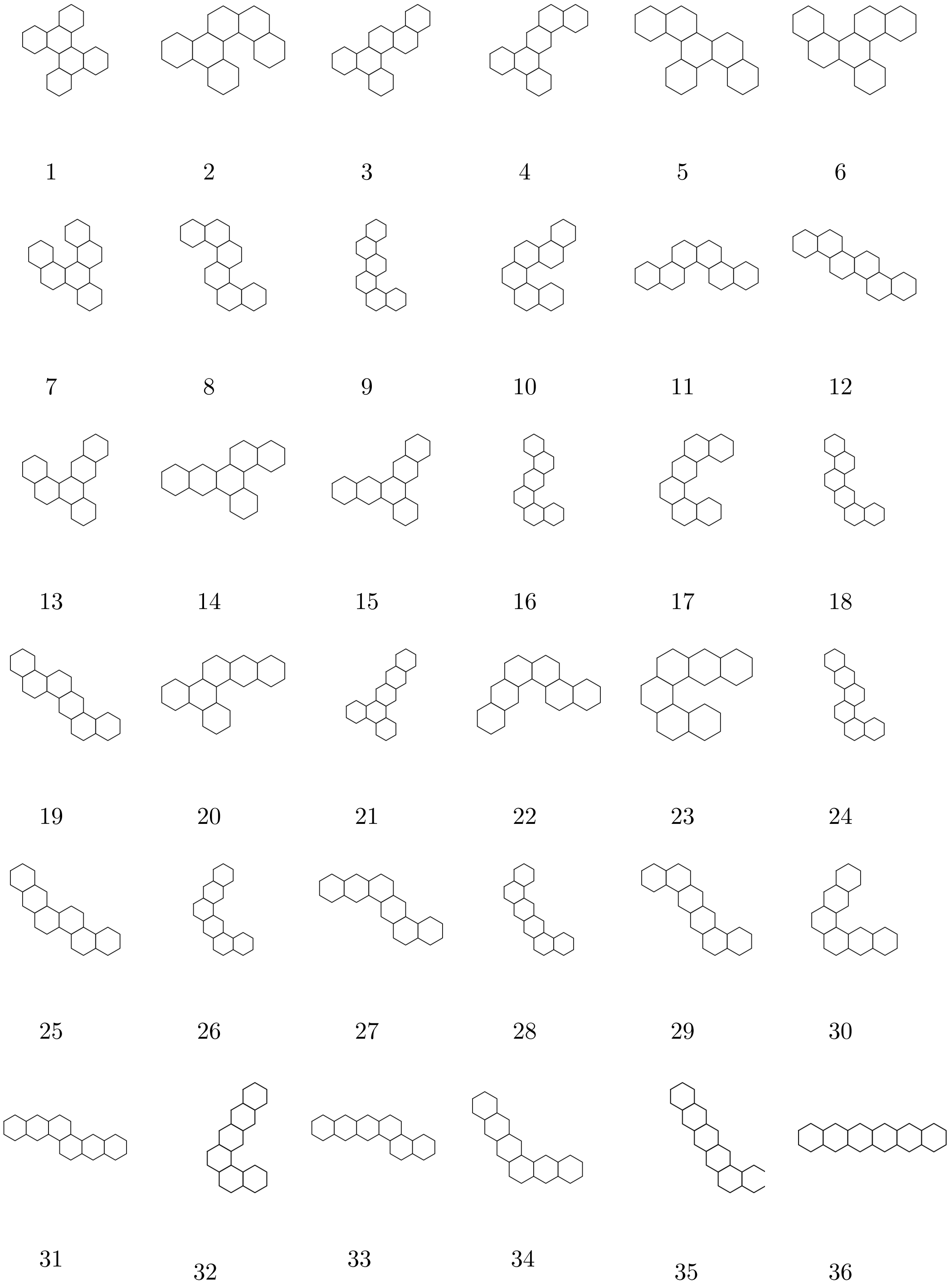
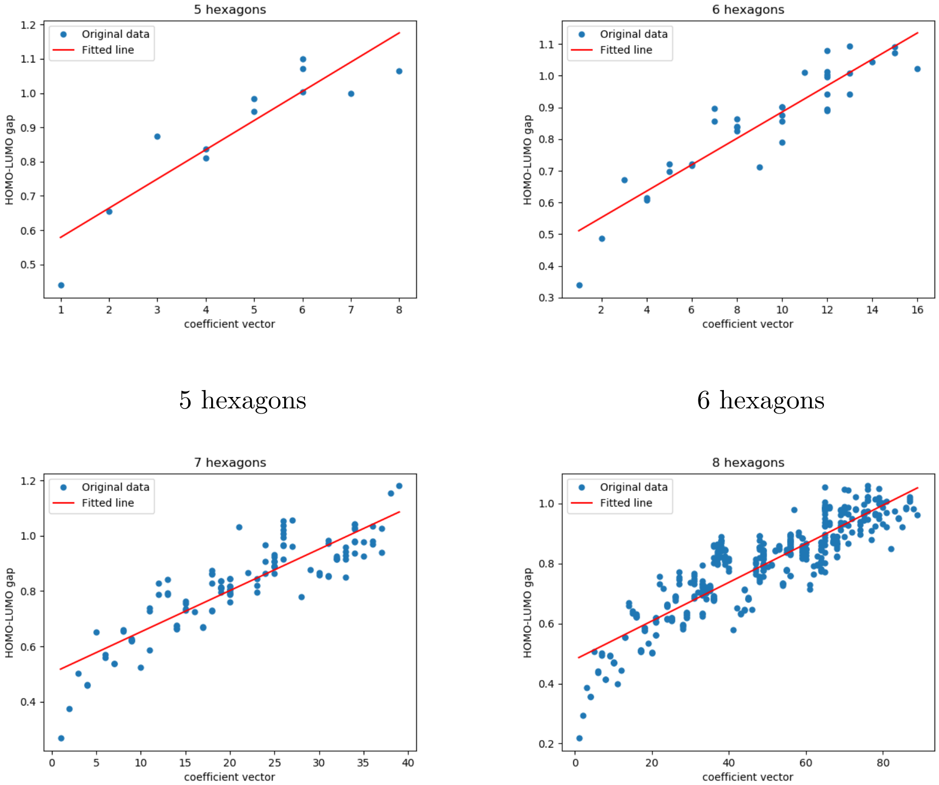


Figure 17: 36 cata-condensed benzenoid systems with 6 hexagons



5 hexagons

6 hexagons

7 hexagons

8 hexagons

Figure 18: Fit the coefficient vectors and HOMO-LOMO gaps of benzenoid systems using least square

CHAPTER 3

CONSTRUCTION OF CATA-CONDENSED BENZENOID SYSTEMS

3.1 Introduction

In the previous chapter, we calculated many parameters of all cata-condensed benzenoid systems giving the number of hexagons. However, in many situations, we are only interested in the benzenoid systems with maximum clar number. In this chapter, we design an algorithm to construct all cata-condensed benzenoid systems with maximum clar number when their Clar number is $n/6$ where n is the number of vertices.

Let H be a cata-condensed hexagonal system. The *inner dual* H^* of H is a graph such that each vertex of H^* represents a hexagon of H and each edge of H^* connects two vertices if their corresponding hexagons share an edge in H. A *degree* of a vertex in H^* is the number of edges that are incident to this vertex. Let $V_i = \{v|\text{degree}_{H^*}(v) = i\}$ for $i = 1, 2, 3$. Let $cl(H)$ be the Clar number of H. A vertex set of S of H is independent if and only if any two vertices of S are not incident. Let S^c be a vertex set that consists of all the vertices in H but not in S. Zhai, Alrowaili and Ye proved the following characterization for cata-condensed hexagonal systems maximizing the Clar number[68].

Theorem 3.12. *Let H be a cata-condensed hexagonal system with n vertices and let H^* be the inner dual of H. Then the Clar number of H is $\frac{n}{6}$ if and only if H^* has an independent set S which contains all vertices of degree at most 2.*

In section 3.2, we will design an algorithm based on the above theorem. In section 3.3, we will show the implementation result of our algorithm when the number of hexagons is 7, 10, and 13.

3.2 Benzenoid System with Maximum Clar Number Construction Algorithm

Let h be the number of hexagons and n be the number of vertices in a cata-condensed hexagonal system H with Clar number $\frac{n}{6}$. It is easy to show that

$$n = 4h + 2. \tag{9}$$

Since H^* is a tree and the degree of every vertex in the independent set S^c is 3, we have the following equation:

$$3|S^c| = |S| + |S^c| - 1 \tag{10}$$

The number of hexagons in H equals the number of vertices in H^*. So we have another equation:

$$|S^c| + |S| = h \tag{11}$$

Based on equations (9),(10) and (11), we can obtain that:

$$\begin{cases} |S^c| = \frac{n-6}{12}, \\ \\ |S| = \frac{n}{6}, \end{cases} \tag{12}$$

This indicates that the number of vertices in S = the number of hexagons in the corresponding forcing the resonant set of H = the Clar number of H = $\frac{n}{6}$.

Let x_1 be the number of vertices with degree 1 in S. Let x_2 be the number of vertices with degree 2 in S. Let x_3 be the number of vertices with degree 3 in S. Let x_3' be the number of vertices with degree 3 in S^c. Then we can obtain the following two equations:

$$\begin{cases} x_1 + x_2 + x_3 = |S|, \\ \\ x_1 + 2x_2 + 3x_3 = h - 1, \end{cases} \tag{13}$$

Combing with equations (9), (12), the solution for (13) is:

$$\begin{cases} x_1 = \frac{n}{12} + \frac{3}{2} + x_3, \\ \\ x_2 = \frac{n}{12} - \frac{3}{2} - 2x_3, \end{cases} \tag{14}$$

Let $|S_c| = k$, then we can get the following results:

$$\begin{cases} n = 12k + 6, \\ h = 3k + 1, \\ |S| = 2k + 1, \\ x_1 = k + 2 + x_3, \\ x_2 = k - 1 - 2x_3, \end{cases} \tag{15}$$

Then, we can compute the degree sequence of the inner dual H^* of cata-condensed benzenoid systems H with $cl(H) = \frac{n}{6} = 2k+1$. For example, when

$n = 42$, $k = 3, h = 10, x_1 = 6, x_2 = 0, x_3 = 1, x_3' = 3$. So the degree sequence of H^* is $\{1, 1, 1, 1, 1, 1, 3, 3', 3', 3'\}$ which is showed in Figure 19 where each dot represents one hexagon. The corresponding cata-condensed benzenoid systems are showed in Figure 20.

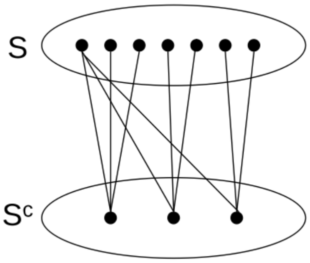


Figure 19: A inner dual H^* with degree sequence $\{1, 1, 1, 1, 1, 1, 3, 3', 3', 3'\}$

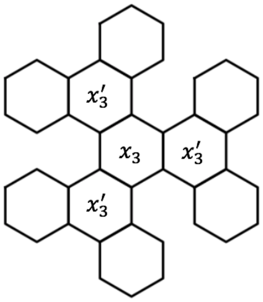


Figure 20: A hexagonal system H with inner dual H^* having degree sequence $\{1, 1, 1, 1, 1, 1, 3, 3', 3', 3'\}$

We can simply use the following modified brute force method to filter all cata-condensed benzenoid systems with maximum Clar number based on these three steps:

- Step 1: Search all benzenoid systems whose number of vertices with degree 3 equal to $|x_3| + |x'_3|$.

- Step 2: Given $Clar(H) = \frac{n}{6}$, if a vertex v belongs to S^c, where S is the maximum independent resonant set of H and the degree of one child of v is 3, we can conclude that degrees of three children of this child of v are all 3, as explained in Figure 20.

- Step 3: Set of all children of each vertex in $S^C \bigcup V(S^C) = V(H^*)$.

The corresponding pseudocode is as following:

Algorithm 3: Construction of cata-condensed Hexagonal System with Maximum Clar Number

```
Input: $h$ − Number of hexagons
       $H$ − All cata−condensed hexagonal systems with
             $h$ hexagons
Output: All cata−condensed hexagonal system with
        $h$ hexagons and maximum Clar number
for $H_i \in H$ do
    if $V_3 = x_3 + x'_3$ then
        // If one child of $x'_3$ is degree 3,
        // all children of this child must be degree 3
        for $v \in V_3$ do
            if degree(one child of $V$) = 3 and
               degree(all children of $V$) = 3 then
               $v \in x'_3$
            end
        end
        if set($x'_3$) covers $V(H)$ then
            save this hexagonal system
        end
    end
end
```

3.3 Implementation Result and Conclusion

Figure 21 shows two cata-condensed benzenoid systems of 7 hexagons with maximum Clar number, Figure 22 shows six cata-condensed benzenoid systems of 10 hexagons with maximum Clar number, and Figure 23 shows 32 cata-condensed benzenoid systems of 13 hexagons with maximum Clar number. Given a hexagonal system H with n vertices, based on our algorithm, it needs $O(1)$ time to check if a vertex is in set of x'_3 and $O(n)$ time to check if the set of x'_3 covers the whole graph. So the overall time needed for a single loop over all hexagonal systems is $O(n)$.

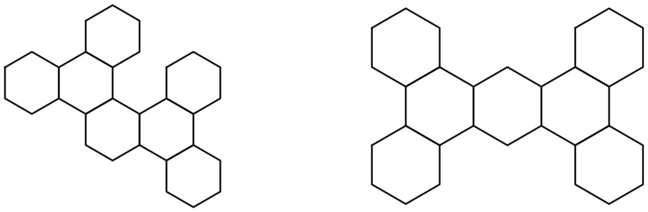


Figure 21: Two cata-condensed benzenoid systems of 7 hexagons with maximum Clar number

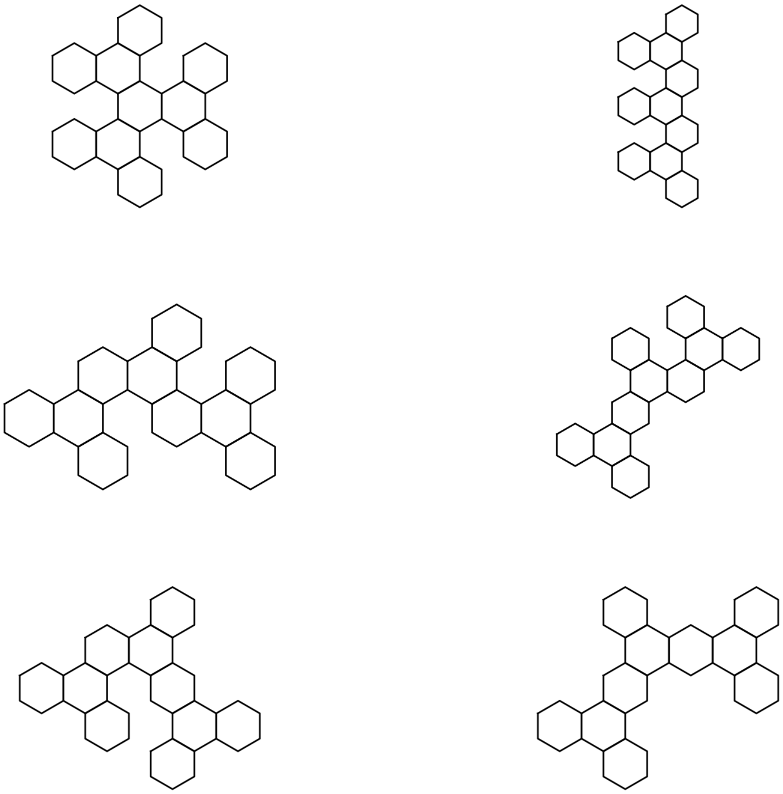


Figure 22: Six cata-condensed benzenoid systems of 10 hexagons with maximum Clar number

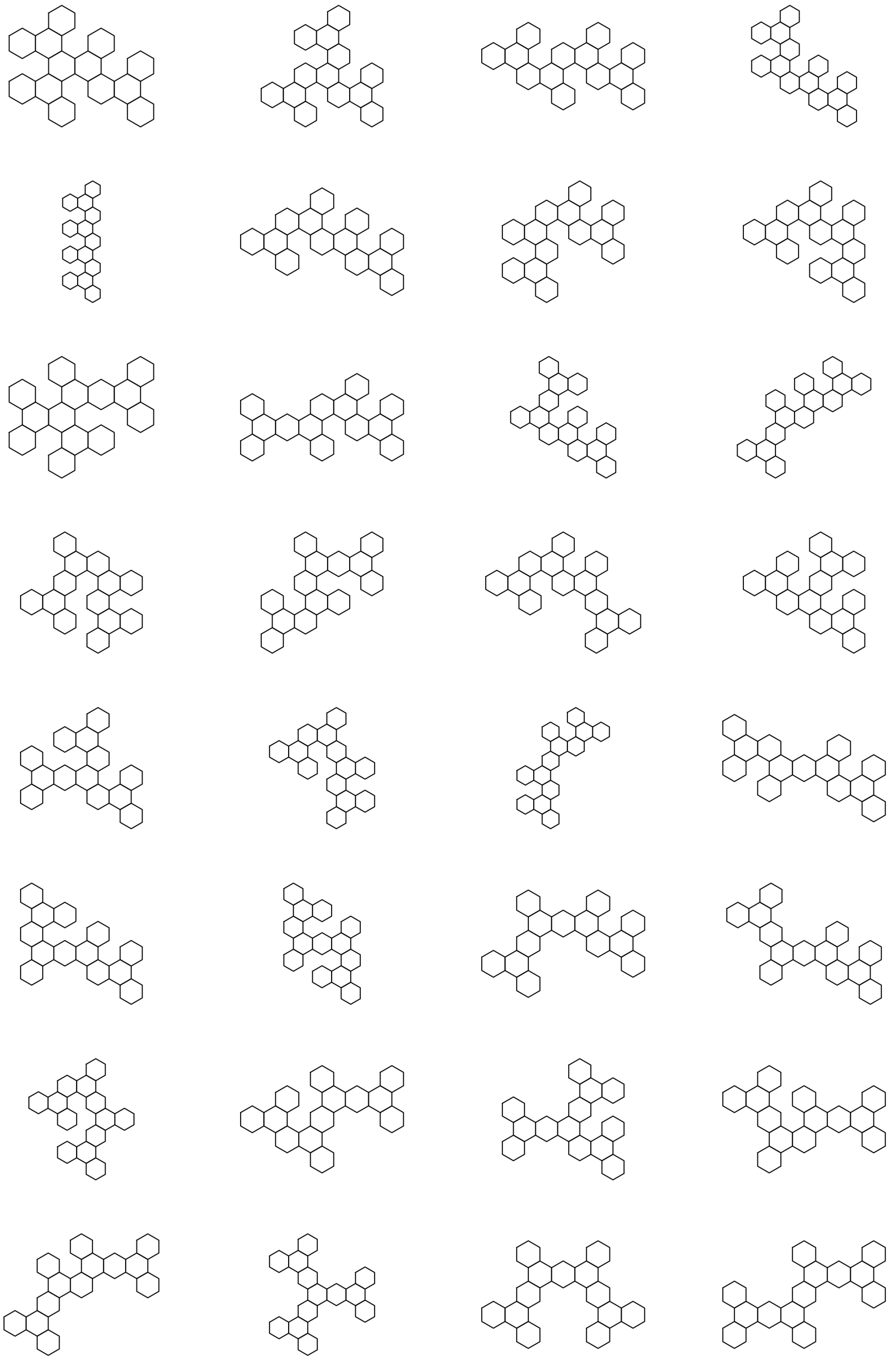


Figure 23: 32 cata-condensed benzenoid systems of 13 hexagons with maximum clar number

CHAPTER 4

CONSTRUCT GRAPHS WITH FORCING FACE

4.1 Introduction

Let S be a set of edges. S is named as forcing set if $G - V(S)$ has a unique perfect matching. More and more mathematicians have been attracted by the concept of *Forcing set* [10] in the past few decades since Harary, Klein, and Zivkovic [31] introduced the definitions of *forcing edge* and *forcing number* in 1991. This forcing concept is originated from the research on the application of Kekulé structure in the resonance theory in chemistry by Klein and Randić [39]. Zhang, Zhang, and Li published a series of papers [72, 73, 42] in 90s that applied the forcing edge idea in hexagonal systems. Motivated by this work, Che and Chen proposed the concept of forcing hexagon in hexagonal systems [9] in 2006. They continued to generalize the idea of forcing hexagon in hexagonal systems to *forcing faces* in plane bipartite graphs [11] in 2013. In recent years, many studies focus on a variety of forcing related topics and applications such as forcing polynomial [80], forcing and anti-forcing numbers in fullerene [57], Clar set and maximum forcing numbers of hexagonal systems [84], and anti-forcing polynomial in [82].

Let G be a graph. A *forcing edge* is an edge that the remaining graph after deleting this edge and all incident edges to this edge from graph G

has a unique perfect matching. A finite *face* of G is a forcing face if the remaining graph of G after deleting all edges of this face and incident edges to this face has a unique perfect matching. A *bridge* is an edge whose removal will increase the number of connected components of graph G. A connected component is *even* if there are even number of vertices in this component, and *odd* otherwise. A bridge is *even* if two new connected components after its removal are all even. A bridge is *odd* if two new connected components after its removal are all odd. A connected component formed by deletion of all bridges is called as *2-edge-connected component*.

In [41], Kotzig proved this following well-known theorem:

Theorem 4.13. *Let G be a connected graph with unique perfect matching. Then G has a bridge that belongs to this matching.*

Every component of a graph with a perfect matching must have an even number of vertices. Hence every even bridge of a graph G is not in M, and every odd bridge of G is in M if G has a unique perfect matching M. In [22], Gabow, Kaplan, and Tarjan designed a polynomial time algorithm to determine if a graph contains a unique perfect matching based on Kotzig's theorem. The idea is to consecutively add an odd bridge e to the unique perfect matching M of G while deleting e and all edges incident to e after e is added to M. Then G has a unique perfect matching when the final graph is empty. They refined this idea by an observation that every new bridge created after deletion of an edge $\{v, w\}$ is on every path from v to w. In order to make the algorithm efficiently, they dynamically add new bridges, instead of exploring all bridges, after removing edges from G. This algorithm can achieve time $O(m\log^4 n)$ by using top tree data structure. The pseudocode of their algorithm is as following:

Algorithm 4: Unique Prefect Matching Algorithm

```
**Initialize** $M = \emptyset$ and $R$ to be the set of all bridges of $G$.
**while** $R \neq \emptyset$ repeat the following steps:
    Delete an edge $\{x, y\}$ from $R$.
    **if** $x, y$ is an odd bridge, delete $\{x, y\}$ from $G$,
    add $x, y$ to $M$, and
    repeat the following steps
    **for** each edge $\{v, w\}$ incident to $x$ or $y$:
        Delete $\{v, w\}$ from $G$, and from $R$ **if** it is in $R$
        If $v$ and $w$ are still connected but are
        in different 2-edge-connected components, **then**
        find a path $P(v, w)$ connecting $v$ and $w$ and add
        every bridge on $P(v, w)$ to $R$.
```

A graph G is planar if G can be drawn in the plane without crossing edges. A planar graph divides the plane into multiple regions, which are also named as faces. A face f of graph G is forcing if $G - V(f)$ has a unique perfect matching. Che and Chen defined that a graph G with a perfect matching is said to be elementary if the union of all perfect matchings forms a connected subgraph. They proved that any connected plane bipartite graph with a forcing face is elementary [11]. However, there does not exist an algorithm to check the existence of forcing face in a general graph. Inspired by this unique perfect matching algorithm, in section 4.2, we gave an algorithm to discover all forcing faces of a graph G with m edges and n vertices in time $O(m^2\log^4 n)$. In section 4.3, we apply the algorithm to hexagonal systems.

In each while loop of the unique perfect matching algorithm, an odd bridge is added to a set M of the graph G. When the while loops stop, the set M would be the unique perfect matching if there is a unique perfect matching in G. In other words, the set M is constructed by iteratively deleting odd bridges and edges incident to odd bridges from the graph one by one until the graph becomes empty. So, by reversely thinking this unique perfect matching algorithm, we could construct a graph with unique perfect matching from empty through iteratively adding odd bridges and edges incident to odd bridges one by one back to the graph. Chaplick *et al* provided a constructive characterization of the claw-free graphs with a unique perfect matching in [8]. In section 4.4, we propose an algorithm to construct general graphs with a unique perfect matching through odd bridges. In section 4.5, we present a forcing face construction algorithm based on the proposed unique perfect matching construction algorithm.

4.2 Forcing Face Detection Algorithm

Let G be a plane graph. A finite face f of G is a forcing face if the remaining graph $G - V(f)$ has a unique perfect matching. By the definition, a straightforward forcing face detection algorithm for a face f in G is to check if there is a unique perfect matching on $G - V(f)$ using algorithm 4. In order to make this straightforward algorithm more efficient, we need to detect the new bridges generated by deleting f and edges incident to f. In Algorithm 4, the key observation in the for loop is that every bridge newly created by deletion of an edge $\{v, w\}$ is on every path from v to w. This observation is still partially true in our forcing face detection algorithm, since we have another observation that some other new bridges are created by deletion of the face f and ever such newly created bridge is on every path from one vertex x on f to another vertex y on f. The graphical example is showed in Figure 24, where the solid circle represents the face f.

The pseudocode of this algorithm is as following:

Algorithm 5: Forcing Face Detection Algorithm

```
Initialize $M = \emptyset$, $R$ to be the set of all bridges of $G$, and
$C$ to be the set of all inner faces of $G$.

while $C \neq \emptyset$ do:
    Restore $G$ and $R$ as the original unchanged $G$ and $R$.
    Delete a face $f$ from $C$.
    Delete all edges of this face from $G$, and
```

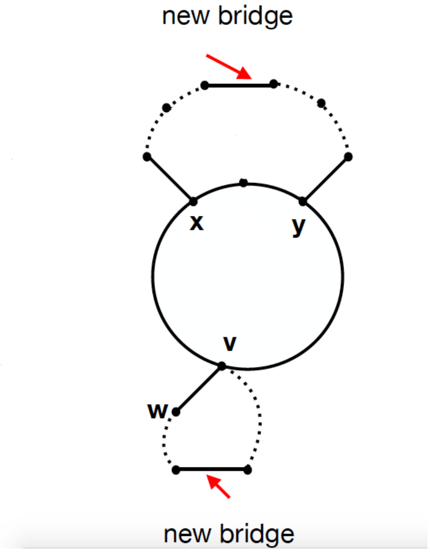


Figure 24: Forcing face algorithm

```
    from $R$ if the edge is in $R$.
for every pair of vertices $x$ and $y$ on this face do
    if $x$ and $y$ are still connected but
    are in different 2-edge-connected components,
    then find a path $P(x,y)$ connecting $x$ and $y$ and
    add every bridge on this path to $R$,
    except the edges incident to this face.
end
for each edge $v,w$ incident to this face do
    Delete $v,w$ from $G$, and from $R$ if it is in $R$.
    if $v$ and $w$ are still connected but
    are in different 2-edge-connected components,
    then find a path $P(v,w)$ connecting $v$ and $w$ and
    add every bridge on this path to $R$.
end
Remove duplication of newly added bridges from $R$.
```

```
    Use unique perfect matching algorithm to
        find a unique perfect matching
        on the remaining graph $G$.
end
```

We have following lemmas to prove the correctness of our proposed forcing face algorithm.

Lemma 4.14. *In the step of the first for loop in Algorithm 5, every bridge newly created by deletion of a face is on every path from x to y, where x and y are a pair of vertices on this face.*

Proof. Suppose the newly created bridge is not on any path containing any pair of vertices of this face. We know that there is no circle in this graph containing this bridge after the deletion. Adding this face back to the graph will not form a circle containing this bridge. This means that this bridge should exist before deleting this face. A contradiction is found and thus this claim is correct. □

Lemma 4.15. *In the step of the second for loop in Algorithm 5, every bridge newly created by deletion of an edge v, w is on every path from v to w.*

Proof. Suppose the newly created bridge in the second for loop is not on any path from v to w. We know that there is no circle in this graph containing this bridge after the deletion. Adding edge v, w back to the graph will not form a circle containing this bridge since this bridge is not on any path from

v to w. This means that this bridge should exist before deleting edge v, w. A contradiction is found and thus this claim is correct. □

Lemma 4.16. *Let R be the set of all bridges of a graph G. R contains all remaining bridges of G after deleting a face with all its incident edges in this algorithm.*

Proof. A bridge will remain a bridge once it is created until it is deleted since no new edges and vertices will be added to G during the whole algorithm. Based on Lemma 4.14 and 4.15, every newly created bridges will be added to R. A newly created bridge is not a bridge before the deletion of face or edges since this bridge is on a circle of G before the deletion. The second last step in the while loop of this algorithm removes all duplication of newly added bridges from R. Based on all above, we can conclude that R contains all remaining bridges of G after deleting a face with all its incident edges in this algorithm.

□

Now we show that Algorithm 5 runs in time $O(m^2\text{log}^4 n)$. Based on paper [22], it will take $O(m\text{log}^4 n)$ to identify if there is an unique perfect matching in a graph with m edges and n vertices. The number of faces in a graph is less than the number of edge m, so the Algorithm 5 will run in time $O(m^2\text{log}^4 n)$.

4.3 Implementation Result of Forcing Face Detection Algorithm

A non-linear hexagonal system has 0, 1, or 2 forcing face , while a linear hexagonal has all hexagons as forcing faces [11]. Figure 25 shows a linear hexagonal system with 5 hexagons. All five hexagons are forcing faces. Figure 26 shows one forcing face of this linear hexagonal system with the edges of the unique perfect matching labeled using green lines after this forcing face deleted.

Figure 28 shows a hexagonal system with only one forcing face. Figure 27 shows a hexagonal system with only two forcing faces.

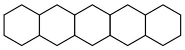


Figure 25: A linear hexagonal system

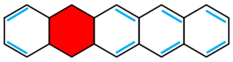


Figure 26: A forcing face in a linear hexagonal system

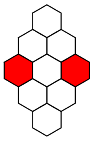


Figure 27: A hexagonal system with two forcing faces

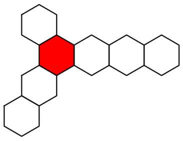


Figure 28: A hexagonal system with one forcing face

4.4 Unique Perfect Matching Construction Algorithm

First of all, we define a series of terminologies and operations before discussing the algorithm. We define an *element* as a combination of a letter L or R and a positive integer as subscript. A letter S is also an element. For example, S, L_1 and R_2 are all elements. We define a *sequence* as a list of elements starting from S and S only appear once in the list. For example, S, SL_1, and $SR_1L_1L_2R_1L_3$ are all sequences. We define following operations on a sequence s:

- $|s|$: the number of elements, length of s
- $s[n]$: the nth element in s
- $s[:n]$: a *subsequence* of s from the first element to the nth element

- Letter($s[n]$): the letter of the nth element in s

For example, given a sequence $s = SR_1L_1L_2R_1L_3$, $|s| = 6$, $s[2] = R_1$, Letter($s[4]$) $= L$, and $s[: 3] = SR_1L_1$, so we can say SR_1L_1 is a subsequence of s.

Let s_1, s_2 be two sequences. We call s_1 is an *ancestor* of s_2 (or s_2 is a *descendant* of s_1 otherwise) if s_1 is a subsequence of s_2. For example, S, SR_1, and $SR_1L_1L_2R_1$ are all ancestors of $SR_1L_1L_2R_1L_3$.

We define an operation vertex(s_1, s_2) between ancestor s_1 and descendant s_2 as vertex(s_1, s_2) $=$ Letter($s_2[|s_1| + 1]$). In other words, vertex(s_1, s_2) is the letter after s_1 on s_2. For example, vertex($SR_1L_1L_2$, $SR_1L_1L_2R_1L_3$) is R.

Now we can follow these steps to construct a connected graph G with unique perfect matching M:

- Step 1, add two vertices x, y and an edge xy to an empty graph G. Assign sequence S to edge xy. Add edge xy to an empty set M. Name x as left vertex and name y as right vertex.
- Step 2, add two vertices v, w and an edge vw to G. Assign a sequence s_{vw} to edge vw following these two substeps:
    - Step 2.1, choose one edge e from M. Suppose s is the sequence assigned to e.

  - Step 2.2, append an element L_k or R_k to sequence s as s_{vw}, where k is a positive integer and to make sure that the sequence s_{vw} should not have been assigned to any other edges before.

  Add edge vw to M. Name v as left vertex and name w as right vertex.

- Step 3, choose an edge m from M such that the sequence s_m of m is ancestor of s_{vw}. Add one edge from v or w to a vertex p, where p is the left vertex of m if vertex(s_m, s_{vw}) is L and is the right vertex of m otherwise. Add this new edge to G.

- Repeat Step 3 to add more edges to G.

- Go back to Step 2.

Let's see one example in Figure 29 constructed by this method as following:

1. Add edge 1 with sequence S to M and to G. We call the vertex on the left/right position as left/right vertex.

2. Add edge 2 to G. Assign a sequence SL_1 based on edge 1. Add edge 2 to M.

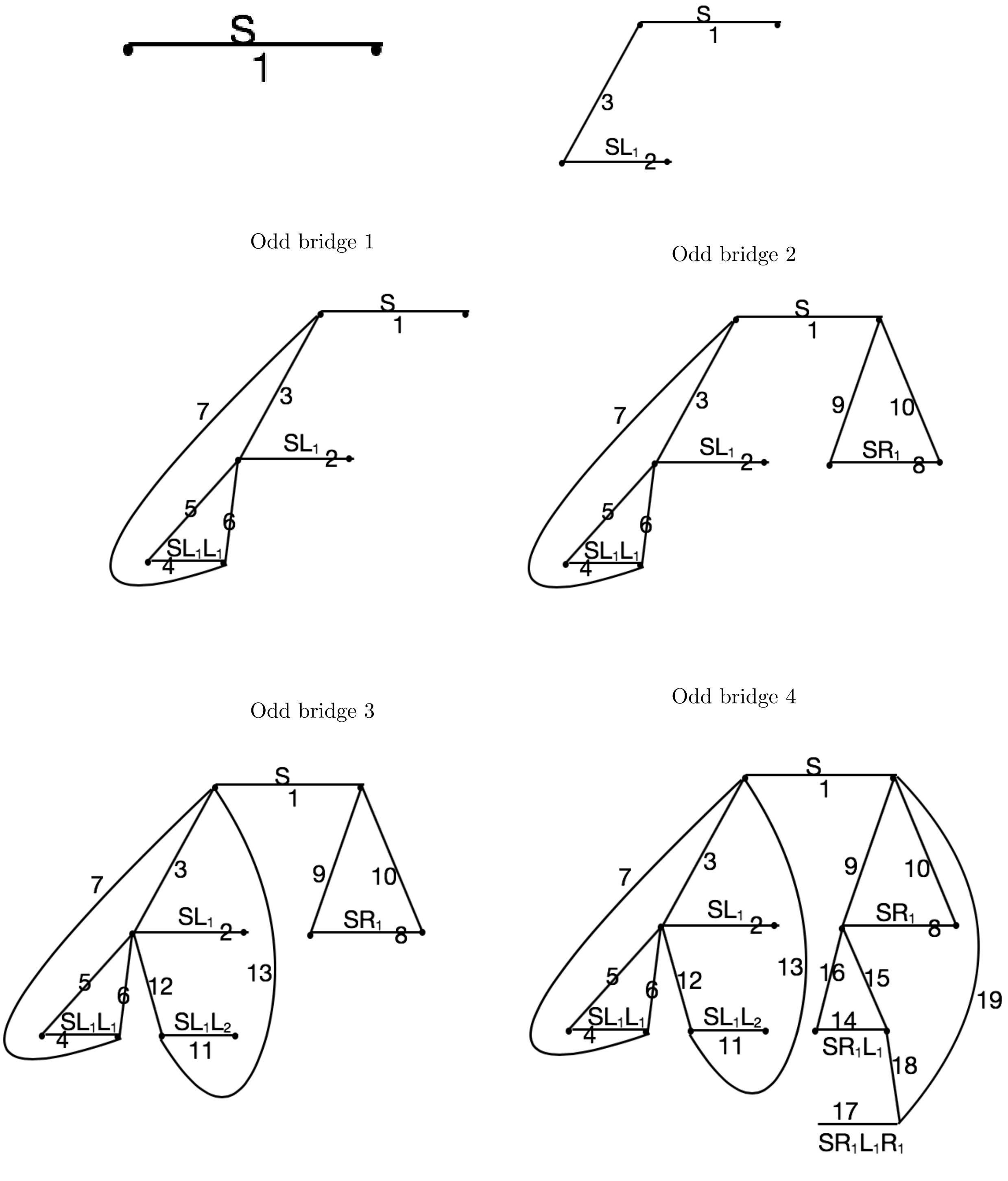
3. Add edge 3 from one endpoint of edge 2 to the left vertex of edge 1, because the sequence of edge 1 is an ancestor of the sequence of edge 2 and vertex(S, SL_1) $= L$. Add edge 3 to G.

4. Add edge 4 to G. Assign a sequence SL_1L_1 to edge 4 based on edge 2. Add edge 4 to M.
5. Add edge 5, 6, 7 from edge 4 to its ancestors. Add edge 5, 6, 7 to G.
6. Add edge 8 to G and then to M. Add edge 9, 10 to G.
7. Continue to add edges until edge 19 is added.

We prove the correctness of our method as following.

Lemma 4.17. *A graph G constructed by above algorithm has a unique perfect matching.*

Proof. Firstly, we prove that the initial edge xy with sequence S is an odd bridge. Based on the operation vertex(), all the edges connected to the left vertex of xy have a sequence starting as SL and all the edges connected to the right vertex of xy have a sequence starting as SR. A sequence starting from SL does not have ancestor/descendant relationship with a sequence starting from SR. So there is no edge between the group of edges connected to the x (left side) and the the group of edges connected to the y (right side). All other edges without a sequence assigned are connected between two vertices in one side. So edge xy is a bridge.



Odd bridge 1

Odd bridge 2

Odd bridge 3

Odd bridge 4

Odd bridge 5

Odd bridge 6 & 7

Figure 29: Example of construction of a graph with unique perfect matching

Throughout this algorithm, we always add a pair of new vertices v and w in Step 2. The edge vw will be connected to only one vertex of the edge xy. So the vertices in either side is an odd number. So, the edge xy is an odd bridge.

Based on the unique perfect matching algorithm, we delete the edge xy and all edges incident to xy. Now we prove that all edges with length 2 sequences are odd bridges. For instance, we have a bridge vw with sequence SL_k. All the edges from vw to its ancestor xy has been deleted. So, we can recursively deem SL_k like the initial edge S to build all its descendants. This is saying that edge vw is an odd bridge.

After all odd bridges with length 2 sequences are removed from the graph. All edges with length 3 sequences are odd bridges. We can recursively repeat these steps and finally all edges with sequences assigned are deleted, and consequently all edges without sequences assigned are also deleted. Thus the final graph is empty. We get a unique perfect matching M of G. □

In Step 2 of our method, it needs $O(1)$ time to add an edge with sequence assigned. In Step 3 of our method, it also needs $O(1)$ time to add an edge without sequence assigned. So, the method overall needs $O(m)$ time to construct a graph with m edges and a unique perfect matching.

4.5 Forcing Face Construction Algorithm

A face f of a plan graph G is a forcing face if $G - V(f)$ has a unique perfect matching. In section 4.4, we present an algorithm to construct graphs with a unique perfect matching. So, we can intuitively construct a graph with a forcing face by adding a face to a graph with a unique perfect matching.

We can use an algorithm of Galil *et al* [23] to test the planarity of the graph constructed in the algorithm. Galil *et al* algorithm can test whether an edge could be added to the graph without violating planarity in time $O(n^{2/3})$, where n is the number of vertices of G.

Here are the steps to construct a graph G with a forcing face f:

- Step 1, construct a graph G with a unique perfect matching based on the algorithm in section 4.4. Galil *et al* algorithm is applied in Step 3 of the algorithm in section 4.4 to maintain the planarity of G.
- Step 2, add an edge between one vertex of f and one vertex of G. Galil *et al* algorithm is applied after adding this edge to maintain the planarity of G.
- Repeat Step 2 to add more edges.

CHAPTER 5

FUTURE WORK

In chapter 2, we only calculate the forcing polynomial for cata-condensed benzenoid system. In the future, we want to design an algorithm for calculating all benzenoid systems. We also want to build relationships between forcing resonant polynomial and other counting polynomials such as Clar polynomials. For example, a forcing resonant set is always maximal, but a maximal resonant set may be not forcing if the benzenoid system is not cata-condensed. We want to find a sufficient and necessary condition between maximal and forcing of a resonant set.

In chapter 3, we only propose a method to enumerate all cata-condensed benzenoid systems with maximum Clar number $n/6$, where n is the number of vertices of benzenoid systems. As a future work, we will continue to design methods to discover hexagonal systems with maximum Clar number as $(n-2)/6$ and $(n-4)/6$. Our current enumeration algorithm is based on the results obtained from Brinkmann et al. algorithm [7]. Our algorithm only works for small number of hexagons. In the future, we would like to enumerate such benzenoid systems attaining the upper bound directly, instead of enumerating all benzenoid systems at first.

We also can observe that all cata-condensed benzenoid systems with maximum Clar number $n/6$ are comprised of ”star” shaped structures. So, we may define this structure as a basic unit just like a single hexagon. And then we can build all systems by combing multiple such basic units. For example, in Fugure 21, two cata-condensed benzenoid system of 7 hexagons are combined by two basic star units in two different ways. The core of this kind of algorithm is to detect if two graphs are isomorphism or not.

In chapter 4, we prove that our proposed forcing face detection algorithm runs in time $O(m^2\log^4 n)$, but we could implement this algorithm using a much more efficient dynamical data structure - top tree. Top tree can be used to maintain many information of a tree dynamically when updating the tree through adding, deleting, etc. The idea of top tree is to represent a tree using clusters, where each cluster is a subtree. The implementation in chapter 4 is not coded in top tree structure. In the future, we will implement our algorithm using top tree.

In chapter 4, we also propose another algorithm to construct graphs with unique perfect matchings using odd bridges. We prove that every graph constructed by our algorithm will have a unique prefect matching. In the future, we would like to prove a conjecture that every graph with a unique perfect matching can be constructed by our algorithm.

In the last section of chapter 4, we present a forcing face construction algorithm. In the future, we want to construct graphs according to their forcing resonant polynomials.

APPENDIX

Table 3: All cata-condensed Benzenoid Systems with 7 hexagons

Graph index	Polynomial	Clar #	Coefficient vector	x	HOMO-LUMO gap	Number of perfect matching
1	$x^5 + 2x^3 + x^2$	5	[1, 0, 2, 1, 0]	39	1.1801	41
2	$x^5 + 2x^3$	5	[1, 0, 2, 0, 0]	38	1.1548	40
3	$3x^4 + 2x^3$	4	[0, 3, 2, 0, 0]	37	1.0255	38
4	$3x^4 + 2x^3$	4	[0, 3, 2, 0, 0]	37	0.941	38
5	$3x^4 + x^3 + x^2$	4	[0, 3, 1, 1, 0]	36	1.0356	37
6	$3x^4 + x^3 + x^2$	4	[0, 3, 1, 1, 0]	36	0.9799	37
7	$3x^4 + x^3 + x^2$	4	[0, 3, 1, 1, 0]	36	0.9698	37
8	$3x^4 + 2x^2$	4	[0, 3, 0, 2, 0]	35	0.979	35
9	$3x^4 + 2x^2$	4	[0, 3, 0, 2, 0]	35	0.9268	35
10	$2x^4 + 4x^3$	4	[0, 2, 4, 0, 0]	34	1.0431	36
11	$2x^4 + 4x^3$	4	[0, 2, 4, 0, 0]	34	1.0406	36
12	$2x^4 + 4x^3$	4	[0, 2, 4, 0, 0]	34	1.0291	36
13	$2x^4 + 4x^3$	4	[0, 2, 4, 0, 0]	34	1.0242	36
14	$2x^4 + 4x^3$	4	[0, 2, 4, 0, 0]	34	0.9796	36
15	$2x^4 + 4x^3$	4	[0, 2, 4, 0, 0]	34	0.9782	36
16	$2x^4 + 4x^3$	4	[0, 2, 4, 0, 0]	34	0.9371	36
17	$2x^4 + 3x^3$	4	[0, 2, 3, 0, 0]	33	0.9592	34
18	$2x^4 + 3x^3$	4	[0, 2, 3, 0, 0]	33	0.9424	34
19	$2x^4 + 3x^3$	4	[0, 2, 3, 0, 0]	33	0.9281	34
20	$2x^4 + 3x^3$	4	[0, 2, 3, 0, 0]	33	0.9163	34
21	$2x^4 + 3x^3$	4	[0, 2, 3, 0, 0]	33	0.8499	34
22	$2x^4 + 2x^3 + 2x^2$	4	[0, 2, 2, 2, 0]	32	0.926	33
23	$2x^4 + 2x^3 + 2x^2$	4	[0, 2, 2, 2, 0]	32	0.9204	33
24	$2x^4 + 2x^3 + 2x^2$	4	[0, 2, 2, 2, 0]	32	0.9142	33
25	$2x^4 + 2x^3 + 2x^2$	4	[0, 2, 2, 2, 0]	32	0.9141	33
26	$2x^4 + 2x^3 + x^2$	4	[0, 2, 2, 1, 0]	31	0.9832	32
27	$2x^4 + 2x^3 + x^2$	4	[0, 2, 2, 1, 0]	31	0.9722	32

Table 3: Continued.

28	$2x^4 + 2x^3 + x^2$	4	[0, 2, 2, 1, 0]	31	0.8561	33
29	$2x^4 + 2x^3 + x^2$	4	[0, 2, 2, 1, 0]	31	0.8521	33
30	$2x^4 + x^3 + 3x^2$	4	[0, 2, 1, 3, 0]	30	0.8671	32
31	$2x^4 + x^3 + 3x^2$	4	[0, 2, 1, 3, 0]	30	0.8575	32
32	$2x^4 + x^3 + 2x^2$	4	[0, 2, 1, 2, 0]	29	0.8781	31
33	$2x^4 + x^3 + x^2$	4	[0, 2, 1, 1, 0]	28	0.78	30
34	$x^4 + 7x^3$	4	[0, 1, 7, 0, 0]	27	1.0559	35
35	$x^4 + 7x^3$	4	[0, 1, 7, 0, 0]	27	0.9604	35
36	$x^4 + 6x^3$	4	[0, 1, 6, 0, 0]	26	1.0548	34
37	$x^4 + 6x^3$	4	[0, 1, 6, 0, 0]	26	1.0361	34
38	$x^4 + 6x^3$	4	[0, 1, 6, 0, 0]	26	1.0223	34
39	$x^4 + 6x^3$	4	[0, 1, 6, 0, 0]	26	1.0063	34
40	$x^4 + 6x^3$	4	[0, 1, 6, 0, 0]	26	0.9927	34
41	$x^4 + 6x^3$	4	[0, 1, 6, 0, 0]	26	0.9681	34
42	$x^4 + 6x^3$	4	[0, 1, 6, 0, 0]	26	0.963	34
43	$x^4 + 6x^3$	4	[0, 1, 6, 0, 0]	26	0.9165	34
44	$x^4 + 4x^3 + x^2$	4	[0, 1, 4, 1, 0]	25	0.9317	31
45	$x^4 + 4x^3 + x^2$	4	[0, 1, 4, 1, 0]	25	0.9243	31
46	$x^4 + 4x^3 + x^2$	4	[0, 1, 4, 1, 0]	25	0.9068	31
47	$x^4 + 4x^3 + x^2$	4	[0, 1, 4, 1, 0]	25	0.9063	31
48	$x^4 + 4x^3 + x^2$	4	[0, 1, 4, 1, 0]	25	0.8909	31
49	$x^4 + 4x^3 + x^2$	4	[0, 1, 4, 1, 0]	25	0.8886	31
50	$x^4 + 4x^3 + x^2$	4	[0, 1, 4, 1, 0]	25	0.8885	31
51	$x^4 + 4x^3 + x^2$	4	[0, 1, 4, 1, 0]	25	0.8624	31
52	$x^4 + 3x^3 + x^2$	4	[0, 1, 3, 1, 0]	24	0.9667	29
53	$x^4 + 3x^3 + x^2$	4	[0, 1, 3, 1, 0]	24	0.9077	29
54	$x^4 + 3x^3 + x^2$	4	[0, 1, 3, 1, 0]	24	0.8642	29
55	$9x^3$	3	[0, 0, 9, 0, 0]	23	0.8438	31
56	$9x^3$	3	[0, 0, 9, 0, 0]	23	0.8216	31
57	$9x^3$	3	[0, 0, 9, 0, 0]	23	0.7955	31
58	$8x^3 + x^2$	3	[0, 0, 8, 1, 0]	22	0.8666	29
59	$8x^3 + x$	3	[0, 0, 8, 0, 1]	21	1.0314	28
60	$8x^3$	3	[0, 0, 8, 0, 0]	20	0.8439	30
61	$8x^3$	3	[0, 0, 8, 0, 0]	20	0.8439	30
62	$8x^3$	3	[0, 0, 8, 0, 0]	20	0.818	30

Table 3: Continued.

63	$8x^3$	3	[0, 0, 8, 0, 0]	20	0.8173	30
64	$8x^3$	3	[0, 0, 8, 0, 0]	20	0.8074	30
65	$8x^3$	3	[0, 0, 8, 0, 0]	20	0.8033	30
66	$8x^3$	3	[0, 0, 8, 0, 0]	20	0.7956	30
67	$8x^3$	3	[0, 0, 8, 0, 0]	20	0.7907	30
68	$8x^3$	3	[0, 0, 8, 0, 0]	20	0.7889	30
69	$8x^3$	3	[0, 0, 8, 0, 0]	20	0.7619	30
70	$7x^3 + x^2$	3	[0, 0, 7, 1, 0]	19	0.8371	29
71	$7x^3 + x^2$	3	[0, 0, 7, 1, 0]	19	0.8363	29
72	$7x^3 + x^2$	3	[0, 0, 7, 1, 0]	19	0.8155	29
73	$7x^3 + x^2$	3	[0, 0, 7, 1, 0]	19	0.8128	29
74	$7x^3 + x^2$	3	[0, 0, 7, 1, 0]	19	0.8121	29
75	$7x^3 + x^2$	3	[0, 0, 7, 1, 0]	19	0.8084	29
76	$7x^3 + x^2$	3	[0, 0, 7, 1, 0]	19	0.7957	29
77	$6x^3 + 2x^2$	3	[0, 0, 6, 2, 0]	18	0.8738	27
78	$6x^3 + 2x^2$	3	[0, 0, 6, 2, 0]	18	0.8632	27
79	$6x^3 + 2x^2$	3	[0, 0, 6, 2, 0]	18	0.8622	27
80	$6x^3 + 2x^2$	3	[0, 0, 6, 2, 0]	18	0.8274	27
81	$6x^3 + 2x^2$	3	[0, 0, 6, 2, 0]	18	0.7298	27
82	$6x^3 + 2x^2$	3	[0, 0, 6, 2, 0]	18	0.7288	27
83	$6x^3 + x^2$	3	[0, 0, 6, 1, 0]	17	0.6724	26
84	$6x^3 + x^2$	3	[0, 0, 6, 1, 0]	17	0.6694	26
85	$6x^3 + x$	3	[0, 0, 6, 0, 1]	16	0.7266	25
86	$5x^3 + 3x^2$	3	[0, 0, 5, 3, 0]	15	0.7629	26
87	$5x^3 + 3x^2$	3	[0, 0, 5, 3, 0]	15	0.7558	26
88	$5x^3 + 3x^2$	3	[0, 0, 5, 3, 0]	15	0.7361	26
89	$5x^3 + 3x^2$	3	[0, 0, 5, 3, 0]	15	0.7303	26
90	$5x^3 + 2x^2$	3	[0, 0, 5, 2, 0]	14	0.6756	25
91	$5x^3 + 2x^2$	3	[0, 0, 5, 2, 0]	14	0.6751	25
92	$5x^3 + 2x^2$	3	[0, 0, 5, 2, 0]	14	0.6664	25
93	$5x^3 + 2x^2$	3	[0, 0, 5, 2, 0]	14	0.663	25
94	$4x^3 + 5x^2$	3	[0, 0, 4, 5, 0]	13	0.8412	25
95	$4x^3 + 5x^2$	3	[0, 0, 4, 5, 0]	13	0.7927	25
96	$4x^3 + 5x^2$	3	[0, 0, 4, 5, 0]	13	0.7881	25
97	$4x^3 + 4x^2$	3	[0, 0, 4, 4, 0]	12	0.8284	24

Table 3: Continued.

98	$4x^3 + 4x^2$	3	[0, 0, 4, 4, 0]	12	0.7881	24
99	$4x^3 + 3x^2$	3	[0, 0, 4, 3, 0]	11	0.7383	23
100	$4x^3 + 3x^2$	3	[0, 0, 4, 3, 0]	11	0.7281	23
101	$4x^3 + 3x^2$	3	[0, 0, 4, 3, 0]	11	0.5871	24
102	$4x^3 + x$	3	[0, 0, 4, 0, 1]	10	0.5242	21
103	$3x^3 + 5x^2$	3	[0, 0, 3, 5, 0]	9	0.6286	23
104	$3x^3 + 5x^2$	3	[0, 0, 3, 5, 0]	9	0.6228	23
105	$3x^3 + 5x^2$	3	[0, 0, 3, 5, 0]	9	0.6216	23
106	$3x^3 + 5x^2$	3	[0, 0, 3, 5, 0]	9	0.6199	23
107	$3x^3 + 4x^2$	3	[0, 0, 3, 4, 0]	8	0.6603	22
108	$3x^3 + 4x^2$	3	[0, 0, 3, 4, 0]	8	0.6544	22
109	$3x^3 + 2x^2$	3	[0, 0, 3, 2, 0]	7	0.539	20
110	$3x^3 + 2x^2$	3	[0, 0, 3, 2, 0]	7	0.5381	20
111	$11x^2$	2	[0, 0, 0, 11, 0]	6	0.5698	19
112	$11x^2$	2	[0, 0, 0, 11, 0]	6	0.5597	19
113	$9x^2 + x$	2	[0, 0, 0, 9, 1]	5	0.6512	17
114	$9x^2$	2	[0, 0, 0, 9, 0]	4	0.4623	17
115	$9x^2$	2	[0, 0, 0, 9, 0]	4	0.4598	17
116	$8x^2 + x$	2	[0, 0, 0, 8, 1]	3	0.5021	16
117	$5x^2 + x$	2	[0, 0, 0, 5, 1]	2	0.3741	13
118	$7x$	1	[0, 0, 0, 0, 7]	1	0.2684	8

Table 4: All cata-condensed Benzenoid Systems with 8 hexagons

Graph index	Polynomial	Clar #	Coefficient vector	x	HOMO-LUMO gap	Number of perfect matching
1	$2x^5 + 2x^4 + x^3$	5	[2, 2, 1, 0, 0]	89	0.961	66
2	$2x^5 + 2x^4 + x^2$	5	[2, 2, 0, 1, 0]	88	0.982	65
3	$2x^5 + x^4 + 3x^3$	5	[2, 1, 3, 0, 0]	87	1.0215	64
4	$2x^5 + x^4 + 3x^3$	5	[2, 1, 3, 0, 0]	87	1.0136	64
5	$2x^5 + x^4 + 3x^3$	5	[2, 1, 3, 0, 0]	87	1.0066	64
6	$2x^5 + x^4 + 2x^3$	5	[2, 1, 2, 0, 0]	86	0.9875	62
7	$2x^5 + x^4 + 2x^3$	5	[2, 1, 2, 0, 0]	86	0.9824	62
8	$2x^5 + x^4 + x^3 + x^2$	5	[2, 1, 1, 1, 0]	85	0.9233	61
9	$2x^5 + 3x^3 + x^2$	5	[2, 0, 3, 1, 0]	84	0.9509	59
10	$2x^5 + 3x^3 + x^2$	5	[2, 0, 3, 1, 0]	84	0.9495	59
11	$2x^5 + 3x^3$	5	[2, 0, 3, 0, 0]	83	0.9738	58
12	$2x^5 + 2x^3$	5	[2, 0, 2, 0, 0]	82	0.8495	56
13	$x^5 + 4x^4 + x^3$	5	[1, 4, 1, 0, 0]	81	1.0078	62
14	$x^5 + 4x^4 + x^3$	5	[1, 4, 1, 0, 0]	81	0.9698	62
15	$x^5 + 4x^4 + x^3$	5	[1, 4, 1, 0, 0]	81	0.9216	62
16	$x^5 + 3x^4 + 4x^3$	5	[1, 3, 4, 0, 0]	80	1.0015	61
17	$x^5 + 3x^4 + 4x^3$	5	[1, 3, 4, 0, 0]	80	0.9877	61
18	$x^5 + 3x^4 + 4x^3$	5	[1, 3, 4, 0, 0]	80	0.9312	61
19	$x^5 + 3x^4 + 3x^3$	5	[1, 3, 3, 0, 0]	79	1.0502	60
20	$x^5 + 3x^4 + 3x^3$	5	[1, 3, 3, 0, 0]	79	1.0202	60
21	$x^5 + 3x^4 + 3x^3$	5	[1, 3, 3, 0, 0]	79	1.0091	60
22	$x^5 + 3x^4 + 3x^3$	5	[1, 3, 3, 0, 0]	79	0.9994	60
23	$x^5 + 3x^4 + 3x^3$	5	[1, 3, 3, 0, 0]	79	0.9633	60
24	$x^5 + 3x^4 + 3x^3$	5	[1, 3, 3, 0, 0]	79	0.9494	60
25	$x^5 + 3x^4 + x^3 + x^2$	5	[1, 3, 1, 1, 0]	78	1.015	57
26	$x^5 + 3x^4 + x^3 + x^2$	5	[1, 3, 1, 1, 0]	78	0.9664	57
27	$x^5 + 3x^4 + x^3 + x^2$	5	[1, 3, 1, 1, 0]	78	0.9618	57
28	$x^5 + 3x^4 + x^3 + x^2$	5	[1, 3, 1, 1, 0]	78	0.9256	57
29	$x^5 + 3x^4 + x^3$	5	[1, 3, 1, 0, 0]	77	0.9102	58
30	$x^5 + 3x^4 + x^3$	5	[1, 3, 1, 0, 0]	77	0.88	58

Table 4: Continued.

31	$x^5 + 2x^4 + 5x^3$	5	[1, 2, 5, 0, 0]	76	1.0588	59
32	$x^5 + 2x^4 + 5x^3$	5	[1, 2, 5, 0, 0]	76	1.0481	59
33	$x^5 + 2x^4 + 5x^3$	5	[1, 2, 5, 0, 0]	76	1.0217	59
34	$x^5 + 2x^4 + 5x^3$	5	[1, 2, 5, 0, 0]	76	1.0216	59
35	$x^5 + 2x^4 + 5x^3$	5	[1, 2, 5, 0, 0]	76	1.0122	59
36	$x^5 + 2x^4 + 5x^3$	5	[1, 2, 5, 0, 0]	76	1.0096	59
37	$x^5 + 2x^4 + 5x^3$	5	[1, 2, 5, 0, 0]	76	0.9737	59
38	$x^5 + 2x^4 + 5x^3$	5	[1, 2, 5, 0, 0]	76	0.9461	59
39	$x^5 + 2x^4 + 3x^3$	5	[1, 2, 3, 0, 0]	75	0.9968	56
40	$x^5 + 2x^4 + 3x^3$	5	[1, 2, 3, 0, 0]	75	0.9729	56
41	$x^5 + 2x^4 + 3x^3$	5	[1, 2, 3, 0, 0]	75	0.9711	56
42	$x^5 + 2x^4 + 3x^3$	5	[1, 2, 3, 0, 0]	75	0.9645	56
43	$x^5 + 2x^4 + 3x^3$	5	[1, 2, 3, 0, 0]	75	0.9632	56
44	$x^5 + 2x^4 + 3x^3$	5	[1, 2, 3, 0, 0]	75	0.9463	56
45	$x^5 + 2x^4 + 3x^3$	5	[1, 2, 3, 0, 0]	75	0.9455	56
46	$x^5 + 2x^4 + 3x^3$	5	[1, 2, 3, 0, 0]	75	0.944	56
47	$x^5 + 2x^4 + 2x^3 + x^2$	5	[1, 2, 2, 1, 0]	74	0.9395	55
48	$x^5 + 2x^4 + 2x^3 + x^2$	5	[1, 2, 2, 1, 0]	74	0.9299	55
49	$x^5 + 2x^4 + 2x^3 + x^2$	5	[1, 2, 2, 1, 0]	74	0.9115	55
50	$x^5 + 2x^4 + 2x^3 + x^2$	5	[1, 2, 2, 1, 0]	74	0.8979	55
51	$x^5 + 2x^4 + 2x^3$	5	[1, 2, 2, 0, 0]	73	1.0218	54
52	$x^5 + 2x^4 + 2x^3$	5	[1, 2, 2, 0, 0]	73	0.9822	54
53	$x^5 + 2x^4 + 2x^3$	5	[1, 2, 2, 0, 0]	73	0.9806	54
54	$x^5 + 2x^4 + x^3 + x^2$	5	[1, 2, 1, 1, 0]	72	0.9493	53
55	$x^5 + 2x^4 + x^3 + x^2$	5	[1, 2, 1, 1, 0]	72	0.9048	53
56	$8x^4$	4	[0, 8, 0, 0, 0]	71	1.045	60
57	$8x^4$	4	[0, 8, 0, 0, 0]	71	0.9882	60
58	$8x^4$	4	[0, 8, 0, 0, 0]	71	0.9658	60
59	$8x^4$	4	[0, 8, 0, 0, 0]	71	0.9265	60
60	$8x^4$	4	[0, 8, 0, 0, 0]	71	0.8678	60
61	$7x^4 + x^3$	4	[0, 7, 1, 0, 0]	70	1.0467	58
62	$7x^4 + x^3$	4	[0, 7, 1, 0, 0]	70	0.9893	58
63	$7x^4 + x^3$	4	[0, 7, 1, 0, 0]	70	0.9863	58
64	$7x^4 + x^3$	4	[0, 7, 1, 0, 0]	70	0.984	58
65	$7x^4 + x^3$	4	[0, 7, 1, 0, 0]	70	0.9794	58

Table 4: Continued.

66	$7x^4 + x^3$	4	[0, 7, 1, 0, 0]	70	0.9774	58
67	$7x^4 + x^3$	4	[0, 7, 1, 0, 0]	70	0.9708	58
68	$7x^4 + x^3$	4	[0, 7, 1, 0, 0]	70	0.9349	58
69	$7x^4 + x^3$	4	[0, 7, 1, 0, 0]	70	0.9346	58
70	$7x^4 + x^3$	4	[0, 7, 1, 0, 0]	70	0.9316	58
71	$7x^4 + x^3$	4	[0, 7, 1, 0, 0]	70	0.8896	58
72	$6x^4 + 3x^3$	4	[0, 6, 3, 0, 0]	69	0.9789	57
73	$6x^4 + 3x^3$	4	[0, 6, 3, 0, 0]	69	0.9767	57
74	$6x^4 + 3x^3$	4	[0, 6, 3, 0, 0]	69	0.9631	57
75	$6x^4 + 3x^3$	4	[0, 6, 3, 0, 0]	69	0.9375	57
76	$6x^4 + 3x^3$	4	[0, 6, 3, 0, 0]	69	0.9263	57
77	$6x^4 + 3x^3$	4	[0, 6, 3, 0, 0]	69	0.9125	57
78	$6x^4 + 2x^3$	4	[0, 6, 2, 0, 0]	68	0.8903	54
79	$6x^4 + 2x^3$	4	[0, 6, 2, 0, 0]	68	0.8903	54
80	$6x^4 + 2x^3$	4	[0, 6, 2, 0, 0]	68	0.8899	54
81	$6x^4 + 2x^3$	4	[0, 6, 2, 0, 0]	68	0.8898	54
82	$6x^4 + 2x^3$	4	[0, 6, 2, 0, 0]	68	0.8743	54
83	$6x^4 + 2x^3$	4	[0, 6, 2, 0, 0]	68	0.8661	54
84	$6x^4 + 2x^3$	4	[0, 6, 2, 0, 0]	68	0.8658	54
85	$6x^4 + 2x^3$	4	[0, 6, 2, 0, 0]	68	0.8644	54
86	$6x^4 + 2x^3$	4	[0, 6, 2, 0, 0]	68	0.858	54
87	$6x^4 + 2x^3$	4	[0, 6, 2, 0, 0]	68	0.8501	54
88	$6x^4 + 2x^3$	4	[0, 6, 2, 0, 0]	68	0.8252	54
89	$6x^4 + 2x^3$	4	[0, 6, 2, 0, 0]	68	0.8231	54
90	$6x^4 + x^3 + x^2$	4	[0, 6, 1, 1, 0]	67	0.8927	53
91	$6x^4 + x^3 + x^2$	4	[0, 6, 1, 1, 0]	67	0.8901	53
92	$6x^4 + x^3 + x^2$	4	[0, 6, 1, 1, 0]	67	0.8763	53
93	$6x^4 + x^3 + x^2$	4	[0, 6, 1, 1, 0]	67	0.8722	53
94	$6x^4 + x^3 + x^2$	4	[0, 6, 1, 1, 0]	67	0.8694	53
95	$6x^4 + x^3 + x^2$	4	[0, 6, 1, 1, 0]	67	0.8688	53
96	$6x^4 + 2x^2$	4	[0, 6, 0, 2, 0]	66	0.9894	51
97	$6x^4 + 2x^2$	4	[0, 6, 0, 2, 0]	66	0.984	51
98	$6x^4 + 2x^2$	4	[0, 6, 0, 2, 0]	66	0.9381	51
99	$6x^4 + 2x^2$	4	[0, 6, 0, 2, 0]	66	0.9289	51
100	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	1.0557	55

Table 4: Continued.

101	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	1.0048	55
102	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.9951	55
103	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.9872	55
104	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.9868	55
105	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.9839	55
106	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.9813	55
107	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.9804	55
108	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.9748	55
109	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.9636	55
110	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.9498	55
111	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.9372	55
112	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.9267	55
113	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.9252	55
114	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.8979	53
115	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.8877	55
116	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.8771	53
117	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.8617	53
118	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.8369	53
119	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.8027	53
120	$5x^4 + 4x^3$	4	[0, 5, 4, 0, 0]	65	0.7731	53
121	$5x^4 + 3x^3$	4	[0, 5, 3, 0, 0]	64	0.8732	52
122	$5x^4 + 3x^3$	4	[0, 5, 3, 0, 0]	64	0.869	52
123	$5x^4 + 3x^3$	4	[0, 5, 3, 0, 0]	64	0.8602	52
124	$5x^4 + 3x^3$	4	[0, 5, 3, 0, 0]	64	0.8507	52
125	$5x^4 + 3x^3$	4	[0, 5, 3, 0, 0]	64	0.8478	52
126	$5x^4 + 3x^3$	4	[0, 5, 3, 0, 0]	64	0.8443	52
127	$5x^4 + 3x^3$	4	[0, 5, 3, 0, 0]	64	0.8187	52
128	$5x^4 + 3x^3$	4	[0, 5, 3, 0, 0]	64	0.8071	52
129	$5x^4 + 3x^3$	4	[0, 5, 3, 0, 0]	64	0.8068	52
130	$5x^4 + 3x^3$	4	[0, 5, 3, 0, 0]	64	0.8022	52
131	$5x^4 + 3x^3$	4	[0, 5, 3, 0, 0]	64	0.802	52
132	$5x^4 + 3x^3$	4	[0, 5, 3, 0, 0]	64	0.7831	52
133	$5x^4 + 3x^3$	4	[0, 5, 3, 0, 0]	64	0.7802	52
134	$5x^4 + 3x^3$	4	[0, 5, 3, 0, 0]	64	0.7733	52
135	$5x^4 + 2x^3 + x^2$	4	[0, 5, 2, 1, 0]	63	0.8239	51

Table 4: Continued.

136	$5x^4 + 2x^3 + x^2$	4	[0, 5, 2, 1, 0]	63	0.7973	51
137	$5x^4 + 2x^3 + x^2$	4	[0, 5, 2, 1, 0]	63	0.7943	51
138	$5x^4 + 3x^2$	4	[0, 5, 0, 3, 0]	62	0.791	48
139	$5x^4 + 3x^2$	4	[0, 5, 0, 3, 0]	62	0.7633	48
140	$5x^4 + 2x^2$	4	[0, 5, 0, 2, 0]	61	0.73	47
141	$5x^4 + 2x^2$	4	[0, 5, 0, 2, 0]	61	0.7151	47
142	$4x^4 + 6x^3$	4	[0, 4, 6, 0, 0]	60	0.8666	51
143	$4x^4 + 6x^3$	4	[0, 4, 6, 0, 0]	60	0.8498	51
144	$4x^4 + 6x^3$	4	[0, 4, 6, 0, 0]	60	0.8468	51
145	$4x^4 + 6x^3$	4	[0, 4, 6, 0, 0]	60	0.8459	51
146	$4x^4 + 6x^3$	4	[0, 4, 6, 0, 0]	60	0.8358	51
147	$4x^4 + 6x^3$	4	[0, 4, 6, 0, 0]	60	0.8263	51
148	$4x^4 + 6x^3$	4	[0, 4, 6, 0, 0]	60	0.8136	51
149	$4x^4 + 5x^3$	4	[0, 4, 5, 0, 0]	59	0.8852	49
150	$4x^4 + 5x^3$	4	[0, 4, 5, 0, 0]	59	0.8835	49
151	$4x^4 + 5x^3$	4	[0, 4, 5, 0, 0]	59	0.8583	49
152	$4x^4 + 5x^3$	4	[0, 4, 5, 0, 0]	59	0.8557	50
153	$4x^4 + 5x^3$	4	[0, 4, 5, 0, 0]	59	0.8486	49
154	$4x^4 + 5x^3$	4	[0, 4, 5, 0, 0]	59	0.8434	50
155	$4x^4 + 5x^3$	4	[0, 4, 5, 0, 0]	59	0.8403	50
156	$4x^4 + 5x^3$	4	[0, 4, 5, 0, 0]	59	0.8362	50
157	$4x^4 + 5x^3$	4	[0, 4, 5, 0, 0]	59	0.8298	50
158	$4x^4 + 5x^3$	4	[0, 4, 5, 0, 0]	59	0.828	50
159	$4x^4 + 5x^3$	4	[0, 4, 5, 0, 0]	59	0.8207	50
160	$4x^4 + 5x^3$	4	[0, 4, 5, 0, 0]	59	0.8161	49
161	$4x^4 + 4x^3 + 2x^2$	4	[0, 4, 4, 2, 0]	58	0.9058	48
162	$4x^4 + 4x^3 + 2x^2$	4	[0, 4, 4, 2, 0]	58	0.8941	48
163	$4x^4 + 4x^3 + x^2$	4	[0, 4, 4, 1, 0]	57	0.9804	47
164	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8982	50
165	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8982	50
166	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8927	50
167	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8901	50
168	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8827	48
169	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8795	48
170	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8753	50

Table 4: Continued.

171	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8751	50
172	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8721	50
173	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8664	50
174	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8648	50
175	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8595	48
176	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8556	50
177	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8549	50
178	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8548	50
179	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8509	50
180	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8356	48
181	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.8307	50
182	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.7597	48
183	$4x^4 + 4x^3$	4	[0, 4, 4, 0, 0]	56	0.7596	48
184	$4x^4 + 3x^3 + x^2$	4	[0, 4, 3, 1, 0]	55	0.8478	47
185	$4x^4 + 3x^3 + x^2$	4	[0, 4, 3, 1, 0]	55	0.8376	47
186	$4x^4 + 3x^3 + x^2$	4	[0, 4, 3, 1, 0]	55	0.7799	47
187	$4x^4 + 3x^3 + x^2$	4	[0, 4, 3, 1, 0]	55	0.779	47
188	$4x^4 + 3x^3 + x^2$	4	[0, 4, 3, 1, 0]	55	0.7781	47
189	$4x^4 + 3x^3 + x^2$	4	[0, 4, 3, 1, 0]	55	0.778	47
190	$4x^4 + 3x^3$	4	[0, 4, 3, 0, 0]	54	0.7341	46
191	$4x^4 + 3x^3$	4	[0, 4, 3, 0, 0]	54	0.7335	46
192	$4x^4 + 3x^3$	4	[0, 4, 3, 0, 0]	54	0.7323	46
193	$4x^4 + 3x^3$	4	[0, 4, 3, 0, 0]	54	0.7279	46
194	$4x^4 + 2x^3 + 3x^2$	4	[0, 4, 2, 3, 0]	53	0.8605	46
195	$4x^4 + 2x^3 + 3x^2$	4	[0, 4, 2, 3, 0]	53	0.8574	46
196	$4x^4 + 2x^3 + 3x^2$	4	[0, 4, 2, 3, 0]	53	0.8507	46
197	$4x^4 + 2x^3 + 3x^2$	4	[0, 4, 2, 3, 0]	53	0.847	46
198	$4x^4 + 2x^3 + 2x^2$	4	[0, 4, 2, 2, 0]	52	0.845	45
199	$4x^4 + 2x^3 + 2x^2$	4	[0, 4, 2, 2, 0]	52	0.8425	45
200	$4x^4 + 2x^3 + x^2$	4	[0, 4, 2, 1, 0]	51	0.7962	44
201	$4x^4 + 2x^3 + x^2$	4	[0, 4, 2, 1, 0]	51	0.7955	44
202	$4x^4 + x^3 + 2x^2$	4	[0, 4, 1, 2, 0]	50	0.7883	43
203	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.8678	49
204	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.8677	49
205	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.8452	49

Table 4: Continued.

206	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.8446	49
207	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.8441	49
208	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.8394	49
209	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.8349	49
210	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.8315	49
211	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.8306	49
212	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.8259	49
213	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.8251	49
214	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.8195	49
215	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.8153	49
216	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.8136	49
217	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.8006	49
218	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.7936	49
219	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.7916	49
220	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.7901	49
221	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.7889	49
222	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.7767	49
223	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.7761	49
224	$3x^4 + 7x^3$	4	[0, 3, 7, 0, 0]	49	0.7562	49
225	$3x^4 + 5x^3$	4	[0, 3, 5, 0, 0]	48	0.8919	46
226	$3x^4 + 5x^3$	4	[0, 3, 5, 0, 0]	48	0.8919	46
227	$3x^4 + 5x^3$	4	[0, 3, 5, 0, 0]	48	0.8758	46
228	$3x^4 + 5x^3$	4	[0, 3, 5, 0, 0]	48	0.855	46
229	$3x^4 + 5x^3$	4	[0, 3, 5, 0, 0]	48	0.8539	46
230	$3x^4 + 5x^3$	4	[0, 3, 5, 0, 0]	48	0.8534	46
231	$3x^4 + 5x^3$	4	[0, 3, 5, 0, 0]	48	0.8474	46
232	$3x^4 + 5x^3$	4	[0, 3, 5, 0, 0]	48	0.8191	46
233	$3x^4 + 5x^3$	4	[0, 3, 5, 0, 0]	48	0.7572	46
234	$3x^4 + 5x^3$	4	[0, 3, 5, 0, 0]	48	0.7535	46
235	$3x^4 + 5x^3$	4	[0, 3, 5, 0, 0]	48	0.746	46
236	$3x^4 + 5x^3$	4	[0, 3, 5, 0, 0]	48	0.7402	46
237	$3x^4 + 4x^3 + x^2$	4	[0, 3, 4, 1, 0]	47	0.8135	45
238	$3x^4 + 4x^3 + x^2$	4	[0, 3, 4, 1, 0]	47	0.8036	45
239	$3x^4 + 4x^3 + x^2$	4	[0, 3, 4, 1, 0]	47	0.7992	45
240	$3x^4 + 4x^3 + x^2$	4	[0, 3, 4, 1, 0]	47	0.7825	45

Table 4: Continued.

241	$3x^4 + 4x^3 + x^2$	4	[0, 3, 4, 1, 0]	47	0.7755	45
242	$3x^4 + 4x^3 + x^2$	4	[0, 3, 4, 1, 0]	47	0.7655	45
243	$3x^4 + 4x^3$	4	[0, 3, 4, 0, 0]	46	(0.6474+0j)	44
244	$3x^4 + 3x^3 + 2x^2$	4	[0, 3, 3, 2, 0]	45	0.6864	43
245	$3x^4 + 3x^3 + 2x^2$	4	[0, 3, 3, 2, 0]	45	0.683	43
246	$3x^4 + 3x^3 + 2x^2$	4	[0, 3, 3, 2, 0]	45	0.6824	43
247	$3x^4 + 3x^3 + 2x^2$	4	[0, 3, 3, 2, 0]	45	0.682	43
248	$3x^4 + 3x^3 + x^2$	4	[0, 3, 3, 1, 0]	44	0.7152	42
249	$3x^4 + 3x^3 + x^2$	4	[0, 3, 3, 1, 0]	44	0.713	42
250	$3x^4 + 3x^3 + x^2$	4	[0, 3, 3, 1, 0]	44	0.6457	43
251	$3x^4 + 3x^3 + x^2$	4	[0, 3, 3, 1, 0]	44	0.6448	43
252	$3x^4 + x^3 + 4x^2$	4	[0, 3, 1, 4, 0]	43	0.6354	41
253	$3x^4 + x^3 + 4x^2$	4	[0, 3, 1, 4, 0]	43	0.6328	41
254	$3x^4 + x^3 + 3x^2$	4	[0, 3, 1, 3, 0]	42	0.6508	40
255	$3x^4 + x^3 + x^2$	4	[0, 3, 1, 1, 0]	41	0.5795	38
256	$2x^4 + 10x^3$	4	[0, 2, 10, 0, 0]	40	0.8167	48
257	$2x^4 + 10x^3$	4	[0, 2, 10, 0, 0]	40	0.8089	48
258	$2x^4 + 10x^3$	4	[0, 2, 10, 0, 0]	40	0.7851	48
259	$2x^4 + 10x^3$	4	[0, 2, 10, 0, 0]	40	0.7756	48
260	$2x^4 + 9x^3$	4	[0, 2, 9, 0, 0]	39	0.8465	47
261	$2x^4 + 9x^3$	4	[0, 2, 9, 0, 0]	39	0.8443	47
262	$2x^4 + 9x^3$	4	[0, 2, 9, 0, 0]	39	0.8442	47
263	$2x^4 + 9x^3$	4	[0, 2, 9, 0, 0]	39	0.8441	47
264	$2x^4 + 9x^3$	4	[0, 2, 9, 0, 0]	39	0.8257	47
265	$2x^4 + 9x^3$	4	[0, 2, 9, 0, 0]	39	0.8222	47
266	$2x^4 + 9x^3$	4	[0, 2, 9, 0, 0]	39	0.8185	47
267	$2x^4 + 9x^3$	4	[0, 2, 9, 0, 0]	39	0.8178	47
268	$2x^4 + 9x^3$	4	[0, 2, 9, 0, 0]	39	0.8165	47
269	$2x^4 + 9x^3$	4	[0, 2, 9, 0, 0]	39	0.8131	47
270	$2x^4 + 9x^3$	4	[0, 2, 9, 0, 0]	39	0.8083	47
271	$2x^4 + 9x^3$	4	[0, 2, 9, 0, 0]	39	0.8073	47
272	$2x^4 + 9x^3$	4	[0, 2, 9, 0, 0]	39	0.7976	47
273	$2x^4 + 7x^3 + x^2$	4	[0, 2, 7, 1, 0]	38	0.8904	44
274	$2x^4 + 7x^3 + x^2$	4	[0, 2, 7, 1, 0]	38	0.8782	44
275	$2x^4 + 7x^3 + x^2$	4	[0, 2, 7, 1, 0]	38	0.878	44

Table 4: Continued.

276	$2x^4 + 7x^3 + x^2$	4	[0, 2, 7, 1, 0]	38	0.8685	44
277	$2x^4 + 7x^3 + x^2$	4	[0, 2, 7, 1, 0]	38	0.8576	44
278	$2x^4 + 7x^3 + x^2$	4	[0, 2, 7, 1, 0]	38	0.843	44
279	$2x^4 + 7x^3 + x^2$	4	[0, 2, 7, 1, 0]	38	0.8418	44
280	$2x^4 + 7x^3 + x^2$	4	[0, 2, 7, 1, 0]	38	0.8211	44
281	$2x^4 + 6x^3 + x^2$	4	[0, 2, 6, 1, 0]	37	0.8645	43
282	$2x^4 + 6x^3 + x^2$	4	[0, 2, 6, 1, 0]	37	0.8557	43
283	$2x^4 + 6x^3 + x^2$	4	[0, 2, 6, 1, 0]	37	0.8395	43
284	$2x^4 + 6x^3 + x^2$	4	[0, 2, 6, 1, 0]	37	0.8326	43
285	$2x^4 + 6x^3 + x^2$	4	[0, 2, 6, 1, 0]	37	0.8156	43
286	$2x^4 + 6x^3 + x^2$	4	[0, 2, 6, 1, 0]	37	0.8013	43
287	$2x^4 + 6x^3 + x^2$	4	[0, 2, 6, 1, 0]	37	0.7977	43
288	$2x^4 + 5x^3 + x^2$	4	[0, 2, 5, 1, 0]	36	0.8623	41
289	$2x^4 + 5x^3 + x^2$	4	[0, 2, 5, 1, 0]	36	0.8559	41
290	$2x^4 + 5x^3 + x^2$	4	[0, 2, 5, 1, 0]	36	0.8207	41
291	$2x^4 + 5x^3 + x^2$	4	[0, 2, 5, 1, 0]	36	0.8024	41
292	$2x^4 + 5x^3 + x^2$	4	[0, 2, 5, 1, 0]	36	0.6921	41
293	$2x^4 + 5x^3 + x^2$	4	[0, 2, 5, 1, 0]	36	0.6908	41
294	$2x^4 + 5x^3 + x^2$	4	[0, 2, 5, 1, 0]	36	0.6846	41
295	$2x^4 + 5x^3 + x^2$	4	[0, 2, 5, 1, 0]	36	0.6833	41
296	$2x^4 + 5x^3 + x^2$	4	[0, 2, 5, 1, 0]	36	0.6819	41
297	$2x^4 + 5x^3 + x^2$	4	[0, 2, 5, 1, 0]	36	0.6812	41
298	$2x^4 + 5x^3 + x^2$	4	[0, 2, 5, 1, 0]	36	0.6803	41
299	$2x^4 + 5x^3 + x^2$	4	[0, 2, 5, 1, 0]	36	0.6765	41
300	$2x^4 + 4x^3 + x^2$	4	[0, 2, 4, 1, 0]	35	0.7264	39
301	$2x^4 + 4x^3 + x^2$	4	[0, 2, 4, 1, 0]	35	0.7207	39
302	$2x^4 + 4x^3 + x^2$	4	[0, 2, 4, 1, 0]	35	0.7182	39
303	$2x^4 + 4x^3 + x^2$	4	[0, 2, 4, 1, 0]	35	0.7101	39
304	$14x^3$	3	[0, 0, 14, 0, 0]	34	0.723	42
305	$14x^3$	3	[0, 0, 14, 0, 0]	34	0.7076	42
306	$14x^3$	3	[0, 0, 14, 0, 0]	34	0.7026	42
307	$14x^3$	3	[0, 0, 14, 0, 0]	34	0.6816	42
308	$13x^3$	3	[0, 0, 13, 0, 0]	33	0.7934	41
309	$13x^3$	3	[0, 0, 13, 0, 0]	33	0.7425	41
310	$13x^3$	3	[0, 0, 13, 0, 0]	33	0.7316	41

Table 4: Continued.

311	$13x^3$	3	[0, 0, 13, 0, 0]	33	0.731	41
312	$13x^3$	3	[0, 0, 13, 0, 0]	33	0.7244	41
313	$13x^3$	3	[0, 0, 13, 0, 0]	33	0.7188	41
314	$13x^3$	3	[0, 0, 13, 0, 0]	33	0.708	41
315	$13x^3$	3	[0, 0, 13, 0, 0]	33	0.7074	41
316	$13x^3$	3	[0, 0, 13, 0, 0]	33	0.7025	41
317	$13x^3$	3	[0, 0, 13, 0, 0]	33	0.6969	41
318	$13x^3$	3	[0, 0, 13, 0, 0]	33	0.6955	41
319	$13x^3$	3	[0, 0, 13, 0, 0]	33	0.6341	40
320	$13x^3$	3	[0, 0, 13, 0, 0]	33	0.6305	40
321	$13x^3$	3	[0, 0, 13, 0, 0]	33	0.6253	40
322	$12x^3 + 2x^2$	3	[0, 0, 12, 2, 0]	32	0.7037	39
323	$12x^3 + x^2$	3	[0, 0, 12, 1, 0]	31	0.766	40
324	$12x^3 + x^2$	3	[0, 0, 12, 1, 0]	31	0.7594	40
325	$12x^3 + x^2$	3	[0, 0, 12, 1, 0]	31	0.7493	40
326	$12x^3 + x^2$	3	[0, 0, 12, 1, 0]	31	0.7421	40
327	$12x^3 + x^2$	3	[0, 0, 12, 1, 0]	31	0.7328	40
328	$12x^3 + x^2$	3	[0, 0, 12, 1, 0]	31	0.6906	38
329	$12x^3 + x^2$	3	[0, 0, 12, 1, 0]	31	0.6842	38
330	$12x^3 + x$	3	[0, 0, 12, 0, 1]	30	0.7366	37
331	$12x^3$	3	[0, 0, 12, 0, 0]	29	0.6364	39
332	$12x^3$	3	[0, 0, 12, 0, 0]	29	0.6364	39
333	$12x^3$	3	[0, 0, 12, 0, 0]	29	0.6284	39
334	$12x^3$	3	[0, 0, 12, 0, 0]	29	0.628	39
335	$12x^3$	3	[0, 0, 12, 0, 0]	29	0.622	39
336	$12x^3$	3	[0, 0, 12, 0, 0]	29	0.6192	39
337	$11x^3$	3	[0, 0, 11, 0, 0]	28	(0.5964+0j)	38
338	$11x^3$	3	[0, 0, 11, 0, 0]	28	0.5939	38
339	$11x^3$	3	[0, 0, 11, 0, 0]	28	0.5903	38
340	$11x^3$	3	[0, 0, 11, 0, 0]	28	0.5827	38
341	$10x^3 + 3x^2$	3	[0, 0, 10, 3, 0]	27	0.773	37
342	$10x^3 + 3x^2$	3	[0, 0, 10, 3, 0]	27	0.7552	37
343	$10x^3 + 3x^2$	3	[0, 0, 10, 3, 0]	27	0.7459	37
344	$10x^3 + 3x^2$	3	[0, 0, 10, 3, 0]	27	0.7327	37
345	$10x^3 + 2x^2$	3	[0, 0, 10, 2, 0]	26	0.6914	36

Table 4: Continued.

346	$10x^3 + 2x^2$	3	[0, 0, 10, 2, 0]	26	0.6904	36
347	$10x^3 + 2x^2$	3	[0, 0, 10, 2, 0]	26	0.6831	36
348	$10x^3 + 2x^2$	3	[0, 0, 10, 2, 0]	26	0.6762	36
349	$10x^3 + x^2$	3	[0, 0, 10, 1, 0]	25	0.6194	37
350	$10x^3 + x^2$	3	[0, 0, 10, 1, 0]	25	0.6187	37
351	$10x^3 + x^2$	3	[0, 0, 10, 1, 0]	25	0.6136	37
352	$10x^3 + x^2$	3	[0, 0, 10, 1, 0]	25	0.6122	37
353	$10x^3 + x^2$	3	[0, 0, 10, 1, 0]	25	0.6119	37
354	$10x^3 + x^2$	3	[0, 0, 10, 1, 0]	25	0.611	37
355	$10x^3 + x^2$	3	[0, 0, 10, 1, 0]	25	0.6091	37
356	$9x^3 + 2x^2$	3	[0, 0, 9, 2, 0]	24	0.6648	35
357	$9x^3 + 2x^2$	3	[0, 0, 9, 2, 0]	24	0.6613	35
358	$9x^3 + 2x^2$	3	[0, 0, 9, 2, 0]	24	0.6603	35
359	$9x^3 + 2x^2$	3	[0, 0, 9, 2, 0]	24	0.6572	35
360	$9x^3 + 2x^2$	3	[0, 0, 9, 2, 0]	24	0.617	35
361	$9x^3 + 2x^2$	3	[0, 0, 9, 2, 0]	24	0.6132	35
362	$9x^3 + x$	3	[0, 0, 9, 0, 1]	23	0.7175	33
363	$8x^3 + 4x^2$	3	[0, 0, 8, 4, 0]	22	0.7563	33
364	$8x^3 + 4x^2$	3	[0, 0, 8, 4, 0]	22	0.7323	33
365	$8x^3 + 3x^2$	3	[0, 0, 8, 3, 0]	21	0.6196	34
366	$8x^3 + 3x^2$	3	[0, 0, 8, 3, 0]	21	0.6183	34
367	$8x^3 + 3x^2$	3	[0, 0, 8, 3, 0]	21	0.6066	34
368	$8x^3 + 3x^2$	3	[0, 0, 8, 3, 0]	21	0.6045	34
369	$8x^3 + 3x^2$	3	[0, 0, 8, 3, 0]	21	0.5622	34
370	$8x^3 + 3x^2$	3	[0, 0, 8, 3, 0]	21	0.562	34
371	$8x^3 + x^2$	3	[0, 0, 8, 1, 0]	20	0.505	32
372	$8x^3 + x^2$	3	[0, 0, 8, 1, 0]	20	0.5025	32
373	$8x^3 + x$	3	[0, 0, 8, 0, 1]	19	0.534	31
374	$7x^3 + 4x^2$	3	[0, 0, 7, 4, 0]	18	0.5876	33
375	$7x^3 + 4x^2$	3	[0, 0, 7, 4, 0]	18	0.5851	33
376	$7x^3 + 4x^2$	3	[0, 0, 7, 4, 0]	18	0.5776	33
377	$7x^3 + 4x^2$	3	[0, 0, 7, 4, 0]	18	0.5764	33
378	$7x^3 + 2x^2$	3	[0, 0, 7, 2, 0]	17	0.5105	31
379	$7x^3 + 2x^2$	3	[0, 0, 7, 2, 0]	17	0.5103	31
380	$7x^3 + 2x^2$	3	[0, 0, 7, 2, 0]	17	0.5073	31

Table 4: Continued.

381	$7x^3 + 2x^2$	3	[0, 0, 7, 2, 0]	17	0.5065	31
382	$6x^3 + 6x^2$	3	[0, 0, 6, 6, 0]	16	0.6326	32
383	$6x^3 + 6x^2$	3	[0, 0, 6, 6, 0]	16	0.6259	32
384	$6x^3 + 6x^2$	3	[0, 0, 6, 6, 0]	16	0.6226	32
385	$6x^3 + 6x^2$	3	[0, 0, 6, 6, 0]	16	0.621	32
386	$6x^3 + 5x^2$	3	[0, 0, 6, 5, 0]	15	0.6428	31
387	$6x^3 + 5x^2$	3	[0, 0, 6, 5, 0]	15	0.634	31
388	$6x^3 + 4x^2$	3	[0, 0, 6, 4, 0]	14	0.6688	30
389	$6x^3 + 4x^2$	3	[0, 0, 6, 4, 0]	14	0.6598	30
390	$6x^3 + 3x^2$	3	[0, 0, 6, 3, 0]	13	0.5548	29
391	$6x^3 + 3x^2$	3	[0, 0, 6, 3, 0]	13	0.5532	29
392	$5x^3 + 4x^2$	3	[0, 0, 5, 4, 0]	12	0.4451	29
393	$5x^3 + x$	3	[0, 0, 5, 0, 1]	11	0.3992	25
394	$4x^3 + 6x^2$	3	[0, 0, 4, 6, 0]	10	0.4711	28
395	$4x^3 + 6x^2$	3	[0, 0, 4, 6, 0]	10	0.4693	28
396	$4x^3 + 6x^2$	3	[0, 0, 4, 6, 0]	10	0.4688	28
397	$4x^3 + 6x^2$	3	[0, 0, 4, 6, 0]	10	0.4684	28
398	$4x^3 + 5x^2$	3	[0, 0, 4, 5, 0]	9	0.4938	27
399	$4x^3 + 5x^2$	3	[0, 0, 4, 5, 0]	9	0.4924	27
400	$4x^3 + 2x^2$	3	[0, 0, 4, 2, 0]	8	0.413	24
401	$4x^3 + 2x^2$	3	[0, 0, 4, 2, 0]	8	0.4127	24
402	$15x^2$	2	[0, 0, 0, 15, 0]	7	0.5021	24
403	$15x^2$	2	[0, 0, 0, 15, 0]	7	0.4939	24
404	$14x^2$	2	[0, 0, 0, 14, 0]	6	0.4411	23
405	$14x^2$	2	[0, 0, 0, 14, 0]	6	0.4375	23
406	$12x^2 + x$	2	[0, 0, 0, 12, 1]	5	0.5068	21
407	$11x^2$	2	[0, 0, 0, 11, 0]	4	0.3575	20
408	$11x^2$	2	[0, 0, 0, 11, 0]	4	0.3566	20
409	$10x^2 + x$	2	[0, 0, 0, 10, 1]	3	0.3854	19
410	$6x^2 + x$	2	[0, 0, 0, 6, 1]	2	0.2949	15
411	$8x$	1	[0, 0, 0, 0, 8]	1	0.2176	9

REFERENCES

[1] Abeledo, H., and Atkinson, G. W. Unimodularity of the Clar number problem. *Linear Algebra and Its Applications 420*, 2-3 (2007), 441–448.

[2] Aihara, J. I. Reduced HOMO-LUMO Gap as an Index of Kinetic Stability for Polycyclic Aromatic Hydrocarbons. *Journal of Physical Chemistry A 103*, 37 (1999), 7487–7495.

[3] Alstrup, S., Holm, J., Lichtenberg, K. D., and Thorup, M. Maintaining Information in Fully Dynamic Trees with Top Trees. *ACM Transactions on Algorithms 1*, 2 (2005), 243–264.

[4] Austin, S. J., Fowler, P. W., Hansenb, P., Monolopoulos, D. E., and Zheng, M. Fullerene isomers of C_{60}. Kekulé counts versus stability. *Chemical Physics Letters 228*, 4-5 (1994), 478–484.

[5] Bašić, N., Estélyi, I., Škrekovski, R., and Tratnike, N. On the clar number of benzenoid graphs. *Match 80*, 1 (2018), 173–188.

[6] Bernáth, A., and R, B.-K. E. NP-hardness of the Clar number in general plane graphs. *EGRES Quick-Proof 7* (2011), 1–3.

[7] Brinkmann, G., Caporossi, G., and Hansen, P. A constructive enumeration of fusenes and benzenoids. *Journal of Algorithms 45*, 2 (2002), 155–166.

[8] Chaplick, S., Fürst, M., Maffray, F., and Rautenbach, D. On some graphs with a unique perfect matching. *Information Processing Letters 139* (2018), 60–63.

[9] Che, Z., and Chen, Z. Forcing hexagons in hexagonal systems. *Match 56*, 3 (2006), 649–668.

[10] Che, Z., and Chen, Z. Forcing on perfect matchings - A survey. *Match 66*, 1 (2011), 93–136.

[11] Che, Z., and Chen, Z. Forcing faces in plane bipartite graphs. *Discrete Applied Mathematics 161*, 1-2 (2013), 71–80.

[12] Chou, C. P., Kang, J. S., and Witek, H. A. Closed-form formulas for the Zhang-Zhang polynomials of benzenoid structures: Prolate rectangles and their generalizations. *Discrete Applied Mathematics 198* (2016), 101–108.

[13] Chou, C. P., Li, Y., and Witek, H. A. Zhang-Zhang polynomials of various classes of benzenoid systems. *Match 68*, 1 (2012), 31–64.

[14] Cioslowski, J. A unified theory of the stability of benzenoid hydrocarbons. *International Journal of Quantum Chemistry 31*, 4 (1987), 581–590.

[15] Clar, E. *The Aromatic Sextet.* John Wiley & Sons, London, 1972.

[16] Clar, E., and Macpherson, I. A. The significance of Kekulé structures for the stability of aromatic systems-II. *Tetrahedron 18*, 12 (1962), 1411–1416.

[17] Cyvin, S. J., and Gutman, I. Kekulé structures in benzenoid hydrocarbons. In *Lecture Notes in Chemistry*, vol. 46. Springer-Verlag, Berlin Heidelberg, 1988.

[18] Deng, K., and Zhang, H. Extremal anti-forcing numbers of perfect matchings of graphs. *Discrete Applied Mathematics 224* (2017), 69–79.

[19] Ding, L. P., Zhang, F. H., Zhu, Y. S., Lu, C., Kuang, X. Y., Lv, J., and Shao, P. Understanding the structural transformation, stability of medium-sized neutral and charged silicon clusters. *Scientific Reports 5* (2015), 1–12.

[20] Dixit, V. A., and Singh, Y. Y. How much aromatic are naphthalene and graphene? *Computational and Theoretical Chemistry 1162*, 41 (2019), 1–12.

[21] Fukuda, K., and Matsui, T. Finding all the perfect matchings in bipartite graphs. *Applied Mathematics Letters 7*, 1 (1994), 15–18.

[22] Gabow, H. N., Kaplan, H., and Tarjan, R. E. Unique Maximum Matching Algorithms. *Journal of Algorithms 40*, 2 (2001), 159–183.

[23] Galil, Z., and Italiano, G. F., and Sarnak, N. Fully Dynamic Planarity Testing with Applications. *Journal of the ACM 46*, 1 (1999), 28–91.

[24] Gao, Y., Li, Q., and Zhang, H. Fullerenes with the maximum Clar number. *Discrete Applied Mathematics 202* (2016), 58–69.

[25] Geim, A. K., and Novoselov, K. S. The rise of graphene. *Nature Materials 6* (2007), 183–191.

[26] Graver, J. E., and Hartung, E. J. Internal Kekulé structures for graphene and general patches. *Match 76*, 3 (2016), 693–705.

[27] Griffith, J. S., and Orgel, L. E. Ligand-field theory. *Quarterly Reviews, Chemical Society 11* (1957), 381–393.

[28] Gutman, I., Obenland, S., and Schmidt, W. Clar Formulas and Kekule Structures. *Match 17* (1985), 75–90.

[29] Hansen, P., and Zheng, M. Upper bounds for the clar number of a benzenoid hydrocarbon. *Journal of the Chemical Society, Faraday Transactions 88*, 12 (1992), 1621–1625.

[30] Hansen, P., and Zheng, M. The Clar number of a benzenoid hydrocarbon and linear programming. *Journal of Mathematical Chemistry 15*, 1 (1994), 93–107.

[31] Harary, F., Klein, D., and Zivkovic, T. Graphical properties of polyhexes: perfect matching vector and forcing. *Journal of Mathematical Chemistry 6*, 1 (1991), 295–306.

[32] Holm, J., Lichtenberg, K. D., and THORUP, M. Poly-Logarithmic Deterministic Fully-Dynamic Algorithms for Connectivity, Minimum Spanning Tree, 2-Edge, and Biconnectivity. *Journal of the ACM 48*, 4 (2001), 723–760.

[33] Hosoya, H., and Yamaguchi, T. Sextet polynomial. A new enumeration and proof technique for the resonance theory applied to the aromatic hydrocarbons. *Tetrahedron Letters 16*, 52 (1975), 4659–4662.

[34] Hückel, E. Quanstentheoretische beiträge zum benzolproblem II. quantentheorie der induzierten poläritaten. *Zeitschrift für Physik 72*, 5-6 (1931), 310–337.

[35] Hückel, E. Quantentheoretische beiträge zum benzolproblem I. die elektronenkonfiguration des benzols und verwandter verbindungen. *Zeitschrift für Physik 70*, 3-4 (1931), 204–286.

[36] Hückel, E. Quantentheoretische beiträge zum problem der aromatischen und ungesättigten verbindungen. III. *Zeitschrift für Physik 76*, 9-10 (1932), 628–648.

[37] Kekulé, A. Sur la constitution des substances aromatiques. *Bulletin de la Société Chimique de Paris 22* (1865), 98–110.

[38] Klavžar, S., Žigert, P., and Gutman, I. Clar number of catacondensed benzenoid hydrocarbons. *Journal of Molecular Structure: THEOCHEM 586*, 1-3 (2002), 235–240.

[39] Klein, D. J., and Randić, M. Innate degree of freedom of a graph. *Journal of Mathematical Chemistry 8* (1987), 516–521.

[40] Klein, D. J., Yang, Y., Ye, D., and Klein, D. J. HOMO – LUMO gaps for sub-graphenic and sub-buckytubic species. *Proceedings of the Royal Society A 471*, 2180 (2015), 1–16.

[41] Kotzig, A. On the theory of finite graphs with a linear factor I. *Mat.-Fyz. Casopis. Slovensk. 9*, 3 (1959), 136–159.

[42] Li, X. Hexagonal systems with forcing single edges. *Discrete Applied Mathematics 72*, 3 (1997), 295–301.

[43] Misra, A., Klein, D. J., and Morikawa, T. Clar theory for Molecular benzenoids. *The Journal of Physical Chemistry A 113*, 6 (2009), 1151–1158.

[44] Misra, A., Schmalz, T. G., and Klein, D. J. Clar theory for radical benzenoids. *Journal of Chemical Information and Modeling 49*, 12 (2009), 2670–2676.

[45] Nishina, N., Makino, M., and Aihara, J. I. Aromatic Character of Irregular-Shaped Nanographenes. *Journal of Physical Chemistry A 120*, 15 (2016), 2431–2442.

[46] Novoselov, K. S., Geim, A. K., Morozov, S. V., Jiang, D., Zhang, Y., Dubonos, S. V., Grigorieva, I. V., and Firsov, A. A. Electric field effect in atomically thin carbon films. *Science 306*, 5696 (2004), 666–669.

[47] Ohkami, N., and Hosoya, H. Topological dependency of the aromatic sextets in polycyclic benzenoid hydrocarbons. recursive relations of the sextet polynomial. *Theoretica chimica acta 64*, 3 (1983), 153–170.

[48] Petersen, R., Pedersen, T. G., and Jauho, A. P. Clar sextet analysis of triangular, rectangular, and honeycomb graphene antidot lattices. *ACS Nano 5*, 1 (2011), 523–529.

[49] Petersen, R., Pedersen, T. G., and Jauho, A. P. Clar sextets in square graphene antidot lattices. *Physica E: Low-Dimensional Systems and Nanostructures 44*, 6 (2012), 967–970.

[50] Popov, I. A., Bozhenko, K. V., and Boldyrev, A. I. Is graphene aromatic? *Nano Research 5*, 2 (2012), 117–123.

[51] R., B.-K. E., and Bernáth, A. The complexity of the Clar number problem and an exact algorithm. *Journal of Mathematical Chemistry 56*, 2 (2018), 597–605.

[52] Randić, M., El-Basil, S., Nikolić, S., and Trinajstić, N. Clar polynomials of large benzenoid systems. *Journal of Chemical Information and Computer Sciences 38*, 4 (1998), 563–574.

[53] Rispoli, F. J. Counting Perfect Matchings in Hexagonal Systems Associated with Benzenoids. *Mathematics Magazine 74*, 3 (2001), 194–200.

[54] Ruiz-Morales, Y. HOMO-LUMO gap as an index of molecular size and structure for polycyclic aromatic hydrocarbons (PAHs) and asphaltenes: A theoretical study. *Journal of Physical Chemistry A 106*, 46 (2002), 11283–11308.

[55] Sachs, H. Perfect matchings in hexagonal systems. *Combinatorica 4*, 1 (1984), 89–99.

[56] Sereni, J. S., and Stehlík, M. On the sextet polynomial of fullerenes. *Journal of Mathematical Chemistry 47*, 3 (2010), 1121–1128.

[57] Shi, L., and Zhang, H. Forcing and anti-forcing numbers of (3,6)-fullerenes. *Match 76*, 3 (2016), 597–614.

[58] Solà, M. Forty years of Clar's aromatic π-sextet rule. *Frontiers in Chemistry 1* (2013), 4–11.

[59] Tarjan, R. E., and Werneck, R. F. Self-adjusting top trees. *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms* (2005), 813–822.

[60] Uno, T. Algorithms for enumerating all perfect, maximum and maximal matchings in bipartite graphs. In *Algorithms and Computation: 8th International Symposium*, H. W. Leong, H. Imai, and S. Jain, Eds. 1997, pp. 92–101.

[61] Vöge, M., Guttmann, A. J., and Jensen, I. On the number of benzenoid hydrocarbons. *Journal of Chemical Information and Computer Sciences 42*, 3 (2002), 456–466.

[62] Wai Chee Shiu, Peter Che Bor Lama, H. Z. Clar and sextet polynomials of buckminsterfullerene. *Journal of Molecular Structure: Theochem 622*, 3 (2003), 239–248.

[63] Wassmann, T., Seitsonen, A. P., Saitta, A. M., Lazzeri, M., and Mauri, F. Clar's Theory, π-Electron Distribution, and Geometry of Graphene Nanoribbons. *Journal of the American Chemical Society 132*, 10 (2010), 3440–3451.

[64] Witek, H. A., Langner, J., Moś, G., and Chou, C. P. Zhang-Zhang polynomials of regular 5-tier benzenoid strips. *Match 78*, 2 (2017), 487–504.

[65] Wu, D., Gao, X., Zhou, Z., and Chen, Z. Understanding aromaticity of graphene and graphene nanoribbons by the clar sextet rule. In *Graphene Chemistry: Theoretical Perspectives*. Wiley, 2013, ch. 3, pp. 29–49.

[66] Ye, D., and Zhang, H. Extremal fullerene graphs with the maximum Clar number. *Discrete Applied Mathematics 157*, 14 (2009), 3152–3173.

[67] Zdetsis, A. D., and Economou, E. N. A Pedestrian Approach to the Aromaticity of Graphene and Nanographene: Significance of

Huckel's (4n+2)π Electron Rule. *Journal of Physical Chemistry C 119*, 29 (2015), 16991–17003.

[68] Zhai, S., Alrowaili, D., and Ye, D. Clar structures vs Fries structures in hexagonal systems. *Applied Mathematics and Computation 329* (2018), 384–394.

[69] Zhang, F., Chen, R., and Guo, X. Perfect matchings in hexagonal systems. *Graphs and Combinatorics 1*, 1 (1985), 383–386.

[70] Zhang, F., Chen, R., Guo, X., and Cutman, I. Benzenoid systems whose (x,y,z)-invariants have $x = 1$ and $x = 2$. *Match*, 26 (1991), 229–241.

[71] Zhang, F., Guo, X., and Zhang, H. Advances of Clar's Aromatic Sextet Theory and Randić's Conjugated Circuit Model. *The Open Organic Chemistry Journal 5*, 1 (2011), 87–111.

[72] Zhang, F., and Li, X. Hexagonal systems with forcing edges. *Discrete Mathematics 140*, 1-3 (1995), 253–263.

[73] Zhang, F., and Zhang, H. A new enumeration method for Kekulé structures of hexagonal systems with forcing edges. *Journal of Molecular Structure: THEOCHEM 331*, 3 (1995), 255–260.

[74] Zhang, F. J., Zhang, H. P., and Liu, Y. T. The Clar covering polynomial of hexagonal systems II. An application to resonance energy

of condensed aromatic hydrocarbons. *Chinese Journal of Chemistry 14*, 4 (1996), 321–325.

[75] Zhang, G., and Musgrave, C. B. Comparison of DFT methods for molecular orbital eigenvalue calculations. *The Journal of Physical Chemistry 111* (2007), 1554–1561.

[76] Zhang, H., and Ye, D. An upper bound for the Clar number of fullerene graphs. *Journal of Mathematical Chemistry 41*, 2 (2007), 123–133.

[77] Zhang, H., Ye, D., and Liu, Y. A combination of Clar number and Kekulé count as an indicator of relative stability of fullerene isomers of C_{60}. *Journal of Mathematical Chemistry 48*, 3 (2010), 733–740.

[78] Zhang, H., and Zhang, F. The clar covering polynomial of hexagonal systems. *Discrete Mathematics 212*, 3 (2000), 261–269.

[79] Zhang, H., and Zhang, F. The clar covering polynomial of hexagonal systems III. *Discrete Mathematics 212*, 3 (2000), 261–269.

[80] Zhang, H., Zhao, S., and Lin, R. The forcing polynomial of catacondensed hexagonal systems. *Match 73*, 2 (2015), 473–490.

[81] Zhang, X., Xin, J., and Ding, F. The edges of graphene. *Nanoscale 5* (2013), 2556–2569.

[82] Zhao, S., and Zhang, H. Anti-forcing polynomials for benzenoid systems with forcing edges. *Discrete Applied Mathematics 250* (2018), 342–356.

[83] Zheng, M., and Chen, R. A maximal cover of hexagonal systems. *Graphs and Combinatorics 1*, 1 (1985), 295–298.

[84] Zhou, X., and Zhang, H. Clar sets and maximum forcing numbers of hexagonal systems. *Match 74*, 1 (2015), 161–174.