
Electronic Theses and Dissertations, 2004-2019

2009

Almost Regular Graphs And Edge Face Colorings Of Plane Graphs

Lisa Macon
University of Central Florida



Part of the [Mathematics Commons](#)

Find similar works at: <https://stars.library.ucf.edu/etd>

University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Macon, Lisa, "Almost Regular Graphs And Edge Face Colorings Of Plane Graphs" (2009). *Electronic Theses and Dissertations, 2004-2019*. 3923.

<https://stars.library.ucf.edu/etd/3923>

**ALMOST REGULAR GRAPHS AND EDGE-FACE COLORINGS OF
PLANE GRAPHS**

by

LISA FISCHER MACON

B.S. Hofstra University, 1991
M.S. University of Central Florida, 1999

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Mathematics
in the College of Sciences
at the University of Central Florida

Spring Term

2009

Major Professor: Yue Zhao

ABSTRACT

Regular graphs are graphs in which all vertices have the same degree. Many properties of these graphs are known. Such graphs play an important role in modeling network configurations where equipment limitations impose a restriction on the maximum number of links emanating from a node. These limitations do not enforce strict regularity, and it becomes interesting to investigate nonregular graphs that are in some sense close to regular. This dissertation explores a particular class of almost regular graphs in detail and defines generalizations on this class. A linear-time algorithm for the creation of arbitrarily large graphs of the discussed class is provided, and a polynomial-time algorithm for recognizing graphs in the class is given. Several invariants for the class are discussed.

The edge-face chromatic number χ_{ef} of a plane graph G is the minimum number of colors that must be assigned to the edges and faces of G such that no edge or face of G receives the same color as an edge or face with which it is incident or adjacent. A well-known result for the upper bound of χ_{ef} exists for graphs with maximum degree $\Delta \geq 10$. We present a tight upper bound for plane graphs with $\Delta = 9$.

To my children: Tracy-Lynn, Sheridan, and Bryce

ACKNOWLEDGMENTS

My most heartfelt gratitude goes to:

Dr. Nick Zhao, my advisor and mentor, for his help and support in getting me through this entire process.

Dr. Robert Brigham, one of the world's best teachers, for making it so easy to fall in love with graph theory many years ago.

My friends, family, and Valencia colleagues, for all of their help, encouragement and support.

My two youngest children, Sheridan and Bryce, for being as understanding as small children can possibly be about the hours away from them required to do this work, and for all the extra hugs and kisses given exactly when I needed them.

My parents, for approximately a million reasons including but not limited to babysitting, financial support, and love.

My eldest daughter Tracy-Lynn, who also serves as my best friend and closest confidant, for providing me with love and inspiration for 14 years.

My deepest thanks are for the one great true love of my life, my husband Brian Macon, for loving me, for always believing in me, and for doing the best he could to try to pick up my slack to help me achieve a lifelong dream.

TABLE OF CONTENTS

LIST OF FIGURES	vii
LIST OF TABLES	viii
CHAPTER 1 INTRODUCTION AND LITERATURE REVIEW	1
Notation	1
Almost Regular Graphs – Literature Review	5
Edge-Face Coloring – Literature Review	12
CHAPTER 2 AMOST REGULAR GRAPHS	16
Motivation	16
Objective	17
Outline	20
Strongly $k-\varepsilon$ Almost Regular Graphs: Previous Results	20
Definitions and Preliminaries	20
The Order of Extremal Strongly $k-\varepsilon$ Almost Regular Graphs	23
Invariants of Extremal Strongly $k-\varepsilon$ Almost Regular Graphs	27
The Size of Extremal Strongly $k-\varepsilon$ Almost Regular Graphs	30
Strongly $k-\varepsilon$ Almost Regular Graph Algorithms	39
A Linear-Time Algorithm for Creation	39
A Polynomial-Time Algorithm for Recognition	41
CHAPTER 3 EDGE-FACE COLORINGS OF PLANE GRAPHS	48
Motivation and Notation	48
Objective	48
Outline	49
A Precise Upper Bound for the Edge-Face Chromatic Number of Plane Graphs with $\Delta = 9$...	49
CHAPTER 4 SUGGESTIONS FOR FUTURE WORK	66
Almost Regular Graphs	66
Edge-Face Colorings of Plane Graphs	67

APPENDIX A: CONSTRUCTION ALGORITHM IMPLEMENTATION AND EXAMPLES	.69
APPENDIX B: RECOGNITION ALGORITHM IMPLEMENTATION AND EXAMPLES80
LIST OF REFERENCES 102

LIST OF FIGURES

Figure 1: A Simple Graph	2
Figure 2: Paths	2
Figure 3: Isomorphic Graphs	3
Figure 4: A graph and two subgraphs	3
Figure 5: The first five complete graphs	4
Figure 6: 2-regular and 3-regular graphs on six vertices	4
Figure 7: Graphs in which every vertex has degree t or $t+1$	5
Figure 8: Degree uniform graphs	6
Figure 9: Locally $(k-\varepsilon)$ almost regular graph	8
Figure 10: Highly irregular graphs	9
Figure 11: Distance-2-regular graph	10
Figure 12: K_3 -regular graph of degree 3	11
Figure 13: P_3 -irregular graph	11
Figure 14: A graph with chromatic number 2	12
Figure 15: A graph with chromatic index 3	13
Figure 16: A graph with $\Delta(G) = 4$ edge-face colored with 6 colors	15
Figure 17: Graphical Representations of Computer Networks	16
Figure 18: 2-regular and 3-regular graphs on six vertices	18
Figure 19: Strongly $k-\varepsilon$ almost regular graphs	19
Figure 20: Strongly 1-2 almost regular graph with vertices grouped by degrees	21
Figure 21: Strongly $k-\varepsilon$ Almost Regular Graph with $\Delta = \delta + \varepsilon$	27
Figure 22: Strongly 1-2 almost regular graph	42
Figure 23: XML Representation of a Strongly 1-2 Almost Regular Graph	44
Figure 24: Forbidden subgraphs of a minimal pseudograph	52
Figure 25: Subgraphs referenced in Lemma 3.3	54

LIST OF TABLES

Table 1: Almost Regular Classes7
Table 2: Classes of k - ϵ Almost Regular Graphs19

CHAPTER 1 INTRODUCTION AND LITERATURE REVIEW

Notation

Graphs are mathematical objects which are useful in modeling problems from a range of disciplines, including computer network design. Some of the many areas where the use of graphs has found application are engineering, computer science, chemistry, and sociology. We review the aspects of graph theory that are pertinent to this dissertation.

A graph $G(V,E)$ is comprised of a set $V(G)$ of **vertices** and a set $E(G)$ of **edges** between pairs of vertices in $V(G)$. The **order** of a graph G is the number of vertices in G , while the **size** of G is the number of edges in G . Two vertices are **adjacent** if they are connected by an edge. For the majority of this study, we will assume simple graphs in which no edge joins a vertex to itself and at most one edge joins two vertices. In the graph in Figure 1-1, vertex A is adjacent to B and C, but not adjacent to D or E. The **neighborhood** of a vertex v , denoted $N(v)$, is the set of vertices that are adjacent to v . In the graph in Figure 1-1, vertices B and C are in the neighborhood of A, but neither D nor E is in $N(A)$. The **degree** of a vertex is the number of edges connected to that vertex. The **degree set** of a graph G , denoted D_G , is the set of all the different degrees of vertices in G . The **minimum degree** of any vertex in a graph G is denoted $\delta(G)$, and the **maximum degree** is denoted $\Delta(G)$. In the graph G in Figure 1, vertices B and E have degree 3; vertices A, C, and D have degree 2; $D_G = \{2, 3\}$; $\delta(G) = 2$; and $\Delta(G) = 3$.

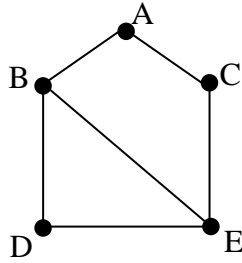


Figure 1: A Simple Graph

A **path** is a connected sequence of edges in a graph and the **length** of the path is the number of edges traversed. A path of n vertices is denoted P_n . Figure 1-2 shows paths of various lengths. If u and v are vertices, the **distance** from u to v , written $d(u,v)$, is the minimum length of any path from u to v . In Figure 1 above, $d(A,C) = 1$ while $d(A,D) = 2$. A graph is **connected** if there exists a path between any pair of vertices in the graph.

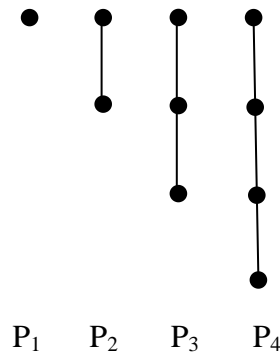


Figure 2: Paths

Two graphs G and H are **isomorphic** if there exists a one-to-one correspondence between their vertex sets which preserves adjacency. For example, G and H in Figure 3 are isomorphic under the correspondence $v_i \leftrightarrow u_i$.

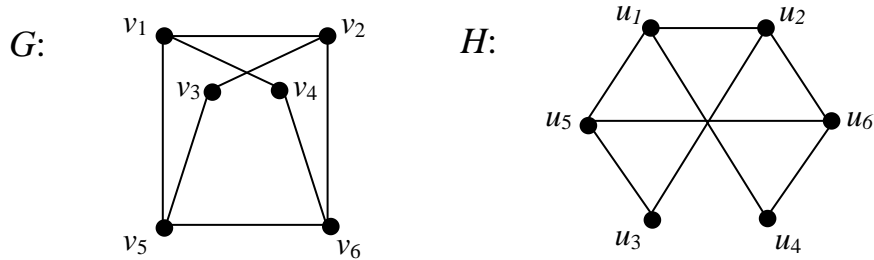


Figure 3: Isomorphic Graphs

A **numerical invariant** of a graph G is a number associated with G which has the same value for any graph isomorphic to G . For example, the number of vertices in a graph is an invariant. An **extremal** graph is one which has a certain prescribed property and for which an invariant has a maximum or minimum possible value.

A **subgraph** of a graph G is a graph having all of its vertices and edges in G . Figure 4 shows a graph G and two of its subgraphs, G_1 and G_2 . A **spanning subgraph** of a graph G is a subgraph containing all the vertices in G . In Figure 4, G_1 is a spanning subgraph of G while G_2 is not. A **factor** of a graph G is a spanning subgraph of G which is not totally disconnected. For any set S of vertices in G , the **induced subgraph** $\langle S \rangle$ is the subgraph of G with vertex set S and whose edge set contains all edges in G between the vertices in S .

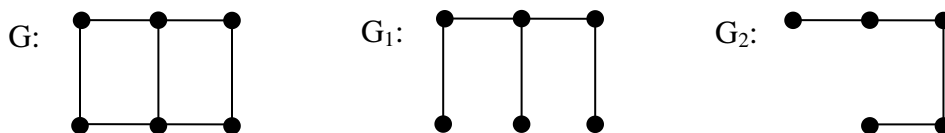


Figure 4: A graph and two subgraphs

A **complete** graph with n vertices (denoted K_n) is a graph in which each vertex is connected to all of the others. Figure 5 shows the first five complete graphs. Note that each vertex in K_n has degree $n-1$.

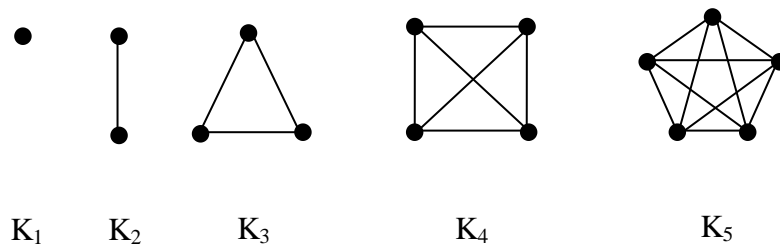


Figure 5: The first five complete graphs

A **regular** graph is a graph in which all vertices have the same degree. If all vertices in a regular graph G have degree r , then G is said to be **r -regular**. The graphs in Figure 6 are 2-regular and 3-regular, each with six vertices.



Figure 6: 2-regular and 3-regular graphs on six vertices

Almost Regular Graphs – Literature Review

The literature sometimes refers to graphs we would designate almost regular by the names “nearly regular” or “irregular”. One of the earliest appearances was in 1983 in a paper by Alon, Friendland, and Kalai [3] which considered graphs in which all vertex degrees are either t or $t+1$ and at least one vertex has each degree. Figure 7 illustrates examples of two such graphs. It was shown that graphs of this type for which $t \geq 2q - 2$ and q is a prime power contain a q -regular subgraph and also an r -regular subgraph for all $r < q$ where $r \equiv q \pmod{2}$. It is also proved that every graph with maximal degree $\delta \geq 2q - 2$ and average degree (mathematical mean of the degrees of all vertices in the graph) $d > [(2q - 2)/(2q - 1)](\delta + 1)$, where q is a prime power, contains a q -regular subgraph (and also an r -regular subgraph for all $r < q$ where $r \equiv q \pmod{2}$). This paper also briefly examines regular subgraphs contained in graphs in which every vertex has degree t , $t+1$, or $t+2$. Note that in both types of graphs discussed in this paper (those whose vertices have degree t or $t+1$, and those whose vertices have degree t , $t+1$, or $t+2$) each vertex differs in degree from each of its neighbors by at most 1 for the former type and at most 2 for the latter type. However, there is no restriction on the number of neighbors a vertex of a certain degree can have.

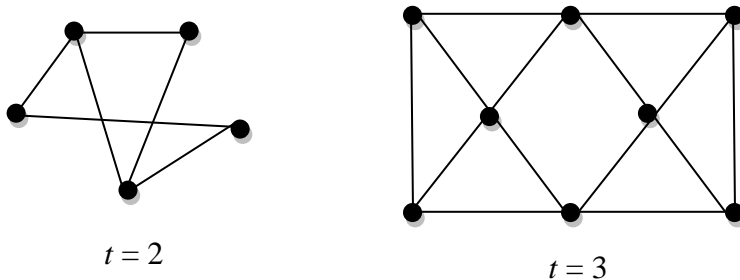


Figure 7: Graphs in which every vertex has degree t or $t+1$

In 1988 Chartrand, Lesniak, Mynhardt, and Ollerman [6] defined an **n -degree uniform graph** as one which contains exactly n vertices of each degree that appears in a graph. For example, Figure 8(a) is a 2-degree uniform graph with degrees 1, 2 and 3, and Figure 1-8(b) is a 4-degree uniform graph with vertex degrees 2 and 4. The authors show that for every graph H , not necessarily connected, there exists a degree uniform graph G such that H is an induced subgraph of G and that $D_G = D_H$. They present a stronger version of this result for connected graphs. Assume H is a connected graph with degree set $D_H = \{d_1, d_2, \dots, d_k\}$, $k \geq 2$, where $d_1 < d_2 < \dots < d_k$, such that H contains p_i vertices of degree d_i ($1 \leq i \leq k$). Then there exists a connected, degree uniform graph G containing H as an induced subgraph with $D_G = D_H$ unless one of the following conditions hold: $D_H = \{1, 2\}$ and $p_1 < p_2$; $D_H = \{1, 3\}$ and $p_1 < p_3$; or $D_H = \{1, 2, 3\}$ and $p_1 < p_2$.

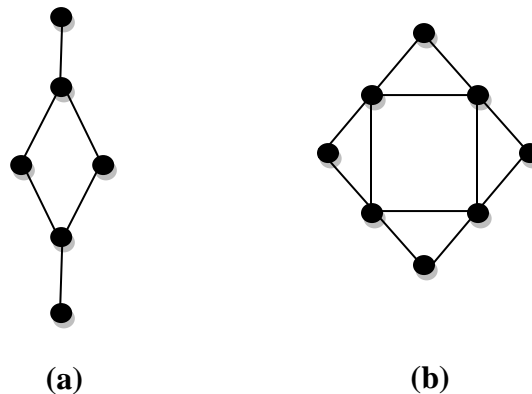


Figure 8: Degree uniform graphs

Yet another definition of almost regularity appears in a 1991 paper by Joentgen and Volkmann [14]. A graph G is said to be **locally r -almost regular** if, for every pair of vertices v and w in G , $|d(v) - d(w)| \leq r$ where $d(v)$ and $d(w)$ are the degrees of v and w , respectively. The

paper discusses two concepts – the factors of r -almost regular graphs and the classification of spanning subgraphs for r -almost regular graphs. Note that this class of graphs is a weak restriction on the degrees in an almost regular graph in that the difference of degrees between adjacent vertices is *bounded* by some number rather than restricted to a particular value as in previous definitions of almost regularity.

Some of the most extensive work done on almost regular graphs appears in a 1993 preprint by Linda Lawson [17]. In it, four classes of almost regular graphs, summarized in Table 1-1, are defined. In a graph in any of the four classes, every vertex has the same fixed degree d except for a prescribed number of vertices. The classes are defined by the way in which the number and degrees of these special vertices differ from the vertices of degree d .

Table 1: Almost Regular Classes

CLASS	Each vertex has degree d except...	The degrees of those t vertices differ from d by...
Almost Regular Class 1	exactly t vertices	exactly ε
Almost Regular Class 2	exactly t vertices	at most ε
Almost Regular Class 3	at most t vertices	exactly ε
Almost Regular Class 4	at most k vertices	at most ε

Note that we will use these basic definitions to define the classes of almost-regular graphs discussed in this dissertation.

Lawson discusses the construction of all four classes of graphs. She proves that the complement of an Almost Regular Class 4 graph is also Almost Regular Class 4 and that every graph is an induced subgraph of some Almost Regular Class 1 graph. In addition, she discusses the possible applications of almost regular graphs in the computer networking environment.

Some of the ideas she presents are expanded upon by Haynes, Lawson and Boland [12]. In this paper, a **locally $(k-\epsilon)$ almost regular** graph G is defined as a graph in which every vertex v in G is adjacent to all but at most k vertices of degree $d(v)$ and those k differ in degree from v by at most ϵ . This definition permits the same range of degrees as found in Almost Regular Class 1. However, Almost Regular Class 1 graphs have strict properties in that exactly k vertices which differ in degree from v must differ by exactly ϵ . An example of a locally $(1-1)$ almost regular graph which is not Almost Regular Class 1 is given in Figure 9. Vertices a and b demonstrate one reason why this graph is not Almost Regular Class 1 since they are adjacent only to vertices of degree three.

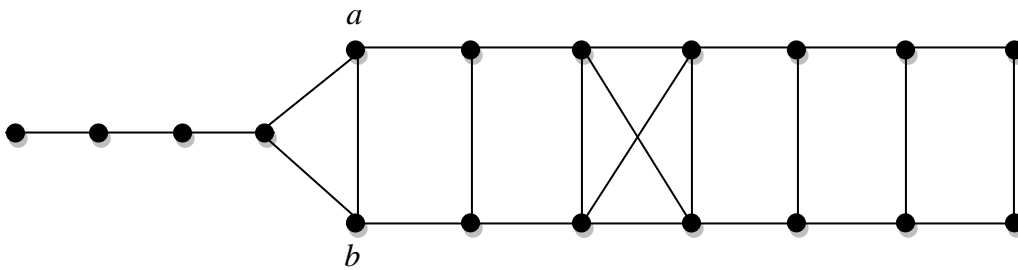


Figure 9: Locally $(k-\epsilon)$ almost regular graph

In contrast to graphs that are near to being regular, there are graphs that are very much not regular. The earliest references to such graphs date back to the late 1980's. These papers focus on graphs in which all vertices are adjacent to vertices with **distinct** degrees. It is not possible for all vertices of a graph to be distinct. To see this, suppose conversely that all vertices have different degrees, that is $D_G = \{0, 1, 2, \dots, n-1\}$. Then the vertex with degree $n-1$ must be adjacent to all other vertices in the graph, including the vertex of degree 0. But this is

impossible. Therefore, at least two vertices must have the same degree [1]. However, same degree vertices need not be adjacent to one another. Graphs in which all neighbors of a vertex have distinct degrees are called **highly irregular**. Figure 10 shows two highly irregular graphs.



Figure 10: Highly irregular graphs

In a 1987 paper by Alavi, Chartrand, Chung, Erdos, Graham and Oellerman [1], many interesting properties of highly irregular graphs are discussed. Some that follow directly from the definition of highly irregular graphs include: if v is a vertex of maximum degree Δ in a highly irregular graph H , then v is adjacent to only one degree t vertex for $1 \leq t \leq \Delta$; a highly irregular graph H with maximum degree Δ has at least 2Δ vertices; there are no highly irregular graphs of order 3 or 5; for $n \neq 3, 5, \text{ or } 7$ there is a highly irregular graph of order n ; the size of a highly irregular graph of order n is at most $\frac{n(n+2)}{8}$, with equality possible for n even. The paper also examines highly irregular graphs containing a given graph as an induced subgraph. The main theorem presented on this topic shows that every graph of order $n \geq 2$ is an induced subgraph of a highly irregular graph of order $4n - 4$.

In 1988 Chartrand, Erdős, and Oellerman [6], in another paper on irregular graphs, refer to graphs that may be deemed almost regular. Particularly, they discuss the notions of distance- t -

regular and locally irregular graphs. In a **distance- t -regular** graph G , every vertex v has the same number of vertices at distance t from it. Figure 11 shows a distance-2-regular graph that is not regular. The idea of distance- t -regularity differs from other ideas of irregularity in that the degrees of the vertices play only a minor role in the classification of these graphs. The paper also defines a **locally irregular** graph H to be a graph in which every vertex v in H has the property that all of its neighbors have distinct degrees. This is identical to the definition of highly irregular in [1].

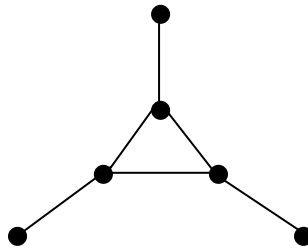


Figure 11: Distance-2-regular graph

The authors of this paper discuss another alternative class of irregular graphs. The concept of the degree of a vertex is generalized as follows: Given a graph F , the **F -degree** of a vertex v in G is the number of subgraphs of G , isomorphic to F , to which v belongs. A graph is **F -regular of degree r** if every vertex of G has F -degree r . Note that the ordinary degree of a vertex v is the K_2 -degree of v . Figure 12 illustrates a K_3 -regular graph of degree 3 that is not regular.

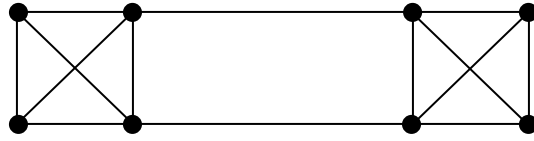


Figure 12: K_3 -regular graph of degree 3

The concept of F -regularity suggests another type of irregularity. For a graph F , a graph G is **F -irregular** if every vertex in G has a distinct F -degree. For example, the graph in Figure 13 is P_3 -irregular. The numbers next to the vertices represent the P_3 -degree of each vertex.

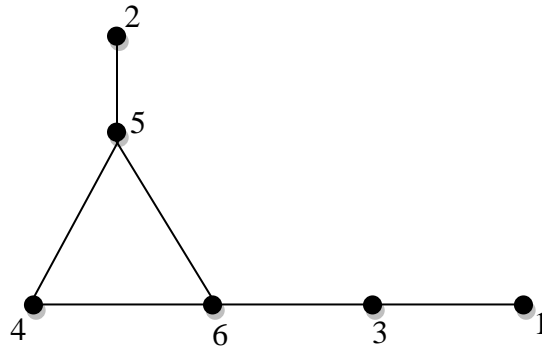


Figure 13: P_3 -irregular graph

A more recent (1997) paper by Albertson [2] defines a concept of irregularity in terms of the edges of a graph. The **imbalance** of an edge (x, y) is defined as $|\deg(x) - \deg(y)|$. The **irregularity** of the graph is then the sum of all edge imbalances. The paper shows that, given any graph G with n vertices: the irregularity is $O(n^3)$; the irregularity of G is even; if G is bipartite, its irregularity is strictly bounded from above by $\frac{n^3}{6\sqrt{3}}$; if G is triangle-free, its

irregularity is bounded above by $\frac{n^3}{9}$. While unrelated to most other concepts of irregularity, this does give insight to qualities that separate a graph from regular graphs.

Edge-Face Coloring – Literature Review

The idea of coloring the vertices of a graph in such a way that no two adjacent vertices receive the same color is not new. Its origins may be traced by to 1852 when Hamilton received a letter from his friend de Morgan which stated that one of his students had observed that when coloring the counties on a map of England, only four colors were necessary to ensure that adjacent counties were given different colors. This letter led to the more formally posed problem: “What is the least possible number of colors needed to fill any map (real or invented) on the plane?” Although it was generally believed at that time that four colors would be sufficient to color any planar graph, this conjecture was not proved until 1976 with the help of computers. The number of colors needed to color the vertices of a graph G is referred to as the **chromatic number** of G and is denoted $\chi(G)$ [16]. Figure 14 depicts a graph with eight vertices that has been colored with two colors.

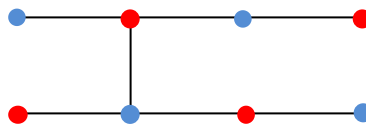


Figure 14: A graph with chromatic number 2

The problem of edge-coloring a graph arose in an 1889 paper by Tait who proved that the four-color problem is equivalent with the problem of edge-coloring every planar 3-connected cubic graph with three colors. The number of colors needed to color the edges of a graph G is called the **chromatic index** of G and is denoted $\chi'(G)$. Many properties of the chromatic index of graphs have been proved since that time. The most famous of these is Vizing's theorem, which states that a graph can be edge-colored in either Δ or $\Delta+1$ colors, where Δ is the maximum degree of the graph [21]. Figure 15 shows a graph with $\Delta = 3$ and chromatic index 3.

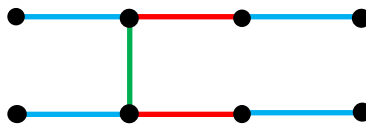


Figure 15: A graph with chromatic index 3

In the twentieth century, various coloring problems were considered involving some combination of vertices, edges, and faces of plane graphs. In 1968, Ringel studied the problem of coloring the vertices and faces of plane graphs [22]. In 1973, Kronk and Mitchem colored the vertices, edges, and faces of plane graphs [15].

Melnikov is credited as being the first to formally consider the problem of coloring the edges and faces of plane graphs. At the Graph Theory Symposium held in Prague in June 1974, he conjectured that the edges and faces of any plane graph G can be colored with $\Delta(G)+3$ colors in such a way that any two adjacent or incident edges or faces would receive different colors, where $\Delta(G)$ is the maximum degree of G . Note that the minimum number of colors required to edge-face color a graph is referred to as the graph's edge-face chromatic number and is denoted χ_{ef} .

In 1992, Hu and Zhang considered the problem of coloring the edges and faces of a plane graph, and showed Melnikov's conjecture to be true for outerplanar graphs [13]. Borodin [4], in 1994, used Kotzig's Theorem on the minimal weight of edges in plane graphs to show that for graphs with maximum degree 10 or higher, $\chi_{ef} \leq \Delta + 1$, and that this bound is tight. Lin, Hu, and Zhang proved in 1995 that Melnikov's conjecture is true for any plane graph G with $\Delta(G) \leq 3$ [21]. This result was improved by Sanders and Zhao [24] and Waller [28] independently in 1997, when Melnikov's conjecture was proved to be true in general via proofs that made use of the Four-Color Theorem. In fact, it was proved in Sanders and Zhao's paper that for every plane graph G with $\Delta(G) \geq 8$, G is $(\Delta(G) + 2)$ -edge-face colorable, and again in 2000 in a note that shows that for a plane graph G with maximum degree three, the edges and faces can be simultaneously colored with five colors ($\Delta(G) + 2$ colors) [25]. A 2001 paper, also by Sanders and Zhao [26], improved the results by proving Melnikov's conjecture with a more direct approach, and in the process also showing that for maximum degree $\Delta \leq 5$, the theorem extends to multigraphs, and for $\Delta \geq 7$, a graph can be edge-face colored with $\Delta + 2$ colors. In 2002, Wang and Lih [29] presented a new proof of Melnikov's conjecture that was independent of the Four-Color Theorem, and conjectured that in fact for any plane graph G with $\Delta(G) \geq 3$, $\chi_{ef}(G) \leq \Delta(G) + 2$. Figure 16 shows a plane graph with $\Delta(G) = 4$ edge-face colored with 6 colors.

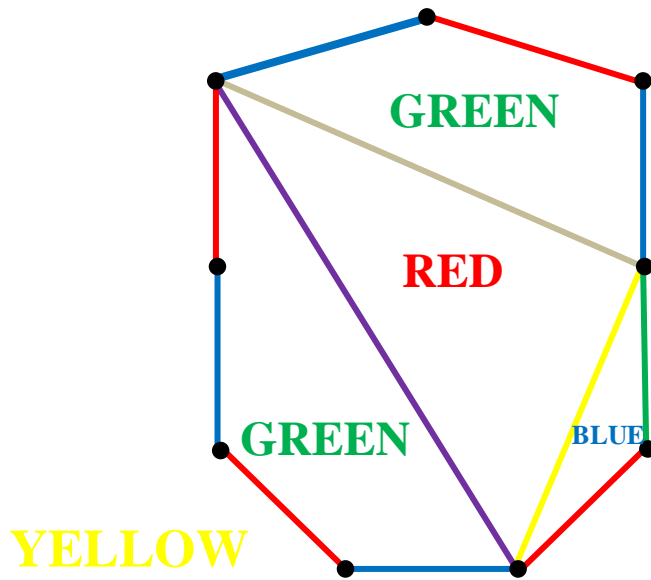


Figure 16: A graph with $\Delta(G) = 4$ edge-face colored with 6 colors

In recent years, studies have been conducted involving the edge-face chromatic number for particular classes of graphs. In 2005, Luo and Zhang [19] showed that for any 2-connected simple plane graph G with $\Delta(G) \geq 24$, $\chi_{ef} = \chi_e = \Delta(G)$. In 2007, Wang [30] proved several results for graphs embedded on surfaces of characteristic zero – specifically, that for a graph G with this property, $\chi_{ef}(G) \leq \Delta + 1$ if $\Delta \geq 13$, $\chi_{ef}(G) \leq \Delta + 2$ if $\Delta \geq 12$, $\chi_{ef}(G) \leq \Delta + 3$ if $\Delta \geq 4$, and $\chi_{ef}(G) \leq 7$ if $\Delta \leq 3$. Many classes of graphs and their extensions have yet to be studied with a focus on edge-face coloring, and many open problems remain in this area.

CHAPTER 2 AMOST REGULAR GRAPHS

Motivation

A computer network is comprised of a set of computers linked by a communication media such as copper cable or fiber-optic cable for the purpose of sharing information and/or resources. Figure 17 shows two examples of pictorial representation for computer networks.

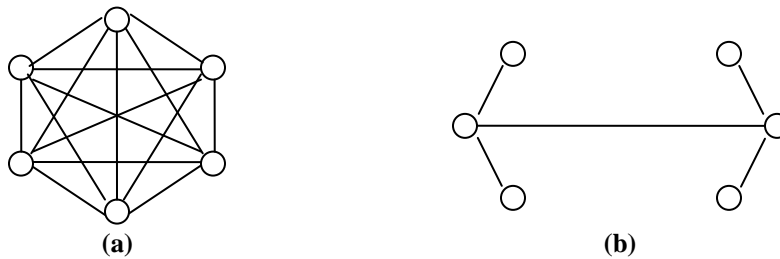


Figure 17: Graphical Representations of Computer Networks

In these representations, computers are represented by circles and the communications paths between them are represented by lines. Various network configurations are utilized to achieve different purposes. Some configurations emphasize speed and reliability by including as many communication paths as possible, while others are designed to minimize cost by providing the minimum number of paths needed to make the network function. The network shown in Figure 17 (a) emphasizes speed and reliability. Since each node is directly connected to every other node, the communication path between any arbitrary pair of nodes is as short as possible. With so many connections, this network also has the property that failure of a single line would not destroy communication between any arbitrary pair of nodes. On the other hand, the network

shown in Figure 17 (b) may require a longer communication path between any arbitrary pair of nodes, and failure of a single line may prevent communication between a pair of nodes.

However, without any such removal, this network does provide for communication between any pair of nodes, while maintaining a lower cost in terms of the number of connections between nodes.

It is essential in practice to maintain sufficient performance and reliability in computer networks while minimizing cost. Rather than haphazardly constructing a network of nodes with no particular form, it is important to design networks for which connectivity properties are well understood such that performance and cost can easily be determined. Graphs are mathematical objects that can be used to model and study computer networks. Each vertex in a graph can be used to represent either a single computer on the network or an entire sub-network, while each edge in the graph can be used to represent a communication path. We examine a particular class of graphs that could possibly be used to model networks and examine the connectivity properties of these graphs.

Objective

A *regular* graph is a graph in which all vertices have the same degree, i.e., each vertex is connected to the same number of vertices. If all vertices in a regular graph G have degree r , then G is said to be r -regular. The graphs in Figure 18 are 2-regular and 3-regular, each with six vertices.

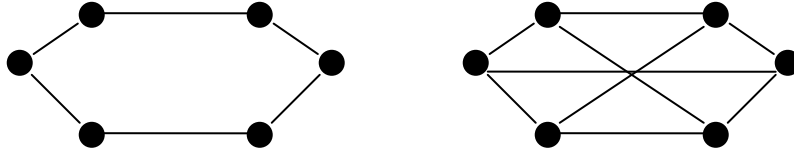


Figure 18: 2-regular and 3-regular graphs on six vertices

Regular graphs have many interesting connectivity properties that are well known. These properties are useful in light of the requirements of computer networks. However, often a regular graph contains more edges than are necessary for network applications, leading to unnecessarily high costs. It therefore becomes interesting to study graphs which are not quite regular, but differ from regular graphs in some well-defined manner. Graphs of this nature are referred to as “almost regular” or “nearly regular” graphs. Various “classes” of almost regular graphs have been devised, and their definitions can be divided into “global” forms which restrict variations throughout the entire graph and “local” forms which place limitations on the neighborhoods of vertices. Wide area networks are often restricted by local limitations, and, in light of this fact, we will be concerned with the latter type of definition. Consideration is restricted to connected graphs.

Each class is delineated by the number of vertices that keep the graph from being regular, the amount by which these vertices differ in degree from the rest of the vertices in the local subgraph, and whether or not these restrictions are *strong* or *weak*. We will concentrate our study on one of the classes of almost regular graphs known as strongly k - ϵ almost regular graphs.

Table 2 outlines the four classes of graphs considered to be almost regular.

Table 2: Classes of k - ε Almost Regular Graphs

Name of Class	Each vertex has a degree different from this quantity of its neighbors:	Each vertex differs in degree from the defined number of neighbors by:
Strongly k - ε Almost Regular	exactly k	exactly ε
Semi-Strong k - ε Almost Regular	exactly k	at most ε
Semi-Weak k - ε Almost Regular	at most k	exactly ε
Weakly k - ε Almost Regular	at most k	at most ε

Varying the integers k and ε yields a wide variety of graphs in each class. Figure 19 depicts three strongly k - ε almost regular graphs.

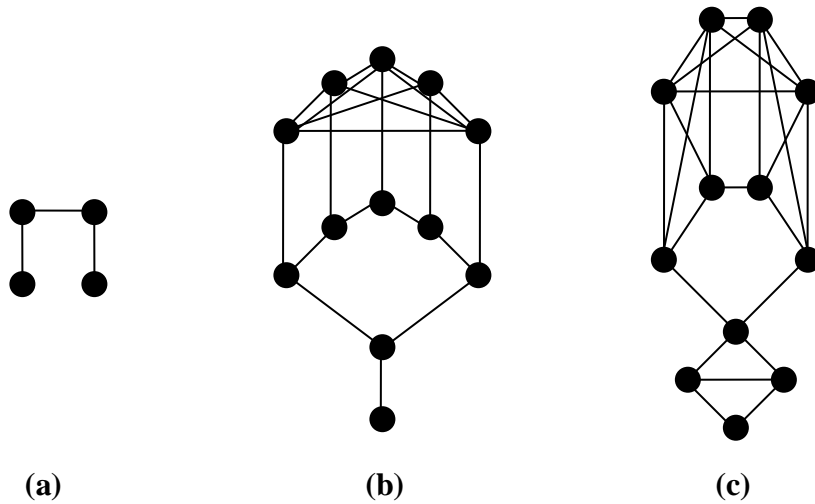


Figure 19: Strongly k - ε almost regular graphs

Figure 19 (a) shows a strongly 1-1 almost regular graph with minimum degree $\delta = 1$ and maximum degree $\Delta = 2$. Note that every vertex v has the same degree as each of its neighbors except for exactly $k = 1$, and v differs from this one vertex by exactly $\varepsilon = 1$. Figure 19 (b) depicts a strongly 1-2 almost regular graph with minimum degree $\delta = 1$ and maximum degree $\Delta = 5$. Note that every vertex in the graph has odd degree. Indeed, it is a property of strongly k - ε almost

regular graphs that all vertices have the same degree parity when ε is even. Figure 19 (c) shows a strongly 2-1 almost regular graph with minimum degree $\delta = 2$ and maximum degree $\Delta = 5$.

Outline

Section 4 of this chapter focuses on previous results concerning the order of strongly k - ε almost regular graphs and invariant values for some extremal graphs in this class. In section 5, the results for this class are extended to compute the size of extremal graphs in this class. Section 6 presents algorithms for constructing and recognizing graphs in the strongly k - ε almost regular class. These algorithms have been implemented using JavaScript and code listings with output samples are presented in the Appendices.

Strongly k - ε Almost Regular Graphs: Previous Results

Definitions and Preliminaries

An extremal strongly k - ε almost regular graph in $G_k^\varepsilon(\delta, \Delta)$ is a strongly k - ε almost regular graph with minimum degree δ and maximum degree Δ which contains the smallest possible number of vertices. The extremal strongly k - ε almost regular graph for a particular choice of k , ε , δ , and Δ is not unique in that there may be more than one configuration of vertices that gives the minimum number.

When considering any strongly k - ϵ almost regular graph, we can group the vertices by degrees. For example, in the 1-2 almost regular graph shown in Figure 20, there is one vertex of degree 1, six vertices of degree 3, and five vertices of degree 5.

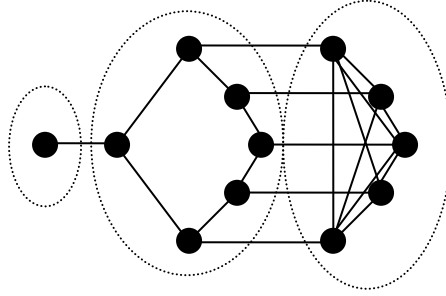


Figure 20: Strongly 1-2 almost regular graph with vertices grouped by degrees

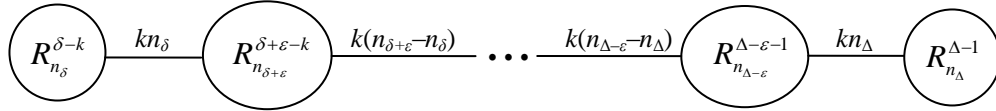
If we consider the subgraph induced by the vertices of degree i , we see that it is $(i-k)$ -regular since each vertex of degree i must be adjacent to exactly $i-k$ vertices of degree i . For example, in Figure 20, the six vertices of degree 3 form a subgraph that is 2-regular.

The vertices of degree i are said to be at *position* i in the graph. For example, in Figure 20, the vertex of degree 1 is at position 1 in the graph, while the vertices of degree 3 are at position 3. We now give the following definitions to facilitate the discussion of strongly k - ϵ almost regular graphs.

Definition: The symbol R_i^j represents any regular subgraph on i vertices with degree j , $i \geq j+1$, where i is even if j is odd.

Definition: The symbol $S_i = (p_i, R_{n_i}^{i-k}, q_i)$ represents the structure at position i in the graph. If $n_i = i - k + 1$, we can replace $R_{n_i}^{i-k}$ with K_{i-k+1} . The letters p_i and q_i represent positive integers such that $p_i + q_i = kn_i$ and refer to the number of edges between $R_{n_i}^{i-k}$ and $R_{n_{i-\varepsilon}}^{i-k-\varepsilon}$ and between $R_{n_i}^{i-k}$ and $R_{n_{i+\varepsilon}}^{i-k+\varepsilon}$, respectively. Note that R_i^j is not unique for a particular i and j . For our purposes any graph of the form R_i^j , for fixed i and j , will suffice.

Using these new symbols, we can represent a strongly k - ε almost regular graph as follows:



The circles represent the subgraphs at each position and each line encapsulates all of the edges between the structures to which it is attached. The numbers above the lines represent the total number of edges between successive structures. Because there are n_δ vertices of degree δ , each of which must be adjacent to k vertices in subgraph at position $\delta+\varepsilon$, there are kn_δ edges leaving the structure at position δ and entering the structure at position $\delta+\varepsilon$. Since there are $n_{\delta+\varepsilon}$ vertices at position $\delta+\varepsilon$, each of which is adjacent to k vertices in other structures, there must be $kn_{\delta+\varepsilon}$ edges leaving the structure at position $\delta+\varepsilon$. Because kn_δ edges are incident on the structure at position $\delta+\varepsilon$ from position δ , there must be $k(n_{\delta+\varepsilon}-n_\delta)$ edges leaving the structure at position $\delta+\varepsilon$ for vertices in the structure at position $\delta+2\varepsilon$. The parameters for the rest are found by continuing this reasoning.

In order for a subgraph of vertices all of degree r at a particular position to exist, it is necessary that either r is even or the number of vertices of degree r is even. This is because the number of vertices of odd degree in a graph must be even. Since vertices at position i are regular with degree $i-k$, n_i must be even if $i-k$ is odd.

Since every vertex of degree i must have $i-k$ neighbors of degree i , at least $i-k+1$ vertices must exist at position i . The **excess** e_i at position i is defined by $e_i = n_i - (i-k+1)$, the number of vertices at that position beyond the minimum number $i-k+1$ required.

The Order of Extremal Strongly k - ε Almost Regular Graphs

Fischer presents five fundamental lemmas that lead to the development of formulae for the number of vertices in an extremal strongly k - ε almost regular graph with minimum degree δ and maximum degree Δ [8]. We restate these lemmas and the resulting theorem without proof.

Lemma 2.1 (Reduction Lemma) Suppose a strongly k - ε almost regular graph contains adjacent structures $S_i = (p, R_{n_i}^{i-k}, q)$ and $S_{i+\varepsilon} = (q, R_{n_{i+\varepsilon}}^{i-k+\varepsilon}, r)$ for which there is an even integer x such that $kx < q$, $e_i \geq x$, and $e_{i+\varepsilon} \geq x$. Then there is a smaller strongly k - ε almost regular graph identical to the first in all positions except i and $i+\varepsilon$ whose structures are replaced by $S_i = (p, R_{n_i-kx}^{i-k}, q-kx)$ and $S_{i+\varepsilon} = (q-kx, R_{n_{i+\varepsilon}-x}^{i-k+\varepsilon}, r)$.

Lemma 2.2 For any minimum strongly k - ε almost regular graph in $G_k^\varepsilon(\delta, \Delta)$,

$$S_\Delta = (k(\Delta - k + 1), K_{\Delta-k+1}, -)$$

if k and Δ have opposite parity, and

$$S_\Delta = (k(\Delta - k + 2), R_{\Delta-k+2}^{\Delta-k}, -) \text{ or } S_\Delta = (k(\Delta - k + 1), K_{\Delta-k+1}, -)$$

if k and Δ have the same parity.

Lemma 2.3 (Push Lemma) Suppose a strongly k - ε almost regular graph G contains structures

$$S_i = (p, R_{n_i}^{i-k}, q), S_{i+\varepsilon} = (q, R_{n_{i+\varepsilon}}^{i-k+\varepsilon}, r), \text{ and } S_{i+2\varepsilon} = (r, R_{n_{i+2\varepsilon}}^{i-k+2\varepsilon}, s),$$

and also suppose that there is a positive integer x such that $q > kx$, $e_i \geq x$, and x is even if $i-k$ is odd. Then there is a different strongly k - ε almost regular graph having the same number of vertices as G and which is identical to G in all positions except i , $i+\varepsilon$, and $i+2\varepsilon$. These three positions now have the structures

$$S_i = (p, R_{n_i-x}^{i-k}, q-kx), S_{i+\varepsilon} = (q-kx, R_{n_{i+\varepsilon}}^{i-k+\varepsilon}, r+kx), \text{ and } S_{i+2\varepsilon} = (r+kx, R_{n_{i+2\varepsilon}+x}^{i-k+2\varepsilon}, s).$$

Lemma 2.4 Any not necessarily minimum strongly k - ε almost regular graph can be converted to one having the same number of vertices and for which

$$S_i = (p_i, K_{i-k+\varepsilon}, q_i) \quad \text{for } i = \delta, \delta+\varepsilon, \dots, \delta-2\varepsilon,$$

$$S_{\Delta-\varepsilon} = (p_{\Delta-\varepsilon}, R_{n_{\Delta-\varepsilon}}^{\Delta-k-\varepsilon}, q_{\Delta-\varepsilon}), \text{ and}$$

$$S_\Delta = (p_\Delta, R_{n_\Delta}^{\Delta-k}, -)$$

where p_δ is

undefined and

$$p_i = q_{i-\varepsilon} = \begin{cases} k \binom{i-\delta}{2} & \text{if } \frac{i-\delta}{\varepsilon} \text{ is even} \\ k \left(\frac{\delta-2k+i-\varepsilon+2}{2} \right) & \text{if } \frac{i-\delta}{\varepsilon} \text{ is odd} \end{cases}$$

for $i = \delta + \varepsilon, \dots, \Delta - 2\varepsilon$.

Lemma 2.5 In any minimum graph in $G_k^\varepsilon(\delta, \Delta)$ obtained by the procedure in Lemma 2.4 where

k and Δ have the same parity,

$S_\Delta = (k(\Delta - k + 1), K_{\Delta - k + 1}, -)$ if

1) ε is even

2) ε is odd, δ is of opposite parity to k and Δ , and

a) $\Delta - \delta \equiv 1 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is odd, or

b) $\Delta - \delta \equiv 3 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is even

3) ε is odd, k, δ and Δ have the same parity and $\Delta - \delta \equiv 2 \pmod{4}$

and $S_\Delta = (k(\Delta - k + 2), R_{\Delta - k + 2}^{\Delta - k}, -)$ if

1) ε is odd, k, δ and Δ have the same parity and $\Delta - \delta \equiv 0 \pmod{4}$

2) ε is odd, δ is of opposite parity to k and Δ , and

a) $\Delta - \delta \equiv 1 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is even, or

b) $\Delta - \delta \equiv 3 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is odd

Theorem 2.1. (The Order of Extremal Strongly k - ε Almost Regular Graphs)

In an extremal strongly k - ε almost regular graph with minimum degree δ and maximum degree

Δ , the number of vertices is:

$$\text{I. } \frac{\Delta^2 - \delta^2 + \varepsilon^2 + 2\varepsilon(\Delta - k) + 2k(\delta - \Delta) + 2(\Delta - \delta + \varepsilon)}{2\varepsilon} \text{ if}$$

1) ε is odd, k and δ have the same parity, and Δ has the opposite parity

2) ε is odd, δ has the opposite parity to k and Δ , and either

a) $\Delta - \delta \equiv 1 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is odd, or

b) $\Delta - \delta \equiv 3 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is even

3) ε is even, k has the same parity as δ and Δ , and $\frac{\Delta - \varepsilon - \delta}{\varepsilon}$ is even

4) ε is even, k has the opposite parity to δ and Δ , and $\frac{\Delta - \varepsilon - \delta}{\varepsilon}$ is even

$$\text{II. } \frac{\Delta^2 - \delta^2 + 2\varepsilon(\Delta + \delta - 2k) + 2k(\delta - \Delta) + 2(\Delta - \delta + 2\varepsilon)}{2\varepsilon} \text{ if}$$

1) ε is odd, k , δ , and Δ have the same parity, and $\Delta - \delta \equiv 2 \pmod{4}$

2) ε is odd, δ and Δ have the same parity and k has the opposite parity

3) ε is even, k has the same parity as δ and Δ , and $\frac{\Delta - \varepsilon - \delta}{\varepsilon}$ is odd

4) ε is even, k has the opposite parity to δ and Δ , and $\frac{\Delta - \varepsilon - \delta}{\varepsilon}$ is odd

$$\text{III. } \frac{\Delta^2 - \delta^2 + \varepsilon^2 + 2\varepsilon(\Delta - k) + 2k(\delta - \Delta) + 2(\Delta - \delta + 3\varepsilon)}{2\varepsilon} \text{ if}$$

ε is odd, δ has the opposite parity to k and Δ , and either

a) $\Delta - \delta \equiv 1 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is even, or

b) $\Delta - \delta \equiv 3 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is odd

IV. $\frac{\Delta^2 - \delta^2 + 2\varepsilon(\Delta + \delta - 2k) + 2k(\delta - \Delta) + 2(\Delta - \delta + 4\varepsilon)}{2\varepsilon}$ if

ε is odd, k , δ , and Δ have the same parity, and $\Delta - \delta \equiv 0 \pmod{4}$

Invariants of Extremal Strongly k - ε Almost Regular Graphs

In general, the computation of invariant values for extremal strongly k - ε almost regular graphs is difficult for general k and ε . However, graphs in which δ and Δ differ by exactly ε will be most useful as network models and several results are known for graphs of this type. Figure 21 depicts a strongly 3-3 almost regular graph with $\delta = 4$ and $\Delta = 7$. Note that every vertex in the graph has either degree 4 (δ) or degree 7 ($\Delta = \delta + \varepsilon$).

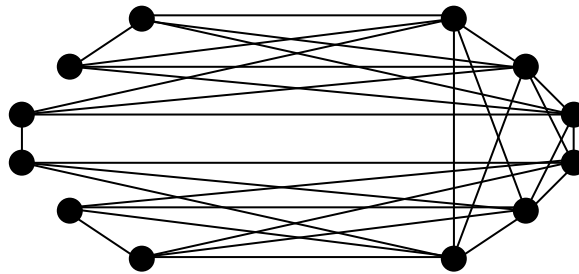


Figure 21: Strongly k - ε Almost Regular Graph with $\Delta = \delta + \varepsilon$

Because the focus is on graphs which may be useful network topologies, the most important invariants to examine may be those related to routing properties, specifically, vertex connectivity κ and edge connectivity κ_1 . The *vertex connectivity* of a graph G is the minimum

number of vertices whose removal results in a disconnected graph or the trivial graph (K_1).

Similarly, the *edge connectivity* of G is the minimum number of edges whose removal results in a disconnected graph. Due to the difficulty of evaluating these invariants in general for graphs in $G_k^\varepsilon(\delta, \Delta)$, previous results were partial in nature and assumed a large connectivity, consistent with the goal that large connectivity in networks is desirable in order to minimize disruptions when a component fails. The following result, stated without proof, gives sufficient conditions for a maximal value of κ equal to δ .

Theorem 2.2 Let positive integers k , ε , δ , and Δ be given such that $\Delta = \delta + \varepsilon$, $\delta - k$ is even, and $\varepsilon \geq k - 1$. Then there is an extremal strongly k - ε almost regular graph G having minimum degree δ and maximum degree Δ and such that $\kappa(G) = \delta$.

A similar statement can be made for edge connectivity. Note that this theorem follows immediately from Theorem 1.2 since it is known that $\kappa \leq \kappa_1 \leq \delta$.

Theorem 2.3 Let positive integers k , ε , δ , and Δ be given such that $\Delta = \delta + \varepsilon$, $\delta - k$ is even, and $\varepsilon \geq k - 1$. Then there is an extremal strongly k - ε almost regular graph G having minimum degree δ and maximum degree Δ and such that $\kappa_1(G) = \delta$.

The **diameter** d of a graph G is the maximum distance between any two vertices of G . Computer network applications favor graph models having small diameter so messages between

machines can be distributed throughout the network quickly. The next theorem, stated without proof, shows that the diameter is small for the graphs under consideration.

Theorem 2.4 In an extremal strongly k - ε almost regular graph G with $\Delta = \delta + \varepsilon$, $d \leq 3$ if

$$n_{\Delta} = \Delta - k + 1 \text{ and } d \leq 4 \text{ if } n_{\Delta} = \Delta - k + 2.$$

The **radius** r of a graph G is defined as $\min_{v \in V(G)} \left(\max_{u \in V(G)} d(u, v) \right)$. Like the diameter, the radius is

important for network considerations. The following theorem gives upper bounds for the radius of the graphs under consideration.

Theorem 2.5 In an extremal strongly k - ε almost regular graph G with $\Delta = \delta + \varepsilon$, $r \leq 2$ if

$$n_{\Delta} = \Delta - k + 1. \text{ If } n_{\Delta} = \Delta - k + 2, \text{ then } r \leq 2 \text{ if } k \geq 2 \text{ and } r \leq 3 \text{ if } k = 1.$$

A clique is a complete subgraph. The maximum clique size ω of a graph G is the number of vertices making up the largest clique in G and in a network topology refers to the largest concentration of components all of which directly interconnect. The following theorem presents bounds for the maximum clique size of the graphs under consideration.

Theorem 2.6 Assume G is a strongly k - ε almost regular graph with $\Delta = \delta + \varepsilon$. If $n_{\Delta} = \Delta - k + 1$,

then $\Delta - k + 1 \leq \omega \leq \delta + 1$ if $\varepsilon < k$ and $\omega = \Delta - k + 1$ if $\varepsilon \geq k$. If $n_{\Delta} = \Delta - k + 2$, then

$$\frac{\Delta - k + 2}{2} \leq \omega \leq \delta + 1 \text{ if } \varepsilon < \delta + k \text{ and } \omega = \frac{\Delta - k + 2}{2} \text{ if } \varepsilon \geq \delta + k.$$

The Size of Extremal Strongly k - ε Almost Regular Graphs

From the previous work reviewed in Section 4 of this chapter, we can determine the number of vertices in the extremal graphs. A related topic is the calculation of the number of edges required by these graphs. This is a straight-forward calculation based on the material from the previous chapter, but also a useful one if these graphs were applied to networks since the number of connections between various parts of the network could be crucial due to a network's requirements.

We will use the definitions, representations, lemmas and the theorem presented in Section 4 in the course of the proof that is the main focus of this chapter.

Consider any minimum graph in $G_k^\varepsilon(\delta, \Delta)$. Using the technique described in Lemma 2.4, convert the graph to one with the properties described in the Lemma. Denote the converted graph G . The number of edges in G can be calculated as the number of edges in the individual regular subgraphs of G , added to the number of edges between regular subgraphs of G . The number of edges within a regular subgraph R_i^j can be found easily, given that we know the number of vertices in this subgraph. The number of edges between adjacent regular subgraphs in G is also found using the order of each regular subgraph R_i^j . Using the results of Lemmas 2.4 and 2.5, we present the following trio of fundamental lemmas that will lead to the development of the formulae needed to calculate number of edges in an extremal strongly k - ε almost regular graph with minimum degree δ and maximum degree Δ .

Lemma 2.6 In the converted graph G , the number of edges between regular subgraphs is:

$$\text{I. } \frac{k(\Delta^2 - \delta^2 + \varepsilon^2 + 2\varepsilon(\Delta - k) + 2k(\delta - \Delta) + 2(\Delta - \delta + \varepsilon))}{4\varepsilon} \text{ if}$$

1) ε is odd, k and δ have the same parity, and Δ has the opposite parity

2) ε is odd, δ has the opposite parity to k and Δ , and either

a) $\Delta - \delta \equiv 1 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is odd, or

b) $\Delta - \delta \equiv 3 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is even

3) ε is even, k has the same parity as δ and Δ , and $\frac{\Delta - \varepsilon - \delta}{\varepsilon}$ is even

4) ε is even, k has the opposite parity to δ and Δ , and $\frac{\Delta - \varepsilon - \delta}{\varepsilon}$ is even

$$\text{II. } \frac{k(\Delta^2 - \delta^2 + 2\varepsilon(\Delta + \delta - 2k) + 2k(\delta - \Delta) + 2(\Delta - \delta + 2\varepsilon))}{4\varepsilon} \text{ if}$$

1) ε is odd, k , δ , and Δ have the same parity, and $\Delta - \delta \equiv 2 \pmod{4}$

2) ε is odd, δ and Δ have the same parity and k has the opposite parity

3) ε is even, k has the same parity as δ and Δ , and $\frac{\Delta - \varepsilon - \delta}{\varepsilon}$ is odd

4) ε is even, k has the opposite parity to δ and Δ , and $\frac{\Delta - \varepsilon - \delta}{\varepsilon}$ is odd

$$\text{III. } \frac{k(\Delta^2 - \delta^2 + \varepsilon^2 + 2\varepsilon(\Delta - k) + 2k(\delta - \Delta) + 2(\Delta - \delta + 3\varepsilon))}{4\varepsilon} \text{ if}$$

ε is odd, δ has the opposite parity to k and Δ , and either

a) $\Delta - \delta \equiv 1 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is even, or

b) $\Delta - \delta \equiv 3 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is odd

$$\text{IV. } \frac{k(\Delta^2 - \delta^2 + 2\varepsilon(\Delta + \delta - 2k) + 2k(\delta - \Delta) + 2(\Delta - \delta + 4\varepsilon))}{4\varepsilon} \text{ if}$$

ε is odd, k , δ , and Δ have the same parity, and $\Delta - \delta \equiv 0 \pmod{4}$

Proof. In the converted graph G , there are k edges connecting pairs of vertices between regular subgraphs. Therefore, the number of edges joining regular subgraphs in the G will be one-half the order of the graph multiplied by k . \square

Lemma 2.7 The number of edges in regular subgraphs in G from position δ up through position $\Delta - 2\varepsilon$, inclusive, is

$$-\frac{1}{6\varepsilon} \left((\delta - \Delta + \varepsilon)(6k^2 + 2\delta^2 + 3\Delta + 2\Delta^2 + \delta(3 + 2\Delta - 5\varepsilon) - 6k(1 + \delta + \Delta - 2\varepsilon) - 6\varepsilon - 7\Delta\varepsilon + 6\varepsilon^2) \right)$$

Proof. The number of vertices in a graph of order n that is regular with degree r is $\frac{nr}{2}$.

Therefore, the number of edges in the regular subgraphs in positions $\delta, \delta + \varepsilon, \delta + 2\varepsilon, \dots, \Delta - 3\varepsilon, \Delta - 2\varepsilon$ is

$$\sum_{i=0}^{\frac{\Delta - 2\varepsilon - \delta}{\varepsilon}} \frac{(\varepsilon i + \delta - k + 1)(\varepsilon i + \delta - k)}{2}$$

$$\begin{aligned}
&= \varepsilon^2 \sum_{i=0}^{\frac{\Delta-2\varepsilon-\delta}{\varepsilon}} i^2 + [2\varepsilon(\delta-k) + \varepsilon] \sum_{i=0}^{\frac{\Delta-2\varepsilon-\delta}{\varepsilon}} i + [(\delta-k)^2 + \delta - k] \sum_{i=0}^{\frac{\Delta-2\varepsilon-\delta}{\varepsilon}} 1 \\
&= \varepsilon^2 \left[\frac{1}{3} \left(\frac{\Delta-\varepsilon-\delta}{\varepsilon} \right)^3 - \frac{1}{2} \left(\frac{\Delta-\varepsilon-\delta}{\varepsilon} \right)^2 + \frac{1}{2} \left(\frac{\Delta-\varepsilon-\delta}{\varepsilon} \right) \right] + [2\varepsilon(\delta-k) + \varepsilon] \left[\frac{1}{2} \left(\frac{\Delta-2\varepsilon-\delta}{\varepsilon} \right) \left(\frac{\Delta-\varepsilon-\delta}{\varepsilon} \right) \right] \\
&\quad + [(\delta-k)^2 + \delta - k] \left[\frac{\Delta-\varepsilon-\delta}{\varepsilon} \right] \\
&= -\frac{1}{6\varepsilon} \left((\delta-\Delta+\varepsilon)(6k^2 + 2\delta^2 + 3\Delta + 2\Delta^2 + \delta(3+2\Delta-5\varepsilon) - 6k(1+\delta+\Delta-2\varepsilon) - 6\varepsilon - 7\Delta\varepsilon + 6\varepsilon^2) \right)
\end{aligned}$$

□

Lemma 2.8 The number of edges in the subgraphs in positions $\Delta-\varepsilon$ and Δ is:

$$\frac{1}{4} (4k^2 + 5\Delta^2 + \varepsilon(-2 + \delta + \varepsilon) + k(-4 + \delta - 9\Delta + 3\varepsilon) - \Delta(-4 + \delta + 4\varepsilon)) \text{ if}$$

1) ε is odd, k and δ have the same parity, and Δ has the opposite parity

2) ε is odd, δ has the opposite parity to k and Δ , and either

c) $\Delta - \delta \equiv 1 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is odd, or

d) $\Delta - \delta \equiv 3 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is even

3) ε is even, k has the same parity as δ and Δ , and $\frac{\Delta-\varepsilon-\delta}{\varepsilon}$ is even

4) ε is even, k has the opposite parity to δ and Δ , and $\frac{\Delta-\varepsilon-\delta}{\varepsilon}$ is even

$$\frac{1}{4} (2(-k+\Delta)(1-k+\Delta) + (k-\Delta+\varepsilon)(-4+4k-\delta-3\Delta+2\varepsilon)) \text{ if}$$

- 1) ε is odd, k , δ , and Δ have the same parity, and $\Delta - \delta \equiv 2 \pmod{4}$
- 2) ε is odd, δ and Δ have the same parity and k has the opposite parity
- 3) ε is even, k has the same parity as δ and Δ , and $\frac{\Delta - \varepsilon - \delta}{\varepsilon}$ is odd
- 4) ε is even, k has the opposite parity to δ and Δ , and $\frac{\Delta - \varepsilon - \delta}{\varepsilon}$ is odd

$$\frac{1}{4}(2(-k + \Delta)(2 - k + \Delta) + (k - \Delta + \varepsilon)(-4 + 2k + \delta - 3\Delta + \varepsilon)) \text{ if}$$

ε is odd, δ has the opposite parity to k and Δ , and either

$$\text{a) } \Delta - \delta \equiv 1 \pmod{4} \text{ and } \left\lfloor \frac{\varepsilon}{2} \right\rfloor \text{ is even, or}$$

$$\text{b) } \Delta - \delta \equiv 3 \pmod{4} \text{ and } \left\lfloor \frac{\varepsilon}{2} \right\rfloor \text{ is odd}$$

$$\frac{1}{4}(2(-k + \Delta)(2 - k + \Delta) + (k - \Delta + \varepsilon)(-6 + 4k - \delta - 3\Delta + 2\varepsilon)) \text{ if}$$

ε is odd, k , δ , and Δ have the same parity, and $\Delta - \delta \equiv 0 \pmod{4}$

Proof. We examine each case separately:

Case 1

Position Δ has $\Delta - k + 1$ vertices regular with degree $\Delta - k$ by Lemma 2.5. Therefore, the subgraph

in that position contains $\frac{(\Delta - k + 1)(\Delta - k)}{2}$ edges.

Position $\Delta-\varepsilon$ has $\Delta-k+1+\frac{\Delta-\delta-\varepsilon}{2}$ vertices regular with degree $\Delta-k-\varepsilon$ by Lemmas 2.4 and 2.5.

Therefore, the subgraph in that position contains $\left(\frac{3\Delta-4k+\delta-2\varepsilon+2}{2}\right)\left(\frac{\Delta-k-\varepsilon}{2}\right)$ edges.

Therefore, positions Δ and $\Delta-\varepsilon$ together contain

$$\begin{aligned} & \frac{(\Delta-k+1)(\Delta-k)}{2} + \left(\frac{3\Delta-4k+\delta-2\varepsilon+2}{2}\right)\left(\frac{\Delta-k-\varepsilon}{2}\right) \\ &= \frac{1}{4}\left(4k^2+5\Delta^2+\varepsilon(-2+\delta+\varepsilon)+k(-4+\delta-9\Delta+3\varepsilon)-\Delta(-4+\delta+4\varepsilon)\right) \text{ edges.} \end{aligned}$$

Case 2

Position Δ has $\Delta-k+1$ vertices regular with degree $\Delta-k$ by Lemma 2.5. Therefore, the subgraph in that position contains $\frac{(\Delta-k+1)(\Delta-k)}{2}$ edges.

Position $\Delta-\varepsilon$ has $\Delta-k+1+\frac{\delta-2k+\Delta-2\varepsilon+2}{2}$ vertices regular with degree $\Delta-k-\varepsilon$ by Lemmas

2.4 and 2.5. Therefore, the subgraph in that position contains $\left(\frac{3\Delta-4k+\delta-2\varepsilon+4}{2}\right)\left(\frac{\Delta-k-\varepsilon}{2}\right)$ edges.

Therefore, positions Δ and $\Delta-\varepsilon$ together contain

$$\frac{(\Delta-k+1)(\Delta-k)}{2} + \left(\frac{3\Delta-4k+\delta-2\varepsilon+4}{2}\right)\left(\frac{\Delta-k-\varepsilon}{2}\right)$$

$$= \frac{1}{4}(2(-k + \Delta)(1 - k + \Delta) + (k - \Delta + \varepsilon)(-4 + 4k - \delta - 3\Delta + 2\varepsilon)) \text{ edges.}$$

Case 3

Position Δ has $\Delta - k + 2$ vertices regular with degree $\Delta - k$ by Lemma 2.5. Therefore, the subgraph

in that position contains $\frac{(\Delta - k + 2)(\Delta - k)}{2}$ edges.

Position $\Delta - \varepsilon$ has $\Delta - k + 2 + \frac{\Delta - \varepsilon - \delta}{2}$ vertices regular with degree $\Delta - k - \varepsilon$ by Lemmas 2.4 and

2.5. Therefore, the subgraph in that position contains $\left(\frac{3\Delta - 2k - \delta - \varepsilon + 4}{2}\right)\left(\frac{\Delta - k - \varepsilon}{2}\right)$ edges.

Therefore, positions Δ and $\Delta - \varepsilon$ together contain

$$\begin{aligned} & \frac{(\Delta - k + 2)(\Delta - k)}{2} + \left(\frac{3\Delta - 2k - \delta - \varepsilon + 4}{2}\right)\left(\frac{\Delta - k - \varepsilon}{2}\right) \\ &= \frac{1}{4}(2(-k + \Delta)(2 - k + \Delta) + (k - \Delta + \varepsilon)(-4 + 2k + \delta - 3\Delta + \varepsilon)) \text{ edges.} \end{aligned}$$

Case 4

Position Δ has $\Delta - k + 2$ vertices regular with degree $\Delta - k$ by Lemma 2.5. Therefore, the subgraph

in that position contains $\frac{(\Delta - k + 2)(\Delta - k)}{2}$ edges.

Position $\Delta-\varepsilon$ has $\Delta-k+2+\frac{\delta-2k+\Delta-2\varepsilon+2}{2}$ vertices regular with degree $\Delta-k-\varepsilon$ by Lemmas

2.4 and 2.5. Therefore, the subgraph in that position contains $\left(\frac{3\Delta-4k+\delta-2\varepsilon+6}{2}\right)\left(\frac{\Delta-k-\varepsilon}{2}\right)$ edges.

Therefore, positions Δ and $\Delta-\varepsilon$ together contain

$$\begin{aligned} & \frac{(\Delta-k+2)(\Delta-k)}{2} + \left(\frac{3\Delta-4k+\delta-2\varepsilon+6}{2}\right)\left(\frac{\Delta-k-\varepsilon}{2}\right) \\ &= \frac{1}{4}(2(-k+\Delta)(2-k+\Delta) + (k-\Delta+\varepsilon)(-6+4k-\delta-3\Delta+2\varepsilon)) \text{ edges. } \square \end{aligned}$$

Theorem 2.7 (The Size of Extremal Strongly $k-\varepsilon$ Almost Regular Graphs)

In an extremal $k-\varepsilon$ almost regular graph with minimum degree δ and maximum degree Δ , the number of edges is:

$$\begin{aligned} \text{I. } & \frac{1}{12} \left(-3k(\delta-\Delta-\varepsilon)\varepsilon(2-2k+\delta+\Delta+\varepsilon) \right. \\ & - \frac{1}{\varepsilon} \left(2(\delta-\Delta+\varepsilon)(6k^2+2\delta^2+3\Delta+2\Delta^2+\delta(3+2\Delta-5\varepsilon))-6k(1+\delta+\Delta-2\varepsilon)-6\varepsilon-7\Delta\varepsilon+6\varepsilon^2) \right) \\ & \left. + 3(4k^2+5\Delta^2+\varepsilon(-2+\delta+\varepsilon)+k(-4+\delta-9\Delta+3\varepsilon)-\Delta(-4+\delta+4\varepsilon)) \right) \text{ if} \end{aligned}$$

- 1) ε is odd, k and δ have the same parity, and Δ has the opposite parity
- 2) ε is odd, δ has the opposite parity to k and Δ , and either

$$\text{e) } \Delta-\delta \equiv 1 \pmod{4} \text{ and } \left\lfloor \frac{\varepsilon}{2} \right\rfloor \text{ is odd, or}$$

f) $\Delta - \delta \equiv 3 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is even

3) ε is even, k has the same parity as δ and Δ , and $\frac{\Delta - \varepsilon - \delta}{\varepsilon}$ is even

4) ε is even, k has the opposite parity to δ and Δ , and $\frac{\Delta - \varepsilon - \delta}{\varepsilon}$ is even

$$\text{II. } \frac{1}{12} \left(-3k(2 - 2k + \delta + \Delta)(\delta - \Delta - 2\varepsilon)\varepsilon \right.$$

$$\left. - \frac{1}{\varepsilon} (2(\delta - \Delta + \varepsilon)(6k^2 + 2\delta^2 + 3\Delta + 2\Delta^2 + \delta(3 + 2\Delta - 5\varepsilon) - 6k(1 + \delta + \Delta - 2\varepsilon) - 6\varepsilon - 7\Delta\varepsilon + 6\varepsilon^2)) \right)$$

$$+ 3(2(-k + \Delta)(1 - k + \Delta) + (k - \Delta + \varepsilon)(-4 + 4k - \delta - 3\Delta + 2\varepsilon))) \text{ if}$$

1) ε is odd, k , δ , and Δ have the same parity, and $\Delta - \delta \equiv 2 \pmod{4}$

2) ε is odd, δ and Δ have the same parity and k has the opposite parity

3) ε is even, k has the same parity as δ and Δ , and $\frac{\Delta - \varepsilon - \delta}{\varepsilon}$ is odd

4) ε is even, k has the opposite parity to δ and Δ , and $\frac{\Delta - \varepsilon - \delta}{\varepsilon}$ is odd

$$\text{III. } \frac{1}{12} \left(3k\varepsilon(2(-1 + k)\delta - \delta^2 + \Delta^2 + \varepsilon(6 - 2k + \varepsilon) + \Delta(2 - 2k + 2\varepsilon)) \right.$$

$$\left. - \frac{1}{\varepsilon} (2(\delta - \Delta + \varepsilon)(6k^2 + 2\delta^2 + 3\Delta + 2\Delta^2 + \delta(3 + 2\Delta - 5\varepsilon) - 6k(1 + \delta + \Delta - 2\varepsilon) - 6\varepsilon - 7\Delta\varepsilon + 6\varepsilon^2)) \right)$$

$$+ 3(2(-k + \Delta)(2 - k + \Delta) + (k - \Delta + \varepsilon)(-4 + 2k + \delta - 3\Delta + \varepsilon))) \text{ if}$$

ε is odd, δ has the opposite parity to k and Δ , and either

a) $\Delta - \delta \equiv 1 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is even, or

b) $\Delta - \delta \equiv 3 \pmod{4}$ and $\left\lfloor \frac{\varepsilon}{2} \right\rfloor$ is odd

IV. $\frac{1}{12} (3k\varepsilon(\delta^2 - \Delta^2 + 2\Delta(-1+k-\varepsilon)) + 4(-2+k)\varepsilon - 2\delta(-1+k+\varepsilon))$

$-\frac{1}{\varepsilon} (2(\delta - \Delta + \varepsilon)(6k^2 + 2\delta^2 + 3\Delta + 2\Delta^2 + \delta(3 + 2\Delta - 5\varepsilon) - 6k(1 + \delta + \Delta - 2\varepsilon) - 6\varepsilon - 7\Delta\varepsilon + 6\varepsilon^2))$

$+ 3(2(-k + \Delta)(2 - k + \Delta) + (k - \Delta + \varepsilon)(-6 + 4k - \delta - 3\Delta + 2\varepsilon))$ if

ε is odd, k , δ , and Δ have the same parity, and $\Delta - \delta \equiv 0 \pmod{4}$

Proof. Add the results from Lemmas 2.6, 2.7 and 2.8 for each case. Simplification leads to the results shown for the number of edges in the graph G . Since Lemma 2.4 shows that the conversion process does not alter the number of edges and vertices in the minimum graph, the minimum graph has the same number of edges as G . \square

Strongly k - ε Almost Regular Graph Algorithms

A Linear-Time Algorithm for Creation

The structure of a minimum strongly k - ε almost regular graph with minimum degree δ and maximum degree Δ after conversion using the fundamental lemmas from Section 4 leads to the development of an linear-time algorithm to create such a graph for given k , ε , δ , and Δ .

Algorithm 2.1 (Creation of extremal strongly k - ε almost regular graphs with given k , ε , δ , and Δ)

Input: Integers k , ε , δ and Δ .

Idea: Set positions $i = \delta$ through $i = \Delta - 2\varepsilon$ to the structure $S_i = K_{i-k+1}$. Determine whether position Δ has the structure $S_\Delta = K_{\Delta-k+1}$ or $S_\Delta = K_{\Delta-k+2}$. Determine the structure at position $i = \Delta - \varepsilon$ based on the structures at positions $i = \Delta - 2\varepsilon$ and $i = \Delta$.

Initialization: Create three arrays to store the number of vertices at each position i (\mathbf{v}), the number of edges from each position adjacent to vertices at position $i - \varepsilon$ (\mathbf{p}) and the number of edges from each position adjacent to vertices at position $i + \varepsilon$ (\mathbf{q}). Note that $\mathbf{p}[\delta]$ and $\mathbf{q}[\Delta]$ are zero.

Process:

1. Set positions $i = \delta$ through $i = \Delta - 2\varepsilon$ to the structure $S_i = K_{i-k+1}$.
2. Assume that $S_\Delta = K_{i-k+1}$. Examine the parity $p1$ of $\mathbf{v}[\Delta] = \Delta - k + 1 + \mathbf{q}[\Delta - 2\varepsilon]/k$, which represents the number of vertices that must be at position $i = \Delta - \varepsilon$ under the assumption. Examine the parity $p2$ of $\Delta - k - \varepsilon$, which represents the degree of the regular subgraph at position $i = \Delta - \varepsilon$. If both $p1$ and $p2$ are odd, add one vertex at position Δ so that $S_\Delta = K_{i-k+2}$, since we cannot have a regular subgraph on an odd number of vertices of odd degree.
3. Set $S_{\Delta-\varepsilon} = K_{\mathbf{q}[\Delta-2\varepsilon]+\mathbf{p}[\Delta]}$.

4. Return **v**, **p** and **q**.

Analysis: This algorithm is linear in the number of positions in the graph, $\frac{\Delta - \delta}{\varepsilon} + 1$. At most five operations are performed for each position in the graph.

An implementation of this algorithm in JavaScript with sample output for various parameter values is given in Appendix I. This program is available on the Web at <http://web.valencia.cc.fl.us/cpa/lmacon/ARke.html>.

A Polynomial-Time Algorithm for Recognition

Recognition of strongly k - ε almost regular graphs, including the recognition of the values for k , ε , δ , and Δ in a particular graph, depends in part on the representation of the graphs within the program. Noting that these graphs, if used as network topology models, can be very large in general, we avoid storing the adjacency matrix and instead store arbitrary graphs using an adjacency list stored as an XML (eXtensible Markup Language) document.

Developed in 1996 and released in 1998 by the World Wide Web Consortium, XML is a markup language designed to store information for use by Web applications. Like Hypertext Markup Language (HTML) documents, XML documents consist of tags (elements) that enclose content (data). Unlike HTML, there is no fixed set of tags. Rather, the developer defines tags suitable to the application.

XML is particularly suited for adjacency list storage, as an XML document itself is stored in RAM after being loaded as a node tree. We can use arbitrary tag names to identify elements within the document and thus store only information about adjacencies, providing the means to store dense graphs, yet there is no need to store more information than necessary about sparse graphs.

As an example, consider the graph shown in Figure 22. This graph is a strongly 1-2 almost regular graph. Each vertex has been assigned a label for identification. Note that the labeling begins at zero. This is simply to accommodate the looping mechanisms of most high-level languages.

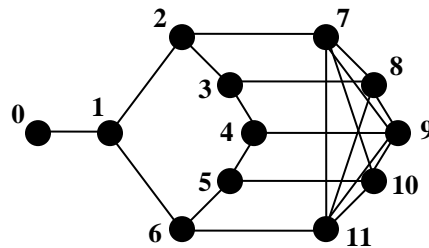


Figure 22: Strongly 1-2 almost regular graph

This graph can be represented in XML using the code shown in Figure 23, where the element `<graph>` stores the top-level element for the document and the element `<vertex>` stores data about each vertex. The `order` attribute, used within the `<graph>` tag, provides the number of vertices for the graphs. This attribute is used for looping purposes within the code. The `label` attribute, used within the `<vertex>` tag, delineates the vertex that is currently being described. The `<neighbor>` element, located within the `<vertex>` element, stores adjacency information for each neighbor. Note that these choices for element and attribute names are arbitrary and can be changed easily.

```

<graph order="12">
  <vertex label="0">
    <neighbor label="1"/>
  </vertex>
  <vertex label="1">
    <neighbor label="0"/>
    <neighbor label="2"/>
    <neighbor label="6"/>
  </vertex>
  <vertex label="2">
    <neighbor label="1"/>
    <neighbor label="3"/>
    <neighbor label="7"/>
  </vertex>
  <vertex label="3">
    <neighbor label="2"/>
    <neighbor label="4"/>
    <neighbor label="8"/>
  </vertex>
  <vertex label="4">
    <neighbor label="3"/>
    <neighbor label="5"/>
    <neighbor label="9"/>
  </vertex>
  <vertex label="5">
    <neighbor label="4"/>
    <neighbor label="6"/>
    <neighbor label="10"/>
  </vertex>
  <vertex label="6">
    <neighbor label="1"/>
    <neighbor label="5"/>
    <neighbor label="11"/>
  </vertex>
  <vertex label="7">
    <neighbor label="2"/>
    <neighbor label="8"/>
    <neighbor label="9"/>
    <neighbor label="10"/>
    <neighbor label="11"/>
  </vertex>
  <vertex label="8">
    <neighbor label="3"/>
    <neighbor label="7"/>
    <neighbor label="9"/>
    <neighbor label="10"/>
    <neighbor label="11"/>
  </vertex>
  <vertex label="9">
    <neighbor label="4"/>
    <neighbor label="7"/>
    <neighbor label="8"/>
    <neighbor label="10"/>
    <neighbor label="11"/>
  </vertex>

```

```

</vertex>
<vertex label="10">
  <neighbor label="5"/>
  <neighbor label="7"/>
  <neighbor label="8"/>
  <neighbor label="9"/>
  <neighbor label="11"/>
</vertex>
<vertex label="11">
  <neighbor label="6"/>
  <neighbor label="7"/>
  <neighbor label="8"/>
  <neighbor label="9"/>
  <neighbor label="10"/>
</vertex>
</graph>

```

Figure 23: XML Representation of a Strongly 1-2 Almost Regular Graph

Given a graph represented using the XML scheme shown in Figure 23, the following algorithm will check the graph for the properties representative of strongly k - ε almost regular graphs. If the graph satisfies the properties for any k and ε , this information will be reported, along with the values for the parameters. If the graph does not satisfy the properties, this will be indicated on output.

Algorithm 2.2 (Recognition of extremal strongly k - ε almost regular graphs for any k and ε)

Input: XML representation of a simple, connected graph G .

Idea: Choose one vertex v_0 in the graph. Compare degrees of adjacent vertices. Count the number with different degrees (k) and the amount by which the degrees differ from the degree of v_0 – verify that this is the same in absolute value for all neighbors (this value is ε). We can do this because k and ε must be the same for all vertices by definition.

Examine degrees of adjacent vertices for each vertex in the graph. If each vertex v has exactly k neighbors whose degrees differ from the degree of v by exactly ε , and all other neighbors of v have the same degree as v , the graph is k - ε almost regular.

Initialization: Create an array \mathbf{s} to store the number of adjacent vertices with the same degree for each vertex v and an array \mathbf{d} to store the number of vertices with degrees differing from the degree of v by exactly ε . Initialize all elements of both arrays to zero.

Create an array to store the degrees of each vertex in the graph and compute these degrees for each vertex based on the adjacency matrix.

Iteration: For each vertex v

1. Verify that k neighbors differ in degree from v .
2. Examine the degree of each vertex w adjacent to v . If the degree of w is the same as the degree of v , then $\mathbf{s}[v]=\mathbf{s}[v]+1$. If the degree of w is $\text{degree}(v) - \varepsilon$ or $\text{degree}(v) + \varepsilon$, then $\mathbf{d}[v]=\mathbf{d}[v]+1$. Otherwise, break – the graph is not k - ε almost regular.
3. If $\mathbf{d}[v]$ is not equal to k , break. Otherwise, continue.

Return true if break does not occur after all vertices have been examined.

Analysis: This algorithm is polynomial in the order of the graph G . It takes at most $O(p^2)$ iterations to examine the neighbors of each vertex in the graph for a graph of order p .

An implementation of this algorithm in JavaScript with sample output for various parameter values is given in Appendix II. This program is available on the Web at <http://web.valencia.cc.fl.us/cpa/lmacon/recognize2xml.html>. The appendix includes the XML document representations for various graphs. Images are provided for each represented graph.

Note that if it is desirable to recognize the strongly k - ε almost regular graph properties in a graph for *particular* values of k and ε , the algorithm is a simplification of Algorithm 2.2. A polynomial time algorithm with short-circuiting is given below.

Algorithm 2.3 (Recognition of strongly k - ε almost regular graph properties for given values of k and ε)

Input: Integers k and ε , and a simple, connected graph G .

Idea: Examine degrees of adjacent vertices for each vertex in the graph. If each vertex v has exactly k neighbors whose degrees differ from the degree of v by exactly ε , and all other neighbors of v have the same degree as v , the graph is k - ε almost regular.

Initialization: Create an array \mathbf{s} to store the number of adjacent vertices with the same degree for each vertex v and an array \mathbf{d} to store the number of vertices with degrees differing from the degree of v by exactly ε . Initialize all elements of both arrays to zero.

Create an array to store the degrees of each vertex in the graph and compute these degrees for each vertex based on the adjacency matrix.

Iteration: For each vertex v

1. Examine the degree of each vertex w adjacent to v . If the degree of w is the same as the degree of v , then $s[v] = s[v] + 1$. If the degree of w is $\text{degree}(v) - \epsilon$ or $\text{degree}(v) + \epsilon$, then $\mathbf{d}[v] = \mathbf{d}[v] + 1$. Otherwise, break – the graph is not $k - \epsilon$ almost regular.
2. If $\mathbf{d}[v]$ is not equal to k , break. Otherwise, continue.

Return true if break does not occur after all vertices have been examined.

Analysis: This algorithm is polynomial in the order of the graph G .

CHAPTER 3 EDGE-FACE COLORINGS OF PLANE GRAPHS

Motivation and Notation

In 1994, Borodin [4] proved that for any plane graph G with $\Delta(G) \geq 10$, $\chi_{ef} \leq \Delta + 1$ and this bound is sharp. In the same paper, Borodin posed the problem of finding a precise upper bound for plane graphs G with $\Delta(G) \leq 9$.

The word *pseudograph* is used in this study to allow loops and multiple edges, while the word *graph* serves to prohibit them. Let $G = (V(G), E(G), F(G))$ be a plane pseudograph where $V(G)$, $E(G)$ and $F(G)$ are the vertex set, edge set and face set of G respectively. Let $x, y \in V(G) \cup E(G) \cup F(G)$. For convenience, we say that x and y are *adjacent* if x and y are either adjacent or incident in a conventional sense. An *edge-face coloring* of a plane pseudograph G is a function $f: E(G) \cup F(G) \rightarrow \{1, \dots, k\}$ such that $f(x) \neq f(y)$ if x and y are adjacent. When needed, we allow loops and self-adjacent faces, and ignore the contact of a color to itself in this exceptional case. A plane pseudograph is *edge-face k -colorable* if there is an edge-face coloring of the graph with colors from $\{1, \dots, k\}$. We use $\chi_{ef}(G)$, $\Delta(G)$, and $\delta(G)$ to denote the *edge-face chromatic number*, *maximum degree* and *minimum degree* of G , respectively.

Objective

It is not difficult to see that if G is a plane graph with $\Delta(G) = 2$, then $\chi_{ef}(G) \leq 5$ and this bound is sharp. Sanders and Zhao [25] showed in 1998 that if G is a plane graph with $\Delta(G) = 3$,

then $\chi_{ef}(G) \leq 5$, and this bound is sharp as well. We now turn our attention to the case where $\Delta(G) = 9$ and show that for graphs with this property, $\chi_{ef}(G) \leq 10$, and that this bound is sharp.

Outline

To show that if G is a plane graph with $\Delta(G) = 9$ then $\chi_{ef}(G) \leq 10$, we consider plane pseudographs instead of plane graphs and prove that for a graph G in this category with $\Delta(G) = 9$ and minimum degree $\delta(G) \geq 2$, if G has no ≤ 2 -faces (loops or multiple edges), then $\chi_{ef}(G) \leq 10$. When that theorem is applied to plane graphs, the main result is proved.

A Precise Upper Bound for the Edge-Face Chromatic Number of Plane Graphs with $\Delta = 9$

Theorem 3.1. Let G be a plane pseudograph with $\Delta(G) = 9$ and $\delta(G) \geq 2$. If G has no ≤ 2 -faces, then $\chi_{ef}(G) \leq 10$.

Before this theorem is proved, we will introduce further notation. A plane pseudograph G is called a *minimal pseudograph* if G is a counterexample to our theorem with a minimal number of edges. Let an *i - j edge* be an edge that is adjacent to an i -vertex and a j -vertex. Let $e \in E(G)$ and let f be a face adjacent to e . Then, the other face adjacent to e is denoted by f_e , noting that it is possible that $f_e = f$. Furthermore, for $e \in E(G)$, let G/e denote the graph that is obtained by contracting e in G . A *k -vertex*, $\geq k$ -vertex or $\leq k$ -vertex is a vertex of degree k , at least k , or at most k . Similarly, we can define a *k -face*, $\geq k$ -face or $\leq k$ -face. For $x \in V(G) \cup F(G)$, $d(x)$ shall denote the degree of x if $x \in V(G)$ or the length of the facial walk of x if $x \in F(G)$. Let x be a

vertex of G . We denote the number of i -faces adjacent to x by $f_i(x)$. Similarly, we can define $f_{\geq i}(x)$. A *partial edge-face coloring* of a plane pseudograph G is a coloring of a subset of $E(G) \cup F(G)$ such that distinct, adjacent, colored elements of $E(G) \cup F(G)$ have different colors.

The following lemmas prove useful properties of minimal pseudographs.

Lemma 3.1. Let G be a minimal pseudograph. The G satisfies the following properties:

- (1) G contains no i - j edge e if $i + j \leq 9$ and e is adjacent to a ≤ 5 -face. Moreover, G contains no 2 - j edge e if $j \leq 7$.
- (2) G contains no i - j edge if $i + j = 10$ and e is adjacent to a ≤ 4 -face. Moreover, G contains no 2 - j edge e if $j \leq 8$ and e is adjacent to a ≤ 5 -face.
- (3) G contains no i - j edge e if $i + j = 11$ and e is adjacent to a ≤ 4 -face and a ≤ 5 -face. Moreover, G contains no 2 - j edge e if e is adjacent to two ≤ 5 -faces.

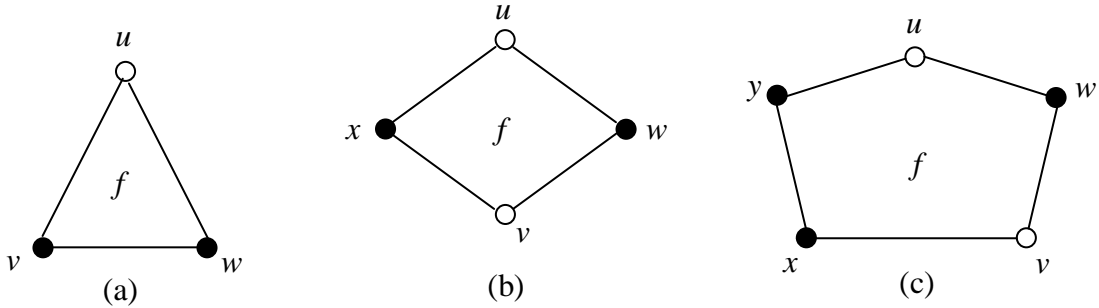
Proof.

(1) For the first part of (1), suppose to the contrary that G contains an i - j edge e that is adjacent to an i -vertex v , a j -vertex u and a ≤ 5 -face f . Let $G_0 = G - e$ and G_i be obtained from G_{i-1} by deleting a 1-vertex of G_{i-1} if such a vertex exists. Clearly, there is an integer $k \geq 0$ such that either $k = 0$, $\delta(G_0) \geq 2$ or $k \geq 1$, $\delta(G_k) \geq 2$, but $\delta(G_{k-1}) = 1$. By the minimality of G , G_k has an edge-face 10-coloring. This induces a partial edge-face 10-coloring of G , where f_e inherits the color of the face of G_k not in G , and only edges in $E(G) - E(G_k)$ and f are uncolored. Since each of the uncolored edge and f are adjacent to at most nine elements in $E(G) \cup F(G) - (E(G) - E(G_k))$, we can color f first, then color these uncolored edges one by one. Hence G is edge-face

10-colorable, a contradiction. Thus the first part of (1) is true. For the second part of (1), suppose to the contrary that G contains a $2-j$ edge e with $j \leq 7$. By the first part of (1), e is not adjacent to any ≤ 5 -faces. Consider G/e . Since G is minimal, G/e is edge-face 10-colorable. An edge-face 10-coloring of G/e induces a partial edge-face 10-coloring of G with only e uncolored. Since e is adjacent to at most nine elements in $E(G) \cup F(G)$, it is clear that we can color e . Thus G is edge-face 10-colorable, another contradiction.

(2) and (3) Suppose that G contains an $i-j$ edge e that is adjacent to an i -vertex v , a j -vertex u and a ≤ 4 -face f . Moreover, $d(f_e) \leq 5$. Using the technique used in (1), we obtain G_k . By the minimality of G , G_k has an edge-face 10-coloring. This induces a partial edge-face 10-coloring of G , where f_e inherits the color of the face of G_k not in G , and only edges in $E(G) - E(G_k)$ and f are uncolored. In (2), since each of the uncolored edges and f are adjacent to at most nine colored elements, we can color these uncolored edges one by one, and then f . In (3), remove the color from f_e . Since each of the uncolored edges, f_e and f are adjacent to at most nine colored elements, we can first color those uncolored edges one by one, then f_e , and then f . Hence G has an edge-face 10-coloring, a contradiction. Thus, the first parts of (2) and (3) are true. For the second parts of (2) and (3), suppose to the contrary that G contains a $2-j$ edge e . Let f be a 5-face adjacent to e in (2), and f and g be ≤ 5 -faces adjacent to e in (3). By the above proof, f and g are 5-faces. Consider G/e . Since G is minimal, G/e is edge-face 10-colorable. An edge-face 10-coloring of G/e induces a partial edge-face 10-coloring of G with e uncolored. Remove the colors from f in (2), and f and g in (3). In (2), we color e first followed by f , and in (3) we color e first followed by f and g . Thus G is edge-face 10-colorable, a contradiction. \square

Lemma 3.2. Let G be a minimal pseudograph. The G does not contain any of the configurations in Figure 24.



u, v, w, x, y are vertices and f is a face

Each hollow circle represents a 2-vertex

Figure 24: Forbidden subgraphs of a minimal pseudograph

Proof. Suppose that G contains at least one of (a), (b), or (c) in Figure 24. We consider the following two cases.

Case 1. $f_{uw} = f_{vw}$.

Since $f_{uw} = f_{vw}$, w is a cut vertex. Let G_1 and G_2 be the connected subgraphs of G such that

$$V(G) = V(G_1) \cup V(G_2),$$

$$E(G) = E(G_1) \cup E(G_2),$$

$$V(G_1) \cap V(G_2) = \{w\}, \text{ and}$$

$$E(G_1) \cap E(G_2) = \emptyset.$$

Assume that $u, v \in V(G_1)$. Suppose that G_2 has no ≤ 2 -face. By the minimality of G , G_1 and G_2 are edge-face 10 colorable. Without loss of generality, we assume that uv, vw, f and f_{uv} of G_1 are

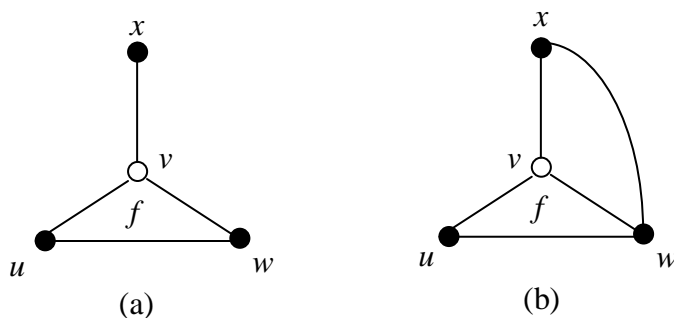
colored with 1, 2, 3, and 4. Since the degree of w in G_2 is at most seven, we can always permute colors of edges and faces in G_2 in such a way that we can combine edge-face 10-colorings of G_1 and G_2 to obtain an edge-face 10-coloring of G .

Suppose that G_2 has one ≤ 2 -face f' . Construct a graph H from G_2 by adding vertices $\alpha, \beta \notin V(G)$ and edges $\alpha w, \beta w$ and $\alpha\beta$ in f' . Then H has no ≤ 2 -face and satisfies $\delta(H) \geq 2$. Since $|E(H)| < |E(G)|$, by the minimality of G , H is edge-face 10-colorable. Thus G_2 is edge-face 10-colorable. By the earlier result shown in the proof of this lemma, G is edge-face 10-colorable, a contradiction.

Case 2. $f_{uw} \neq f_{vw}$.

First, assume G contains (a) in Figure 24. Let $w_i \notin \{u, v\}$, $v_i \notin \{u, w\}$ for $i = 1, \dots, 7$ be the vertices adjacent to w and v , respectively. Consider $G - u$. By the minimality of G , $G - u$ has an edge-face 10-coloring. This induces a partial edge-face 10-coloring of G , where f_{uw} inherits the color of the face of $G - u$ not in G , and only uw, uv , and f are uncolored. Since uv is adjacent to at most nine colored elements, we can color uv . Without loss of generality, we assume that vw is colored with 1, ww_i for $i = 1, \dots, 7$ is colored with $i + 1$, uv is colored with 9 and f_{uw} is colored with 10. If we can recolor uv with a color from $\{2, \dots, 8\}$, then we can color uw with 9. Hence G is edge-face 10-colorable, a contradiction. Thus, vv_i for $i = 1, \dots, 7$ is colored with a color from $\{2, \dots, 8\}$. Since the partial edge-face 10-coloring of G is induced from the edge-face 10-coloring of $G - u$ and since $f_{uw} \neq f_{vw}$, f_{vw} is not colored with 10. Hence we can recolor vw with 10, color uw with 1 and finally color f . Thus G is edge-face 10-colorable, a contradiction. We can treat the cases where G contains either (b) or (c) in Figure 3-1 similarly. \square

Lemma 3.3. Let G be a minimal pseudograph. If G contains (a) in Figure 25, then either $d(f_{uv}) \geq 6$ or $d(f_{vw}) \geq 6$ and if G contains (b), then $d(f_{uw}) \geq 6$.



$u, v, w,$ and x are vertices
 $d(v) = 3$
 f is a face
 In (b), $d(u) = 8$

Figure 25: Subgraphs referenced in Lemma 3.3

Proof. Suppose that G contains the configuration (a) in Figure 25 with $d(f_{uv}) \leq 5$ and $d(f_{vw}) \leq 5$. Consider $G - vw$. By the minimality of G , $G - vw$ has an edge-face 10-coloring. This induces a partial edge-face 10-coloring of G , where f_{vw} inherits the color of the face of $G - vw$ not in G and only vw and f are uncolored. Remove the colors from f_{uv} and f_{vw} . Let $u_i \notin \{v, w\}$ for $i = 1, \dots, 7$ be a vertex adjacent to u . Similarly, define $w_i \notin \{u, v\}$ for $i = 1, \dots, 7$, and $x_i \notin \{v\}$ for $i = 1, \dots, 8$. Since G is minimal, without loss of generality, we may assume that uw is colored with 1, ww_i for $i = 1, \dots, 7$ is colored with $i + 1$, and uv and vx are colored with 9 and 10, respectively. Suppose that we can recolor uv with a color from $\{2, \dots, 8\}$. Then, we can color vw with 9. Since $d(f_{uv}) \leq 5$, $d(f_{vw}) \leq 5$ and f is uncolored, we can color f_{uv} and f_{vw} first, followed by f . Hence G is edge-face 10 colorable, a contradiction. Thus uu_i for $i = 1, \dots, 7$ is colored with a color

from $\{2, \dots, 8\}$. Similarly, we can show that xx_i for $i = 1, \dots, 8$ is colored with a color from $\{1, \dots, 8\}$. Now we can either recolor uw with 10 or swap the colors of uv and vx and recolor uw with 9, and then color vw with 1. Since $d(f_{uv}) \leq 5$, $d(f_{vw}) \leq 5$ and f is uncolored, we can color f_{uv} and f_{vw} first, followed by f . Hence G is edge-face 10 colorable, a contradiction.

Now, suppose that G contains the configuration (b) in Figure 25 with $d(f_{uw}) \leq 5$.

Consider $G - vw$. By the minimality of G , $G - vw$ has an edge-face 10-coloring. This induces a partial edge-face 10-coloring of G , where f_{vw} inherits the color of the face of $G - vw$ not in G , and only vw and f are uncolored. Remove the colors from f_{uw} and f_{vw} . Let $u_i \notin \{v, w\}$ for $i = 1, \dots, 6$ be a vertex adjacent to u . Similarly, we define $w_i \notin \{u, v\}$ for $i = 1, \dots, 6$. Since G is minimal, without loss of generality we assume that uw is colored with 1, ww_i for $i = 1, \dots, 6$ is colored with $i + 1$, and wx , uv , and vx are colored with 8, 9, and 10 respectively. The recoloring argument used in the previous paragraph can be used to show that uu_i for $i = 1, \dots, 6$ is colored with a color from $\{2, \dots, 8\}$. Now we can recolor uw with 10 and color vw with 1. Since $d(f_{uw}) \leq 5$ and f is uncolored, we can color f_{uw} first, and then f and f_{vw} . Hence G is edge-face 10-colorable, a contradiction. \square

Lemma 3.4. Let G be a minimal pseudograph and v be a 3-vertex of G that is adjacent to vertices u , w , and x such that u , v , and w are adjacent to a ≤ 4 -face and $d(w) = 8$. Then $d(x) \geq 7$.

Proof. Suppose that G contains a 3-vertex v that is adjacent to vertices u , w and x such that u , v and w are adjacent to a ≤ 4 -face f , and $d(w) \leq 8$. Suppose $d(x) \leq 6$ and consider $G - vw$. By the minimality of G , $G - vw$ has an edge-face 10-coloring. This induces a partial edge-face 10-coloring of G , where f_{vw} inherits the color of the face of $G - vw$ not in G , and only vw and f are

uncolored. Let $w_i \notin \{v, w\}$ for $i = 1, \dots, 7$ be a vertex adjacent to w . Similarly, we define $x_i \notin \{v\}$ for $i = 1, \dots, 6$. Since G is minimal, without loss of generality we assume that $w w_i$ for $i = 1, \dots, 7$ is colored with i , while uv and vx are colored with 8 and 9 respectively, and f_{vw} is colored with 10. Since $d(x) \leq 6$, we can recolor vx with a color from $\{1, \dots, 7\}$. Then we color vw with 9, and color f . Hence G is edge-face 10-colorable, a contradiction. \square

The **discharging method** is a technique used to prove theorems and lemmas in structural graph theory. The technique is used to prove that every graph in a specifically defined class contains at least one subgraph from a prescribed list. The presence of the subgraph is then used to prove some result. Application of the technique begins by assigning a **charge** to each vertex and face in the graph. Assignment of the charges provides that the sum of all charges is a small positive number. During the **discharging phase**, charges may be redistributed to nearby vertices and faces, according to a defined set of **discharging rules**. Each rule must maintain the sum of the charges. Design of the rules will lead to the positive charge lying in one of the desired subgraphs after the discharging phase. Since the sum of the charges is positive, at least one vertex or face in the graph must have a positive charge, so the desired subgraph must be present. Successful application of this technique depends on the creative design of the discharging rules.

We use the discharging method to prove our main result, and we now present the discharging rules that will be integral to the proof.

Let G be a plane pseudograph. By Euler's Formula $|V(G)| - |E(G)| + |F(G)| = 2$,

$$\sum_{x \in V(G) \cup F(G)} (4 - d(x)) = 8$$

We call $M(x) = 4 - d(x)$ the *initial charge of x* . We will reassign a *new charge* denoted by $M'(x)$ to each $x \in V(G) \cup F(G)$ according to the discharging rules defined below.

Discharging Rules:

First we redistribute charges of 3-faces to vertices of G .

R1. Let x be a 3-face. Then x sends $\frac{1}{3}$ to each vertex adjacent to it.

After we redistribute charges of 3-faces to vertices, we redistribute charges of the vertices of G .

Let x be a 2-vertex. We redistribute the charge of x as follows:

R2.1. x sends $\frac{1}{6}$ to each adjacent vertex v via each face adjacent to xv .

R2.2. x sends $\frac{2}{3}$ to each ≥ 6 -face adjacent to it.

R2.3. x sends $\frac{1}{3}$ to each 5-face adjacent to it.

After the first three steps, if the charge a of x is still positive, then we take the fourth step.

R2.4. x sends $\frac{a}{2}$ directly to each vertex y adjacent to it along each xy edge.

Let x be a k -vertex with $3 \leq k \leq 5$ with a positive charge. We redistribute the charge of x as follows:

R3.1. x sends $\frac{1}{3}$ to each ≥ 5 -face adjacent to it.

After the first step, if the charge of x is still positive, we take the second step.

R3.2. x sends $\frac{1}{6}$ to each ≥ 6 -vertex v adjacent to it via each ≥ 4 -face adjacent to xv .

After the first two steps, if the charge a of x is still positive, then we take the third step.

R3.3. x sends $\frac{a}{j}$ directly to each ≥ 7 -vertex y along each xy edge, where j is the number

of edges that join x to ≥ 7 -vertices.

We are now well equipped to prove the theorem presented earlier in this chapter.

Proof of Theorem 3.1. Let G be a minimal pseudograph with $\Delta(G) = 9$ and $\delta(G) \geq 2$. According to the discharging rules defined previously, we redistribute $M(x)$ for each $x \in V(G) \cup F(G)$. Then we check $M'(x)$ and show that $M'(x) \leq 0$ for each $x \in V(G) \cup F(G)$. Hence we obtain a contradiction.

Let x be a 3-face. By R1, clearly $M'(x) = M(x) - 3 \times \frac{1}{3} = 0$.

Let x be a 4-face. By our discharging rules, $M'(x) = M(x) = 0$.

Let x be a 5-face. Then $M(x) = -1$. By Lemma 3.1, x is not adjacent to vertices u or v such that uv is adjacent to x and $d(u) + d(v) \leq 9$. If x is adjacent to two 5-vertices u and v such that uv is adjacent to x , then by Lemma 3.1, uv is adjacent to two ≥ 5 -faces. Hence by our discharging rules, each of u and v receives at most 1 from the adjacent 3-faces and sends nothing to x . Since x receives at most $\frac{1}{3}$ from each ≤ 5 -vertex adjacent to it and since there are no

vertices u and v such that uv is adjacent to x and both u and v send positive charges to x , we have $M'(x) \leq 0$.

Let x be a k -face with $k \geq 6$. Then $M(x) = 4 - k$. By our discharging rules, x receives at most $\frac{1}{3}$ from each j -vertex adjacent to it where $3 \leq j \leq 5$ and receives $\frac{2}{3}$ from each 2-vertex adjacent to it. Since each 2-vertex v adjacent to x is adjacent to two ≥ 8 -vertices, we can think of v sending half of this $\frac{2}{3}$ directly to x and half of the remaining $\frac{1}{3}$ via each vertex adjacent to it. Thus each vertex adjacent to x sends at most $\frac{1}{3}$ to x . Hence we have

$$M'(x) \leq M(x) + \frac{k}{3} = \frac{2(6-k)}{3} \leq 0.$$

Let x be a 2 vertex. By R2.4, $M'(x) \leq 0$.

Let x be a k -vertex with $3 \leq k \leq 5$. Then $M'(x) = 4 - k$. By Lemma 3.1, x is not adjacent to any 2-vertex. If x is adjacent to a ≥ 7 -vertex, by R3.3, $M'(x) = 0$. Hence x is not adjacent to any ≥ 7 -vertex. Assume $k = 3$. By Lemma 3.1, x is not adjacent to any ≤ 5 -face. Hence, x receives no charge by R1. By R3.1, x sends out 1 to the faces adjacent to it and thus $M'(x) = 0$. Assume $k = 4$. By Lemma 3.1, x is not adjacent to any ≤ 4 -face. Hence, x receives no charge by R1, and thus $M'(x) = 0$. Assume $k = 5$. By Lemma 3.1, x is adjacent to at most two ≤ 4 -faces.

Hence, x receives at most $\frac{2}{3}$ by R1, and thus $M'(x) < 0$.

Let x be a 6-vertex. Then $M(x) = -2$. By Lemma 3.1, x is not adjacent to any 2-vertex. Since x does not receive any charge directly from vertices adjacent to it and since each face adjacent to x sends at most $\frac{1}{3}$ to x , we have $M'(x) \leq M(x) + 6 \times \frac{1}{3} = 0$.

Let x be a 7-vertex. Then $M(x) = -3$. By our discharging rules, each face adjacent to x sends at most $\frac{1}{3}$ to x . Now we estimate how much charge each vertex u can send to x directly along each ux edge.

By Lemma 3.1, x is not adjacent to any 2-vertex.

If x is adjacent to a 3-vertex u , by Lemma 3.1, ux is not adjacent to any ≤ 4 -face. Hence u is adjacent to at most one 3-face. According to whether u is adjacent to a 3-face or not, u either sends nothing to x or sends nothing directly to x . Thus no 3-vertex sends any charge directly to x .

If x is adjacent to a 4-vertex u , then ux is adjacent to at least one ≥ 5 -face. If u is adjacent to a ≤ 6 -vertex, then by Lemma 3.1, u is adjacent to at least two ≥ 5 -faces. Thus, by our discharging rules, u sends nothing directly to x . If u is not adjacent to any ≤ 6 -vertex, then u is adjacent to four ≥ 7 -vertices and at most three 3-faces. Hence by $R1$ and $R3.1-3.3$, u sends at most $\frac{1}{4} \times (\frac{3}{3} - \frac{2}{3}) = \frac{1}{12}$ directly to x .

If x is adjacent to a 5-vertex u that is adjacent to a ≤ 6 -vertex, then u is adjacent to at least one ≥ 5 -face. By $R1$, $R3.1$, and $R3.3$, u sends nothing directly to x . If u is adjacent to five 3-faces, then by $R1$ and $R3.1-3.3$, u sends at most $\frac{1}{5} \times (\frac{5}{3} - 1) = \frac{2}{15}$ directly to x .

Now we estimate $M'(x)$. Clearly, if x is not adjacent to any 5-vertex, then $M'(x) \leq M(x) + \frac{7}{3} + \frac{7}{12} < 0$. Thus, we assume that x is adjacent to a 5-vertex u that sends $\frac{2}{15}$ directly to x . Let v and w be vertices adjacent to both x and u . Then $d(v), d(w) \geq 7$. If we think of u sending half of this $\frac{2}{15}$ directly to x , and sending half of the remaining $\frac{1}{15}$ via each of v and w , then each vertex adjacent to x sends at most $\frac{1}{12}$ directly to x . Thus $M'(x) \leq M(x) + \frac{7}{3} + \frac{7}{12} < 0$.

Let x be an 8-vertex. Then $M(x) = -4$. We estimate how much each charge each vertex u can send to x along each ux edge.

If x is adjacent to a 2-vertex u , by Lemma 3.1, ux is not adjacent to any ≤ 5 -face. By R2.1-2.4, x receives nothing directly from u .

If x is adjacent to a 3-vertex u , by Lemma 3.1, ux is adjacent to at least one ≥ 5 -face. If u is adjacent to a ≤ 6 -vertex, then u is not adjacent to any ≤ 4 -face by Lemmas 3.1 and 3.4. Hence, u sends nothing directly to x . Assume that u is not adjacent to any ≤ 6 -vertex. If u is adjacent to a 7-vertex, then u is adjacent to at least two ≥ 5 -faces. Thus by R1 and R3.1-3.3, u sends nothing directly to x . If u is not adjacent to any ≤ 7 -vertex, then by Lemma 3.3, u is adjacent to at most two 3-faces. Thus by R1 and R3.1-3.3, u sends either $\frac{1}{3}$ or at most $\frac{1}{9}$ or nothing directly to x according to whether $f_3(u) = 2$ or $f_3(u) = 1$ or $f_3(u) = 0$.

If x is adjacent to a 4-vertex u that is adjacent to a ≤ 6 -vertex, then by Lemma 3.1, u is adjacent to at least two ≥ 5 -faces. Since $M(u) = 0$, by R1 and R3.1-3.3, u sends nothing to x .

Assume that u is not adjacent to any ≤ 6 -vertex. Hence, by *R1* and *R3.1-3.3*, u sends either $\frac{1}{3}$ or at most $\frac{1}{6}$ or nothing directly to x according to whether $f_3(u) = 4$ or $f_3(u) = 3$ or $f_3(u) = 2$.

If x is adjacent to a 5-vertex u that is adjacent to a ≤ 6 -vertex, then by Lemma 3.1, u is adjacent to at least one ≥ 5 -face. Since $M(u) = -1$, by *R1* and *R3.1-3.3*, u sends nothing to x .

Assume that u is not adjacent to any ≤ 6 -vertex. Hence, by *R1* and *R3.1-3.3*, u sends either $\frac{2}{15}$ or nothing directly to x according to whether $f_3(u) = 5$ or $f_3(u) \leq 4$.

Now we estimate $M'(x)$. Each 3-vertex u that sends $\frac{1}{3}$ directly to x is adjacent to a 3-face f which is adjacent to a ≥ 8 -vertex v such that by Lemma 3.3, $d(f_{vx}) \geq 6$. Hence, one can think of u sending half of this $\frac{1}{3}$ directly to x and sending the remaining $\frac{1}{6}$ via v to x . Each 4-vertex w that sends $\frac{1}{3}$ directly to x is adjacent to two 3-faces that are adjacent to x and two ≥ 7 -vertices y and z . Thus, one can think of w as sending half of this $\frac{1}{3}$ directly to x and sending half of the remaining $\frac{1}{6}$ via each of y and z to x . Since each vertex adjacent to x sends x at most $\frac{1}{6}$, we have

$$M'(x) \leq M(x) + \frac{8}{3} + 8 \times \frac{1}{6} = 0.$$

Let x be a 9-vertex. Then $M(x) = -5$. We estimate how much charge each vertex u can send to x along each ux edge.

If x is adjacent to a 2-vertex u , then by Lemmas 3.1 and 3.2, u is not adjacent to any ≤ 7 -vertex or any 3-face. If u is not adjacent to any $\leq k$ -face with $4 \leq k \leq 5$, then by *R2.1-2.4*, u sends

nothing directly to x . Otherwise, u sends either $\frac{1}{3}$ or $\frac{1}{6}$ directly to x according to whether $f_4(u) = 1$ or $f_5(u) = 1$.

If x is adjacent to a 3-vertex u that is adjacent to a ≤ 6 -vertex, then by Lemma 3.1, u is adjacent to at least two ≥ 6 -faces. By $R1$ and $R3.1-3.3$, according to whether u is adjacent to a 3-face or not, u sends either at most $\frac{1}{6}$ or nothing to x . Assume that u is not adjacent to any ≤ 6 -vertex. Since each 3-face adjacent to u sends $\frac{1}{3}$ to u and since u sends out $\frac{1}{3}$ via each ≥ 4 -face adjacent to it and $\frac{1}{3}$ to each ≥ 5 -face adjacent to it, we have that u sends at most

$$\text{Max} \left\{ \frac{1}{3} \left(1 + \frac{f_3(u)}{3} - \frac{f_4(u)}{3} - \frac{2f_{\geq 5}(u)}{3} \right), 0 \right\} \leq \frac{1}{3} \text{ directly to } x, \text{ where } 0 \leq f_3(u) \leq 2, \text{ and if } f_3(u) \neq 0,$$

then by Lemma 3.3, $f_{\geq 6}(u) \geq 1$, and if $f_3(u) = 2$, then u is not adjacent to any ≤ 7 -vertex.

If x is adjacent to a 4-vertex u that is adjacent to a ≤ 6 -vertex, then by Lemma 3.1, u is adjacent to at least two ≥ 5 -faces. Since $M(u) = 0$, by $R1$ and $R3.1-3.3$, u sends nothing to x . Assume that u is not adjacent to any ≤ 6 -vertex. Then u sends at most

$$\text{Max} \left\{ \frac{1}{4} \left(\frac{f_3(u)}{3} - \frac{f_4(u)}{3} - \frac{2f_{\geq 5}(u)}{3} \right), 0 \right\} \leq \frac{1}{3} \text{ directly to } x, \text{ and if } f_3(u) = 4, \text{ then } u \text{ is not adjacent to}$$

any ≤ 7 -vertex.

If x is adjacent to a 5-vertex u that is adjacent to a ≤ 6 -vertex, then by Lemma 3.1, u is adjacent to at least one ≥ 5 -face. Since $M(u) = -1$, by $R1$ and $R3.1-3.3$, u sends nothing to x .

Assume that u is not adjacent to any ≤ 6 -vertex. Then u sends at most

$$\text{Max}\left\{\frac{1}{5}\left(\frac{f_3(u)}{3} - 1 - \frac{f_4(u)}{3} - \frac{2f_{\geq 5}(u)}{3}\right), 0\right\} \leq \frac{2}{15} \text{ directly to } x.$$

Now we estimate $M'(x)$. Since each 3-vertex u that sends $\frac{1}{3}$ directly to x is adjacent to a 3-face f which is adjacent to ux and a ≥ 8 -vertex v , we can think of u sending two-thirds of this $\frac{1}{3}$ directly to x and sending the remaining $\frac{1}{9}$ via v to x . Since each 4-vertex u that sends $\frac{1}{3}$ directly to x is adjacent to two 3-faces f and g such that f is adjacent to ux and a ≥ 8 -vertex v , and g is adjacent to ux and a ≥ 8 -vertex w , we can think of u sending two-thirds of this $\frac{1}{3}$ directly to x and sending $\frac{1}{18}$ to x via each of v and w . Let u be a 2-vertex that is adjacent to x . If u sends $\frac{1}{3}$ directly to x , then u is adjacent to a 4-face f . Let v be a vertex that is adjacent to x and f . By Lemma 3.2, $d(v) \geq 3$. If v is a 4-vertex, then v sends at most $\frac{1}{6}$ or nothing directly to x according to whether $f_3(v) = 3$ or $f_3(v) \leq 2$. If v is a 3-vertex, then v sends at most $\frac{1}{9}$ or nothing directly to x according to $f_3(v) = 1$ or $f_3(v) = 0$. In the case of $d(v) = 4$, one can think of u as sending two-thirds of this $\frac{1}{3}$ directly to x and sending the remaining $\frac{1}{9}$ temporarily to v . For v , one can think of v as sending two-thirds of this $\frac{1}{3} > \frac{1}{6} + \frac{1}{9}$ directly to x and sending the remaining $\frac{1}{9}$ to x via the ≥ 8 -vertex w adjacent to both v and x . Hence, the eventual charge that each of u , v , and w sends to x is no more

than $\frac{2}{9}$. In the case of $d(v) = 3$, one can think of u as sending two-thirds of this $\frac{1}{3}$ directly to x and sending the remaining $\frac{1}{9}$ via v to x . Let w be the vertex adjacent to both x and f_{vx} . If $d(f_{vx}) = 3$ or $d(f_{vx}) \geq 6$, w sends nothing via v to x . If $4 \leq d(f_{vx}) \leq 5$, then by Lemma 3.3, $f_3(v) = 0$ and v sends nothing directly to x . Hence v sends no more than $\frac{2}{9}$ to x in the case of $d(v) = 3$. Since each vertex sends at most $\frac{2}{9}$ directly to x , we have $M'(x) \leq M(x) + \frac{9}{3} + 9 \times \frac{2}{9} = 0$.

Hence we have

$$8 = \sum_{x \in V(G) \cup F(G)} M(x) = \sum_{x \in V(G) \cup F(G)} M'(x) \leq 0,$$

a contradiction. \square

We now conclude with the proof of our main result.

Theorem 3.2. If G is a plane graph with $\Delta(G) = 9$, then $\chi_{ef}(G) \leq 10$.

Proof. Since isolated vertices do not affect edge-face colorings, we assume that G has no isolated vertices. Let $G_0 = G$ and G_i be obtained from G_{i-1} by deleting a 1-vertex if such a vertex exists. Clearly, there is an integer $k \geq 0$ such that either $k = 0$, $\delta(G_0) \geq 2$ or $k \geq 1$, $\delta(G_k) \geq 2$, but $\delta(G_{k-1}) = 1$. By Theorem 3.1, G_k is edge-face 10-colorable. By our construction of G_k , clearly, we can color all these uncolored edges in $E(G) - E(G_k)$ one by one. Thus G is edge-face 10-colorable. \square

CHAPTER 4 SUGGESTIONS FOR FUTURE WORK

Almost Regular Graphs

Future work in the study of Almost Regular graphs could proceed along at least two paths: (1) research into the graphical properties other than those previously examined, and (2) studies regarding the use of these graphs for network topologies. Here we will emphasize the latter.

Considering the importance of regular graphs in network design, it would be interesting to determine measures for evaluating the difference between a k - ϵ almost regular graph and a regular graph. For example, one could ask for the smallest regular graph containing an almost regular graph. “Smallest” could be determined by the number of added edges and/or vertices. Conversely, it would be interesting to determine when it would be possible to remove edges from a regular graph to form a k - ϵ almost regular graph and to develop an efficient algorithm for doing so. The ability to accomplish this could have application in employing k - ϵ almost regular graphs in network topologies where the cost of the robust connectivity of a regular graph could be reduced using a k - ϵ almost regular graph.

Following a slightly different idea, it may be instructional to examine how one might sequentially expand k - ϵ almost regular graphs in such a manner as to ensure that at each step one maintains the k - ϵ almost regular property. Any network topology based on a k - ϵ almost regular model might at some future time require such an expansion. Applying this idea to the ordinary processes involved when maintaining a computer network, it would be meaningful to investigate

the effects of vertex (workstation) removal, edge (network connection) removal, and vertex and edge additions in these graphs [17].

In any evaluation of the usefulness of the k - ε almost regular graph model as a network topology, it would be necessary to determine the routing properties of these graphs. These include the number of vertex disjoint paths between two vertices, the ease of algorithmically determining them, and the ease of rerouting when a component failure occurs [17]. All of this is connected to the graph's connectivity properties.

This dissertation presents a linear-time algorithm for creating k - ε almost regular graphs. This algorithm may be improved slightly, and modifications could be made to produce graphs with desirable connectivity properties.

Edge-Face Colorings of Plane Graphs

Sharp bounds are well known for the edge-face chromatic number of plane graphs for small Δ and values of Δ greater than or equal to 9. Specifically, if G is a plane graph with $\Delta(G) = 2$, then $\chi_{ef}(G) \leq 5$ and this bound is sharp, and the same sharp bound of 5 holds for plane graphs with $\Delta(G) = 3$ [25]. In this dissertation, we focused on the case where $\Delta(G) = 9$ and showed that for graphs with this property, $\chi_{ef}(G) \leq 10$, and that this bound is sharp. What is left at this point is to find sharp upper bounds for $\chi_{ef}(G)$ for the cases $\Delta(G) = 4, 5, 6, 7,$ and 8 .

Although work has been done to generate algorithms for coloring the vertices or edges of plane graphs with various properties, no work has of yet been published to introduce an algorithm for coloring the edges and faces of plane graphs. A polynomial-time algorithm for this

should be achievable, although finding an optimal algorithm would potentially be quite challenging.

APPENDIX A: CONSTRUCTION ALGORITHM IMPLEMENTATION AND EXAMPLES

ARke.html

```
<html>
<head>
<title>Extremal k-epsilon Almost Regular Graphs</title>

<style type="text/css">
em    {color: #aa3333}
body   {background-color: lightgray; color: black}
strong {font-weight: extra-bold; color: navy}
.grsty {color: navy}
.edsty {color: red}
</style>

<script language="JavaScript">

function validate_existence()
{
    smdelta = parseInt(Params.smdel.value);
    lgdelta = parseInt(Params.lgdel.value);
    k = parseInt(Params.thek.value);
    epsilon = parseInt(Params.eps.value);

    if (lgdelta <= smdelta)
    {
        alert ("The maximum degree must be larger than the
minimum degree!");
        return 0;
    }
    if ((lgdelta-smdelta) % epsilon != 0)
    {
        alert ("The difference between the minimum and maximum
degrees must be a multiple of epsilon!");
        return 0;
    }
    if (smdelta < k)
    {
        alert ("The minimum degree must be at least k!");
        return 0;
    }
    return 1;
}
```

```

function displaygraph(sdel,ldel,k,epsilon,v,p,q)
{
    outputString="<font color=red size=+1><b>";

    for (j=sdel; j <=ldel; j+=epsilon)
    {
        if (j == sdel)
        {
            outputString += "<font color=navy>[R(";
            outputString += v[j]+", "(j-k)+") ]</font>--";
            outputString += q[j]+"--";
        }
        else
        {
            if (j == ldel)
            {
                outputString += "<font color=navy>[R(";
                outputString += v[j]+", "(j-
k)+") ]</font></b></font>";
            }
            else
            {
                outputString += "<font color=navy>[R(";
                outputString += v[j]+", "(j-k)+") ]</font>--
"+q[j]+"--";
            }
        }
        outputString += "<br /><br /><em>Note the structure
<strong>R(a,b)</strong> represents a regular";
        outputString += " subgraph of degree <strong>b</strong> on
<strong>a</strong>";
        outputString += " vertices.</em><br /><br />";
        outputString += "<em>So <strong>R(k,k-1)</strong> would be
the complete graph on <strong>k</strong>";
        outputString += " vertices.</em>";
        graphArea.innerHTML=outputString;
    }
}

```

```

function drawgraph()
{
    smdelta = parseInt(Params.smdel.value);
    lgdelta = parseInt(Params.lgdel.value);
}

```



```

k = parseInt(Params.thek.value);
epsilon = parseInt(Params.eps.value);

if (validate_existence() == 0)
    alert ("The graph does not exist!");
else
    // Create the graph
    {
        // Create and initialize arrays
        v = new Array (lgdelta+1);
        p = new Array (lgdelta+1);
        q = new Array (lgdelta+1);
        for (n = smdelta; n <= lgdelta; n++)
            {
                v[n] = n;
                p[n] = n;
                q[n] = n;
            }

        // Set position smdelta

        v[smdelta] = smdelta-k+1;
        q[smdelta] = (smdelta-k+1)*k;

        // Determine positions smdelta+epsilon through
        lgdelta-2*epsilon
        for (j = smdelta+epsilon; j <= lgdelta-2*epsilon;
j+=epsilon)
            {
                v[j] = j-k+1;
                p[j] = q[j-epsilon];
                q[j] = v[j]*k - p[j];
            }
        // Determine structure at position lgdelta
        if (((((lgdelta-k+1)*k + q[lgdelta-2*epsilon])/k) % 2
!= 0) && ((lgdelta-k-epsilon) % 2 != 0))
            {
                v[lgdelta] = lgdelta-k+2;
                p[lgdelta] = (lgdelta-k+2)*k;
            }
        else
            {
                v[lgdelta] = lgdelta-k+1;
                p[lgdelta] = (lgdelta-k+1)*k;
            }
    }

```

```

// Determine structure at position lgdelta-epsilon
if ((lgdelta-epsilon) == smdelta)
{
    v[lgdelta-epsilon]= v[lgdelta];
    p[lgdelta-epsilon]= 0;
    q[lgdelta-epsilon] = p[lgdelta];
}
else
{
    v[lgdelta-epsilon] = (p[lgdelta] + q[lgdelta-
2*epsilon])/k;
    p[lgdelta-epsilon] = q[lgdelta-2*epsilon];
    q[lgdelta-epsilon] = p[lgdelta];
}
displaygraph(smdelta,lgdelta,k,epsilon,v,p,q);
}
return false;
}
</script>

```

</head>

<body>

<h2 align="center">Extremal k - ϵ Almost Regular Graphs</h2>

<form name="Params">

<table cellpadding="15" cellspacing="0" width="740" border="6" align="center">

<tr>

<td colspan="4">

A k - ϵ Almost Regular graph is a graph in which each vertex v has the same degree as all but k of its neighbors, and v differs in degree

from these neighbors by exactly ϵ . Enter values below for the minimum

and maximum degrees, k and ϵ , and an extremal graph will be outlined for you.

Note that the difference between the maximum and minimum degrees must be an exact

multiple of ϵ and that the minimum degree must be at least $k!$

</td>

</tr>

```

<tr>
<td>
Input minimum degree for the graph:
</td>
<td>
<input name="smdel" size="4">
</td>
<td>
Input maximum degree for the graph:
</td>
<td>
<input name="lgdel" size="4">
</td>
<tr>
<td>
Input <em>k</em> for the graph:
</td>
<td>
<input name="thek" size="4">
</td>
<td>
Input <em>epsilon</em> for the graph:
</td>
<td>
<input name="eps" size="4">
</td>
</tr>
<tr>
<td colspan="4" align="center">
<input type="button" name="subbut" value="Draw Graph"
onclick="drawgraph();" >
</td>
</tr>
</table>
</form>

<br /><br />
<p id="graphArea">&nbsp;</p>
</body>
</html>

```

SAMPLE OUTPUT

Extremal k - ϵ Almost Regular Graphs

A k - ϵ Almost Regular graph is a graph in which each vertex v has the same degree as all but k of its neighbors, and v differs in degree from these neighbors by exactly ϵ . Enter values below for the minimum and maximum degrees, k and ϵ , and an extremal graph will be outlined for you.

Note that the difference between the maximum and minimum degrees must be an exact multiple of ϵ and that the minimum degree must be at least k !

Input minimum degree for the graph:	<input style="width: 80%;" type="text" value="4"/>	Input maximum degree for the graph:	<input style="width: 80%;" type="text" value="18"/>
Input k for the graph:	<input style="width: 80%;" type="text" value="3"/>	Input ϵ for the graph:	<input style="width: 80%;" type="text" value="2"/>

[R(2,1)]--6--[R(4,3)]--6--[R(6,5)]--12--[R(8,7)]--12--[R(10,9)]--18--[R(12,11)]--18--[R(22,13)]--48--[R(16,15)]

Note the structure $R(a,b)$ represents a regular subgraph of degree b on a vertices.

So $R(k,k-1)$ would be the complete graph on k vertices.

Extremal k - ϵ Almost Regular Graphs

A k - ϵ Almost Regular graph is a graph in which each vertex v has the same degree as all but k of its neighbors, and v differs in degree from these neighbors by exactly ϵ . Enter values below for the minimum and maximum degrees, k and ϵ , and an extremal graph will be outlined for you.

Note that the difference between the maximum and minimum degrees must be an exact multiple of ϵ and that the minimum degree must be at least k !

Input minimum degree for the graph:	<input style="width: 80%;" type="text" value="10"/>	Input maximum degree for the graph:	<input style="width: 80%;" type="text" value="15"/>
Input k for the graph:	<input style="width: 80%;" type="text" value="1"/>	Input ϵ for the graph:	<input style="width: 80%;" type="text" value="5"/>

[R(16,9)]--16--[R(16,14)]

Note the structure $R(a,b)$ represents a regular subgraph of degree b on a vertices.

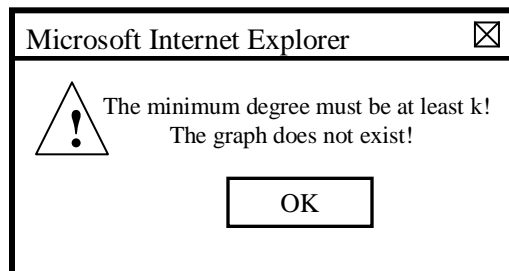
So $R(k,k-1)$ would be the complete graph on k vertices.

Extremal k - ϵ Almost Regular Graphs

A k - ϵ Almost Regular graph is a graph in which each vertex v has the same degree as all but k of its neighbors, and v differs in degree from these neighbors by exactly ϵ . Enter values below for the minimum and maximum degrees, k and ϵ , and an extremal graph will be outlined for you.

Note that the difference between the maximum and minimum degrees must be an exact multiple of ϵ and that the minimum degree must be at least k !

Input minimum degree for the graph:	<input type="text" value="2"/>	Input maximum degree for the graph:	<input type="text" value="18"/>
Input k for the graph:	<input type="text" value="3"/>	Input ϵ for the graph:	<input type="text" value="2"/>

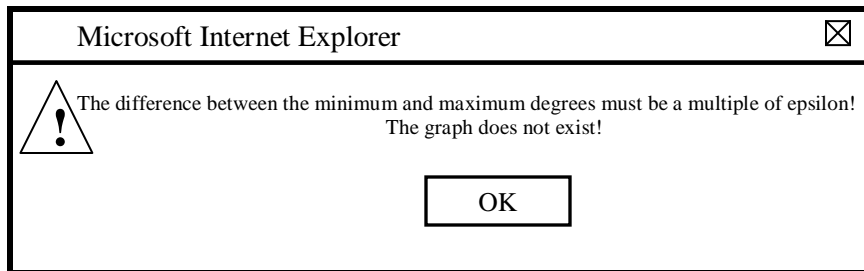


Extremal k - ϵ Almost Regular Graphs

A k - ϵ Almost Regular graph is a graph in which each vertex v has the same degree as all but k of its neighbors, and v differs in degree from these neighbors by exactly ϵ . Enter values below for the minimum and maximum degrees, k and ϵ , and an extremal graph will be outlined for you.

Note that the difference between the maximum and minimum degrees must be an exact multiple of ϵ and that the minimum degree must be at least k !

Input minimum degree for the graph:	<input type="text" value="1"/>	Input maximum degree for the graph:	<input type="text" value="18"/>
Input k for the graph:	<input type="text" value="3"/>	Input ϵ for the graph:	<input type="text" value="2"/>



**APPENDIX B: RECOGNITION ALGORITHM IMPLEMENTATION AND
EXAMPLES**

recognize2xml.html

```
<html>
<head>
<title>k-epsilon Almost Regular Graphs</title>

<style type="text/css">
em    {color: #aa3333}
body   {background-color: lightgray; color: black}
strong {font-weight: extra-bold}
</style>

<script language="JavaScript">

function checkgraph()
{
    var outputstring="";

    // Load adjacency matrix from XML document

    var gDoc = new ActiveXObject("MSXML2.DOMDocument");    //
Create XML doc object
    gDoc.async = false;                                     //
Stop program while file is read

    gDoc.load(Params.matrixpath.value);                    // Load
graph from XML document provided

    var graph = gDoc.documentElement;                      // graph is
the root object
    var graphSize = eval(graph.getAttribute ("order"));
    // graphSize is the value of the order attribute for
element graph

    // Create arrays to count number of neighbors with same or
epsilon-different degrees
    // Also create array to store degree of each vertex.
Initialize all arrays to zero.

    s = new Array(graphSize);
    d = new Array(graphSize);
    degree=new Array(graphSize);
    for (i = 0;i < graphSize; i++) {
        s[i]=0;
```

```

        d[i]=0;
        degree[i]=0;
    }

    am = new Array (graphSize);           // create array of
vertices
    for (i = 0; i < graphSize; i++)
        am[i] = new Array(graphSize);    // for each vertex,
create an array for adjacency information
    for (i = 0; i < graphSize; i++)
        for (j = 0; j < graphSize; j++)
            am[i][j] = 0;                // Initialize all
adjacencies to zero

    // Iterate through the vertices, and then through the
neighbors, to assign 1's in the matrix for all adjacencies

    for (i = 0; i < graph.childNodes.length; i++) {
        var vLabel =
eval(graph.childNodes.item(i).getAttribute("label"));
        for (j = 0; j <
graph.childNodes.item(i).childNodes.length; j++) {
            var nbr =
eval(graph.childNodes.item(i).childNodes.item(j).getAttribute("l
abel"));
            am[vLabel][nbr]=1;
        }
    }

outputstring += "The adjacency matrix is:<br/><br/>"
for (i = 0; i < graphSize; i++) {
    for (j = 0; j < graphSize; j++) {
        outputstring += " " + am[i][j];
    }
    outputstring += "<br/>";
}

graphArea.innerHTML=outputstring;

// Compute degrees for each vertex

for (i = 0; i < graphSize; i++)
    for (j = 0; j < graphSize; j++)
        degree[i]+=am[i][j];
var k = 0;

```

```

var epsilon = 0;
flag = 0;

// Check first vertex against all others to fix k and epsilon.
// These values have to be the same for all vertices!
// Note that if the difference in degrees between vertex 0 and
// ANY two other vertices is DIFFERENT, the graph can't
// be k-epsilon almost regular!

for (j = 1; j < graphSize; j++) {
    if (am[0][j] == 1) {
        if (degree[0]==degree[j])
            s[0]++;
        else {
            d[0]++;
            tempeps = Math.abs(degree[0]-degree[j]);
            if (tempeps != epsilon)
                if (epsilon == 0)
                    epsilon = tempeps;
            else
                flag = 1;
        }
    }
}

if (flag == 1)
    alert ("This graph cannot be k-epsilon almost regular");
else
    k=d[0];

flag = 0;
for (i = 1; i < graphSize; i++) {
    for (j = 0; j < graphSize; j++) {
        if (am[i][j] == 1) {
            if ((i!=j) && (degree[i]==degree[j]))
                s[i]++;
            else if ((i != j) && ((degree[i] == degree[j]-epsilon) ||
(degree[i] == degree[j]+epsilon)))
                d[i]++;
            else if (i != j) {
                flag = 1;
                break;
            }
        }
    }
}
}

```

```

}

for (i = 0; i < graphSize; i++) {
    if (d[i] != k)
        flag = 1;
}
if (flag == 1)
    alert ("The graph is not k-epsilon almost regular");
else
    alert ("The graph is "+k+"-"+epsilon+" almost regular");
}

function setFocus() {
Params.matrixpath.focus();
}

</script>

</head>

<body onLoad="setFocus();" >
<h2 align="center"><em>k</em>-<em>epsilon</em> Almost Regular
Graphs</h2>
<form name="Params">
<table cellpadding="15" cellspacing="0" width="740" border="6"
align="center">
<tr>
<td colspan="2">
A <em>k</em>-<em>epsilon</em> Almost Regular graph is a graph in
which each vertex <strong>v</strong> has the
same degree as all but <em>k</em> of its neighbors, and
<strong>v</strong> differs in degree
from these neighbors by exactly <em>epsilon</em>. Enter the
pathname for XML file that stores the graph data,
and we will tell you whether your graph is <em>k</em>-
<em>epsilon</em> Almost Regular for some value of <em>k</em> and
<em>epsilon</em>.
<br /><br />
</td>
</tr>

<tr>
<td width = "420">Enter pathname for XML file:<br/></td>
<td>

```

```
<input size="50" name=matrixpath>
</td>
</tr>
<tr>
<td colspan="2" align="center">
<input type="button" name="subbut" value="Recognize"
onclick="checkgraph();">
</td>
</table>
</form>

<br /><br />
<p id="graphArea">&nbsp;</p>
</body>
</html>
```

SAMPLE OUTPUT

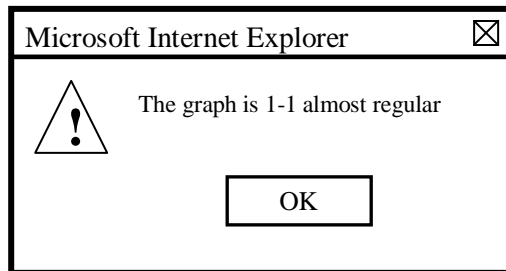
k-epsilon Almost Regular Graphs

A *k-epsilon* Almost Regular graph is a graph in which each vertex v has the same degree as all but k of its neighbors, and v differs in degree from these neighbors by exactly *epsilon*. Enter the pathname for XML file that stores the graph data, and we will tell you whether your graph is *k-epsilon* Almost Regular for some value of k and *epsilon*.

Enter pathname for XML file:

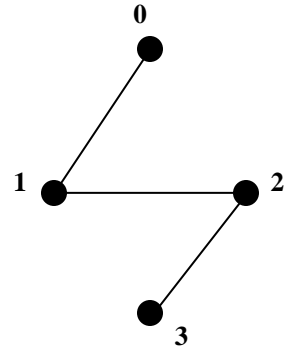
The adjacency matrix is:

```
0 1 0 0
1 0 1 0
0 1 0 1
0 0 1 0
```



graph.xml

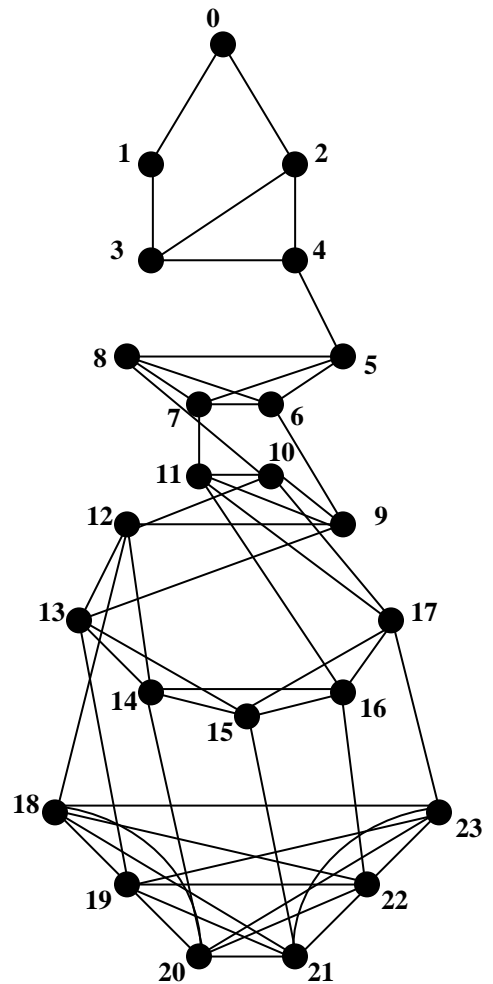
```
<graph order="4">
  <vertex label="0">
    <neighbor label="1"/>
  </vertex>
  <vertex label="1">
    <neighbor label="0"/>
    <neighbor label="2"/>
  </vertex>
  <vertex label="2">
    <neighbor label="1"/>
    <neighbor label="3"/>
  </vertex>
  <vertex label="3">
    <neighbor label="2"/>
  </vertex>
</graph>
```




```

<graph order="24">
  <vertex label="0">
    <neighbor label="1"/>
    <neighbor label="2"/>
  </vertex>
  <vertex label="1">
    <neighbor label="0"/>
    <neighbor label="3"/>
  </vertex>
  <vertex label="2">
    <neighbor label="0"/>
    <neighbor label="3"/>
    <neighbor label="4"/>
  </vertex>
  <vertex label="3">
    <neighbor label="1"/>
    <neighbor label="2"/>
    <neighbor label="4"/>
  </vertex>
  <vertex label="4">
    <neighbor label="2"/>
    <neighbor label="3"/>
    <neighbor label="5"/>
  </vertex>
  <vertex label="5">
    <neighbor label="4"/>
    <neighbor label="6"/>
    <neighbor label="7"/>
    <neighbor label="8"/>
  </vertex>
  <vertex label="6">
    <neighbor label="5"/>
    <neighbor label="7"/>
    <neighbor label="8"/>
    <neighbor label="9"/>
  </vertex>
  <vertex label="7">
    <neighbor label="5"/>
    <neighbor label="6"/>
    <neighbor label="8"/>
    <neighbor label="11"/>
  </vertex>
  <vertex label="8">
    <neighbor label="5"/>
  </vertex>

```



```

        <neighbor label="6"/>
        <neighbor label="7"/>
        <neighbor label="10"/>
</vertex>
<vertex label="9">
    <neighbor label="6"/>
    <neighbor label="10"/>
    <neighbor label="11"/>
    <neighbor label="12"/>
    <neighbor label="13"/>
</vertex>
<vertex label="10">
    <neighbor label="8"/>
    <neighbor label="9"/>
    <neighbor label="11"/>
    <neighbor label="12"/>
    <neighbor label="17"/>
</vertex>
<vertex label="11">
    <neighbor label="7"/>
    <neighbor label="9"/>
    <neighbor label="10"/>
    <neighbor label="16"/>
    <neighbor label="17"/>
</vertex>
<vertex label="12">
    <neighbor label="9"/>
    <neighbor label="10"/>
    <neighbor label="13"/>
    <neighbor label="14"/>
    <neighbor label="18"/>
</vertex>
<vertex label="13">
    <neighbor label="9"/>
    <neighbor label="12"/>
    <neighbor label="14"/>
    <neighbor label="15"/>
    <neighbor label="19"/>
</vertex>
<vertex label="14">
    <neighbor label="12"/>
    <neighbor label="13"/>
    <neighbor label="15"/>
    <neighbor label="16"/>
    <neighbor label="20"/>

```

```

</vertex>
<vertex label="15">
  <neighbor label="13"/>
  <neighbor label="14"/>
  <neighbor label="16"/>
  <neighbor label="17"/>
  <neighbor label="21"/>
</vertex>
<vertex label="16">
  <neighbor label="11"/>
  <neighbor label="14"/>
  <neighbor label="15"/>
  <neighbor label="17"/>
  <neighbor label="22"/>
</vertex>
<vertex label="17">
  <neighbor label="10"/>
  <neighbor label="11"/>
  <neighbor label="15"/>
  <neighbor label="16"/>
  <neighbor label="23"/>
</vertex>
<vertex label="18">
  <neighbor label="12"/>
  <neighbor label="19"/>
  <neighbor label="20"/>
  <neighbor label="21"/>
  <neighbor label="22"/>
  <neighbor label="23"/>
</vertex>
<vertex label="19">
  <neighbor label="13"/>
  <neighbor label="18"/>
  <neighbor label="20"/>
  <neighbor label="21"/>
  <neighbor label="22"/>
  <neighbor label="23"/>
</vertex>
<vertex label="20">
  <neighbor label="14"/>
  <neighbor label="18"/>
  <neighbor label="19"/>
  <neighbor label="21"/>
  <neighbor label="22"/>
  <neighbor label="23"/>

```

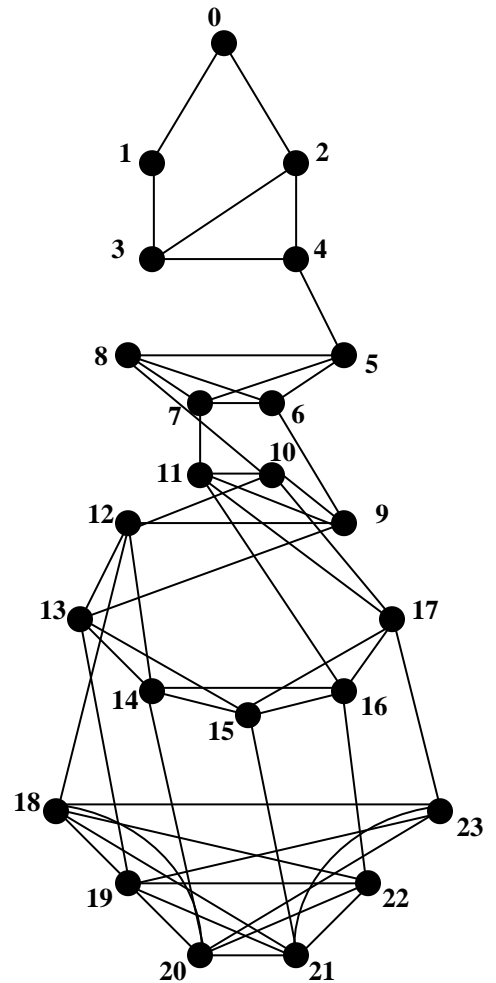
```
</vertex>
<vertex label="21">
  <neighbor label="15"/>
  <neighbor label="18"/>
  <neighbor label="19"/>
  <neighbor label="20"/>
  <neighbor label="22"/>
  <neighbor label="23"/>
</vertex>
<vertex label="22">
  <neighbor label="16"/>
  <neighbor label="18"/>
  <neighbor label="19"/>
  <neighbor label="20"/>
  <neighbor label="21"/>
  <neighbor label="23"/>
</vertex>
<vertex label="23">
  <neighbor label="17"/>
  <neighbor label="18"/>
  <neighbor label="19"/>
  <neighbor label="20"/>
  <neighbor label="21"/>
  <neighbor label="22"/>
</vertex>
</graph>
```



```

<graph order="24">
  <vertex label="0">
    <neighbor label="1"/>
    <neighbor label="2"/>
  </vertex>
  <vertex label="1">
    <neighbor label="0"/>
    <neighbor label="3"/>
  </vertex>
  <vertex label="2">
    <neighbor label="0"/>
    <neighbor label="3"/>
    <neighbor label="4"/>
  </vertex>
  <vertex label="3">
    <neighbor label="1"/>
    <neighbor label="2"/>
    <neighbor label="4"/>
  </vertex>
  <vertex label="4">
    <neighbor label="2"/>
    <neighbor label="3"/>
    <neighbor label="5"/>
  </vertex>
  <vertex label="5">
    <neighbor label="4"/>
    <neighbor label="6"/>
    <neighbor label="7"/>
    <neighbor label="8"/>
  </vertex>
  <vertex label="6">
    <neighbor label="5"/>
    <neighbor label="7"/>
    <neighbor label="8"/>
    <neighbor label="9"/>
  </vertex>
  <vertex label="7">
    <neighbor label="5"/>
    <neighbor label="6"/>
    <neighbor label="8"/>
    <neighbor label="11"/>
  </vertex>
  <vertex label="8">
    <neighbor label="5"/>

```




```

        <neighbor label="6"/>
        <neighbor label="7"/>
        <neighbor label="10"/>
</vertex>
<vertex label="9">
    <neighbor label="6"/>
    <neighbor label="10"/>
    <neighbor label="11"/>
    <neighbor label="12"/>
    <neighbor label="13"/>
</vertex>
<vertex label="10">
    <neighbor label="8"/>
    <neighbor label="9"/>
    <neighbor label="11"/>
    <neighbor label="12"/>
    <neighbor label="17"/>
</vertex>
<vertex label="11">
    <neighbor label="7"/>
    <neighbor label="9"/>
    <neighbor label="10"/>
    <neighbor label="16"/>
    <neighbor label="17"/>
</vertex>
<vertex label="12">
    <neighbor label="9"/>
    <neighbor label="10"/>
    <neighbor label="13"/>
    <neighbor label="14"/>
    <neighbor label="18"/>
</vertex>
<vertex label="13">
    <neighbor label="9"/>
    <neighbor label="12"/>
    <neighbor label="14"/>
    <neighbor label="15"/>
    <neighbor label="19"/>
</vertex>
<vertex label="14">
    <neighbor label="12"/>
    <neighbor label="13"/>
    <neighbor label="15"/>
    <neighbor label="16"/>
    <neighbor label="20"/>

```

```

</vertex>
<vertex label="15">
  <neighbor label="13"/>
  <neighbor label="14"/>
  <neighbor label="16"/>
  <neighbor label="17"/>
  <neighbor label="21"/>
</vertex>
<vertex label="16">
  <neighbor label="11"/>
  <neighbor label="14"/>
  <neighbor label="15"/>
  <neighbor label="17"/>
  <neighbor label="22"/>
</vertex>
<vertex label="17">
  <neighbor label="10"/>
  <neighbor label="11"/>
  <neighbor label="15"/>
  <neighbor label="16"/>
  <neighbor label="23"/>
</vertex>
<vertex label="18">
  <neighbor label="12"/>
  <neighbor label="19"/>
  <neighbor label="20"/>
  <neighbor label="21"/>
  <neighbor label="22"/>
  <neighbor label="23"/>
</vertex>
<vertex label="19">
  <neighbor label="13"/>
  <neighbor label="18"/>
  <neighbor label="20"/>
  <neighbor label="21"/>
  <neighbor label="22"/>
  <neighbor label="23"/>
</vertex>
<vertex label="20">
  <neighbor label="14"/>
  <neighbor label="18"/>
  <neighbor label="19"/>
  <neighbor label="21"/>
  <neighbor label="22"/>
  <neighbor label="23"/>

```

```
</vertex>
<vertex label="21">
  <neighbor label="15"/>
  <neighbor label="18"/>
  <neighbor label="19"/>
  <neighbor label="20"/>
  <neighbor label="22"/>
  <neighbor label="23"/>
</vertex>
<vertex label="22">
  <neighbor label="16"/>
  <neighbor label="18"/>
  <neighbor label="19"/>
  <neighbor label="20"/>
  <neighbor label="21"/>
</vertex>
<vertex label="23">
  <neighbor label="17"/>
  <neighbor label="18"/>
  <neighbor label="19"/>
  <neighbor label="20"/>
  <neighbor label="21"/>
</vertex>
</graph>
```

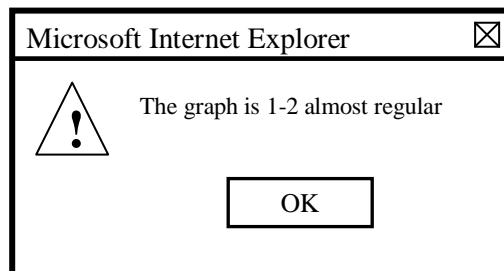
k-epsilon Almost Regular Graphs

A *k-epsilon* Almost Regular graph is a graph in which each vertex v has the same degree as all but k of its neighbors, and v differs in degree from these neighbors by exactly *epsilon*. Enter the pathname for XML file that stores the graph data, and we will tell you whether your graph is *k-epsilon* Almost Regular for some value of k and *epsilon*.

Enter pathname for XML file:

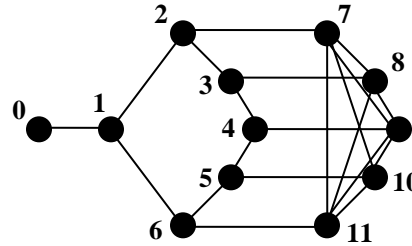
The adjacency matrix is:

```
0 1 0 0 0 0 0 0 0 0 0 0
1 0 1 0 0 0 1 0 0 0 0 0
0 1 0 1 0 0 0 1 0 0 0 0
0 0 1 0 1 0 0 0 1 0 0 0
0 0 0 1 0 1 0 0 0 1 0 0
0 0 0 0 1 0 1 0 0 0 1 0
0 1 0 0 0 1 0 0 0 0 0 1
0 0 1 0 0 0 0 0 1 1 1 1
0 0 0 1 0 0 0 1 0 1 1 1
0 0 0 0 1 0 0 1 1 0 1 1
0 0 0 0 0 1 0 1 1 1 0 1
0 0 0 0 0 0 1 1 1 1 1 0
```



graph4.xml

```
<graph order="12">
  <vertex label="0">
    <neighbor label="1"/>
  </vertex>
  <vertex label="1">
    <neighbor label="0"/>
    <neighbor label="2"/>
    <neighbor label="6"/>
  </vertex>
  <vertex label="2">
    <neighbor label="1"/>
    <neighbor label="3"/>
    <neighbor label="7"/>
  </vertex>
  <vertex label="3">
    <neighbor label="2"/>
    <neighbor label="4"/>
    <neighbor label="8"/>
  </vertex>
  <vertex label="4">
    <neighbor label="3"/>
    <neighbor label="5"/>
    <neighbor label="9"/>
  </vertex>
  <vertex label="5">
    <neighbor label="4"/>
    <neighbor label="6"/>
    <neighbor label="10"/>
  </vertex>
  <vertex label="6">
    <neighbor label="1"/>
    <neighbor label="5"/>
    <neighbor label="11"/>
  </vertex>
  <vertex label="7">
    <neighbor label="2"/>
    <neighbor label="8"/>
    <neighbor label="9"/>
    <neighbor label="10"/>
    <neighbor label="11"/>
  </vertex>
  <vertex label="8">
```



```
    <neighbor label="3"/>
    <neighbor label="7"/>
    <neighbor label="9"/>
    <neighbor label="10"/>
    <neighbor label="11"/>
  </vertex>
  <vertex label="9">
    <neighbor label="4"/>
    <neighbor label="7"/>
    <neighbor label="8"/>
    <neighbor label="10"/>
    <neighbor label="11"/>
  </vertex>
  <vertex label="10">
    <neighbor label="5"/>
    <neighbor label="7"/>
    <neighbor label="8"/>
    <neighbor label="9"/>
    <neighbor label="11"/>
  </vertex>
  <vertex label="11">
    <neighbor label="6"/>
    <neighbor label="7"/>
    <neighbor label="8"/>
    <neighbor label="9"/>
    <neighbor label="10"/>
  </vertex>
</graph>
```

LIST OF REFERENCES

- [1] Alavi Y. Chartrand G. Chung F. R. K. Erdős P. Graham and R. L. Oellermann, O. R. 1987, Highly Irregular Graphs. *Journal of Graph Theory*. Vol 11, No. 2, 235-249.
- [2] Albertson M. 1997, The Irregularity of a Graph. *ARS Combinatoria*. Vol. 46, 219-225.
- [3] Alon N. Friedland Z. and Kalai G. 1984, Regular Subgraphs of Almost Regular Graphs. *Journal of Combinatorial Theory*. Series B, Vol 37, 79-91.
- [4] Borodin O.V. 1994, Simultaneous coloring of edges and faces of plane graphs. *Discrete Mathematics*. 128, 21-33.
- [5] Brigham R. 1998, *Combinatorics and Graph Theory*. (Class Notes).
- [6] Chartrand G. Erdős P. Oellermann O. 1988, How to Define an Irregular Graph. *The College Mathematics Journal*. Vol 19, 36-42.
- [7] Fischer, L.R. 1999, On the Order of Extremal $k-\epsilon$ Almost Regular Graphs. *Congressus Numerantium*. Vol. 138, 79-92.
- [8] Fischer Cleary, L. R. 1999, *Almost Regular Graphs*. Master's Thesis, University of Central Florida.
- [9] Fischer Macon, L. R. 2002, " k -epsilon Almost Regular Graphs." Professor Lisa Fischer Macon. Valencia Community College. URL:
<http://web.valencia.cc.fl.us/cpa/lmacon/recognize2xml.html> (Accessed November 11, 2002).
- [10] Fischer Macon, L. R. 2002, "Extremal k -epsilon Almost Regular Graphs." Professor Lisa Fischer Macon. Valencia Community College. URL:
<http://web.valencia.cc.fl.us/cpa/lmacon/ARke.html> (Accessed November 11, 2002).

- [11] Harary F. 1969, *Graph Theory*. Addison-Wesley Publishing Company, Massachusetts.
- [12] Haynes T. W. Lawson L. M. Boland J. W. 1994, *Research Directions on Almost Regular Graphs and Inclusive Connectivity*. Unpublished Manuscript, East Tennessee State University.
- [13] Hu G. Zhang Z. 1992, On the edge-face total colorings of planar graphs. *J. Tsinghua Univ.* Vol. 32(3), 18-23.
- [14] Joentgen A. Volkmann L. 1991, Factors of Locally Almost Regular Graphs. *Bulletin of the London Mathematical Society*. Vol. 23, 121-122.
- [15] Kronk H. V. Mitchen. J. 1973, A seven-color theorem on the sphere. *Discrete Mathematics*. Vol. 5, 253-260.
- [16] Kubale M. 2004, *Graph Colorings*. American Mathematical Society, Rhode Island.
- [17] Lawson L. M. 1993, *Almost Regular Graphs*. Unpublished Manuscript, East Tennessee State University.
- [18] Lin C. Hu G. Zhag Z. 1995, A six-color theorem for the edge-face coloring of plane graphs. *Discrete Mathematics*. Vol. 141, 291-297.
- [19] Luo R. Zhang C. 2005, Edge-Face Chromatic Number and Ede Chromatic Number of Simple Plane Graphs. *Journal of Graph Theory*. 49, 234-246.
- [20] Mel'nikov L. S. 1995, Problem 9. *Recent Advances in Graph Theory*, Academia Praha, 543.
- [21] Nakano S. Zhou X. Nishizeki T. 1995, *Edge-coloring algorithms*. Springer Berlin / Heidelberg, Germany.

- [22] Ringel. 1968, A six-colour problem on the sphere. *Theory of Graphs*, Academic Press, New York, 265-269.
- [23] Roberts F. S. 1984, *Applied Combinatorics*, Prentice-Hall, New Jersey.
- [24] Sanders D. P. Zhao Y. 1997, On simultaneous edge-face colorings of plane graphs. *Combinatorica*. Vol. 17, 441-445.
- [25] Sanders D. P. Zhao Y. 2000, A five-color theorem. *Discrete Mathematics*. Vol 220, 279-281.
- [26] Sanders D. P. Zhao Y. 2001, On Improving the Edge-Face Coloring Theorem. *Graphs and Combinatorics*. 17, 329-341.
- [27] Sloane N. J. A. Plouffe S. 1995, *The Encyclopedia of Integer Sequences*. Academic Press, San Diego.
- [28] Waller A. O. 1997, Simultaneously Colouring the Edges and Faces of Plane Graphs. *Journal of Combinatorial Theory*, Series B 69, 219-221.
- [29] Wang W. Lih K. 2002, A new proof of Melnikov's conjecture on the edge-face coloring of plane graphs. *Discrete Mathematics*. Vol. 253, 87-95.
- [30] Wang. W. 2007, The edge-face coloring of graphs embedded in a surface of characteristic zero. *Discrete Mathematics*. Article in Press.
- [31] West D. B. 1996, *Introduction to Graph Theory*. Prentice Hall, New Jersey.