Characterization of Realization Graphs

by

Khushboo Seksaria
201511008

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

April, 2017

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Khushboo Seksaria

Certificate

This is to certify that the thesis work entitled CHARACTERIZATION OF REALIZATION GRAPHS has been carried out by KHUSHBOO SEKSARIA for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.

Rahul Muthu
Thesis Supervisor

Acknowledgments

Foremost, I would like to express my sincere gratitude to my enthusiastic thesis guide *Prof. Rahul Muthu* for the continuous support for my research, for his patience, motivation, and immense knowledge to the highest standard inspired me and motivate me. His guidance helped me in all the time of research and writing of this thesis.

I would also like to acknowledge respected faculties of my thesis evaluation com- mittee *Prof. Srikrishnan Divakaran* and *Prof. Puneet Bhateja*, for their very valuable comments on this thesis.

I take this opportunity to sincerely thanks the Dhirubhai Ambani Institute of In- formation and Communication Technology, Gandhinagar, Gujarat for providing me an infrastructural assistance in the form of research lab which buttressed me to perform my work comfortably.

I would also like to thank my family for the support they provided me through out all my studies. Finally, there are my friends. We were not only able to support each other by deliberating over our problems and findings, but also happily by discussing about things other than just our papers.

Thank you everyone!
Khushboo Seksaria

Contents

Abstract

Given a degree sequence d, usually it generates several different realizations belonging to several distinct isomorphism classes. In general, the structure of an arbitrary realization cannot be known from its degree sequence, so it may not be possible to determine whether vertices with specified degrees will be adjacent (or non-adjacent), but in some cases it might be possible. The degree sequence is one of the simplest parameters associated with a graph, however, due to the fact that same degree sequence can belong to several pairwise nonisomorphic graphs (different realizations of a given degree sequence) the utility of a degree sequence in a graph problem is limited.

Studying the structure of these different realizations of a degree sequence and then, to understand the relationships that exist among graphs with the same degree sequence will be our main focus, which can be very well studied with the help of realization graphs. Graphs constructed by applying some specific rule to any given graph are called *auxiliary graphs*. Such graphs are often used to translate one graph problem into another. A *Realization graph*, $G(d)$ is an auxiliary graph defined as the graph (R, δ) where R is the set of realizations of d and δ is the set of edges where two vertices G, G' of R are adjacent if and only if performing a single 2-switch can transform G into G' [3].

This thesis consists of several problems related to different realizations of degree sequences. In this thesis, we have looked at various graph families and studied whether they can be characterized by their degree sequences. Also, we have derived some new characteristics for realization graphs. We have also derived a method for verification of whether or not a given degree sequence is uniquely realizable.

In this thesis, we are considering only simple graphs with finite and nonempty vertex sets.

CHAPTER 1

Introduction

Degree sequence is one of the simplest attributes associated with a graph. However, as most degree sequences belong to several pairwise non-isomorphic graphs, the use of degree sequence in graph problems is limited since the degree sequence does not uniquely determine the structure of a graph. Hence it is desirable to find out the relationship that exists between different graphs with the same degree sequence. The motivation for this work was to study various graph families that can be stated in terms of their degree sequence characterization that will provide us with some foundation for further development. Also such characterization is important because conditions depending only on the degree sequence can often be tested in linear time.

A graph family is said to be *degree determined* if by knowing only the degree sequence we can determine whether a graph belongs to it or not. Most graph classes are not degree determined. However, some are, like: complete graphs, split graphs, threshold graphs. Each of these graph classes have a linear time algorithm for recognizing whether a degree sequence belongs to it or not. We seek to characterize these degree sequence determined classes using 2-switch operations. A 2-switch is an edge deletion and addition operation which changes the adjacencies in the graph [2], as a result, potentially, changing the isomorphism class of the graph. The subgraph induced by the four vertices undergoing a 2-switch operation and their adjacencies with the remaining $n-4$ vertices forms a configuration. We have studied such configurations in which 2-switch operation results in change in the isomorphism class of the graph.

The relationship that exists between various realizations of a degree sequence can be very well studied with the help of realization graphs. The best-known result on a realization graph is that it is a connected graph for any degree sequence

d [3]; this is a consequence of a theorem of Fulkerson, Hoffman, and McAndrew. This naturally motivates us to find characterization of such graph.

In this thesis our main focus is finding several new characteristics of a realization graph and developing efficient algorithm for verification of degree sequence being uniquely realizable. We also look at other related problems.

The thesis is organized as follows. Definitions and notation used is presented in Chapter 1. Basic properties and some new characteristics based on our literature survey are presented in Chapter 2. Various graph families and their degree sequence characterization based on 2-switch operations is presented in Chapter 3. Chapter 4 presents various graph parameters derived for realization graphs. In Chapter 5 we have mentioned various dynamic operations which can be performed on graphs and their effect on the corresponding realization graph. We provide an efficient algorithm for determining whether a degree sequence is uniquely realizable or not, in Chapter 6. We summarize our work and indicate possible future directions for research in the conclusion section.

1.1 Problem Statement

Our work begins with the foundations stated in the previous paragraph and aims to characterize realization graphs completely.

Apart from this major problem we have also worked on the following related problems:

- Degree sequence characterization of well-known graph families in terms of 2-switch operation.
- Studying and determining various graph parameters of realization graph.
- Dynamic operations on graphs and their corresponding effects on the realization graph.

- Providing an efficient method for determining whether a given degree sequence is uniquely realizable or not.

1.2 Definitions and Notation

In this section we present some definitions and notation which we will be using throughout the thesis.

1.2.1 Degree Sequence

Definition 1. Given an undirected graph G on n vertices, a *degree sequence* $d = (d_1, d_2, \ldots, d_n)$ is a monotonically non-increasing sequence of the vertex degrees.

General Characteristics of Degree Sequences

The sum of vertex degrees of a graph is even because of the fact that each edge is counted twice since it connects two vertices.

$$\sum d_i = 2 * number\_of\_edges\_in\_graph$$

Two graphs with different degree sequences cannot be isomorphic. For example, the following two graphs are not isomorphic,

Figure 1.1: Non-Isomorphic Graphs with Different Degree Sequences

Here, G_1 and G_2 have degree sequence (2, 2, 2, 2) and (1, 2, 2, 3) respectively. Since G_1 one has four vertices of degree 2 and G_2 has only two vertices with degree 2.

However, having same degree sequence is not sufficient for isomorphism of graphs. Even simple connected graphs with the same degree sequences can be non-isomorphic. For example,

Figure 1.2: Non-Isomorphic Graphs with Same Degree Sequences

1.2.2 Graphic Sequence

Definition 2. A degree sequence $d = (d_1, d_2, \ldots, d_n)$ is said to be *graphic* if d is the degree sequence of some graph $G = (V, E)$ on the vertex set $V = \{1, 2, \ldots, n\}$ such that $\deg(i) = d_i$ for all i.

In this case, G is a realization of d or it realizes d. While it is fairly straightforward to determine whether a degree sequence is graphic or not, a given graphic sequence can have many different realizations. It is possible that the realizations of same graphic sequences are nonisomorphic to each other. So it is important to the study of properties of such graphic sequences and their overlap with various classes of graphs.

Figure 1.3: Different Realizations of same degree sequence (3,3,3,3,3,3)

Unigraphic Degree Sequence

Definition 3. A degree sequence d, is said to be *unigraphic* if and only if it is uniquely realizable i.e. only one realization can be obtained from d. In order words, d is said to be ***uniquely realizable***. For example, complete graphs are uniquely realizable. Graph family of a unigraphic degree sequence can be determined in linear-time.

Havel-Hakimi construction method

Havel-Hakimi has given a systematic approach [5] for generating a realization from a given non-increasing degree sequence $d = (d_1, d_2, \ldots, d_n)$.

Step 1. Select vertex with highest degree (d_1) and connect it to next $\mid d_1 \mid$ followed vertices in degree sequence and decrement degree of adjacent vertices by 1.

Step 2. If degree sequence contains all $0'$s then terminate else using stable sort rearrange degree sequence into non-increasing order and repeat step 1.

1.2.3 2-Switch

Definition 4. An alternating 4-cycle is an instance of four vertices A, B, C and D in a graph G such that AB and CD are edges in G and BC and AD are not present in G. We denote this alternating 4-cycle by $< A,B : C,D >$.

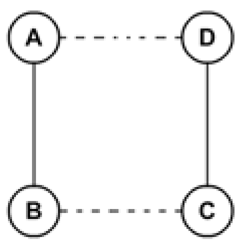


Figure 1.4: An alternating 4-cycle

Definition 5. The 2-switch < A,B : C,D > is the operation of deleting edges AB and CD from G and adding edges AD and BC to G. (This operation is also known as a cycle exchange operation). A 2-switch on $< A, B : C, D >$ is as shown below.

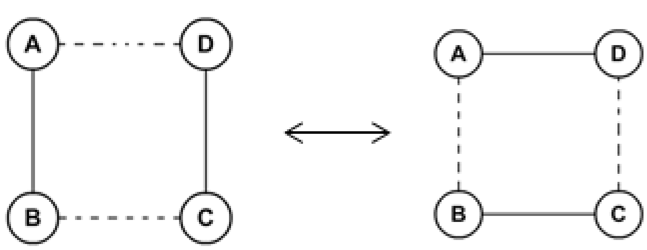


Figure 1.5: 2-Switch operation on < a,b : c,d >

Note that performing a 2-switch operation changes the adjacencies of the graph vertices without changing the degree of any vertex, since the edges are deleted and added at the same vertices. Hence, a 2-switch on G gives us a graph G' where all the vertices have the same degree as they had in G; thus we get another realization G' of degree sequence d. Since the 2-switch operation changes the adjacencies of vertices in the graph, the new realization obtained after performing 2-switch may be non-isomorphic to the previous graph. Also as the 2-switch operation makes only local changes, rest of the graph remains unaffected.

Example 1. *Performing a 2-switch operation on* $< 1, 5 : 3, 4 >$ *in the below figure:*

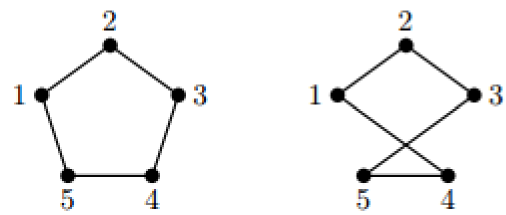


Figure 1.6: Performing 2-switch on (2,2,2,2,2)

Fulkerson Theorem A well-known result of Fulkerson, Hoffman, and McAndrew, links graphs having the same degree sequence.

Theorem 1 (Fulkerson Theorem [4])**.** *Two unlabeled graphs G and H have the same degree sequence if and only if there is a sequence of 2-switches that transforms G into H.*

Hence, it is possible to generate all possible realizations of a degree sequence using 2-switch operations. **Reversibility of 2-switch**

A 2-switch operation is reversible because if given its output graph, it is always possible to determine back its input graph. Suppose a 2-switch $< a, b : c, d >$ is performed on graph G and the output graph is G', then the graph G can be obtained back form G' by performing the following 2-switch on $G' : < a, d : c, b >$.

1.2.4 Degree Sequence Characterization

Definition 6. A graph class C is said to be *degree determined*, or that it has a *degree sequence characterization*, if it is possible to determine whether a graph G belongs to C from just the degree sequence of G [1].

1.2.5 Closure under 2-switch

A graph class C is said to be *closed under 2-switch operation* if and only if after performing a 2-switch on any graph G belonging to C, results in a graph G' also belonging to C. In other words, performing a 2-switch on the graphs of class C produces a graph of the same class C.

1.2.6 Realization Graph

The realization graph $G(d)$ is the graph (R, δ), where R is the set of realizations of d, and δ is the set of edges where two vertices G, G' of R are adjacent if and only if performing some 2-switch changes G into G' [3]. Since the operation of undoing a 2-switch is itself a 2-switch, $G(d)$ may be thought of as an undirected graph. The best-known result on $G(d)$ is that for any degree sequence d its realization graph is a **connected graph**; this is a consequence of a theorem of Fulkerson, Hoffman, and McAndrew.

Chapter 2

Literature Survey

In this chapter we mention various properties and configurations related to 2-switches. In Section 2.1 we derive a proof for various configurations in which a 2-switch can be applied and in Section 2.2 we mention various results known for 2-switch operations.

2.1 Configurations for 2-switch

Theorem 2. *Given a graph G, a 2-switch operation can be performed in G, if and only if an induced sub-graph on any 4 vertices of G lies in one of these three configurations:* $\mathbf{2K_2}$, $\mathbf{P_4}$, $\mathbf{C_4}$.

Proof. The presence of an alternating 4 cycle leads to the possibility of performing a 2-switch operation. Hence, four vertices are required to perform a 2-switch, also at least one perfect matching must be present and one perfect matching must be absent.

Following are the only three configurations ($2K_2$, P_4, C_4) on four vertices in which a two switch can be applied:

1. $2K_2$

   In $2K_2$ since only one perfect matching is present and rest four edges are absent; two distinct 2-switches are possible i.e. $< A, B : C, D >$ and $< A, B : D, C >$ (as shown in below figure). In this case, the result after performing a 2-switch is also a perfect matching i.e. $2K_2$.

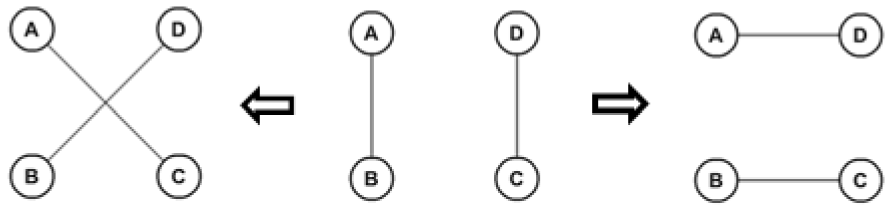


Figure 2.1: Two different possible 2-switches in $2K_2$

2. P_4

   In P_4, only a single 2-switch operation is possible i.e. $< A, B : D, C >$ (as shown in below figure). The result is also a P_4.

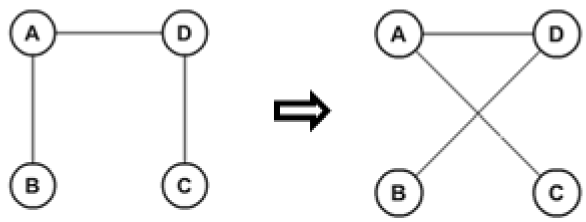


Figure 2.2: Only one possible 2-switch in P_4

3. C_4

   In C_4 since two perfect matching′s are present and rest two edges are absent; two distinct 2-switches are possible i.e. $< A, B : D, C >$ and $< A, D : B, C >$as shown in below figure. In this case, the result after performing a 2-switch is also a C_4.

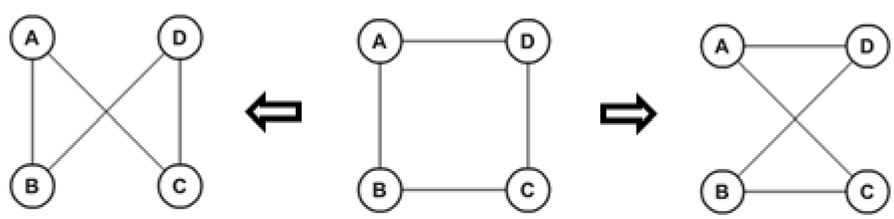


Figure 2.3: Two different possible 2-switches in C_4

Also, there are no other graphs on four vertices on which 2-switch can be applied. Hence, in general to perform 2-switch on a graph G, the induced sub graph on any four vertices must be in one of these three configurations: $\mathbf{2K_2}$, $\boldsymbol{P_4}$, $\boldsymbol{C_4}$.

Hence, in a graph G, if there is no induced sub graph on four vertices containing $2K_2$, P_4, C_4 then we can say that a 2-switch cannot be performed on G.

□

For a given graph there exists a unique degree sequence but from a given degree sequence there exist, in general, more than one graph which are not isomorphic to each other. Using 2-switch operation it is possible to generate all possible realizations of a degree sequence. However the application of a single 2-switch might change the isomorphism class of the graph. So it is necessary to study the configurations in which a 2-switch operation changes the isomorphism class of the graph.

2.2 2-switch and Isomorphism

[2] mentions a necessary and a sufficient condition for a 2-switch to change the isomorphism class of a graph.

Since a 2-switch changes the adjacencies in the graph, intuitively one can say that it will most often change the isomorphism class of the graph. However, this is not always the case. A 2-switch operation may not change the isomorphism class of the graph. For example, performing 2-switch on P_4 will result in another P_4.

2.2.1 Necessary condition for changing isomorphism class

Theorem 3 ([2]). *If a graph G admits a 2-switch that changes the isomorphism class of the graph, then G contains one of the configurations in Figure 2.4, with vertices p, q, r, s as marked, such that the 2-switch is on* $< p,q : r,s >$.

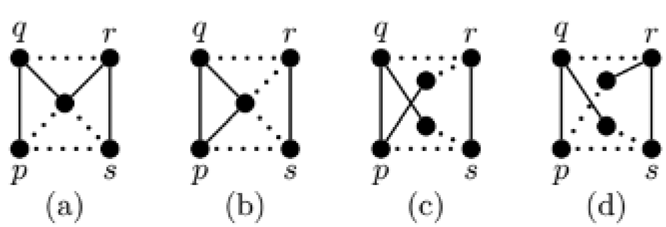


Figure 2.4: Configuration for isomorphism class changing 2-switches [2].

2.2.2 Sufficient condition for changing isomorphism class

Theorem 4 ([2]). *If a graph G contains one of the configurations shown in Figure 2.4, and the 2-switch on < p,q : r,s > produces a graph isomorphic to G, then the vertices of the configuration lie in one of the configurations in Figure 2.5, where the alternating cycle < u,v : w,x > coincides with < p,q : r,s >.*

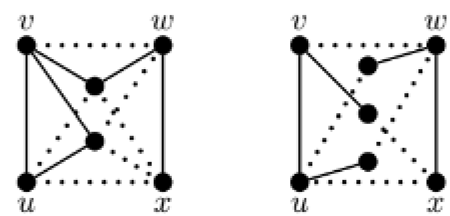


Figure 2.5: Configuration for isomorphism class preserving 2-switches [2].

Corollary 1 [2]: *The converse of Theorem 3 is true for graphs that contain neither of the configurations shown in Figure 2.5.*

CHAPTER 3

Correlation between Graph Classes and Degree Sequence

In this chapter we present some families of graphs and their degree sequence characterization based on 2-switch operation. A graph class C is said to be *degree determined*, or that it has a degree sequence characterization, if it is possible to determine whether a graph G belongs to C from just the degree sequence of G. For example, complete graphs, Split graphs, Threshold graphs. Basically, if all the realizations of a degree sequence belong to the same graph class then it is said to be degree determined. In this chapter we have studied relationship between degree sequence patterns and graph classes. We have mentioned various graph classes and their degree sequence characterization based on 2-switch operation.

3.1 Closure under 2-switch

A graph class C is said to be *closed under 2-switch* operation if and only if performing a 2-switch on any graph G belonging to C, results in a graph G' also belonging to C. In other words, performing a 2-switch on the graphs of class C produces a graph of the same class C. If performing a 2-switch operation on a graph G belonging to graph class C changes the graph class of the new graph to C', then we can say that neither C nor C' are *closed under 2-switch* operation.

1. **Complete Graphs**

   A complete graph is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge.

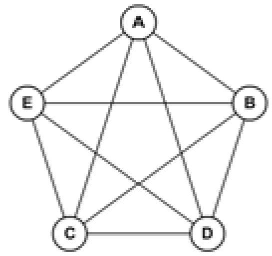


Figure 3.1: Complete graph on 5 vertices.

In case of complete graphs, an induced sub graph on any four vertices will always be complete since all vertices are connected to each other. As, there is no alternating cycle in a complete graph, no 2-switch operation can be performed. Hence, we can say that complete graphs are *degree determined*.

2. **Split Graphs**

A split graph G is a graph in which the vertex set can be partitioned into an independent set I and a clique C. A split graph can have more than one partition into a clique and an independent set.

Proposition 1. *Split graphs are closed under 2-switch operation.*

Proof. (using 2-switch operation):

Case i: All the four vertices of 2-switch are either selected from set I or all are selected from set C

As seen in above section, no 2-switch operation can be performed in a complete graph. Also, an independent set is an empty graph so 2-switch cannot be performed on the vertices of an independent set. So, in case of split graphs to perform a 2-switch we cannot select all four vertices from independent set (or clique).

Case ii: Three vertices are selected from set I and one from set C or three vertices are selected from set C and one from set I

In both the cases, two switch is not possible due to absence of an alternating 4-cycle (Theorem 2).

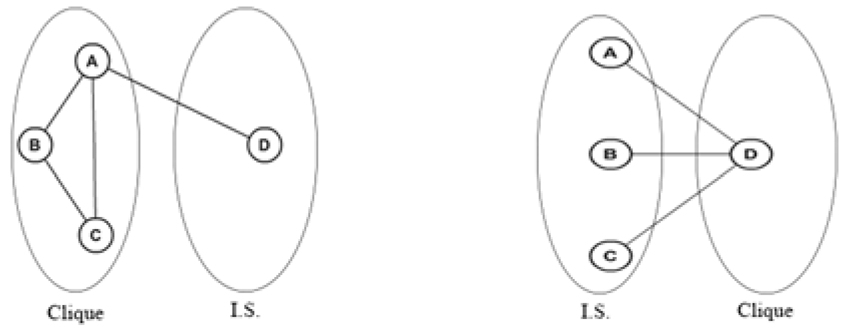


Figure 3.2: Absence of an alternating 4 cycle on selecting three vertices from the same set.

Case iii: Two vertices are selected from set *I* and two from set *C*

Here, two vertices from set *I* will be non-adjacent to each other as they lie in an independent set and the two vertices from set *C* will be adjacent to each other because they lie in a clique. The only configuration in which a 2-switch can be performed in a split graph is as shown in below figure.

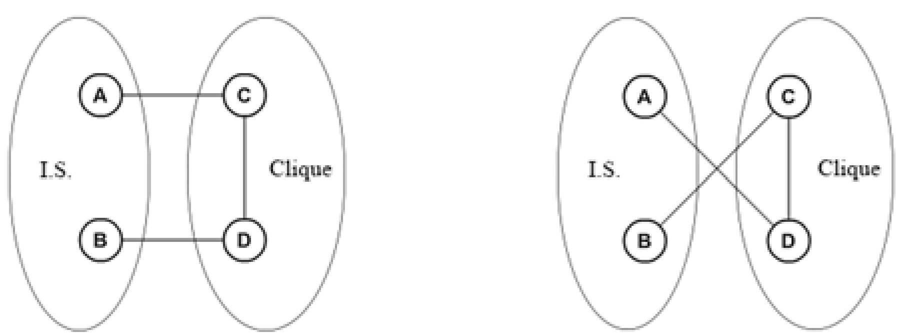


Figure 3.3: 2-switch configuration in split graph.

As we can see, the 2-switch only changes the crossing edges between the independent set and the clique, rest all is unchanged. Hence, we can conclude that the resultant graph is also a split graph.

Planar Graphs

A planar graph is a graph that can be embedded in the plane, i.e., it can be drawn on the plane in such a way that its edges intersect only at their endpoints. In other words, it can be drawn in such a way that no edges cross each other [7]. Such a drawing is called a plane graph or planar embedding of the graph.

Planar graphs are *not closed under 2-switch* operation i.e. are not degree determined. For example,

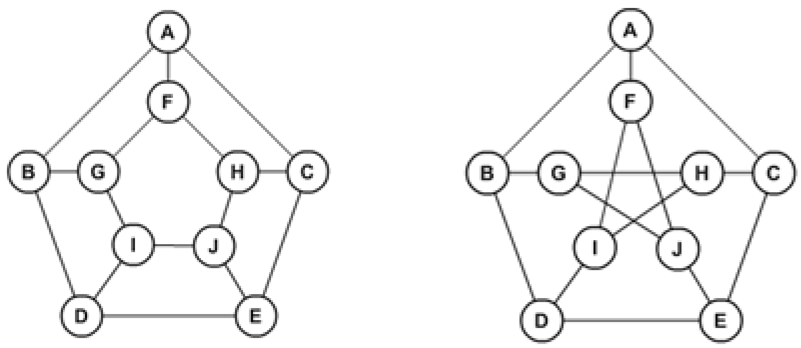


Figure 3.4: Planar and Non-Planar drawing of (3^{10})

Bipartite Graphs

A simple graph $G = (V, E)$ with vertex partition $V = \{V1, V2\}$ is called a bipartite graph if every edge of E joins a vertex in $V1$ to a vertex in $V2$. In general, a Bipartite graph has two sets of vertices, let us say, $V1$ and $V2$, and if an edge is drawn, it should connect any vertex in set $V1$ to any vertex in set $V2$.

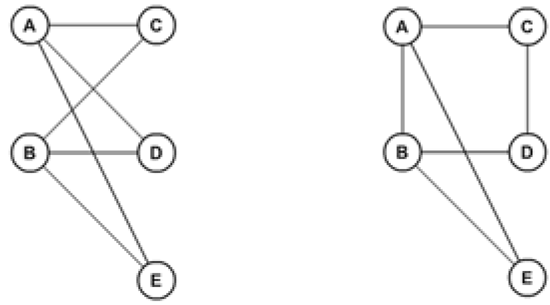


Figure 3.5: 2-switch on $K_{2,3}$

Bipartite graphs are not closed under 2-switch operation since the adjacencies are changed due to which bipartition may not be possible due to formation of odd cycles. Above figure shows class changing 2-switch on $K_{2,3}$. Hence, bipartite graphs are not degree determined classes of graphs.

□

3. **Threshold Graphs**

Threshold graph is a graph which can be constructed from a single vertex graph by repeated application of the following two operations:

(a) Addition of a single isolated vertex to the graph.

(b) Addition of a single dominating vertex i.e. a single vertex that is connected to all other vertices in the graph.

Threshold graph is a graph with no induced P_4, C_4 or $2K_2$ [6]. Hence, by Theorem 2 we can say that 2-switch operation cannot be performed on threshold graph. So, threshold graphs are degree determined.

CHAPTER 4

Parameters of Realization Graph

In this chapter we present results derived by us on various graph parameters of realization graphs.

4.1 Realization Graph

A Realization graph, $G(d)$ is an auxiliary graph defined as the graph (R, δ) where R is the set of realizations of d and two vertices G, G' of R, are adjacent if and only if performing a single 2-switch can transform G into G' [3].

Since the operation of undoing a 2-switch is itself a 2-switch, i.e. reversible, $G(d)$ is an undirected graph.

We are introducing two different versions of realization graphs:

1. **Labeled Version** i.e. allowing isomorphism:

   In the realization graph $G(d)$ of a given degree sequence d, the *labeled realizations* of d will be the vertices, and two vertices of $G(d)$ will be adjacent if and only if one realization can be generated from the other through a 2-switch operation.

2. **Unlabeled Version** i.e. not allowing isomorphism:

   In this case, $G(d)$ will include only unique realizations of a degree sequence up to isomorphism i.e. a new vertex (graph) will be added to $G(d)$ if and only if it can be obtained by performing a 2-switch on an existing vertex (graph) and it is not isomorphic to any of the existing vertices (graphs) in $G(d)$.

Example 2. *Consider the degree sequence (2, 2, 2, 1, 1). Vertices of given degree sequence are shown below:*

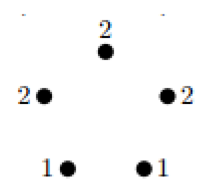


Figure 4.1: Vertices with degree sequence (2,2,2,1,1).

1. **Labeled** $G(d)$ **:**

   All realizations of (2,2,2,1,1) are as shown below. Here only two isomorphically distict realizations are obtained, however, the labeled version allows duplication in terms of isomorphism.

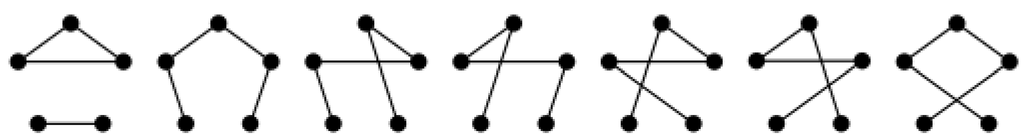


Figure 4.2: All Realizations (not considering isomorphism) of (2,2,2,1,1).

All these realizations can be generated by a sequence of 2-switch operations. An example of a 2-switch transforming one realization into another is shown in below figure.

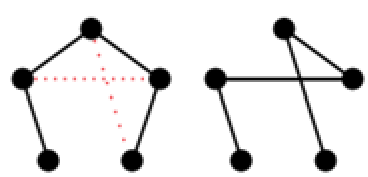


Figure 4.3: 2-Switch Transformation from one realization to another

Realization Graph of d = (2, 2, 2, 1, 1):

$V(G(d))$ = realizations of d, $E(G(d))$ = pairs joined by a 2-switch

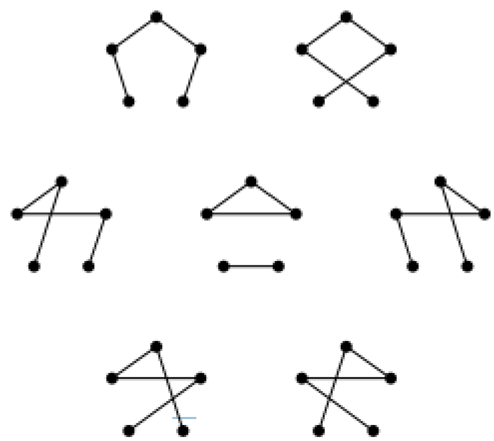


Figure 4.4: Labeled Realization Graph Vertices: $V(G(d))$.

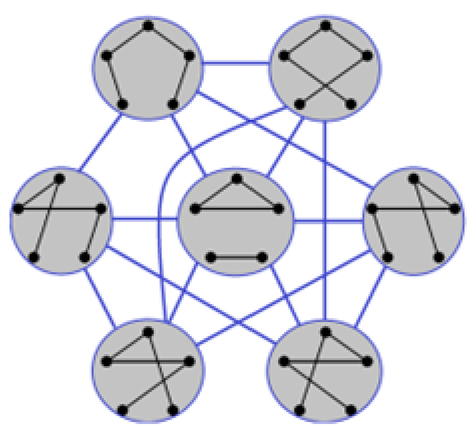


Figure 4.5: Labeled Realization Graph of (2, 2, 2, 1, 1).

2. **Unlabeled** $G(d)$:

The unlabeled version of $G(d)$ for (2,2,2,1,1) will consist only of two vertices which are the non-isomorphic realizations of (2,2,2,1,1).

Figure 4.6: Unlabeled Realization Graph of (2, 2, 2, 1, 1)

As we can see, the unlabeled version of $G(d)$ is much smaller than the labeled version as it eliminates isomorphic realizations of the degree sequence, whereas the labeled version of $G(d)$ is much more complex. Since the unlabeled version does not allow repetition up to isomorphism, our major focus will be on the unlabeled version of realization graphs.

4.2 Properties of $G(d)$

As a consequence of Theorem 1 we can say that:

Theorem 5. *G(d) is connected.*

Theorem 6. *Given a degree sequence d, G(d) $\equiv$ G(d′); where d′ is the complement of the degree sequence d.*

Proof. Given a degree sequence d, the realization graph of d' will be identical to the realization graph of d. Hence, complements of graphs of all the realizations of d will be realizations of d'. So the graphs corresponding to the vertices of $G(d')$ can be obtained by taking complements of the graphs corresponding to the vertices of $G(d)$. Also, since, 2-switch is a reversible operation; every 2-switch in realization graph of $G(d')$ will be the reverse of the corresponding 2-switch operation performed in $G(d)$. Since, in the complement graph the non-adjacent vertices become adjacent and the adjacent vertices become nonadjacent the reverse 2-switch can be applied. Hence, $G(d')$ and $G(d)$ will have the same adjacencies. Also, we can say that $G(d')$ will not contain any new vertices because if a new vertex is added to $G(d')$ then its complement will realize d and must be included in $G(d)$ also. So, $G(d) \equiv G(d')$.

□

Upper Bound on the diameter:

A single 2-switch operation can insert at most two absent edges by deleting two edges present in the graph. So to insert all the edges which are absent in the graph it will require $m/2$ (here m is the number of edges) sequence of 2-switch operations. Since the graph and its complement have the same realization graph we can say that the graph containing minimum edges (minimum of the number of edges in the graph or its complement) will give us an upper bound.

Given a graph with m edges and degree sequence d,

$$Diameter\_of\_G(d) \le \left(\frac{min\{m, \frac{n(n-1)}{2} - m\}}{2}\right)$$

4.3 Adjacency and Non-adjacency in Realization Graph

Theorem 7. *In a Labeled Realization Graph, a clique of size 3 will be formed if and only if the 2-switches involved have at least three vertices in common.*

Proof. Let's say on performing 2-switch ($S1$) on graph $G1$ we get graph $G2$ and graph $G3$ on performing 2-switch ($S2$), as shown in below figure:

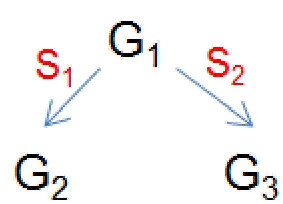


Figure 4.7: Adjacency and Non-adjacency in Realization Graph

Now in order to determine whether $G2$ and $G3$ will be connected or not in the realization graph consider the following four cases:

Case (i): *No vertex is common in $S1$ and $S2$.*

In this case, clearly the edges deleted and inserted by $S1$ and $S2$ will be distinct.

Suppose $S1 : < a, b : c, d >$ and $S2 : < p, q : r, s >$ then $G2$ and $G3$ will have difference of 4 edges. But since a single 2-switch can modify only 2 edges between four vertices it is not possible to achieve the same configuration using a single 2-switch i.e. $G3$ cannot be obtained from $G2$ (or $G2$ cannot be obtained from $G3$) through a single 2-switch. Thus $G3$ and $G2$ will be non-adjacent i.e. clique of size 3 will not be formed.

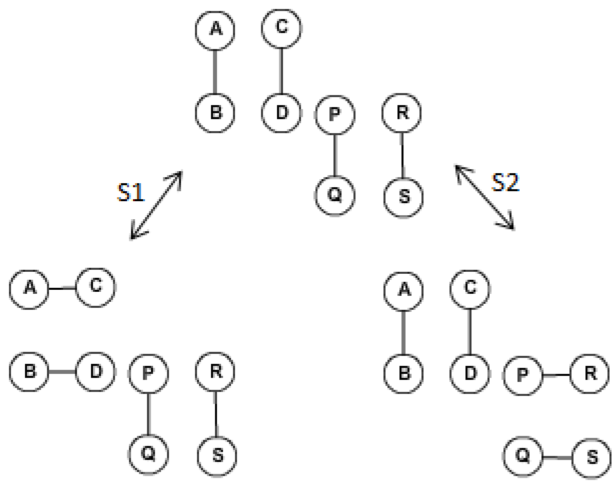


Figure 4.8: No common vertex in S1 and S2

Case (ii): *Only one vertex is common in $S1$ and $S2$.*

Similar to case (i), the edges deleted and inserted by $S1$ and $S2$ will be distinct. So, $G2$ and $G3$ will have difference of 4 edges. So, $G3$ cannot be obtained from $G2$ (or $G2$ cannot be obtained from $G3$) through a single 2-switch. Thus $G3$ and $G2$ will be non-adjacent i.e. clique of size 3 will not be formed.

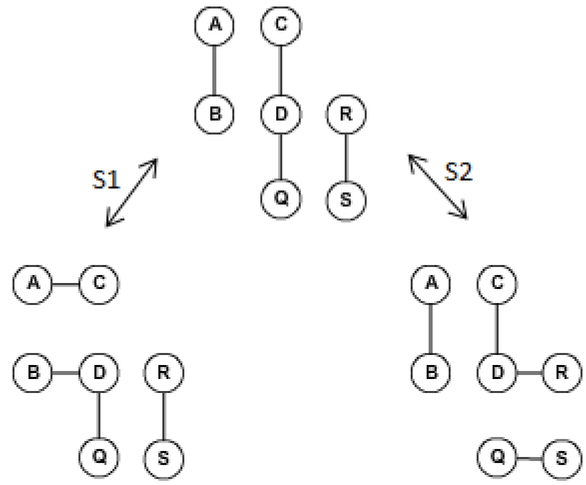


Figure 4.9: Single common vertex in $S1$ and $S2$

Case (iii): *Two vertices are common in* $S1$ *and* $S2$.

Similar to case (i) and (ii), the edges deleted and inserted by $S1$ and $S2$ will be distinct. So, clique will not be formed.

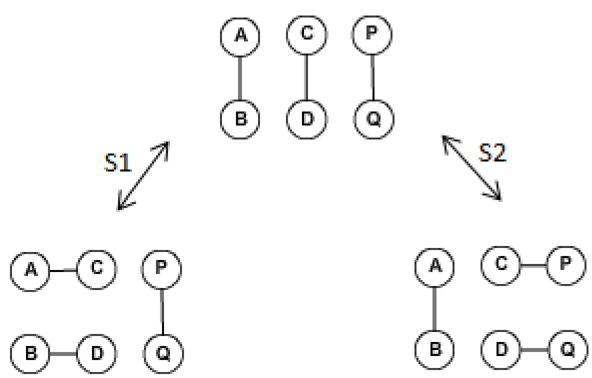


Figure 4.10: Two common vertices in $S1$ and $S2$

Case (iv): *Three vertices are common (at fixed positions) in* $S1$ *and* $S2$.

In this case as 3 vertices are common the difference in edges between $G2$ and $G3$ will be only 2 edges which can be achieved through a single 2-switch. Here, **the three common vertices are fixed** in both the 2-switch operations only one vertex varies which results in creation of one common edge in $G2$ and $G3$ and only one new edge in each of $G2$ and $G3$. Hence only two edges are different in $G2$

and $G3$ which can be added using single 2-switch operation. So, $G2$ and $G3$ will be connected. However if the position of the three common vertices is not same in both the 2-switches then the edges introduced will be different and it won't be possible to generate $G2$ from $G3$ (or $G3$ from $G2$) via single 2-switch operation.

Suppose $S1 : < a, b : c, d >$ and $S2 : < a, b : e, d >$ then $G3$ can be obtained from $G2$ by performing $< c, b : e, d >$.

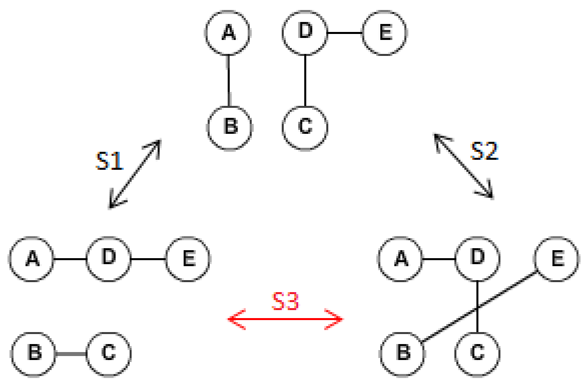


Figure 4.11: Three common vertices in $S1$ and $S2$

Case (iv): *All four vertices are common in $S1$ and $S2$.*

In this case also it is possible to get $G3$ from $G2$ through a single 2-switch since they differ in 2 edges only. Hence, $G3$ and $G2$ will be connected.

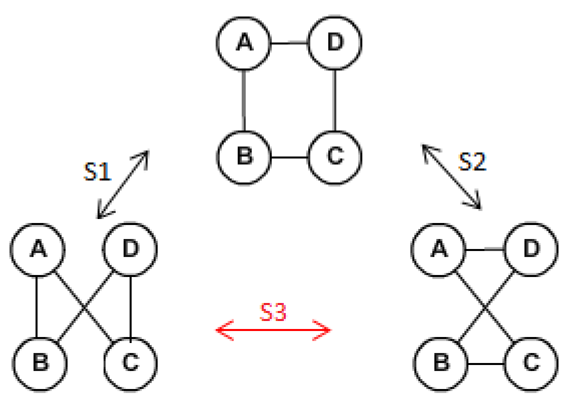


Figure 4.12: Four common vertices in $S1$ and $S2$

So, we can say that if three or four common vertices are common in $S1$ and $S2$ then only $G3$ and $G2$ will be adjacent i.e. clique of size 3 or more will be formed if and only if the 2-switches involved have three or more vertices in common.

□

We can extend the above result to prove that:

Theorem 8. *In a Labeled Realization Graph, a clique of size greater than 3 will be formed if and only if the 2-switches involved have exactly three vertices (at fixed position) in common.*

Proof. According to above result, to form a clique of size 3 we must have 3 or more common vertices, but since if all the four vertices are common then there are only two possibilities for 2-switches. Hence a clique of size greater than 3 cannot be formed if all four vertices are common. Also the position of three vertices which are common should be fixed, because if they are not fixed then it will create a difference of 4 edges which cannot be reached by a single 2-switch operation. Hence, keeping three vertices of a 2-switch operation fixed and varying only one vertex may give us cliques of sizes greater than 3.

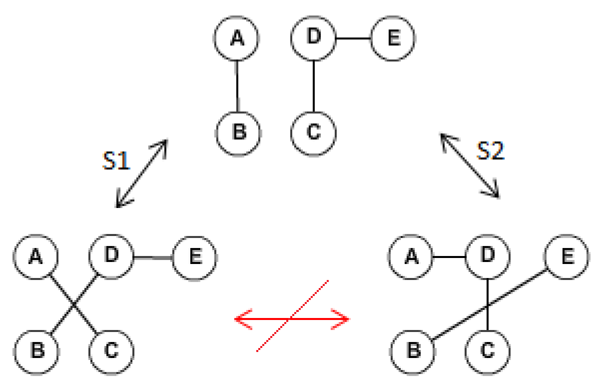


Figure 4.13: Three common vertices but not at fixed position in $S1$ and $S2$

□

Corollary 8.1. *Size of maximum clique in $G(d) \leq d_1$*

Proof. Using above result we can say that, Size of maximum clique in $G(d)$ is upper bounded by the maximum degree in d, because three vertices of 2-switch are fixed and only one varies so the vertex with maximum degree can be fixed and its neighbors can be varied giving us the upper bound on the clique size.

Since no other vertex has more neighbors than the maximum degree vertex it is not possible to get a clique of size greater than △(G).

□

Hence, as the clique size is related to the degree of vertex in the graph, the Realization Graph can have cliques of large sizes.

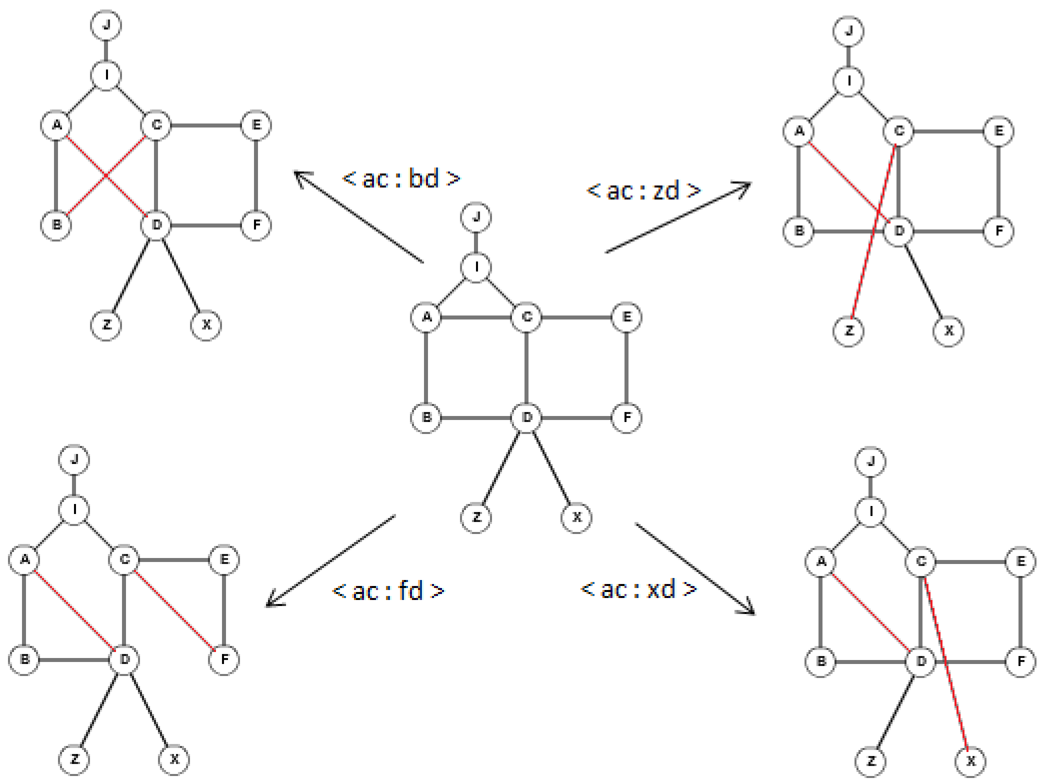


Figure 4.14: 2-switches on maximum degree vertex

Corollary 8.2. *Maximal Clique will be a Maximum clique if it involves 2-switch on a maximum degree vertex.*

This is because no other vertex has more neighbors than a maximum degree vertex.

Corollary 8.3. *In a clique in $G(d)$, union of edges introduced by 2-switches will form a star and a disjoint edge.*

Proof. As shown in Figure 4.14, the union of all the newly introduced edges by 2-switches performed in a clique in $G(d)$ forms a star and a disjoint edge. This is because of the reason that the three vertices are fixed and only one vertex varies.

Lets say we are denoting the 2-switch as $< a, b : x, d >$ where a, b and d are fixed and only x varies. Then for all possible values of x, the 2-switch will create two new edges: ad and bx. Here since a and d both are fixed ad is the disjoint edge which is always obtained whereas since x is varying, the edge bx creates a star like structure upon taking a union over x.

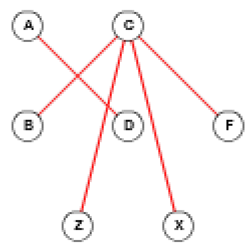


Figure 4.15: Union of edges introduced forms a star and a disjoint edge

$\square$

Proposition 2. *$G(d)$ need not be 2-connected.*

$G(d)$ is not necessarily 2-connected since cut vertex and (or) cut edge may be present. For example,

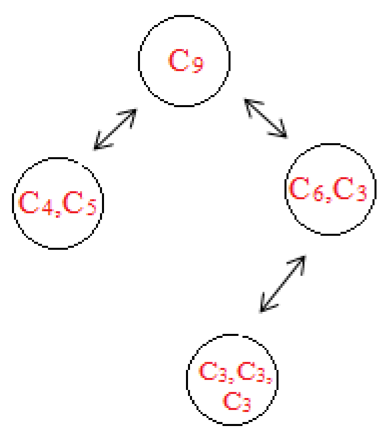


Figure 4.16: Example of a realization graph in which cut-vertex is present.

The results we have derived in this chapter are by no means exhaustive in the context of graph parameters. There is scope to obtain several more results of a similar flavor. A more organic goal of this type of an exercise is to eventually unify these apparently disparate results into a theorem characterizing realization graphs precisely. A natural follow-up to such a characterization is designing efficient algorithms to reconstruct a degree sequence (or all degree sequences) to which the given realization graph corresponds.

CHAPTER 5

Modification of Realization Graph based on Addition and Deletion of Vertices or Edges

By performing dynamic operations like adding a vertex or adding edges we can study relationship between degree sequences having isomorphic realization graphs. In some cases the dynamic operations generate graphs with identical realization graph, whereas some operations change the realization graph structure. If realization graph of a degree sequence is known, then dynamic operations can also be used to generate many other degree sequences which will have the same realization graph. Since generating all possible non-isomorphic realizations of a degree sequence is exponential in terms of n, in some cases using dynamic operations we can delete vertices and/or edges from the graph which makes the problem of generating realization graph easier. Below we present our work on the effect of some dynamic operations on graphs:

Dynamic operations on Graphs:

1. **Adding a vertex:**
   Consider the following operations:

   (a) Addition of a single **isolated vertex** to the graph.

   Since no new edges are introduced, no new 2-switches can be performed. Hence, the realization graph of the new graph does not change.

   (b) Addition of a single **dominating vertex** i.e. a single vertex that is connected to all other vertices in the graph.

Since this vertex is connected to all other vertices, this vertex can never participate in a 2-switch operation because its degree will become 3 in the induced subgraph on 4 vertices. Hence as no new 2-switches are introduced the realization graph of the new graph does not change.

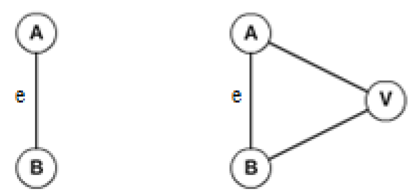
Since addition of an isolated vertex or a dominating vertex does not create any additional 2-switch, the degree sequence changes but the realization graph remains the same.

Hence, we can conclude that:

Lemma 9. *if $G(d)$ is realization graph of degree sequence $d = (d_1, d_2, \ldots, d_n)$ then $d' = (d_1, d_2, \ldots, d_n, 0)$ and $d'' = (d_1 + 1, d_2 + 1, \ldots, d_n + 1, n)$ will also have the same realization graph $G(d)$.*

Also, repeated application of these two operations will not change the realization graph. Hence, all such degree sequences obtained by repeated application of these two operations will have the same realization graph.

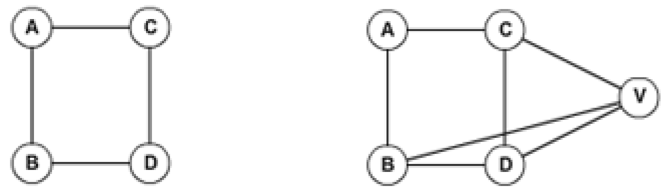
(c) Addition of a single **vertex connecting two adjacent vertices** in the graph :



i. If the edge e between those two adjacent vertices is not involved in any 2-switch operation i.e. it does not have any matching edge such that a 2-switch can be performed on them then, the realization graph does not change.

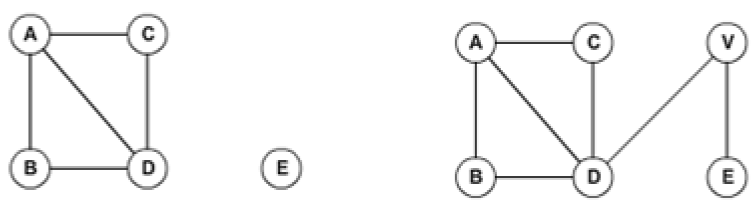
ii. If the edge e between the two adjacent vertices is involved in any 2-switch operation i.e. has a matching edge then, the realization graph changes only if configuration A (Figure 2.4) is false for any such matching edge.

(d) Addition of a single **vertex connecting any subset of the set endpoints of edges which do not have any matching edge.**



The above idea of connecting two endpoints of an edge which is not involved in any two switch can be extended to multiple edges also as all the newly introduced edges won′t have any matching edges. So the addition of a star like structure on the set (or subset) of endpoints of all such edges (not having matching edges) does not change the realization graph.

(e) Addition of a single **vertex connecting an isolated vertex** to the graph :



Since both the isolated vertex and the newly inserted vertex will be nonadjacent to all other vertices in the graph, the edge connecting both of them will be matching to all the other edges in the graph. So this creates many new possibilities for 2-switches and thereby changing the realization graph in most cases. So in this case we need to check all the four configurations (A, B, C, D) (Figure 2.4) for the newly introduced 2-switches.

2. **Deletion of vertex:**

Similar to the addition of vertices in the graph, deleting a vertex can be considered as a reverse operation. So whether the Realization Graph changes or not upon a deletion of a vertex will be similar to the case of adding a vertex.

So the cases in which a vertex is added and the realization graph does not change will be same as the case of deleting that vertex from the graph without changing the realization graph. Also the cases in which the realization graph changes on addition of vertex will also remain same for deletion of that vertex.

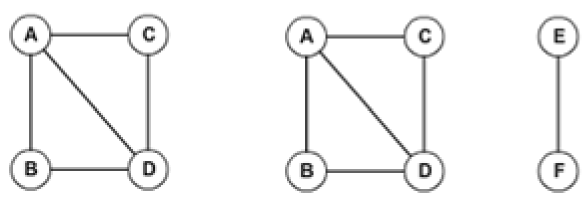
Similar to addition of vertices, if an isolated vertex or a dominating vertex is deleted then also the realization remains the same i.e.

Lemma 10. *if $G(d)$ is realization graph of degree sequence $d = (d_1, d_2, \ldots, d_n, 0)$ then $d' = (d_1, d_2, \ldots, d_n)$ will also give the have realization graph $G(d)$ and if $G(d)$ is realization graph of degree sequence $d = (d_1, d_2, \ldots, d_n, n)$ then and $d'' = (d_1 - 1, d_2 - 1, \ldots, d_n - 1)$ will also give the have realization graph $G(d)$.*
content...

3. **Addition of edge(s) :**

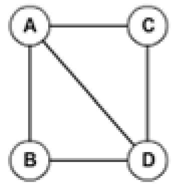
   Consider the following operations:

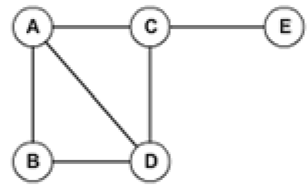
   (a) Adding a **single isolated edge :**



Since this new edge will be a matching for all the edges present in the graph many possibilities for 2-switches will arise. Also since the edge is isolated we only need to check for configurations B and C (Figure 2.4). If any one of them is true isomorphism class will change and so will the realization graph.

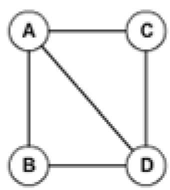
(b) Adding a **single hanging edge :**

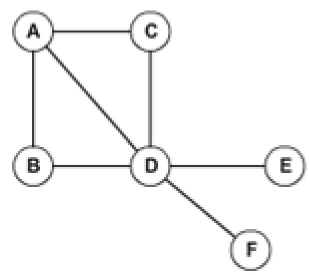




This newly added edge will be a matching for all the edges except the adjacent ones, for them we need to check all the four configurations.

(c) Adding **any number of hanging edges to a vertex with degree** $n-1$ i.e. connected to all other vertices does not change the realization graph.





The methods we have developed in this chapter can be used to generate smaller and larger (and possibly all) degree sequences which have the same realization graph as any given degree sequence.

CHAPTER 6
Unique Realizability

Graphic sequences which have only one realization up to isomorphism are said to be uniquely realizable. Since the Realization Graph of uniquely realizable graphs will consist of a single vertex only, it gives us a large family of degree sequences with same realization graph. Some examples of uniquely realizable graphs include: Complete Graphs, Threshold Graphs, Split Graphs, All 4 vertex Graphs, Perfect Matching, Star Graphs etc.

The fact that the above listed classes of graphs are uniquely realizable is straightforward.

In general regular degree sequences are not uniquely realizable. In particular,

Theorem 11. *For any $r \geq 3$, r-regular graphs on $r + 3$ vertices form more than one isomorphism class. In the case of $r = 2$, the result holds for all $n \geq 6$.*

Proof. For $r = 2$, this corresponds to a vertex disjoint collection of cycles, where each cycle in any such collection consists of at least three vertices (since we are dealing with only simple graphs).

For the case $r \geq 3$, consider 2-regular graphs on $(r + 3)$ vertices. Clearly these form more than one isomorphism class as argued in the case of $r = 2$. It is well known that two graphs are isomorphic to each other (or nonisomorphic to each other) if and only if their complements are. Thus from the fact that there is more than one isomorphism class of 2-regular graphs on $(r + 3)$ vertices, we conclude that there is more than one isomorphism class of r-regular graphs on $(r + 3)$ vertices. $\square$

Some important observations on unique realizability:

1. If a degree sequence d is uniquely realizable then adding an isolated vertex or a dominating vertex (connecting all other vertices) will also result in a uniquely realizable graph.

2. Removing a star or a triangle from a complete graph will be uniquely realizable.

3. In general, removing any uniquely realizable graph from the complete graph will also be uniquely realizable (because graph and its complement have isomorphic realization graphs).

6.1 Decision Algorithm for unique realizability:

Havel-Hakimi gives us an algorithm (Section 1.2.2) for generating one of the realizations of a degree sequence by connecting higher degree vertices first. If we can show that by changing the adjacencies i.e. connecting higher degree vertex with lower ones through 2-switch operation results in a new non-isomorphic realization (to Havel-Hakimi graph) then we can say that the degree sequence is not uniquely realizable. Using this idea of 2-switch on Havel-Hakimi graph and the necessary and sufficient configurations in which the isomorphism class is changed we have developed the following decision algorithm for determining whether a given degree sequence is uniquely realizable or not.

Algorithm 1: From a given degree sequence identify whether the given degree sequence is uniquely realizable or not?

Solution.

Input: Degree sequence $d = (d_1, d_2, \ldots, d_n)$

Assumptions:

Vertex indices are from 1, 2, . . . , n and their corresponding degrees are d_1, d_2 , . . . , d_n; i=1; A set of neighbors of p will be denoted by P and a set of non-neighbors of p will be denoted as P′.

Output:

Yes if the degree sequence is uniquely realizable and **No** if the degree sequence is not uniquely realizable

Step 0: Remove isolated and dominating vertices if any (since it does not change).

Step 1: Construct a realization of d using Havel-Hakimi construction.

- Add edges one by one (edges between lower vertex indices are added first).
- Construct adjacency matrix such that edge weight = i and increment i by 1.

Step 2: Pick the least weighted edge.

- If all the edges are exhausted then go to Step 6.
- Change the edge weight in adjacency matrix to 0.
- Call the least indexed endpoint of the edge as p and the other as q.

Step 3: Find neighbors and non-neighbors of p and q respectively.

- Add neighbors to the set P and Q respectively.
- Add non-neighbors to the set P′ and Q′ respectively.

Step 4: Pick a cross-edge between P' and Q'.

- If there are no cross-edges then go to Step 2.
- Else call the endpoint of the edge from set Q' as r and the other as s.

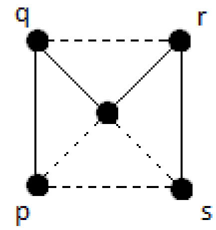
Step 5: Check for configurations.

- Check for configuration A and B.
- If A is true and B is false or if A is false and B is true then
  - Print NO and exit
- Else check for configuration C and D.
  - If C is true and D is false or if C is false and D is true then
    - Print NO and exit
  - description Else pick any other cross-edge (Step 4).

Step 6: Print YES and exit

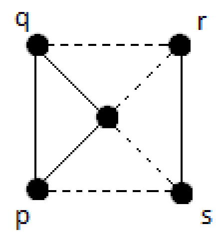
Checking for configuration A:

If $R \cap Q \neq \phi$ then check
for each $v \in R \cap Q$
if $v \in S' \cap P'$
return true
return false



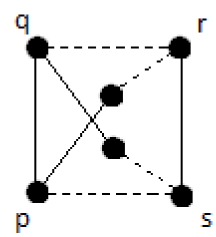
Checking for configuration B:

If $P \cap Q \neq \phi$ then check
for each $v \in P \cap Q$
if $v \in S' \cap R'$
return true
return false



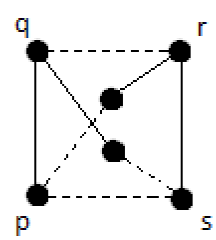
Checking for configuration C:

For each $v \in Q$
If $v \in S'$ then check
for each $u \in P$
if $u \in R'$
return true
else return false
return false



Checking for configuration D:

For each $v \in Q$
If $v \in S'$ then check
for each $u \in R$
if $u \in P'$
return true
else return false
return false



In the above algorithm, we are assigning edge weights according to the order in which they were added using the Havel-Hakimi graph construction method. So, in every iteration we pick the least weighted edge and check whether performing a 2-switch changes the isomorphism class or not. However it is not necessary that one should pick the edges in the order in which they were added.

In Step 0, we are eliminating degree 0 and degree n-1 vertices since they do not introduce any new 2-switch the underlying graph′s uniqueness remains unaffected. Hence if the underlying graph is uniquely realizable the original graph will also be uniquely realizable. We start by constructing a Havel-Hakimi Graph and then picking a least weighted edge we determine all their neighbors and non-neighbors to check for the configurations A,B,C and D. Since if A and B both are present simultaneously then the isomorphism class will not change so we check for them in step 5. If any 2-switch changing isomorphism class is found then the algorithm terminates with a NO instance and if no such 2-switch is found in the end the algorithm terminates when all the edges are exhausted with a YES instance.

Since the algorithm iterates over all edges (i.e. at most m edges) present in the graph and for each edge we are looking at all the neighbors and non-neighbors and finding a parallel edge (i.e. at most m edges) and for each of them we check for configurations (A, B, C or D) which takes O(n) time. Hence the time complexity for this algorithm is O(m^2*n).

CHAPTER 7
Conclusions

We have derived some new characteristics of realization graph .We have designed an efficient algorithm for determining whether a given degree sequence is uniquely realizable or not. We have also discussed degree sequence characterization of various families of graphs using 2-switch operation. We also presented the modifications in the realization graph based on dynamic operations like addition and deletion of edges (or vertices) in the original graph.

We conclude with several directions for future research. A fundamental question left unanswered is how to list all possible realizations of a degree sequence through 2-switch operations. Also many other graph parameters remain undiscovered for realization graphs like its radius, maximum degree and so on. Also studying the relationship between overlapping graph families for degree sequences whose realizations lie in different graph families, can be expanded further.

References

[1] M. D. Barrus. *On induced subgraphs, degree sequences, and graph structure*. University of Illinois at Urbana-Champaign, 2009.

[2] M. D. Barrus. On 2-switches and isomorphism classes. *Discrete Mathematics*, 312(15):2217–2222, 2012.

[3] M. D. Barrus. On realization graphs of degree sequences. *Discrete Mathematics*, 339(8):2146–2152, 2016.

[4] D. R. Fulkerson, A. J. Hoffman, and M. H. McANDREW. Some properties of graphs with multiple edges. In *Selected Papers Of Alan J Hoffman: With Commentary*, pages 75–86. World Scientific, 2003.

[5] S. L. Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph. i. *Journal of the Society for Industrial and Applied Mathematics*, 10(3):496–506, 1962.

[6] N. V. Mahadev and U. N. Peled. *Threshold graphs and related topics*, volume 56. Elsevier, 1995.

[7] D. B. West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.