A STUDY ON GRACEFUL GRAPHS

Project Report Submitted To

MAHATMA GANDHI UNIVERSITY

In partial fulfillment of the requirement

for the award of

THE MASTER DEGREE IN MATHEMATICS

BY

ANEESHA K P

M.Sc.(IV Semester) Reg No. 180011015177

DEPARTMENT OF MATHEMATICS

ST PAUL'S COLLEGE, KALAMASSERY

(AFFLIATED TO M.G. UNIVERSITY, KOTTAYAM)

2018 - 2020

CERTIFICATE

This is to certify that the project entitled " A STUDY ON GRACEFUL GRAPHS " is a bonafide record of studies undertaken by ANEESHA K P (Reg no. 180011015177), in partial fulfillment of the requirements for the award of M.Sc. Degree in Mathematics at Department of Mathematics, St. Paul's College, Kalamassery, during 2018 – 20.

Dr SAVITHA K S	Mr ARAVIND KRISHNAN R
Head of Department	Department of Mathematics
Department of Mathematics	

Examiner :

DECLARATION

I **Aneesha K P** hereby declare that the project entitled **" A Study On Graceful Graphs"** submitted to department of Mathematics St. Paul's College, Kalamassery in partial requirement for the award of M.Sc Degree in Mathematics, is a work done by me under the guidance and supervision of **Mr. Aravind Krishnan R,** Department of Mathematics, St. Pauls's College , Kalamassery during 2018 – 20.

I also declare that this project has not been previously presented for the award of any other degree, diploma, fellowship, etc.

Kalamassery Aneesha K P

ACKNOWLEDGEMENT

I express my heartfelt gratitude to my guide **Mr. Aravind Krishnan R** for providing me necessary stimulus for the preparation of this project.

I would like to acknowledge my deep sense of gratitude to **Dr. Savitha K S**, Head of the department and all other teachers of the department and classmates for their help at all stages.

I also express my sincere gratitude to **Prof. Valentine D'Cruz**, Principal, St. Paul's College, Kalamassery for the support and inspiration rendered to me in this project report.

Kalamassery **Aneesha K P**

TABLE OF CONTENTS

INTRODUCTION

Graceful labeling is one of the best known labeling methods of graphs. Graceful labeling was originally introduced in 1967 by Rosa . The term "graceful" was introduced by Golomb in 1972. A *graceful labeling* of a graph $G = (V,E)$ with n vertices and m edges is a one-to-one mapping Ψ of the vertex set $V(G)$ into the set $\{0, 1, 2, . . . ,m\}$ with this property: if we define, for any edge $e = (u,v) \in E(G)$, the value $\Psi^*(e) = |\Psi(u) - \Psi(v)|$, then Ψ^* is a one-to-one mapping of the set $E(G)$ onto the set $\{1, 2, . . . ,m\}$. A graph is called graceful if it has a graceful labeling.

Rosa proved that if G is graceful and if all vertices of G are of even degrees, then $|E(G)| \equiv 0$ or 3(mod4). Although most graphs are not graceful, graphs that have some sort of regularity of structure are graceful . Many variations of graceful labeling have been introduced in recent years by researchers. All cycles C_n are graceful if and only if $n \equiv 0$ or 3(mod4). All paths P_n, wheels W_n and complete bipartite graphs $K_{m,n}$ are graceful. The complete graphs K_n are graceful only if $n \leq 4$. It has been conjectured that all trees are graceful. Although this conjecture has been the focus of more than 200 papers, it is still an open problem.

Although more than 400 papers have been published on the subject of graph labeling, there are few particular techniques to be used by authors. The graceful labeling problem is to find out whether a given graph is graceful, and if it is graceful, how to label the vertices. The common approach in proving the gracefulness of special classes of graphs is to either provide formulas for gracefully labeling the given graph, or construct desired labeling from combining the famous classes of graceful graphs.

In the Chapter 1 we deals with the basic definitions of graph theory. In Chapter 2, we present the formal definition of graceful labeling of a graph and present the gracefulness of some graph classes as well as some general results about graceful labeling of graphs. In Chapter 3, we focus on results towards the Graceful Tree Conjecture, presenting different approaches to tackle the conjecture. In chapter 4, We deals with the mathematical programming technique is presented to model and solve the graceful labeling problem for different classes of graphs. And finally in Chapter 5 we discuss the applications of graceful labeling of some graphs.

CHAPTER 1

SOME BASIC CONCEPTS OF GRAPH THEORY

Definition 1.1 : A graph is an ordered triple G = (V(G) ,E(G) ,I_G) where V(G) is an non empty set , E(G) is a set disjoint from V(G) ,and I_G is an incidence relation that associates with each element of E(G) an unordered pair of elements (same or distinct) of V(G) .Elements of V(G) are called the vertices of G and elements of E(G) are called the edges of G .V(G) and E(G) are the vertex set and edge set of G respectively .If , for the edge e of G , I_G (e) ={u,v} , we write I_G(e)=uv

Example 1.1: If $V(G) = \{v_1, v_2, v_3, v_4, v_5\}$, $E(G) = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ and I_G is given by $I_G(e_1) = \{v_1, v_5\}$, $I_G(e_2) = \{v_2, v_3\}$, $I_G(e_3) = \{v_2, v_4\}$, $I_G(e_4) = \{v_2, v_5\}$, $I_G(e_5) = \{v_2, v_5\}$, $I_G(e_6) = \{v_3, v_3\}$, then $(V(G), E(G), I_G)$ is a graph.

The diagrammatic representation of the graph is given below.

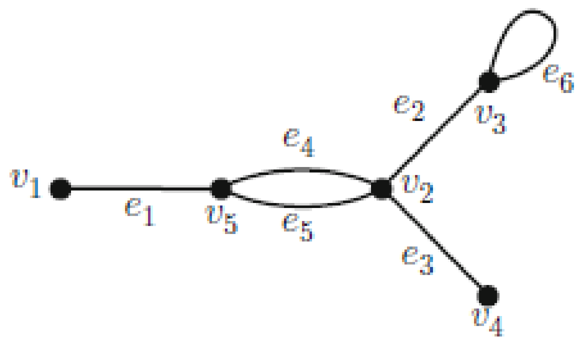


Figure 1.1

Definition 1.2 : If I_G (e) ={u,v} , then the vertices u and v are called the *end vertices* or *ends* of the edge e . Each edge is said to join its ends; in this case, we say that e is *incident* with each one of its ends. Also, the vertices u and v are then incident with e. A set of two or more edges of a graph G is called a set of *multiple* or *parallel edges* if they have the same pair of distinct ends. If e is an edge with end vertices u and v, we write e = uv . An edge for which the two ends are the same is called a *loop* at the common vertex. A vertex u is a *neighbor* of v in G, if uv is an edge of G, and u$\neq$ v. The set of all neighbors of v is the *open neighborhood* of v or the neighbor set of v , and is denoted by N(v) ; the set N [v]=N(v) $\cup$ { v} is the closed neighborhood of v in G. When G needs to be made explicit, these open and closed neighborhoods are denoted by N_G(v) and N_G[v] ,respectively. Vertices u and v are *adjacent* to each other in G if and only if there is an edge of G with u and v as its ends. Two distinct edges e and f are said to be *adjacent* if and only if they have a common end vertex. A graph is *simple* if it has no loops and no parallel edges.

Example 1.2: In the graph of Fig. 1.1, edge e_3 = v_2v_4, edges e_4 and e_5 form multiple edges, e_6 is a loop at v_3; N(v_2) = {v_3, v_4, v_5} N(v_3) = {v_2},N[v_2] = {v_2, v_3, v_4, v_5} and N[v_2] = N(v_2) $\cup${v_2} Further, v_2 and v_5 are adjacent vertices and e_3 and e_4 are adjacent edges

Definition 1.3: A graph is called *finite* if both V(G) and E(G) are finite. A graph that is not finite is called an *infinite* graph.

Definition 1.4: A graph is said to be *labeled* if its *n* vertices are distinguished from one another by labels such as $v_1, v_2, \ldots, v_n$

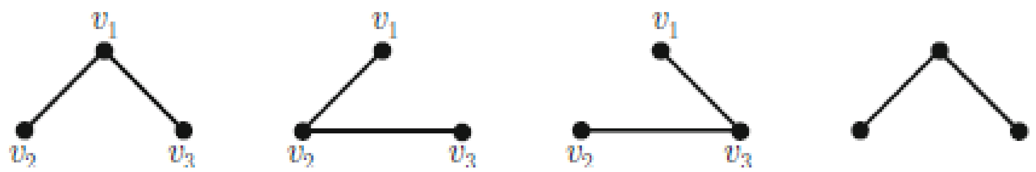


Figure 1.2

Definition 1.5: A simple graph G is said to be *complete* if every pair of distinct vertices of G are adjacent in G and is denoted by K_n .

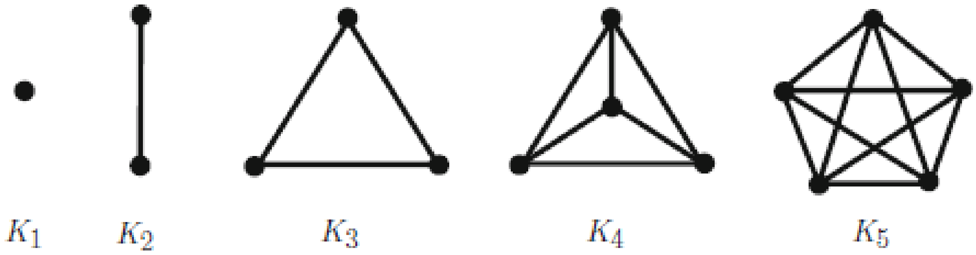


Figure 1.3

Definition 1.6: A graph is *trivial* if its vertex set is a singleton and it contains no edges. A graph is *bipartite* if its vertex set can be partitioned into two nonempty subsets X and Y such that each edge of G has one end in X and the other in Y. The pair (X,Y) is called a *bipartition* of the bipartite graph. The bipartite graph G with bipartition (X,Y) is denoted by G(X,Y) . A bipartite graph G(X,Y) is complete if each vertex of X is adjacent to all the vertices of Y .If G(X,Y) is complete with |X| = p and |Y| =q then G(X,Y) is denoted by $K_{p,q}$. A complete bipartite graph of the form $K_{1,q}$ is called a *star graph*.

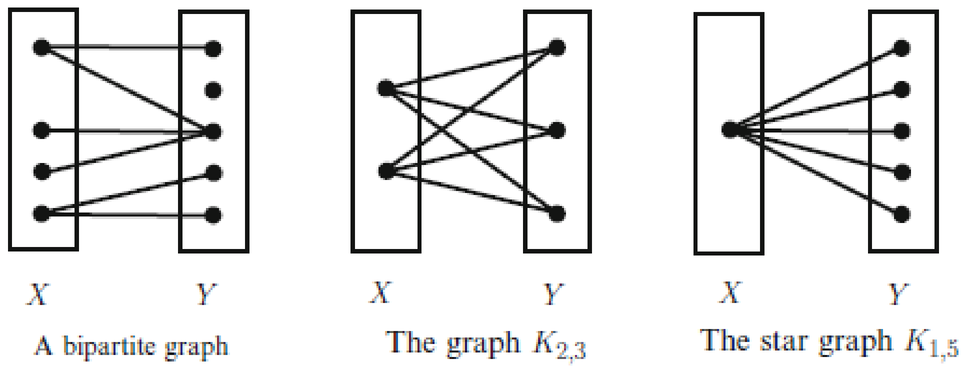


Figure 1.4

Definition 1.7: A k-partite graph is a graph whose vertices can be partitioned into k disjoint sets so that no two vertices within the same set are adjacent.

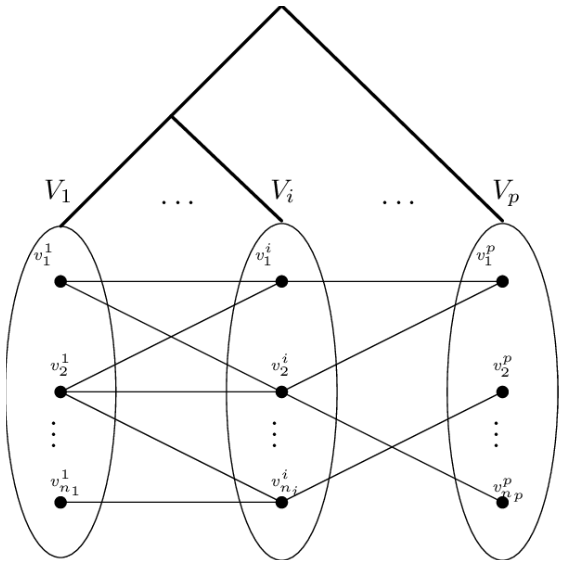


Figure 1.5

Definition 1.8: A *complete k-partite graph* is a *k*-partite graph in which there is an edge between every pair of vertices from different independent sets.

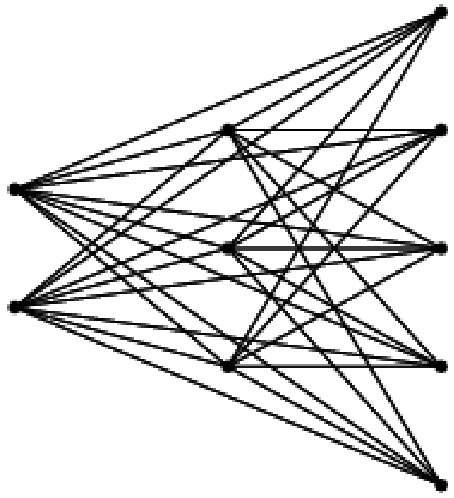


Figure 1.6

Definition 1.9: A graph H is called a *subgraph* of G if V(H) ⊆ V(G), E(H) ⊆ E(G) and I_H is the restriction of I_G to E(H). If H is a subgraph of G, then G is said to be a *supergraph* of H. A subgraph H of a graph G is a *proper subgraph* of G if either V(H) ≠ V(G) or E(H) ≠ E(G) (Hence, when G is given, for any subgraph H of G, the incidence function is already determined so that H can be specified by its vertex and edge sets.) A subgraphH of G is said to be an induced *subgraph* of G if each edge of G having its ends in V(H) is also an edge of H. A subgraph H of G is a *spanning subgraph* of G if V(H)=V(G). The induced subgraph of G with vertex set S ⊆ V(G) is called the subgraph of G induced by S and is denoted by G[S]. Let E′ be a subset of E and let S denote the subset of V consisting of all the end vertices in G of edges in E′. Then the graph (S, E′ ,I_G /E′) is the subgraph of G induced by the edge set E' of G. It is denoted by G[E′]. Let u and v be vertices of a graph G. By G + uv, we mean the graph obtained by adding a new edge uv to G.

Definition 1.10: Let G be a graph and v ∈ V. The number of edges incident at v in G is called the *degree* of the vertex v in G and is denoted by d_G(v), or simply d(v). A loop at v is to be counted twice in computing the degree of v. The minimum (respectively, maximum) of the degrees of the vertices of a graph G is denoted by δ (G) or (respectively, Δ (G) or Δ). A graph G is called *k-regular* if every vertex of G has degree k. A graph is said to be *regular* if it is k-regular for some nonnegative integer k. In particular, a 3-regular graph is called a *cubic graph.*

Definition 1.11: A vertex of degree 0 is an isolated vertex of G. A vertex of degree 1 is called a *pendant vertex* of G, and the unique edge of G incident to such a vertex of G is a *pendant edge* of G. A sequence formed by the degrees of the vertices of G, when the vertices are taken in the same order, is called a *degree sequence* of G.

Definition 1.12: A *walk* in a graph G is an alternating sequence W : $v_0 e_1 v_1 e_2 \ldots\ldots e_p v_p$ of vertices and edges beginning and ending with vertices in which v_{i-1} and v_i are the ends of e_i; v_0 the origin and v_p is the terminus of W. The walk W is said to join v_0 and v_p ; it is also referred to as a $v_0 - v_p$ walk. The walk is *closed* if $v_0 = v_p$ and is *open* otherwise. A walk is called a *trail* if all the edges appearing in the walk are distinct. It is called a *path* if all the vertices are distinct. Thus, a path in G is automatically a trail in G. When writing a path, we usually omit the edges. A *cycle* is a closed trail in which the vertices are all distinct. The *length* of a walk is the number of edges in it. A walk of length 0 consists of just a single vertex . The *distance* between two vertices u and v of a graph G is the length of the shortest u-v path and is denoted by d(u, v).

Example 1.3: In the graph of Fig.1.5, $v_5e_7v_1e_1v_2e_4v_4e_5v_1e_7v_5e_9v_6$ is a walk but not a trail (as edge e_7 is repeated) $v_1e_1v_2e_2v_3e_3v_2e_1v_1$ is a closed walk; $v_1e_1v_2e_4v_4e_5v_1e_7v_5$ is a trail; $v_6e_8v_1e_1v_2e_2v_3$ is a path and $v_1e_1v_2e_4v_4e_6v_5e_7v_1$ is a cycle. Also, $v_6v_1v_2v_3$ is a path, and $v_1v_2v_4v_5v_6v_1$ is a cycle in this graph. Very often a cycle is enclosed by ordinary parentheses.

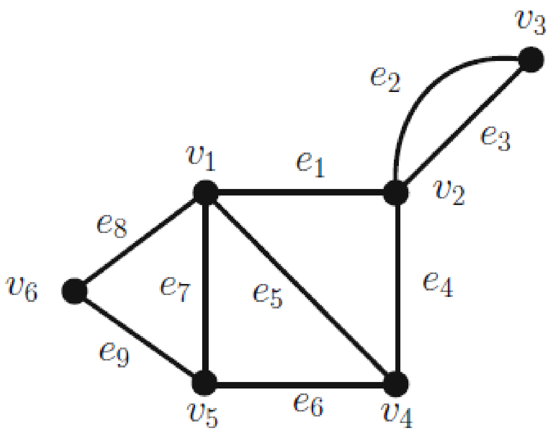


Figure 1.7

Definition 1.13: A cycle of length k is denoted by C_k . Further, P_k denotes a path on k vertices. In particular, C_3 is often referred to as a triangle, C_4 as a square, and C_5 as a pentagon .

Definition 1.14: Let G be a graph. Two vertices u and v of G are said to be connected if there is a u-v path in G: The relation "connected" is an equivalence relation on V(G) . $V_1, V_2, \ldots . V_\omega$be the equivalence classes. The subgraphs $G[V_1], G[V_2], \ldots , G[V_\omega]$, are called the components of G. If $\omega = 1$, the graph G is connected; otherwise, the graph G is disconnected with $\omega \geq 2$ components.

Definition 1.15: An *Euler trail* in a graph G is a spanning trail in G that contains all the edges of G. An *Euler tour* of G is a closed Euler trail of G. G is called *Eulerian* (Fig. 1.6) if G has an *Euler tour*.

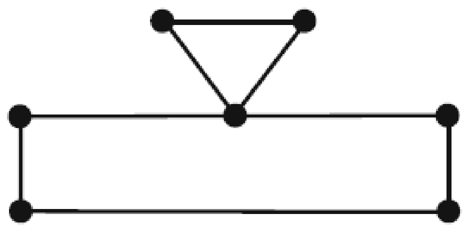


Figure 1.8

Definition 1.16: A connected graph without cycles is defined as a *tree*. A graph without just cycles is called an *acyclic graph* or a *forest*. So each component of a forest is a tree. A forest may consist of a single tree

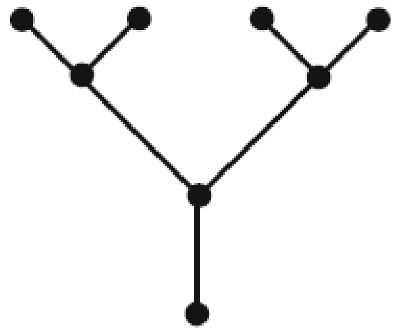


Figure 1.9

Definitions 1.17 Let G be a connected graph.

1. The *diameter* of G is defined as

$$\max\{d(u,v) / u, v \in V(G)\}$$

and is denoted by diam(G).

2. If v is a vertex of G, its *eccentricity* e(v) is defined by

$$e(v) = \max\{d(v, u) / u \in V(G)\}.$$

3. The *radius* r(G) of G is the minimum eccentricity of G; that is,

$$r(G) = \min\{e(v)/ v \in V(G)\}.$$

Note that diam(G) = $\max\{e(v) / v \in . V(G)\}$.

4. A vertex v of G is called a *central vertex* if e(v) = r(G). The set of central vertices of G is called the *center* of G

CHAPTER 2

INTRODUCTION TO GRACEFUL GRAPHS

Graph Labeling

Graph labeling, also known as a valuation of a graph, is a map that carries graph elements onto numbers (usually the positive or nonnegative integers) called *labels* that meet some properties depending on the type of labeling that we are considering. The most common choices for the domain are the set of vertices alone (*vertex labelings*), or edges alone (*edge labelings*), or the set of edges and vertices together (*total labelings*)

Graceful Labeling

A *graceful labeling* of a graph $G = (V,E)$ with $m = |V|$ vertices and $n = |E|$ edges is a one-to-one mapping Ψ of the vertex set $V(G)$ into the set $\{ 0,1,2, \dots ,n \}$ with the following property:

If we define, for any edge $e = \{u,v\} \in E(G)$, the value $\Psi^*(e) = | \Psi(u) - \Psi(v) |$ then Ψ^* is a one-to-one mapping of the set $E(G)$ onto the set $\{ 1,2, \dots ,n\}$. A graph is called *graceful* if it has a graceful labeling.

Although it has been studied for 50 years, not many general results are known about graceful labeling. Most of the results are about asserting the gracefulness of a graph class since it suffices to show a graceful labeling for each graph in the class. On the other hand, results on non-gracefulness of a graph rely basically on a necessary condition only valid for Eulerian graphs or on trying to label the graph gracefully until reaching a contradiction, which is not very effective in most of the cases.

Figure 2.1: Graceful labeling of P_3 and $K_{1,3}$

Proposition 2.1 : The path graph P_n is graceful for all $n \geq 1$.

Proof Take a path graph P_n and let $V(P_n) = \{u_0, u_1, \dots, u_{n-1}\}$ be the set of vertices such that $u_{k-1}u_k \in E(P_n)$ for $0 < k < n$. Since P_n has $m = n-1$ edges, we must label the vertices with numbers from 0 to $n-1$ so that every number in $[1, n-1]$ appears as an edge label. We start with edge label $n-1$ since there is only one way to get an absolute difference equal to $n-1$, which is having a vertex with label 0 adjacent to a vertex with label $n-1$. Thus, let us try labeling u_0 with 0 and u_1 with $n-1$. Next, let us try to get an edge label with value $n-2$. There are only two possible ways to get $n-2$ as an absolute difference:

$$n-2 = |(n-2)-0| = |(n-1)-1|$$

Since u_0 has no more unlabeled adjacent vertices, we can only get the edge label $n-2$ by labeling u_2 with 1. Going on with this strategy, our resulting labeling will be as follows:

$$\psi(u_k) = \begin{cases} \frac{k}{2} & ;\ if\ k\ is\ even \\ \frac{N-(k+1)}{2} & ;\ if\ k\ is\ odd \end{cases}$$

Now, to show that ψ is indeed a graceful labeling of P_n, it suffices to show that the edge label 1 appears, which is expected to appear on the last edge $u_{n-2}u_{n-1}$. If n is even, then $\psi(u_{n-1}) = n/2$ and $\psi(u_{n-2}) = (n-1)/2$. Hence,

$$\psi^*(u_{n-1}u_{n-2}) = n/2 – (n-1)/2 = 1$$

Therefore, the proposition holds.

Proposition 2.2: *If G = (V,E) is graceful, then there exists a partition P = (A,B) of V such that the number of edges with one end in A and the other in B is* $\lceil \frac{m}{2} \rceil$.

Proof Let G = (V,E) be a graph with a graceful labeling ψ and consider the partition P = (A,B) of V such that $A = \{u \in V : \psi(u) \equiv 0 \pmod 2\}$. Since there are $\lceil \frac{m}{2} \rceil$ odd values between 1 and m, and an odd difference is only possible by subtracting an even value from an odd one, the number of edges connecting two vertices with different parities must be exactly $\lceil \frac{m}{2} \rceil$.

Theorem 2.3: *Let G be an Eulerian graph. If $m \equiv 1, 2 \pmod 4$, then G is not graceful.*

Proof Suppose G = (V, E) is a graceful Eulerian graph. Let $\psi : V \to [0, m]$ be a graceful labeling of G and $C = (u_0, u_1, \ldots, u_{m-1}, u_m = u_0)$ be an Eulerian cycle of G. Taking the sum of the edge labels of C modulo 2, we have:

$$\begin{aligned} \sum_{i=1}^{m} \psi^*(u_{i-1}u_i) &= \sum_{i=1}^{m} | \psi(u_{i-1}) - \psi(u_i)| \\ &\equiv \sum_{i=1}^{m} \psi(u_{i-1}) - \psi(u_i) \\ &\equiv 0 \pmod 2 \qquad (2.1) \end{aligned}$$

And, since C is an Eulerian cycle, i.e., the cycle C goes through each edge exactly once, and f is a graceful labeling of G, we have:

$$\sum_{e\in E} \psi * (e) = \sum_{k=1}^{m} k = \frac{m(m+1)}{2} \equiv 0 (\mathrm{mod}\ 2) \qquad (2.2)$$

Thus, we must have $m \equiv 0, 3 \pmod 4$ in order to satisfy equation (2.2).

Theorem 2.4: *Every graph is an induced subgraph of a graceful graph.*

Proof Given a graph $G = (V, E)$, let us construct a graph H from G such that H is graceful and G is an induced subgraph of H. Consider a vertex labeling $\psi : V \to [0, k]$ injective for some $k \geq m$ such that the edge labeling $\psi^* : E \to N$ is also injective, and there exist $u, v \in V$ with $\psi(u) = 0$ and $\psi(v) = k$. Let $\{x_1, x_2, \ldots, x_r\}$ be the set of missing edge labels. Without loss of generality, $x_1, x_2, \ldots, x_s$ are not vertex labels and $x_{s+1}, \ldots, x_r$ are vertex labels. For each x_i, $1 \leq i \leq s$, add a vertex w_i with label x_i and add an edge connecting w_i to u so that $\psi^*(uw_i) = x_i$. For each x_i, $s + 1 \leq i \leq r$, add a vertex wi with label $k + x_i$ and connect w_i to u and v so that $\psi^*(uw_i) = k + x_i$ and $\psi^*(vw_i) = x_i$. Note that the last step might have introduced new missing edge labels by creating vertex labels with values greater than k. However, these new missing edge labels are not vertex labels. So, for each new missing edge label y, add a new vertex z_y with label y and connect z_y to u so that $f(uz_y) = y$. The resulting graph H is graceful and it contains G as an induced subgraph.

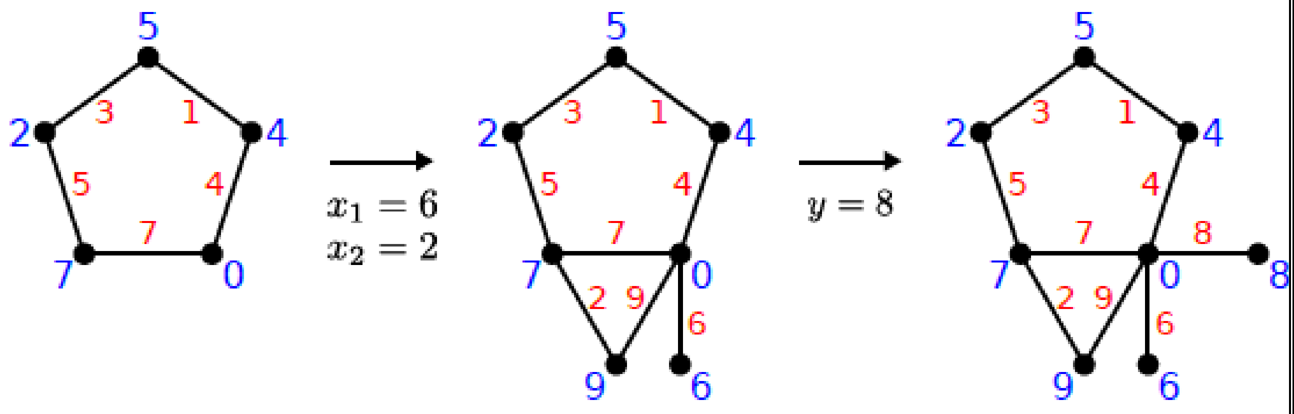


Figure 2.2: Constructing a graceful graph from C_5

Theorem 2.4 says that a graph *G* being non-graceful does not matter for graphs for which *G* is an induced subgraph. It also says that we can always construct a graceful graph from any graph.

Other Graceful Graphs

In this section, we are going to prove the gracefulness of some graph classes. Most of the results asserting the gracefulness of a graph class are given by explicit graceful labelings. For the non-gracefulness of a graph class, there are only a few tools for that. Basically, we only have Proposition 2.2 and theorem 2.3. We can also prove by trying to label the graph and finding a contradiction.

Proposition 2.5: The complete graph K_n is graceful if, and only if, $n \leq 4$

Proof Let us first introduce a property of graceful labelings. Given a graph with a graceful labeling, if we swap every vertex label k with $m - k$, the resulting labeling is also graceful since the edge labels will not have changed: the end vertices of an edge with labels a and b become $m - a$ and $m - b$, and $|a-b| = |(m-a)-(m-b)|$. This is called the complementarity property.

Now, for K_n with $n > 4$, as before, we must have a vertex with label 0 adjacent to a vertex labeled m to get the edge label m. But, in this case, every vertex is adjacent to every other vertex. Thus, we can label any vertex with 0 and any other one with m without loss of generality. To get the edge label $m - 1$, we have two options: $m - 1 = |(m - 1) - 0| = |m - 1|$. However, the complementarity property allows us to choose either one without loss of generality. Choosing to label a vertex with 1, we get edge labels 1 and $m - 1$. Now we need to get the edge label $m - 2 = |(m - 2) - 0| = |(m - 1) - 1| = |m - 2|$. We can not label a vertex with $m - 1$ or 2 because it would create a duplicate edge label. Hence, our only option is to label a vertex with $m - 2$, obtaining edge labels 2, $m - 3$ and $m - 2$.

Since $m - 3$ has already appeared on an edge, the next edge label we must obtain is $m - 4 = |(m - 4) - 0| = |(m - 3) - 1| = |(m - 2) - 2| = |(m - 1) - 3| = |m - 4|$. Again, we only have one option without creating duplicate edge labels, which is to label a vertex with 4, obtaining edge labels 3, 4, $m - 6$ and $m - 4$. At this point, we have labeled five vertices. However, for K_5, we would have $m - 6 = 4$ as a duplicate edge label. For $n \geq 6$, the next edge label to get is $m - 5$. But, all the five possible ways to get $m - 5$ lead to a duplicate edge label. Therefore, there is no way to get label $m - 5$ on an edge and the proposition holds.

Proposition 2.6: The cycle graph C_n is graceful if, and only if, $n \equiv 0, 3 \pmod 4$.

Proof Cycle graphs are Eulerian graphs. Therefore, by the parity condition, if $n \equiv 1, 2 \pmod 4$, then Cn is not graceful. Otherwise, let us call $V(C_n) = \{u_0, u_1, \ldots, u_{n-1}\}$ such that $u_k u_{k+1} \in E(C_n)$ for $0 \leq k \leq n - 1$ and $u_n = u_0$.

If $n \equiv 0 \pmod 4$, then label the vertices according to the following formula:

$$\Psi(u_i) = \begin{cases} \frac{i}{2} & ; \quad i = 0, 2, 4, \ldots, n-2 \\ n - \frac{i-1}{2} & ; \quad i = 1, 3, 5, \ldots, \frac{n}{2} - 1 \\ n - \frac{i-1}{2} - 1 & ; \quad i = \frac{n}{2} + 1, \frac{n}{2} + 3, \ldots, n-1 \end{cases}$$

If $n \equiv 3 \pmod 4$, then label $V(C_n)$ as follows :

$$\Psi(u_i) = \begin{cases} \frac{i}{2} & ; \quad i = 0, 2, 4, \ldots, n-2 \\ n - \frac{i-1}{2} & ; \quad i = 1, 3, 5, \ldots, \frac{n+1}{2} - 1 \\ n - \frac{i-1}{2} - 1 & ; \quad i = \frac{n+1}{2} + 1, \frac{n+1}{2} + 3, \ldots, n-2 \end{cases}$$

Proposition 2.7 : *The wheel graph W_p is graceful for all $p \geq 3$.*

Proof Let $V(W_p) = \{u_0, u_1, \ldots, u_{p-1}, v\}$ be the set of vertices where v is the vertex joined with the cycle and consider the following two cases.

If $p \equiv 0 \pmod 2$, then the following formula gives a graceful labeling:

$$\Psi(v) = 0$$

$$\Psi(u_i) = \begin{cases} 2p & ; \quad i = 0 \\ 2 & ; \quad i = p-1 \\ i & ; \quad i = 1, 3, 5, \ldots, p-3 \\ 2p - i - 1 & ; \quad i = 2, 4, 6, \ldots, p-2 \end{cases}$$

2. If $p \equiv 1 \pmod 2$, then the following formula gives a graceful labeling :

$$\Psi(v) = 0$$

$$\Psi(u_i) = \begin{cases} 2p & ; \quad i = 0 \\ 2 & ; \quad i = 1 \\ p + i & ; \quad i = 2, 4, 6, \ldots, p-1 \\ p + 1 - i & ; \quad i = 3, 5, 7, \ldots, p-2 \end{cases}$$

Proposition 2.8: *All caterpillar trees are graceful*

Proof A *caterpillar* is a tree in which the removal of all leaves results in a path graph.

Draw the caterpillar tree as a planar bipartite representation and label it as shown in Figure 2.3. It is easy to check that such drawing scheme is always possible.

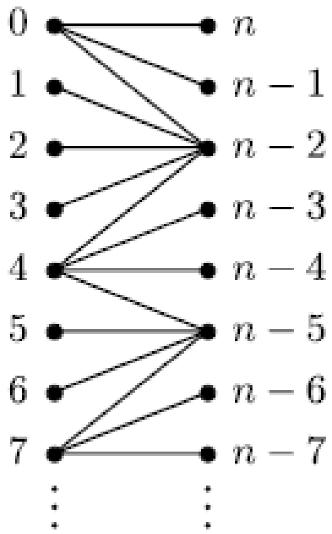


Figure 2.3: Graceful labeling of caterpillar tree.

Note that a path graph P_n is also a caterpillar tree and the labeling scheme given by Proposition 2.8, when applied to a path graph, yields the same labeling constructed before.

Proposition 2.9 : *The complete bipartite graph $K_{p,q}$ is graceful for all $p, q \geq 1$.*

Proof Let $G = (A, B, E)$ be a bipartite graph with $a = |A|$ and $b = |B|$. Assign the vertices from A with numbers $0, 1, \ldots, a-1$, and assign the vertices from B with numbers $a, 2a, \ldots, ba$.

We can generalize the concept of bipartite graph to multipartite graph and, in a similar fashion, we have the complete multipartite graph. It was proven the following proposition regrarding the gracefulness of complete multipartite graph.

Proposition 2.10 : *The complete multipartite graphs $K_{p,q}$, $K_{1,p,q}$, $K_{2,p,q}$, and $K_{1,1,p,q}$ are graceful.*

Proof The graceful labelings are given in Figure 2.4.

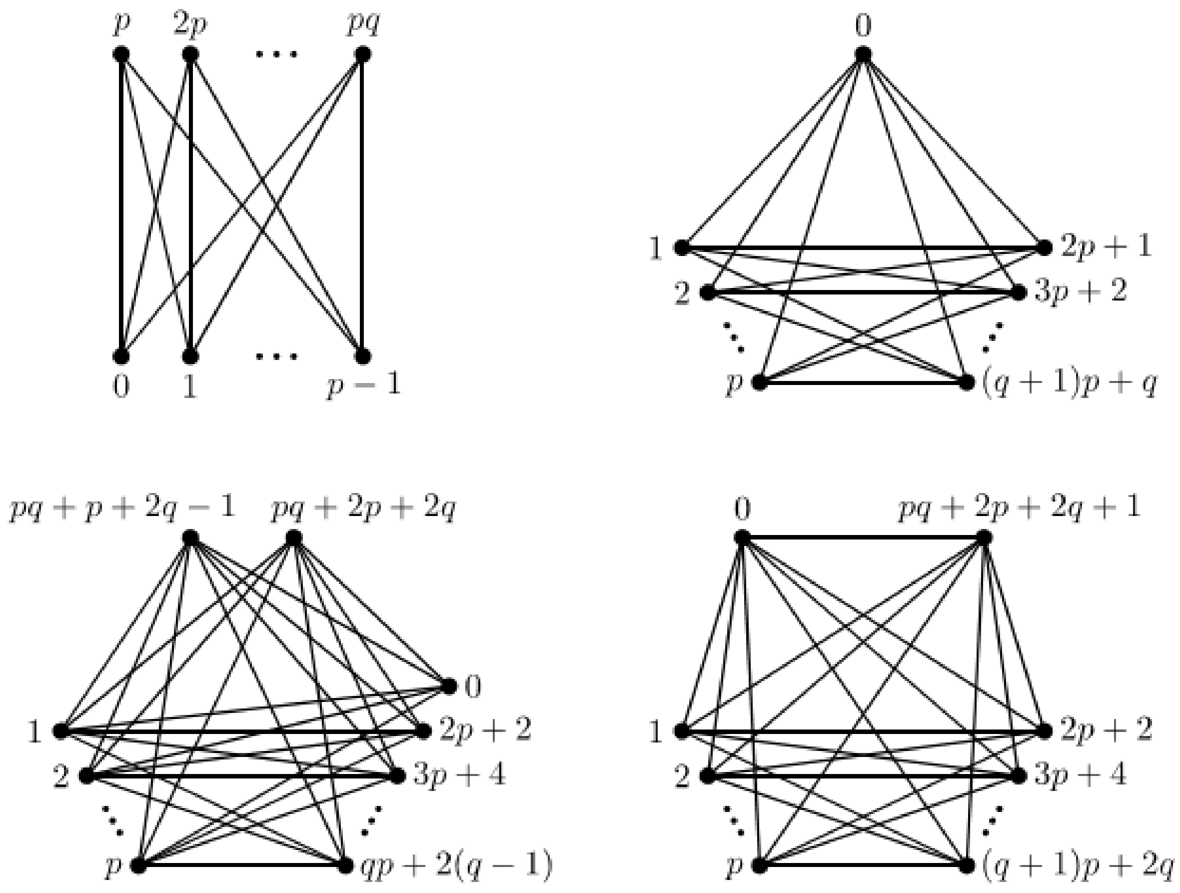


Figure 2.4

Furthermore, Beutner conjectured that these graphs are the only complete multipartite graphs which are graceful, and showed computationally that it is valid for all complete multipartite graphs up to 23 vertices.

CHAPTER 3

TREES

The Graceful Tree Conjecture remains unsolved to these days and there have been a few different approaches researchers have been trying to prove the conjecture. In this section, we have some results on the gracefulness of trees and the different ways in which the conjecture has been tackled.

Conjecture 3.1 (Graceful Tree Conjecture) **:** *Every tree is graceful.*

As shown in Chapter 2, paths and caterpillars are graceful. A first approach would be to extend the definition of caterpillars to new families of trees, i.e., look at the class of trees in which the removal of all leaves results in a caterpillar tree—the *lobsters*—, and so on. However, even the lobster trees have not been characterized yet. Bermond conjectured in 1979 that all lobsters are graceful. This chapter presents others approaches which have shown to be more interesting.

Lemma 3.1: Let T be a tree with a graceful labeling Ψ and let $u \in V(T)$ the vertex with Ψ (u) = 0. If T ‘ is the tree obtained from T by adding a new vertex v only adjacent to u, then T ‘ is graceful.

Proof If m is the number of edges of tree T , then the vertex labeling Ψ' such that $\Psi'|_{V(T)} = \Psi$ and $\Psi'(v) = m + 1$ is a graceful labeling of T ‘ .

Corollary 3.1.1 : *If $w \in V(T)$ has label m, then adding a new vertex only adjacent to w also results in a graceful tree.*

Proof Just consider the complementary graceful labeling of f.

Corollary 3.1.2 : *If $u \in V(T)$ has label* 0 *(or m) and H is a* caterpillar *tree, then adding an edge between u and a vertex of H with maximum eccentricity also results in a graceful tree.*

Proof Apply iteratively Lemma 3.1 giving preference to adding leaves first whenever it is possible. Also note that the corollary is valid for any graceful graph G as long as $u \in V(G)$ has label 0 (or m).

Lemma 3.1 allows us to obtain new graceful graphs from smaller ones by adding a vertex. Then, it is reasonable to ask if this could be used to prove the Graceful Tree Conjecture, i.e., somehow show that for any tree, there is a finite sequence of graceful trees starting from a single vertex such that each tree is the previous one in the sequence plus a vertex, and the last tree of the sequence is the target tree itself.

One sufficient condition to the existence of such sequence is if every tree admits a graceful labeling in which the label 0 can be assigned to any vertex. In the general context, such graphs are called *0-rotatable* graceful graphs. However, it is not true that every tree is 0-rotatable graceful .

Let T be a tree and $uv \in E(T)$. We denote by $T_{u,v}$ the subtree of T containing v after the removal of the edge uv. Precisely, if $S = \{w \in V(T) : v \text{ is on the uw-path }\}$, then $T_{u,v} = T[S]$.

Lemma 3.2: Let T be a tree with a graceful labeling and let $u \in V(T)$ be a vertex adjacent to u_1 and u_2. Consider $T' = T - (V(T_{u,u_1}) \cup V(T_{u,u_2}))$ and let $v \in V(T')$, $v \neq u$

(a) If $u_1 \neq u_2$ and $\Psi(u_1) + \Psi(u_2) = \Psi(u) + \Psi(v)$, then the tree obtained by a disjoint union of T^J, T_{u,u_1} and T_{u,u_2}, and connecting v to u_1 and u_2 is graceful with the same graceful labeling Ψ.

(b) If $u_1 = u_2$ and $2\Psi(u_1) = \Psi(u) + \Psi(v)$, then the tree obtained by a disjoint union of T^J and $T_{u,u1}$, and connecting v to u_1 is graceful with the same graceful labeling Ψ.

Proof It suffices to show that the edge labels of uu_1 and uu_2 are the same as of vu_1 and vu_2.

(a) $|\Psi(u_1) - \Psi(u)| = |\Psi(u) + \Psi(v) - \Psi(u_2) - \Psi(u)| = |\Psi(v) - (u_2)|$

$|\Psi(u_2) - \Psi(u)| = |\Psi(u) + \Psi(v) - \Psi(u_1) - \Psi(u)| = |\Psi(v) - (u_1)|$

(b) $|\Psi(u_1) - \Psi(u)| = \left| \frac{\Psi(u)+\Psi(v)}{2} - \Psi(u) \right| = \left| \frac{\Psi(u)-\Psi(v)}{2} \right|$

$|\Psi(u_1) - \Psi(v)| = \left| \frac{\Psi(u)+\Psi(v)}{2} - \Psi(v) \right| = \left| \frac{\Psi(v)-\Psi(u)}{2} \right|$

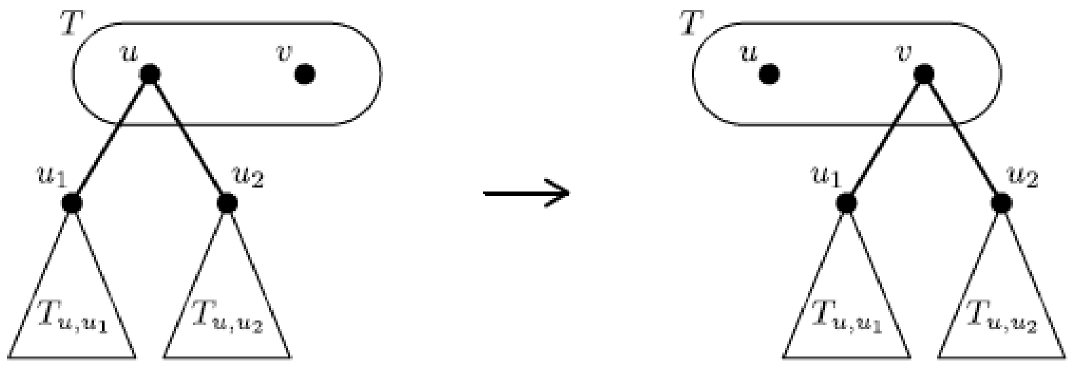
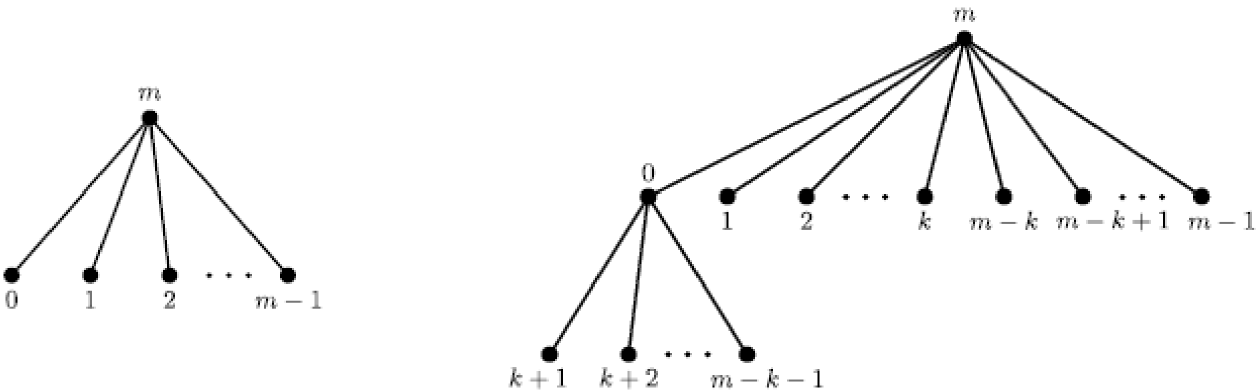


Figure 3.1: Transfer of subtrees from *u* to *v*.

This operation is called a transfer and we mostly do transfers of leaves from one vertex to another. For the remaining of this section, for a graceful tree, we no longer distinguish the vertex label from the vertex itself since in a tree every number from $[0, n-1]$ must appear as a vertex label.

As an example, take the star graph $K_{1,m}$. We can transfer some leaves, which is connected to vertex 0, to the vertex m (see Figure 3.2). For an example, we can transfer k and $m-k$ from 0 to m since $k + (m-k) = 0 + m$. As said before, the subtree being transferred is usually a leaf and we denote a sequence of transfers of leaves adjacent to u to v as $u \to v$. Although the notation is not precise, the context will make clear how many and which leaves are being transferred.



(a) **(b)**

Figure 3.2: Transfer of leaves from *m* to 0 (*m* → 0 transfer)

Proposition 3.3 : *All trees with diameter 4 are graceful.*

Proof. Consider the following types of transfers.

A $u \to v$ transfer is of type 1 if the leaves being transferred are $k, k+1, \ldots, k+s$. This type of transfer can be realized if $u + v = k + (k + s)$. We use this type of transfer when we want to leave an odd number of vertices connected to u.

A $u \to v$ transfer is of type 2 if the leaves being transferred are $k, k+1, \ldots, k+s$ and $l, l+1, \ldots, l+s$ with $k + s < l$. This type of transfer can be realized if

$u + v = k + (l + s)$. We use this type of transfer when we want to leave an even number of vertices connected to u.

By Lemma 3.1, it is sufficient to show that every tree T of diameter 4 with central vertex (which is unique in T) of odd degree has a graceful labeling with the central vertex having the maximum label. This is true because, in a tree of diameter 4, any subtree rooted at one of the children of central vertex is a caterpillar tree.

Let w be the central vertex of T, x be the number of vertices adjacent to w with even degree, and y be the number of vertices adjacent to w with odd degree greater than 1. Let $d(w) = 2k + 1$ and consider the tree of Figure 3.2 b. We can obtain T from that tree by the following sequence of transfers:

$$0 \rightarrow m-1 \rightarrow 1 \rightarrow m-2 \rightarrow 2 \rightarrow m-3 \rightarrow \cdots$$

where the first x transfers (or $x - 1$ if $y = 0$) are of type 1 and the next $y - 1$ transfers (if $y > 1$) are of type 2.

In order to verify that this sequence works, let us analyse the first transfer. Suppose $\{u_1, \ldots, u_x\}$ is the set of vertices adjacent to w with even degree. Starting with the tree on Figure 3.2b, the central Vertex w is the one with label m. The first transfer is $0 \rightarrow m-1$. Then, u_1 is the vertex 0 and we want to leave $d(u_1)-1$ vertices attached to it. Initially, we have the vertices $k+1, k+2, \ldots, m-k-2, m-k-1$ adjacent to 0. Since $0 + (m - 1) = (k + 1) + (m - k - 2)$, it is possible to leave $d(u_1) - 1$ vertices by doing a type 1 transfer of a continuous sequence of vertices to $m - 1$. Going on with an analogous analysis, it can be seen that this sequence works.

Proposition 3.4: *All trees with diameter 5 are graceful.*

The proof of Proposition 3.4 also uses the transfers operations used in the proof of Proposition 3.3.

CHAPTER 4

MATHEMATICAL PROGRAMMING IN GRACEFUL LABELING OF GRAPHS

It has been shown that complete graph K_n, complete bipartite graph $K_{m,n}$, path P_n, wheel graph W_n, cycle C_n are graceful. The graceful labeling problem is to find out whether a given graph is graceful, and if it is graceful, how to label the vertices. The common way to prove the gracefulness of special classes of graphs is to provide formulas for gracefully labeling the given graph. The process of gracefully labeling a particular graph G is a very tedious and difficult task for many classes of graphs. In this chapter, a new approach based on the mathematical programming technique is presented to model and solve the graceful labeling problem for different classes of graphs.

4.1 Mathematical programming model of graceful labeling problem

In modeling the graceful labeling problem, some of our variables cannot take the same value and should be formulated by inequality constraints. For example, assume that we have the below constraints:

$$x_1 \neq x_2,\ x_1, x_2 \geq 0. \tag{4.1}$$

By introducing a new variable w as a nonzero variable, we have

$$x_1 - x_2 - w = 0,\ x_1, x_2 \geq 0,\ w \neq 0. \tag{4.2}$$

Denote the vertices of the graph $G = (V,E)$ by $v_1, v_2, \ldots, v_n$, respectively. Now consider the decision variables of the model are defined as follows:

(i) x_j: the label of vertex vj ;

(ii) x_{ij}: the label of an edge (vi,vj) and a nonzero variable that connects vertices vi and vj, where $x_{ij} \neq 0$ implies that the labels of adjacent vertices v_i and v_j are distinct;

(iii) s_{ijkl} : a nonzero variable, where $s_{ijkl} \neq 0$ implies that the labels of edges (v_i,v_j) and (v_k,v_l) are not equal;

(iv) w_{ijkl} : a nonzero variable, where $w_{ijkl} \neq 0$ implies that the value of an edge label (v_i,v_j) is unequal to the negative value of an edge label (v_k,v_l);

(v) y_{ij} : a nonzero variable, where $y_{ij} \neq 0$ implies that the labels of nonadjacent vertices v_i and v_j are distinct.

The following model has a feasible solution if G is graceful.

Problem 4.1

(1) $x_i - x_j = x_{ij}$ for all i, j, such that $(v_i,v_j) \in E(G)$;

(2) $x_{ij} - x_{kl} = s_{ijkl}$ for all $i, j, k, l, (i, j) \neq (k, l)$, such that (v_i,v_j), $(v_k,v_l) \in E(G)$;

(3) $x_{ij} + x_{kl} = w_{ijkl}$ for all $i, j, k, l, (i, j) \neq (k, l)$, such that (v_i,v_j), $(v_k,v_l) \in E(G)$;

(4) $x_i - x_j = y_{ij}$ for all $i, j, i \neq j$ such that $v_i,v_j \in V(G)$, $(v_i,v_j) \notin E(G)$;

(5) $0 \leq x_i \leq m$, integer, for all i such that $v_i \in V(G)$;

(6) x_{ij} , s_{ijkl} , w_{ijkl}, and y_{ij} are nonzero variables.

In the above model, the first constraint is related to the definition of an edge label as the difference between the corresponding vertex labels. This constraint also causes the edge vertex labels to be distinct. Note that here an edge label is defined as a nonzero, but free in sign variable. Constraints (2) and (3) ensure that the absolute values of edge labels are not equal. Constraint (4) causes the labels of nonadjacent vertices to be distinct. Constraint (5) is related to this fact that the vertex labels are positive integers bounded between 0 and m. Constraints (1)–(5) guarantee that the edge labels are to be distinct and their absolute values generate the set $\{1, 2, \ldots, m\}$. The number of constraints in each equality sets (1) – (4) is m, $(m^2 - m)/2$, $(m^2 - m)/2$, and $(n^2 - n)/2 - m$, respectively. Thus, the total number of constraints (1)–(4) of above problem is $(m^2 + 1/2n^2 - m - n/2)$. Furthermore, in above problem, the total number of variables is equal to the total number of constraints.

4.2 Branching method for solving graceful labeling problem

Branch-and-bound (B&B) algorithm is widely considered to be the most effective method for solving integer programming problems. In this section a special case of B&B algorithm is developed for solving *Problem 4.1*. First, consider the relaxation form of *Problem 4.1* given below.

Problem 4.2

(1) $x_i - x_j = x_{ij}$ for all i, j, such that $(v_i, v_j) \in E(G)$;

(2) $x_{ij} - x_{kl} = s_{ijkl}$ for all $i, j, k, l, (i, j) \neq (k, l)$, such that $(v_i, v_j), (v_k, v_l) \in E(G)$;

(3) $x_{ij} + x_{kl} = w_{ijkl}$ for all $i, j, k, l, (i, j) \neq (k, l)$, such that $(v_i, v_j), (v_k, v_l) \in E(G)$;

(4) $x_i - x_j = y_{ij}$ for all $i, j, i \neq j$ such that $v_i, v_j \in V(G)$, $(v_i, v_j) \notin E(G)$;

(5) $0 \leq x_i \leq m$ for all i such that $v_i \in V(G)$;

(6) x_{ij}, s_{ijkl}, w_{ijkl}, and y_{ij} are free variables.

In the relaxation form of Problem 4.1, the hard constraints are relaxed to produce an easy subproblem. The hard constraints of Problem 4.1 are the integer constraints and the nonzero constraints. First, in Problem 4.1, the integrality constraint is removed and then the sign of the variables is changed from nonzero to free in sign to generate Problem 4.2. It is clear that Problem 4.2 is a linear model and it is much more easier to solve than Problem 4.1. In this branching method for solving Problem 4.2, in each vertex, the corresponding Problem 4.2 is solved, and if the solution satisfies integrality and nonzero constraints, then a feasible solution of Problem 4.1 is found and the algorithm is terminated. If in each vertex, the corresponding Problem 4.2 has no feasible solution, then the related vertex is fathomed. If the corresponding Problem 4.2 has a feasible solution in the current vertex which is not a feasible solution of *Problem 4.1*, then at least one of the following cases occurs:

(1) noninteger values for integer variables;

(2) zero values for nonzero variables.

A vertex in a branching tree is called an active vertex if it has not been fathomed or separated yet. Active vertex are maintained in an active list. Each of the above cases (or both) can be the reason for being a vertex in the active list. Suppose that X^* is the optimal solution of the current *Problem 4.2*. Now define the following sets:

$$N_1 = \{\forall xij\ , yij \in X^* \mid xij = yij = 0\},$$

$$N_2 = \{\forall sijkl\ , wijkl \in X^* \mid sijkl = yijkl = 0\},$$

$$N_3 = \{\forall x_i \in X^* \mid x_i \text{ has noninteger value in } X^*\}.$$

There are two important steps that are the most critical to the performance of this algorithm as follows:

(1) *branching strategy*: selection of the next node from the active list to branch on,

(2) *separation rule*: selection of which variable in the selected node to separate on.

Let N be the total number of variables of the corresponding *Problem 4.2* in the current vertex which are not feasible in constraints (5) or (6) of *Problem 4.1*. Denote the cardinality of set S by $|S|$. In fact, $N = |N_1| + |N_2| + |N_3|$, and N is a degree of infeasibility of the current node regarding

Problem 4.1. If N is very small, then the corresponding solution is very close to a feasible solution for *Problem 4.1.* Here, the “jumptracking strategy” is chosen as branching strategy. In this strategy, a vertex from the active list with the minimum value of N is chosen to branch on. If there is a tie, then a vertex with the minimum value for $|N_1| + |N_2|$ is selected. If there is still a tie, then a vertex with minimum value for $|N_1|$ is selected. Finally, if the tie is not broken, then a vertex from the remaining vertices is selected arbitrarily.

Suppose that according to jumptracking strategy, the current active vertex j is selected. This vertex can have both types of variables causing infeasibility of node j for *Problem 4.1.* In the separation rule, one of the variables such as $x \in N_1$, N_2, N_3 in the selected vertex is chosen to branch on. If the selected variable $x \in N_3$, then the two new subproblems are generated from the selected vertex by using the integer part of x. Denote this strategy of separating current vertex by strategy A. If the selected variable $x \in N_1$ or N_2, then branch it into two different subproblems in which additional constraints $x \geq 1$ and $x \leq -1$ guarantee the nonzero values for x in the next vertices.Denote it by strategy B. From the experimental results, these two methods are applied to more than 100 samples of different types of graphs and it is shown that the second method is much more effective than the first method. Furthermore, according to test problems, separation on variable $x \in N_1$ is more effective than on variable $y \in N_2$. This fact shows that the potential effect of distinct edges is more powerful than that of distinct vertices in gracefully labeling a particular graph. Therefore, in separation rule of branching method, in the process of selection, the next variable, variable $x \in N_1$ has priority to variable $y \in N_2$ and in a similar way, $y \in N_2$ has priority to

$z \in N_3$. Furthermore, when the branching method continues, many branches on the same variable will be generated in different parts of the branching tree. If a variable is chosen many times in different parts of the branching tree, then probably the separation on this variable will not lead us to a feasible solution. Thus, in separation rule of this method, first variable $x \in X^*$ in the selected vertex is chosen according to our priority list, and if there is a tie, then a variable with the minimum number of selections in the other vertices of the branching tree has priority to the other variables. Finally, if there is still a tie, then it is broken arbitrarily.

4.2.1. The branching method for solving Problem 2.1

Step 1 (initializing). Suppose that a graph $G = (V, E)$ with n vertices and m edges is given and we want to know whether or not the graph G is graceful. Furthermore, if G is graceful, we want to know how to label the vertices. A vertex in a branching tree is active if its corresponding problem has not been either solved or subdivided yet. Let the set A denote the list of currently active vertices. Initially, set A = {an active vertex corresponds to the original problem}.

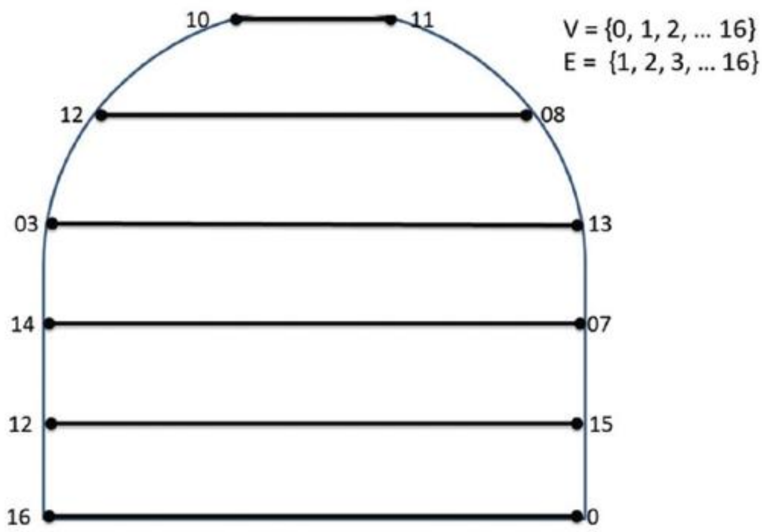
Step 2 (branching). If list A is empty, then stop. G is not graceful. Otherwise, select a vertex j from the active list A according to jumptracking strategy. If its corresponding *Problem 4.2* has a feasible solution in which all the integer variables of *Problem 4.1* have integer values and all nonzero variables of *Problem 4.1* have nonzero values, then $N = 0$, a feasible solution of *Problem 4.1* is found, the graph G is graceful, and the algorithm is terminated. If the corresponding *Problem 4.2* has a feasible solution in the current vertex which is not a feasible solution for *Problem 4.1*, then go to *Step 3.*

Step 3 (selecting). Separate the current vertex into two subproblems according to separa- tion rule described before. In each new vertex, solve its corresponding *Problem 4.2.* Add the new subproblems to the active list if they have feasible solutions for *Problem 4.2.* Go to *Step 2.*

Chapter 5

Application Of Graceful Graphs

Application of graceful graph in dental arch



Each arch consists of a right and left central incisors, lateral incisors, canines, first premolars, second premolars, and molars. Here, it is considered until the first molars. In total there are six teeth on either side of the dental arch summing to a total of 12 teeth.

Each tooth of the arch is considered as a vertex and edges are formed by a line joining the adjacent teeth and the same type of teeth on the left and right side. Graceful labeling is applied in this graph taking the vertex set and the edge set as

V= {0, 1,……. 16} and E = {1, 2,……. 16}.

It is found that while labeling the vertex labels and edge labels are distinct. Furthermore, the vertex labels follow a certain pattern in its arrangement. The relationship between the various teeth and the arch is assessed using graph labeling.

K-Graceful Labeling

A graph G with q edges is k-graceful if there is labeling Ψ from the vertices of G to {0, 1, 2 . . . , q+k-1} such that the set of edge labels induced by the absolute value of the difference of the labels of the adjacent vertices is {k, k+1,. . . ,q+k-1}.

V= {0, 1, 2, . . . , q+k-1}

E= {k, k+1,…., q+k-1}

where

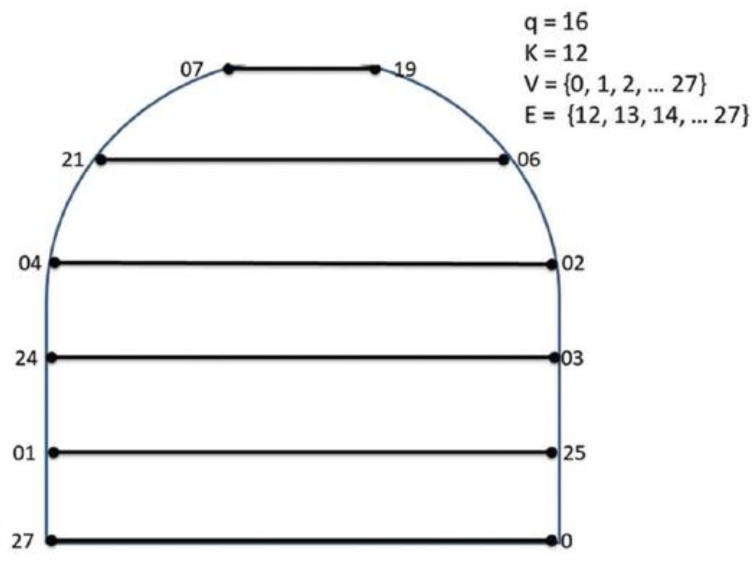
q = number of edges

k = number of vertices.

12 – Graceful Labeling

q = 16

k = 12



The vertex set consist of labels from 0 to q+k-1 = (16+12)-1 = 27

The edge set consists of k, k+1,…., q+k-1 i.e., labels starting from 12 ending with 27

Therefore,

V= {0,1,….27} and

E= {12,13,14,….27}

Thus, the dental arch can be represented by k - Graceful Labeling satisfying its conditions.

Odd Graceful Labeling

A graph G with q edges is considered to be odd-graceful if there is an injection Ψ from V(G) to {0,1,2,…. 2q-1} such that when each edge xy is assigned label | Ψ(x)−Ψ(y) | the resulting edge labels are {1,3,5,….2q-1}.

Now, we try to work out odd graceful labeling in the dental arch. Odd graceful labeling is one of the most widely used labeling methods of the graph while the labeling of graphs is perceived to be a primary theoretical subject in the field of graph theory and it serves as models in a wide range of applications.

Here, p=12, q=16. The vertex set consist of labels from 0 to 2q – 1= (2.16)-1 =31. The edge set consists of all the odd labels < 31

Therefore, V= {0,1,….31} and E = {1,3,5,….31}

Thus, odd graceful labeling is applicable in dental arch models.

Thus the dental arch can be represented by graceful labeling, and we find that there is a certain pattern on doing so. This pattern could be used to analyze the arch and its teeth. Thus, graceful labeling is a powerful tool that makes complicated patterns to be learned easily and conveniently in various fields.

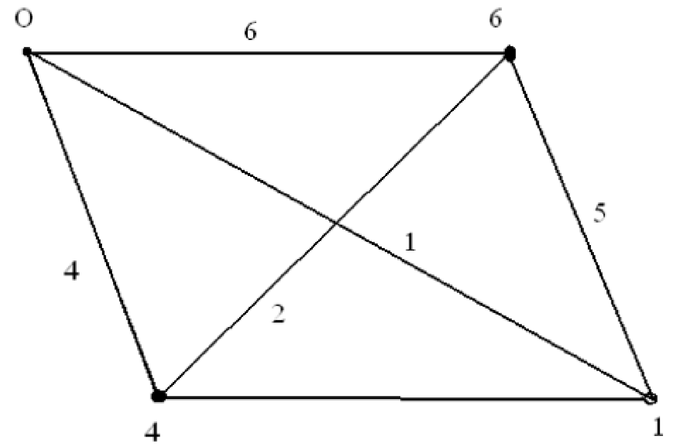
Coding Theory

The design of certain important classes of good non periodic codes for pulse radar and missile guidance is equivalent to numbering of complete graph in such a way that all edge numbers are distinct. The vertex numbers then determine the time positions at which pulse are transmitted. Complete graph Kn is a graph in which every paired vertex are joined by an edge.

In complete graph K4 ,

p = |v| = 4, q = |E| = 4C2 = 6

$\therefore \Psi : V \rightarrow \{0,1,\ldots,6\}$, $\Psi^*: E \rightarrow \{1,2,3,4,5,6\}$



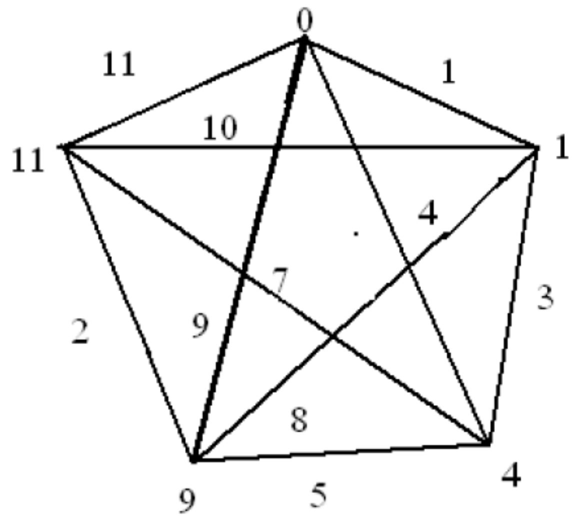
However K5 is semi –graceful labeled, A semi –graceful labeling is defined to be one in which the constraint that the edge lengths need to be consecutive is relaxed , one edge length may be skipped by adding n +1 edge length to the graph.

K5 is semi –graceful labeled as follows.

$p=|V|=5$ $q=|E|=10$

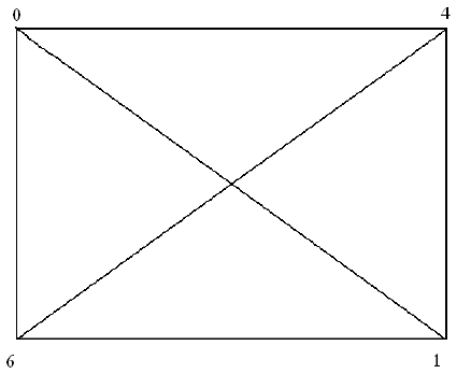
$\Psi: V \rightarrow \{0,1,2,\ldots\ldots 11\}$

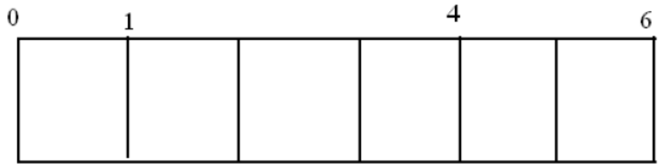
$\Psi^*: E \rightarrow \{1,2,3,4,5,7,8,9,10,11\}$



Quasi – graceful labeling is defined to be when the vertex labels are allowed to be extended beyond the largest edge length value however edge length constraints are left unchanged . Using these type of graceful labeling the extension to coding theory is made possible. Once the graph is gracefully , semi gracefully or quasi-gracefully labeled each vertex label is assigned to the ruler , while using no other tick mark on the ruler.

Ruler to the complete graph K4





Communication Networks

If one had a communication network with a fixed number n+1 of communication centers (i.e. vertex) and they were numbered 0,1,…..,n then the lines between any two centers could be labeled with the difference between two center labels (i.e. vertex labels)

If the communication center grid was laid out in a graceful graph , we would then be able to label the connections between each center such that each connection would have a distinct label.

One good advantage of such a labeling is that if a link goes out , a simple algorithm could detect which two centers are no longer linked.

CONCLUSION

The graceful labeling of graphs has been a topic of research for 50 years and it still has many properties to be found. This project gives a brief overview of the subject, presenting some theoretical results .

In Chapter 2, the problem is presented, as well as the gracefulness of some rather simple graph classes like cycles and wheels. We also show necessary conditions to the existence of a graceful labeling for a graph, and two methods of constructing graceful graphs. In particular, one of them shows that any graph is an induced subgraph of some graceful graph. In Chapter 3, we focus on graceful labeling of trees, more specifically, on different ways to approach the Graceful Tree Conjecture.

In Chapter 4, we move our focus for modeling the graceful labeling problem as a linear programming model. The main goal of this model is to determine how to label the vertices of different classes of graceful graphs. Then a branching strategy was developed to solve the model. The algorithm has been extensively tested on a set of different classes of randomly generated graphs. Moreover, the algorithm described in chapter does not depend on a particular class of graphs and can be easily applied to different types of graphs. Finally in chapter 5 we focus on the application of graceful graphs. Graceful labeling have widespread application in coding theory, dental arch, communication network.

REFERENCES

1. Rodrigo Ming Zhou, Graceful Labeling of Graphs, 2016
2. Kourosh Eshghi and Parham Azimi, Applications Of Mathematical Programming in Graceful Labeling of Graphs, Hindawi Publishing Corporation Journal of Applied Mathematics, 2003
3. P. Lalitha, Application of Graceful Labeling in Dental Arch, Drug Invention Today , Vol 11 , Issue 3 , 2019
4. Ujwala N Deshmukh, Application of Graceful Graphs, International Journal of Engineering Sciences & Research Technology, 2015
5. R. Balakrishnan, K. and Ranganathan, A Textbook of Graph Theory, Springer, New York