Acyclic Edge Colouring of Subgraphs of Hypercubes

by

Maulik Soni
201811057

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

June, 2020

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.

Maulik Soni

Certificate

This is to certify that the thesis work entitled Acyclic Edge Colouring of Subgraphs of Hypercubes has been carried out by Maulik Soni 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.

Dr. Rahul Muthu
Thesis Supervisor

Acknowledgments

First of all, I would like to thank my supervisor/ mentor/ guide, Prof. (Dr.) Rahul Muthu, for believing me that I can work on this topic, for teaching me so many things of the field and for his constant help and support during my entire research period. He guided me at every step from the start till the end about what to do and how to do, whenever I was needed to be. I have made a lot of mistakes during the year, but he always helped me to correct those. I would not have completed this without his help. I am very grateful to him. Thank you so much, sir.

I would like to thank Prof. (Dr.) V. Sunitha, for teaching us graph theory very well as a part of the curriculum. It helped me at many points during the research.

I would like to thank my very good friend, Kishan Patel. Without his help and support, none of these would have been possible, many many thanks to him. Also many thanks to all those friends, whom I can not name them here, who helped me a lot, prior to starting my journey at DA-IICT.

I would also like to thank my friends Jigar Sompura (201811071), Shruti Vaghani (201811072), Dhara Sapara (201811083) and all well wishers, for their constant motivation, support and encouragement throughout the year of study and through the process of researching and writing this thesis. They have performed a major role in my well being.

Above all I am thankful to my family for their support and patience. Thank you!

Contents

Abstract

Acyclic edge colouring is a very important combinatorial optimization problem, which many researchers are currently working on. Acyclic edge colouring is assignment of colours to vertices or edges of the graph such that, no two colour induced subgraph contains any cycle.
In its most general form, it is a very difficult problem and thus research work on this problem, usually focuses on some limited aspect. In particular one approach to this problem is to find exact values for the acyclic chromatic index for special families of graphs. One might also look for corresponding efficient algorithms, for such optimal colourings. However, it is essential to study a few papers on this problem of acyclic edge colouring in order to develop one's techniques and solve a new aspect of this problem.
We aim to obtain optimal bounds and algorithms for the acyclic edge colouring problem on the class of subgraphs of hypercubes. We have given the best colour bound of $\Delta + 1$ (where Δ is maximum degree of graph), for three categories of subgraphs of hypercubes. First is arbitrary subgraph of any dimensional hypercube, second is $(d-1)$-regular subgraphs of d-dimensional hypercube with constraints of limited 2 and 4 cross edges. and third is 3-regular subgraphs of any dimensional hypercube. We have given the some approaches proven effective from mathematical or simulation point of view. All the colour bounds given by us, are supported by well known, Acyclic Edge Colouring Conjecture.
The approach of colouring 3-regular subgraphs of any dimensional hypercube, is not just limited to subgraphs of the hypercube, but it can also be considered for any regular bipartite graph with regularity equals to 3.

Keywords: edge colouring, proper edge colouring, acyclic edge colouring, hypercubes, arbitrary subgraphs of hypercubes, $(d-1)$-regular subgraphs of d-dimensional hypercubes, 3-regular subgraphs of hypercubes.

List of Principal Symbols and Acronyms

$\chi'(G)$	Edge chromatic number of the graph
$\chi(G)$	Chromatic number of the graph
Δ	Maximum degree of the graph
ϕ	Incidence function
$a'(G)$	Acyclic edge chromatic number of the graph
$a(G)$	Acyclic chromatic number of the graph
E	The set of all edges in the graph
Q_n	n-dimensional hypercube
V	The set of all vertices in the graph
a.k.a.	also known as

List of Tables

List of Figures

CHAPTER 1
Introduction

A graph is a structure consisting of vertices and edges, widely studied in mathematics and specifically as a basic element in graph theory. Vertices (a.k.a. nodes) often represents objects of interest for which the graph is drawn. Similarly, edges represents relation between vertices.
For example, if we are talking about people, then each person will be represented by an individual vertex and a person's relation with another person will be represented by an edge between both vertices. If we are talking about a map, then each city can be represented by a vertex and a path to another city can be represented by edge between the two vertices.
There are many real-life problems which can be visualised and solved easily by graph structure. Using graph theory and making a graph based model of the problem, exploring various aspects of the problem becomes quite easy.
Leonhard Euler is called father of graph theory, he solved the very famous Königsberg bridge problem. Before moving further, let us see more about graphs from a mathematical point of view and all the key terms and concepts which are used extensively here.

1.1 Graph

A graph G is denoted as an ordered pair $G = (V, E, \phi)$ or just simply $G = (V, E)$. Where,

- V is set of all vertices in the graph.
- E is set of all edges in the graph.
- ϕ is an *incidence function*, a function that assigns pairs of vertices to edges of the graph.

Based on types of edges in the graph, there are mainly two categories: **directed graph** (Figure 1.1) and **undirected graph** (Figure 1.2). Directed graph edges rep-

resents relation only in direction of the edge, whereas undirected graph edges represents relation in both direction. A common example that distinguishes both are father-son relation and friendship relation respectively.

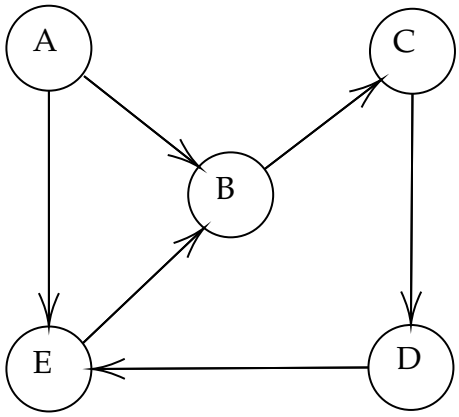

Figure 1.1: Directed graph

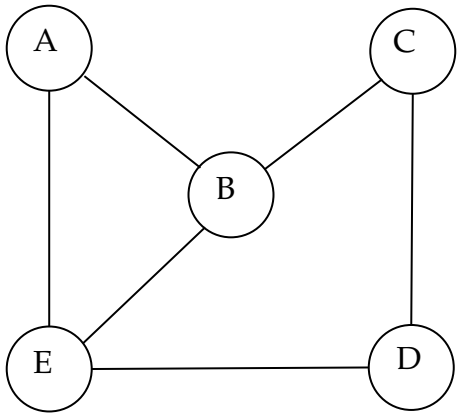

Figure 1.2: Undirected graph

Simple graph (Figure 1.3) is a graph with no loop (a.k.a. self-loop, Figure 1.4, edge that connects one vertex to itself) and no multi-edges (a.k.a. parallel-edges, Figure 1.5, different edges with same endpoints). All graphs discussed throughout this thesis are simple and undirected in nature.

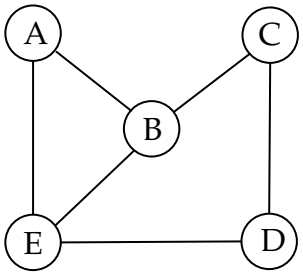

Figure 1.3: Simple graph

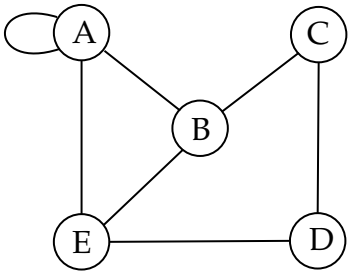

Figure 1.4: Graph with loop

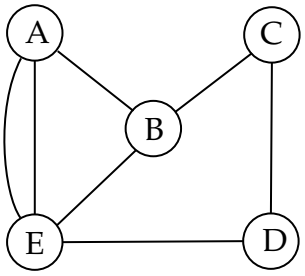

Figure 1.5: Graph with multi-edge

The total number of vertices connected with current vertex is called the degree of the current vertex. For example, in Figure 1.3, degrees of vertices A, C, D are 2; degrees of vertices B, E are 3. The vertex with highest degree among all other vertices is called **maximum degree of the graph** and denoted by $\mathbf{\Delta(G)}$ or simply Δ , hence for above example Δ is 3.

1.2 Bipartite Graph

A bipartite graph is a graph in which, the whole vertex set can be divided into two parts (a.k.a. partite sets) which are disjoint sets and both endpoints of each edge are from different partite sets. In other words, each edge start from one of the vertices of one partite and ends in one of the vertices of another partite. See an example below:

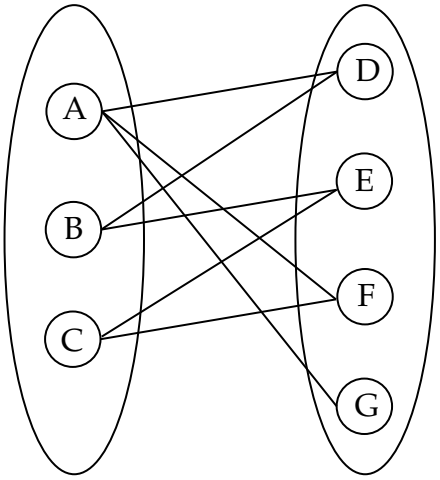

Figure 1.6: Simple bipartite graph

As one can see in the Figure 1.6, one partite contains vertices A, B, C and other partite contains vertices D, E, F, G. All the edges are between vertices of different partite sets. Note that, if there exists any cycle, it must of even length. Since, cycle can be assumed as a path starting from one vertex, traversing through two or more different vertices than starting vertex, and finally coming into starting vertex, making whole cycle. And we have a property that edges are connected between vertices of different partites. So, the path will always alternating between both partites, and finally to form a cycle it will come to the same partite making the length of path (or cycle, if ending vertex of path is same starting vertex) always even. This property will be useful later on.
Some examples of bipartite graphs are trees and hypercubes. Again one should take note that every tree or hypercube will be a bipartite graph, but not every bipartite graph can be tree or hypercube.

1.3 Hypercube (Q_n)

The n-dimensional hypercube, denoted by Q_n, is a special graph structure with 2^n vertices and $2^{n-1} \cdot n$ edges. How this graph is formed is also explained further,

but before that see this basic examples of hypercubes.

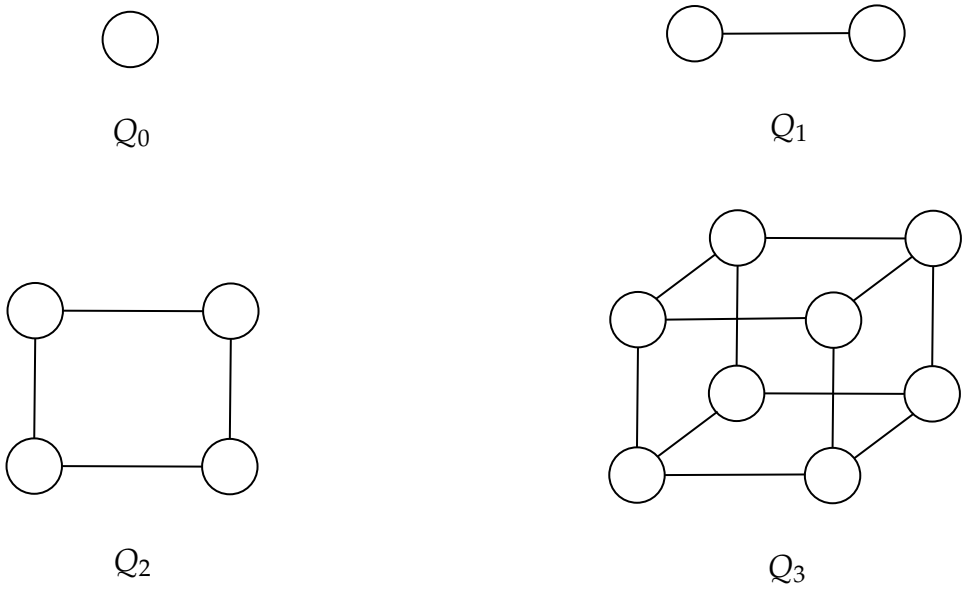

Figure 1.7: Hypercubes Q_0, Q_1, Q_2 and Q_3

As one can see in Figure 1.7, it is clear that n-dimensional hypercube is constructed by having two $(n-1)$-dimensional hypercube and connecting each vertex of first copy to same vertex in second copy, i.e. vertex x in first copy will be connected to vertex x in second copy.
Each vertex in hypercube will have distinct label of n-bit binary string, and any two vertices of the graph will be connected if and only if they differs by only one bit of their label. Adjacency matrix of hypercube be symmetrical and all diagonal elements will be 0, this way we can say that graph has no loop. Hence, studying hypercube using adjacency matrix can be done by considering either upper-triangular or lower-triangular elements.
Hypercubes are bipartite and regular with regularity equal to dimension. Hence, Δ for hypercube will be its dimension. Various important characteristics of hypercubes from theoretical aspects are extensively researched by Harary et al., in [6].

1.4 Proper Colouring of Vertices/Edges

Graph colouring is process of assigning labels to vertices or edges, and those labels are usually colours. If colours are assigned to vertices of the graph, then it is called vertex colouring. Similarly, if colours are assigned to edges of the graph,

then it is called edge colouring.
A maximal subset of V or E, coloured with the same colour is called a **colour class**. In the context of vertex colouring, this colour class is an independent set, whereas in context of edge colouring, the colour class will be a matching.

Proper vertex colouring

Proper vertex colouring is assignment of colours to all the vertices of the graph in such a way that, no pair of adjacent (a.k.a. neighbour) vertices gets the same colour. Hence, all the vertices connected with the current vertex will have different colour than the current vertex. Proper vertex colouring is referred as vertex colouring, unless otherwise stated.
The minimum number of the colours needed to colour all the vertices properly, is called **chromatic number** of the graph, and is denoted by $\chi(G)$. The graph with $\chi(G) = k$ is called k-chromatic graph.

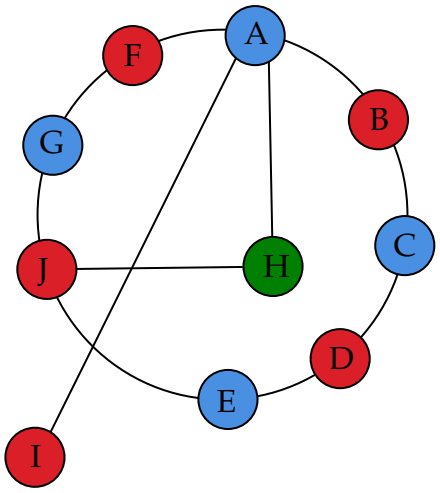

Figure 1.8: Proper vertex colouring of simple graph

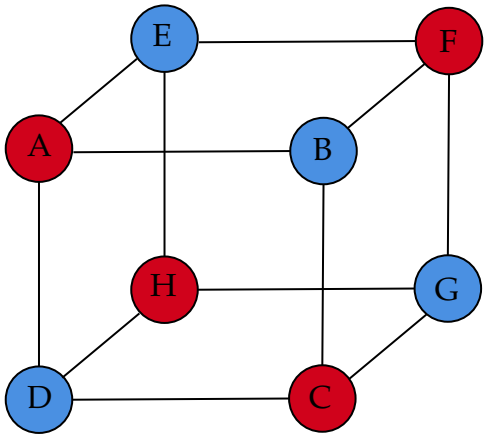

Figure 1.9: Proper vertex colouring of hypercube Q_3

As one can see in Figure 1.8 and Figure 1.9, each and every vertex is coloured using different colour than its neighbours, also with minimum number of colours. Note that, since hypercubes are bipartite graphs in structure, so all vertices in single partite will get same colour, consequently only two colours needed for proper vertex colouring. $\chi(G)$ for Figure 1.8 and Figure 1.9 is 3 and 2 respectively.

Proper Edge colouring

Proper Edge colouring is assignment of colours to all the edges of the graph in such a way that, no pair of adjacent edges gets the same colour. All the edges incident on a single vertex will have different colours. This shows that minimum number of colours needed for proper edge colouring of any graph G will be at least $\Delta(G)$.

The minimum number of the colours needed to colour all the edges properly, is called **edge chromatic number** (a.k.a. chromatic index) of the graph, and is denoted by $\chi'(G)$. The graph with $\chi'(G) = k$ is called k-edge-chromatic graph.

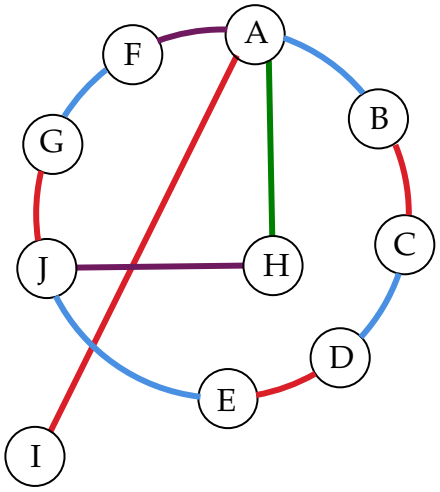

Figure 1.10: Proper edge colouring of simple graph

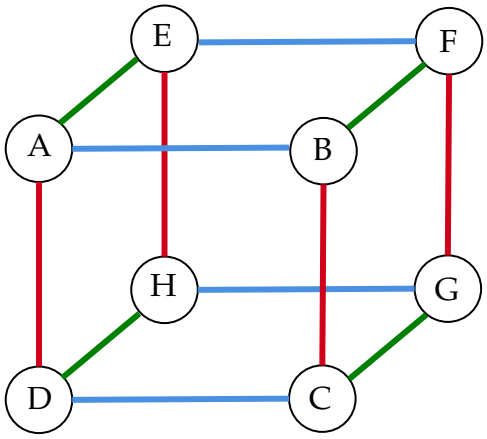

Figure 1.11: Proper edge colouring of hypercube Q_3

As one can see in Figure 1.10 and Figure 1.11, each and every edge is coloured using different colour than other edges incident on it's both endpoints' vertex, also with minimum number of colours. Note that, as mentioned earlier, proper edge colouring of the graph takes at least $\Delta(G)$ colours. For hypercube Q_n , edge chromatic number $\chi'(G)$ will always be n, i.e. each edge of single dimension will be coloured with same colour. $\chi'(G)$ for Figure 1.10 and Figure 1.11 is 4 and 3 respectively.

1.5 Acyclic Colouring of Vertices/Edges

Acyclic vertex/edge colouring is a proper vertex/edge colouring with one additional condition that, the graph should not contain any bichromatic cycle.

Bichromatic cycle is a cycle in a graph in which every vertex/edge is coloured

alternately with only two colours. In other words, if subgraph induced by union of any pair of colour class contains a cycle, then that cycle is called bichromatic cycle.

Acyclic Vertex Colouring

Acyclic vertex colouring of a graph is assignment of colours to each vertex such that it is a properly vertex coloured, with no bichromatic cycle. In Figure 1.8, we can see that there are total three cycles in the graph:

1. $A - B - C - D - E - J - H - A$
2. $A - H - J - G - F - A$
3. $A - B - C - D - E - J - G - F - A$

Note that, first and second cycles are not bichromatic, since in both cycles all the vertices are coloured with either red or blue colour except node H, which is coloured with green colour, hence, making cycle of more than two colours. However, the third cycle is bichromatic, because all the vertices are coloured with either red or blue colour, resulting in a bichromatic cycle. Since the graph does contain a bichromatic cycle, it is properly vertex coloured but not acyclic.

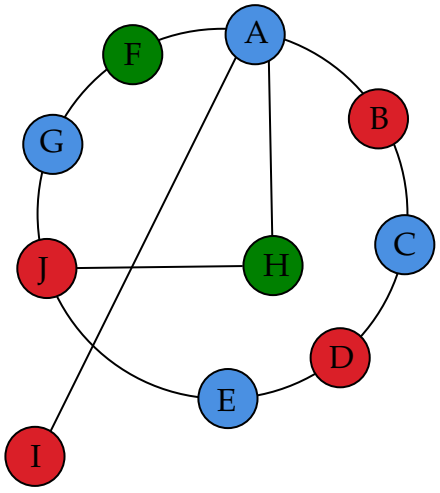

Figure 1.12: Acyclic vertex colouring of simple graph

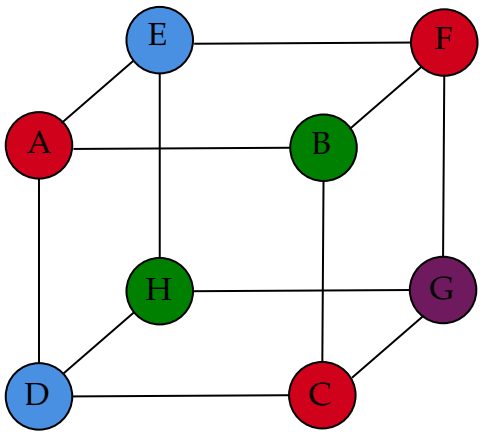

Figure 1.13: Acyclic vertex colouring of hypercube Q_3

The same idea is with properly coloured hypercube Q_3 in Figure 1.9, all the vertices in the graph are coloured with either red or blue colour and graph also contains many cycles, making the vertex colouring to be proper but not acyclic.

Acyclic vertex colouring version of Figure 1.8 and Figure 1.9 is given in Figure 1.12 and Figure 1.13 respectively.
The minimum number of the colours needed to colour all the vertices acyclically, is called **acyclic chromatic number** of the graph, and is denoted by $a(G)$. The value of $a(G)$ for Figure 1.12 and Figure 1.13 is 3 and 4 respectively.

Acyclic Edge Colouring

Acyclic edge colouring of a graph is assignment of colours to each edge such that the graph is properly edge coloured, with no bichromatic cycle. As per Figure 1.10, we can see that all the edges in each cycles are coloured with at least three colours, hence there is no bichromatic cycle and edge colouring is proper as well as acyclic. But the properly edge coloured hypercube Q_3 in Figure 1.11, there exists many cycles in which all edges are coloured with only two colours. So, we need to break all those cycles by introducing one or more colours to convert it's proper edge colouring into acyclic.

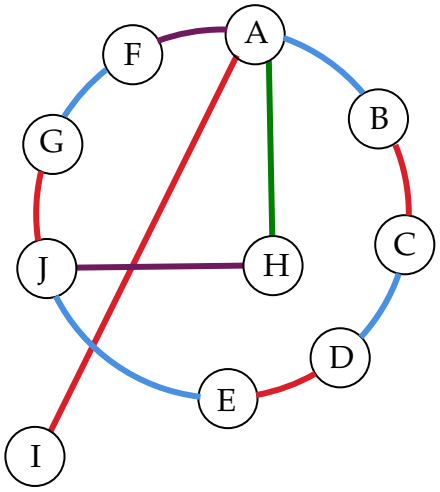

Figure 1.14: Acyclic edge colouring of simple graph

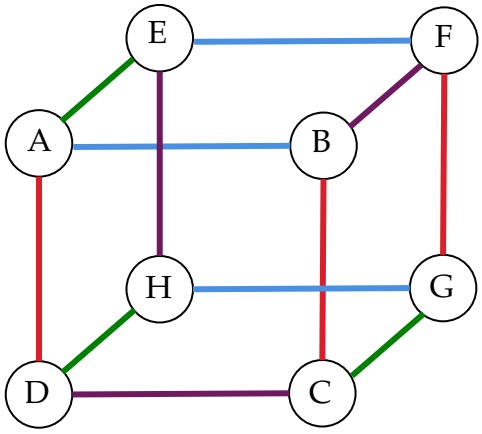

Figure 1.15: Acyclic edge colouring of hypercube Q_3

Acyclic edge colouring version of Figure 1.10 and Figure 1.11 is given in Figure 1.14 and Figure 1.15 respectively.
The minimum number of colours required to colour all the edges of a graph acyclically, is called **acyclic edge chromatic number** (a.k.a. acyclic chromatic index) of the graph, and is denoted by $a'(G)$. The value of $a'(G)$ for both Figure 1.14 and Figure 1.15 is 4.

1.6 Applications

Graph theory is a very vast area of study and extensively researched. In many aspects of real-life problems, graph theory has been proven useful.

Hessians

Computing Hessians of a matrix is one of the important application of acyclic edge colouring.

Sudoku

Each cell in the Sudoku can be represented as a vertex, and the number assigned in the cell will be colour of the vertex. Each vertex will be connected to other vertex if any of the following conditions satisfied:

- Both cells (vertices) are in same row in Sudoku.
- Both cells (vertices) are in same column in Sudoku.
- Both cells (vertices) are in same 3X3 block in Sudoku.

Now, the answer of the Sudoku will be proper vertex colouring of this graph.

Map colouring

Each region in the map will be represented as a vertex, and two regions (vertices) will be connected by an edge, only if they share a common border between them. The same context can be used to solve GSM network distribution problem, where it has been proven that at most four different frequencies are needed.

Scheduling

Most of the scheduling problems can be solved by creating graph for respective context. The final goal is to distribute resources among consumers without any conflict. This scheduling can be for time-tables, air-crafts or system resources etc.

1.7 Problem Statement

The acyclic edge colouring is in general, a very difficult problem. So, it is solved usually for specific class of graph. We are going to acyclically edge colour the sub-

graphs of hypercubes. We have divided the work of colouring subgraphs further into three categories:

1. Arbitrary subgraphs of d-dimensional hypercube.
2. $(d-1)$-regular subgraphs of d-dimensional hypercube, with 2 and 4 cross edges.
3. 3-regular subgraphs of d-dimensional hypercube.

The acyclic edge colouring is optimal in terms of colour bounds. The aim to get $a'(G) = \Delta + 1$, for any G as specified in above three categories, where Δ is maximum degree of the graph G. Note that, The colour bound we are giving here, are also bounded by well-known AECC.
The third category of graphs, are not just limited to subgraphs of hypercubes. The approach we are presenting to acyclically edge colour 3-regular graphs, also expands to solve any 3-regular bipartite graph, which is eventually broader graph class than the graph class discussed in the thesis.

1.8 Motivation

Graph colouring has became a highly studied area of graph theory. Many mathematicians in the history, gave conjectures and theorems for minimum number of colour required to colour the vertices or edges of the graph.
By the time, these colour bounds for general or specific category of graph, were reduced drastically. Today, there exists many problem including some of the applications mentioned in the Section 1.6 above, which uses the graph colouring to easily solve the problems at hand.
Consequently, we have also tried to give acyclic edge colouring of specific class of subgraphs of hypercubes, to achieve the best bounds possible. The goal is to achieve acyclic edge colouring of subgraphs of hypercubes with minimum number of colours, i.e. $\Delta + 1$ total colours, where Δ, as we know, is maximum degree of the graph.

1.9 Thesis Outline

In this section, we give some brief idea about the flow of thesis. We already discussed what is graph and categories of graphs like, simple graph, bipartite graph and hypercube. We also discussed proper vertex/edge colouring as well as acyclic

edge colouring, and completed first chapter of introduction by giving some of the applications of graph colouring, problem statement and thesis motivation.
The second chapter is all about the literature survey, i.e. what we have referred and found some of the already existing solutions or research done in area of graph colouring, as well as some of the techniques we are going to use in the thesis.
The third chapter is about the approached, which we tried but did not give satisfactory results, in terms of acyclic edge colouring or colour bounds, and which approach lacks where. So, if one get some idea about what we faced in those specific approaches, solving those problems might give a good outcome from those approaches.
The fourth, fifth and sixth chapters are about successful approaches, which gave 100% results for acyclic edge colouring of various subgraphs of hypercubes. The fourth chapter is for arbitrary subgraphs of hypercubes, fifth chapter is for $(d-1)$-regular subgraphs of d-dimensional hypercube, with 2 and 4 cross edges, and sixth chapter is for 3-regular subgraphs of hypercubes.
Finally, in seventh chapter we concludes the thesis, with some of the possible future work and scope.

CHAPTER 2
Literature Survey

The concept of acyclic colouring was introduced first by Grünbaum [5]. Since then, it has been a widely researched area of graph theory. Various applications are already discussed earlier. There are many results available for a variety of graph classes. We are going to give a brief survey of some studies, concepts, theorems, conjectures and state of the art results, to compare with our approaches further and get a better understanding.

Determining the acyclic vertex/edge chromatic number ($a(G)$ or $a'(G)$) has been a hard problem from both a theoretical and an algorithmic point of view. Alon and Zaks [2] proved that it is NP-complete to determine if $a'(G) \leq 3$ for an arbitrary graph G. There exists many graph classes, which are highly structured and simple in nature, for which the value of $a'(G)$ is not ascertained yet.

Colouring may be done in many ways, for example by trial and error, analysing the structure, etc. Trial and error approach may be considered as greedy and brute-force way but it helps for analysing the graphs which are covered by a larger graph classes or are highly unstructured, for example, arbitrary graph generated with Erdös-Rényi model which defines $G(n, p)$ as a arbitrary graph with total n vertices and existence probability of each possible edge is p. Similarly, structural analysis can be done if structure of the graph is fixed or very low variability.

It can be noticed that for any arbitrary graph G,

$$\Delta(G) \leq \chi'(G) \leq a'(G) \tag{2.1}$$

since, the proper edge colouring will take at least $\Delta(G)$ colours as discussed earlier, and the acyclic edge colouring will also take at least zero or more extra colours than the total number of colours used in the proper edge colouring.

Let us go through each paper we studied in our literature survey by categorizing into major (used directly and mainly throughout the thesis) and miscellaneous (used indirectly but definitely proven helpful).

2.1 Vizing's theorem

Vizing's theorem [12] states that,

$$a(G) = \{\Delta(G), \Delta(G) + 1\} \tag{2.2}$$

i.e., any arbitrary graph G can be properly edge coloured using at least $\Delta(G)$ colours and at most $\Delta(G) + 1$ colours. He also gives classification of graphs, class one graphs(for example, bipartite graphs) and class two graphs (for example, complete graphs and regular graphs with odd number of vertices). He states that, class one graph can be properly edge coloured with only $\Delta(G)$ colours, but class two graphs requires $\Delta(G) + 1$ colours for proper edge colouring.

2.2 AECC

Alon, Sudakov and Zaks [1] gave a conjecture that for any graph G,

$$a'(G) \leq \Delta(G) + 2 \tag{2.3}$$

always holds, i.e. any graph can be acyclically edge coloured by using at most two extra colours than maximum degree of the graph. This conjecture is widely known as **AECC (Acyclic Edge Colouring Conjecture)**. They [1] also gave examples graphs, which uses $\Delta(G) + 2$ colours for acyclic edge colouring. Many recent studies tried to bound and improve their results and have so far resulted in this conjecture holding true for all graph classes considered.

2.3 Grid-like graphs

Muthu, Narayanan and Subramanian [9], gave a polynomial time algorithm to optimally acyclic edge colour the hypercubes. They proved that for any hypercube G,

$$a'(G) = \Delta(G) + 1 \tag{2.4}$$

We already discussed how the d-dimensional hypercube is formed using two $(d-1)$-dimensional hypercubes. Here [9] the key idea to acyclically edge colour the d-dimensional hypercube is, acyclically colour first copy of $(d-1)$-dimensional hypercube and then use non-fixing permutation to colour the same edges in sec-

ond copy of $(d-1)$-dimensional hypercube, both using total d colours, and then colour the cross-edges connecting corresponding vertices in both copies with last $(d+1)^{th}$ colour. This ensures acyclicness and they [9] have also given a proof for that.

2.4 Subcubic graphs

Skulrattanakulchai [11] gave a linear time algorithm, which performs acyclic vertex colouring using at most 4 colours and acyclic edge colouring using at most 5 colours for subcubic graphs (graph G with $\Delta(G) \leq 3$), i.e.,

$$a(G) \leq \Delta(G) + 1 \tag{2.5a}$$

$$a'(G) \leq \Delta(G) + 2 \tag{2.5b}$$

He [11] has done the colouring and proved by analysing the structure of the graph, with listing all its possible various configurations and recolouring the edges if required in some cases.

2.5 k- Factorization

k-factorization partitions the edges of the graph into k disjoint factors. So, a graph is k-factorable if it admits a k-factorization. A matching in a graph is an independent set of edges without common vertices. A perfect matching is a matching, which matches all the vertices of the graph. A 1-factorization is a collection of edge disjoint perfect matching that span all the edges of a graph. Hence, k-regular graph contains k pairwise edge disjoint perfect matchings if the graph has a 1-factorization. If we remove any 1-perfect matching of a k-regular graph, it becomes a $(k-1)$-regular graph. Figure 1.11 is best example, which is properly edge coloured hypercube of dimension (also regularity) 3 and each perfect matching is given a specific colour.
Alon, Sudakov, Zaks [1] gives a conjecture based on conjecture given by Kotzig [8]. The conjecture states that, any complete graph with even number of vertices (where minimum number of vertices is 4) can be decomposed into $\Delta - 1$ perfect matchings, and union of any two perfect matchings makes a Hamiltonian cycle. They [1] also further shows that, if graph G is properly coloured with $\Delta(G) + 1$ colours, then assigning $(\Delta(G) + 2)^{th}$ colour arbitrarily and independently, while taking care that no two adjacent edges gets the same $(\Delta(G) + 2)^{th}$ colour, there are

only two possibilities:

1. Colouring remains proper (since, no adjacent edges gets the same colour).

2. Colouring becomes acyclic (all cycles contains at least 3 colours).

Since our hypercubes are regular graphs in structure, 1-factorization will give the perfect matching. Hence, for k-regular graphs, there will be k factors, and assigning edges of each factor a same colour will give proper edge colouring. Now, we can use the last $(k+1)^{th}$ colour to break the cycles generated in proper edge colouring stage.
Some work is done in this direction, and the approach is explained in detail in Section 6.2, where the concept of k-factorization has been used and we have given the last $(k+1)^{th}$ colour, arbitrarily at first to speed up the process and then step by step carefully to covert the edge colouring into acyclic completely.

2.6 Other

Nešetřil and Wormald [10] have given $a'(G) = \Delta(G) + 1$ for arbitrary d-regular graphs, which also holds true for AECC conjecture's colour bound of $\Delta(G) + 2$. Note that, hypercubes are also regular graphs, but arbitrary regular graphs covered here [10] are generated with Erdős–Rényi model, which are quite different from the regular graphs addressed in the thesis.

CHAPTER 3
7 ways that won't work

Let's understand the complexity of the colouring first. We follow this basic terminology for this chapter.

- d-dimensional hypercube will be referred as H and its subgraph will be referred as G for this context.
- The current edge will be referred to as e and neighbour/s of the current edge e will be referred as $N(e)$.
- Cycle C_e is α, β-*dangerous* if it has uncoloured edge e, and removing e will create 2-coloured (α and β) odd-length path. Hence, when colouring e using the appropriate colour from $\{\alpha, \beta\}$, it will be still properly coloured but that will become a bichromatic cycle of colours α and β.
- *Available colours* are those which are applicable to current edge and maintain properness of the edge colouring. *Feasible colours* are those which are available to the edge, as well as does not create any bichromatic cycle.

As we have seen in work of Skulrattanakulchai on acyclic edge colouring of subcubic graphs [11], I would like to mention a few points. His work is limited to graphs with maximum degree 3 only. The colour bound given by him is 5, i.e. he used 5 colours to acyclically colour edges of subcubic graph, where every vertex was associated with at most 3 edges. Further, here we are dealing with any Δ-regular subgraph of any dimensional hypercube, with at most $\Delta + 1$ colours.
Also, I would like to draw attention to complexity of structure. As we can see, Figure 3.1 shows the sample graph from Skulrattanakulchai's paper, when we need to colour the edge e, we consider neighbour edges which are a, b, c, d. If all edges are of different colour, we still have one more colour available as well as feasible for current edge e. If 2 of these neighbours are having same colour or 2 pair of opposite neighbours (for example, a and c or a and d, but not a and b) are having same colours, then we just need to check how many out of their two neighbours

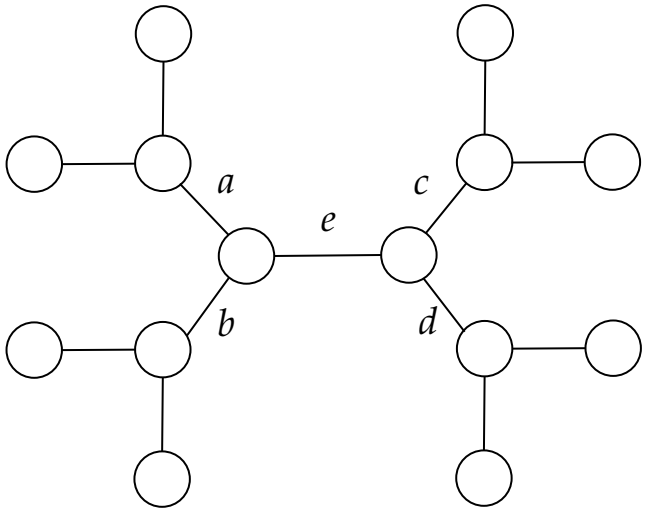

Figure 3.1: subcubic graph with $\Delta = 3$

are same and correspondingly need to change one of the neighbour's colour and we are done. Here, there are at most 4 possible cycles to take care of while colouring, and in worst case changing at most 2 neighbours' colours gives you feasible colour for current edge e. To cut a long story short, we can see the low complexity level of the structure and high availability of colours with respect to Δ, because in this case we can also say that χ' is $2\Delta - 1$, rather than $\Delta + 2$. Although both are same but $\Delta + 2$ sounds lower than $2\Delta - 1$.

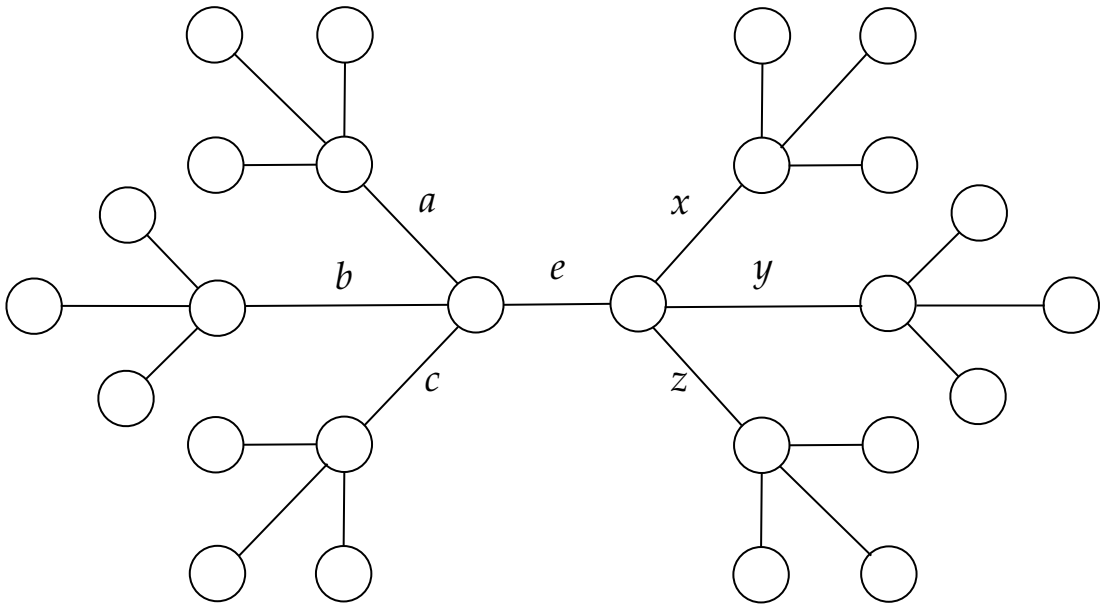

Figure 3.2: arbitrary subgraph of any dimensional hypercube with $\Delta = 4$

Now consider our structure shown in Figure 3.2, for simplicity and sake of ar-

gument, just for $\Delta = 4$ and colour bound of $\chi' = 5$. As one can see, the high complexity of structure and with respect to that, low availability of colours. Each side of edge e gets 3 colours out of 5, and their respective neighbours also get 3 colours out of remaining 4 colours locally. Here we need to take care of 9 cycles in worst case. Now coming to concern of changing neighbour, assume we are about to change colour of an edge a. Since, a also has 3 edges at both sides, making total 6 neighbours, but one of the neighbours is uncoloured (edge e), hence we have 5 other neighbour edges present in it. There is a positive probability that all 5 colours have been used, and we can not change a. Same thing can happen with all neighbours of e, if they are coloured in a certain way. Even if there are less than 5 colours used, then also we still need to take care of cycle not being generated on edge a, while recolouring it. If we take $\Delta = 5$, then total cycles to be taken care of in worst case are 16, and we have just 6 colours to fix. So for each Δ, the number of cycles in worst case goes till $(\Delta - 1)^2$ and total colours are just $\Delta + 1$. To cut a long story short here, if we are not able to change any of the a, b, c, x, y, z in this case, we need to consider changing their neighbours.

All this above discussion is still limited to $\Delta = 4$ and 5. If we go for higher value of $\Delta = 6, 7, 8...$, the structure becomes highly complex and yet we need to colour the edges using only $\Delta + 1$ colours. Again, although it is difficult, but it is possible at some cost of time complexity.

Till now, we have already given some brief introduction on graph theory, vertex and edge colouring, problem statement along with motivation and workflow. We now further demonstrate those approaches which did not give 100% results due to whatever reason (in some cases we have mentioned), which were tried for acyclic edge colouring, algorithms and their outcomes. So if someone is referring this thesis in future, they get an idea which approaches lacks where, and which approaches work in what specific way. Like Thomas Alva Edison said, "I have not failed. I've just found 10,000 ways that won't work."

Approaches covered in this chapter are those, which did not give satisfactory results. Approach 1 to 5 are based on arbitrary subgraphs of hypercube and approach 6, 7 are based on $(d-1)$-regular subgraphs of d-dimensional hypercube.

3.1 Approach-1

This approach was specific for subgraph G, with $\Delta(G) = 3$ of any dimensional hypercube. The approach was somewhat extended from Skulrattanakulchai's paper on acyclic colourings of subcubic graphs [11] to some extent. The idea in the

paper, colours the edges acyclically with restriction of $\Delta(G) = 3$ and $a'(G) = 5$. We did the work with some more careful computations, to achieve $a'(G) = 4$ which is less than $a'(G) = 5$, as we always targeted $a'(G)$ to be $\Delta + 1$.
In the graph with $\Delta = 3$, any edge can have at most 4 neighbour edges, 2-2 on both endpoints. While colouring an edge e, consider several cases,
Case 1: If e has less than 4 neighbours distinctly coloured, nevertheless e can still have 4^{th} colour available as well as feasible to it.
Case 2: If e has less than 4 coloured neighbours, where 2 opposite side neighbours have same colour, those commonly coloured edges can create at most 2 cycles. Hence, e can still have at least 1 colour available as well as feasible to it.
Case 3: If all the neighbours of e are coloured, there are three possibilities:

1. All 4 edges are having different colours, so no colour available to e.
2. Only 2 opposite neighbours of the 4 neighbours having same colour, and those can cause at most 2 cycles, again making no colour available to e.
3. Two pairs of opposite neighbours having same colour, hence each pair causing at most 2 cycles and making no colour available to e.

For case 3 consider further explanation.

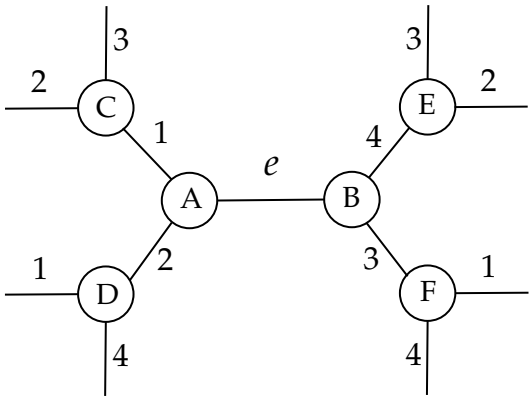

Figure 3.3: Sample instance subgraph consisting edge e, $N(e)$ and $N(N(e))$ for approach-1

In Figure 3.3, the edge AB, represented as e, is yet to be coloured. Coloured edges till now, do not cause any bichromatic cycle. Now, as one can see, there is no colour available for e because all 4 colours have been used on $N(e)$. So in this case we have to recolour at least one of the $N(e)$, in such a way that e gets some colour, available as well as feasible, to be assigned to it. The edge AC can be recoloured with colour 4, but note that it is creating potential 2, 4-dangerous cycle, hence this

edge can not be recoloured. Similarly, the edge AD can be recoloured with colour 3, but again doing so will create potential $1,3$-dangerous cycle, hence this edge also can not be recoloured. The edges BE and BF can be recoloured with 1 and 2 respectively, and that also creates potential $1,3$-dangerous and $2,4$-dangerous cycles respectively. So, in such cases, recolouring of $N(N(e))$ also has to be considered, and at that level the difficulty level rises due to increasing possibilities and low availability of colours.
Doing this way the results were 90%, i.e. 90% of the times, the graph got coloured with just 4 colours, but for rest of the graphs, colouring with only 4 colours was not possible as explained above. Hence, this approach was dropped, although we have worked on one approach, which does recolouring layer by layer of neighbours, which is included later in the thesis.

3.2 Approach-2

This approach was specific for subgraphs G of hypercube, with $\Delta(G) = 3$. The approach starts colouring by selecting an edge, which has four uncoloured neighbour edges. If no such edge is available then any arbitrary uncoloured edge is selected. When there is any uncoloured edge with all neighbours uncoloured, we have the priority to set the colour to minimize the chances of bichromatic cycle. Hence, the edge will be coloured with colour 4 and both side neighbours will be coloured with 1, 2 on one side and 2, 3 on other side, setting based on availability.

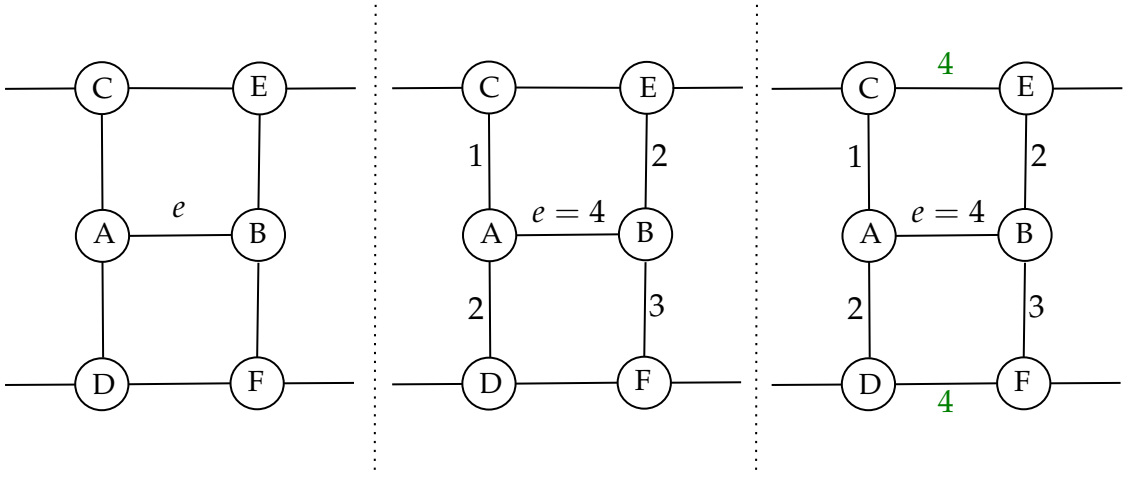

Figure 3.4: Sample instance subgraph for approach-2. First image shows the current edge e to be coloured. Second image shows the initial colouring. Third image shows how colour 4 can be assigned to CE and DF, which are parallel edges to e.

This ensures now, if there is any bichromatic cycle then it would be of colours 2 and 4. Moreover, if there is any edge which is parallel (in this context, an edge is said to be parallel to any other edge, if they both are part of cycle of length 4 and at distance 1 from each other) to the edge with colour 4, it would also get colour 4 as a first priority, if it is not creating bichromatic cycle. Since, as explained above, if such edge is coloured with colour 4 then there is a probability of $1/4$ that it will create bichromatic cycle of colours 2 and 4, and we check for bichromatic cycle before assignment of the colour. See Figure 3.4 and Figure 3.5.

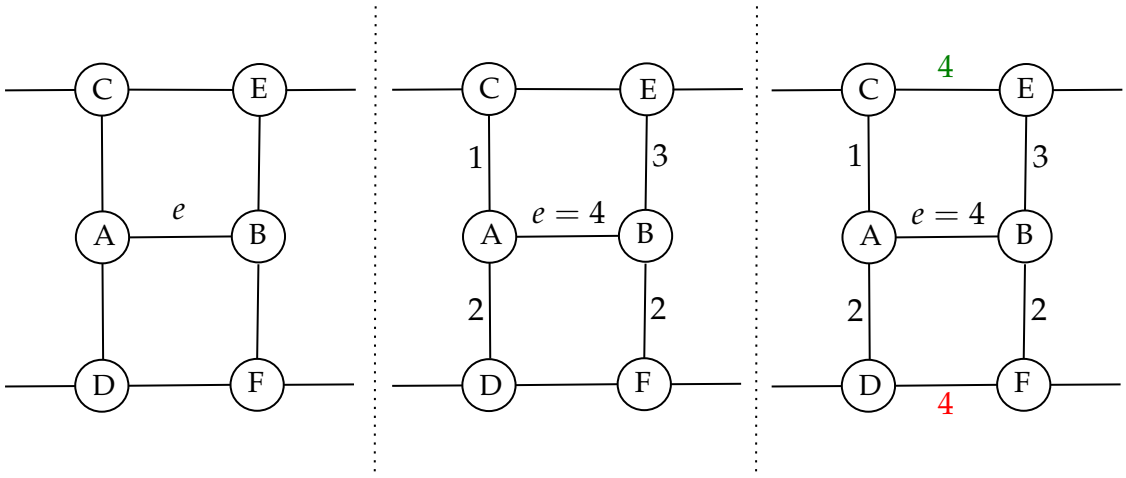

Figure 3.5: Sample instance subgraph for approach-2. First image shows the current edge e to be coloured. Second image shows the initial colouring. Third image shows how colour 4 can be assigned to CE but not DF, which are parallel edges to e.

After first initialization, while there exists any vertex with one coloured and two uncoloured edges incident, we assign the colours to both uncoloured edges based on number of options available to them, i.e. edge with less option will get to choose the colour first. This way, we go vertex by vertex and colour the uncoloured edges. Note that, any sort of recolouring of the edge is not done in this approach. And finally, all the edges which were left uncoloured, are coloured with any colour available to it, if possible. The basic algorithm along with result is given below:

Algorithm 1: Approach-2

```
if uncoloured edge with 4 uncoloured neighbours then
    assign colour 1 and 2 to neighbours on one side;
    assign colour 2 and 3 to neighbours on other side;
else
    assign colour 1 to any uncoloured edge;
while vertex with 1 coloured and 2 uncoloured neighbours do
    assign colour from available colours to the edge which has less options
     compared to second edge;
    assign colour from available colours to the second edge;
while uncoloured edge do
    assign colour from available colours to the uncoloured edge;
```

Result: Worked around 70% of the times; rest caused uncoloured edges

3.3 Approach-3

This approach was based on idea of the paper on optimal acyclic edge colouring of grid like graphs by Muthu, Narayanan and Subramanian [9]. To begin with, we applied the idea for only subgraphs of 4-dimensional hypercubes with $\Delta(G) = 3$. We used the complete 3-dimensional hypercube, coloured using 4 colours, as a **standard hypercube**. Now, consider a subgraph with $\Delta(G) = 3$ of 4-dimensional hypercube. We know that 4-dimensional hypercube will have two sub-hypercubes of dimension 3 and cross edges between them. But since we have restriction of $\Delta(G) = 3$ currently, there will be some edges missing in both sub-hypercubes as well as cross edges. First we coloured the edges, which were available in first sub-hypercube, by looking for the exact same edge in our standard hypercube. For colouring the edges of second sub-hypercube we used non-fixing permutation, for simplicity using this formula,

$$colour = (edge\_colour\_in\_standard\_hypercube \ \% \ total\_colours) + 1 \tag{3.1}$$

For cross edges, we first try to find all feasible colours. If there is no feasible colour available, then we need to change the colour of neighbour edges. For that, we first try to see, if edge in neighbourhood is changeable or not. If not, then

try to change their neighbours' colours. This whole process checks for alternative colours for at most 12 edges (because of $\Delta(G) = 3$). We call this whole process as ***try_to_change_neighbours***. After changing neighbour, we again check if there is any colour available for our current edge, then apply it.

Note that, we also tried to give the same colouring to both sub-hypercubes using standard hypercube directly. This ensures that, there will be always at least one colour available for the current cross edge. But results using this way are also not very promising.

Let's see, what are the drawbacks of this approach. Colouring both of the sub-hypercubes, is straightforward and cause no cycles yet. When colouring the cross edges, each cross edge can have at least zero (if non fixing permutation used for second sub-hypercube) or one (if both sub-hypercubes coloured using standard hypercube directly) and at most two colours available to it, but none of them guarantees the feasibility. Hence, if no feasible colour is available to the edge, then we have to keep looking for alternate colour option till at most $N(N(e))$. At the end, if no feasible colour available, then we assign (if) any of the available colours to the current edge. Note that, initially all cross edges did have at least one colour available to it, but since we recoloured the neighbours of the edge, the available colour may not be available anymore.

Computations and simulations were done for this approach, to get the best results possible. The idea of simple algorithm along with results is given below:

Algorithm 2: Approach-3

```
1  initialization of standard hypercube;
2  for all edges which are not cross edges do
3      if edge is from first hypercube then
4          edge colour = edge colour in standard hypercube;
5      else
6          edge colour = (edge colour in standard hypercube % total colours)
               + 1;
7  for all edges which are cross edges do
8      if feasible colour available for the edge then
9          edge colour = first colour from feasible colours;
10     else
11         try_to_change_neighbours(edge);
12         if any colour available for edge then
13             edge colour = first colour from available colours;
```

Result:

- without cycle: 90.32 %
- with cycle: 8.60 %
- with uncoloured edge/s: 1.08 %

3.4 Approach-4

This approach was straightforward. All the nodes were traversed one by one, starting initially with any arbitrary node and checked, if there is any uncoloured edge incident on the current node. The edges were coloured based on number of options of available colours it has. An edge with having lesser option was coloured first. First priority of colour was feasible colour. If there is no feasible colour available, then *try_to_change_neighbours* method is called and again checked, if there is any feasible colour available. If no feasible colour is available, then if any colour is available for the edge, it is assigned to the edge. If during this process any edge was not able to be coloured, then at the end all these uncoloured edges will be tried once more via the same process.

This approach was intended to work for subgraph with $\Delta(G) = 3$ of any dimensional hypercube. Results of this approach are quite similar to the results of the previous approach. The algorithm for above described approach is given below:

Algorithm 3: Approach-4

```
for all nodes with uncoloured edge do
    edges = sorted edges incident on current vertex based on number of
     available colours;
    for each edge in list of edges do
        if edge has feasible colours then
            edge colour = first colour from feasible colours;
        else
            try_to_change_neighbours(edge);
            if feasible colour available for edge then
                edge colour = first colour from feasible colours;
                continue;
            if any colour available for the edge then
                edge colour = first colour from available colours;
for all edges which are left uncoloured do
    if feasible colour available for the edge then
        edge colour = first colour from feasible colours;
    else
        try_to_change_neighbours(edge);
        if any colour available for main edge then
            edge colour = first colour from available colours;
```

Result:

- without cycle: 89.61 %
- with cycle: 9.62 %
- with uncoloured edge/s: 0.77 %

3.5 Approach-5

This approach is extended from approach-3, based on idea of paper on optimal acyclic edge colouring of grid like graphs [9]. But unlike the approach-3, it works for subgraph of any dimensional hypercube with $\Delta(G) = 3$. Any d-dimensional hypercube can be visualised as collection of multiple 3-dimensional hypercubes with cross edges, to complete the d-dimensional hypercube. This construction is straightforward to understand. 3-dimensional hypercube will occur two times in 4-dimensional hypercube and four times in 5-dimensional hypercube and so on. Each occurrence of the hypercube will have its own dimensional number. Now, first we colour all the edges, which are part of multi-times occurring 3-dimensional hypercubes (denoted as dimensional edges) and then we go for cross edges (denoted as inter-dimensional edges). The desired colour for dimensional edges are counted using the following equation:

$$\begin{aligned} desired\_colour = {} & ((edge\_colour\_in\_standard\_hypercube + dimension\_number) \\ & \% \ total\_colours) + 1 \end{aligned} \tag{3.2}$$

If this desired colour is also feasible colour for the current edge, then it is assigned to the dimensional edge. For all remaining uncoloured dimensional edges and inter-dimensional edges, we first try to find all feasible colours. If there is no feasible colour available, then we use try_to_change_neighbours method and after that again check for feasible colours. This time, if there is any feasible colour available, then we assign it to the current edge, else leave it uncoloured.
The algorithm for this approach is given below:

Algorithm 4: Approach-5

```
initialization of standard hypercube;
for all edges which are not cross edge do
    desired_colour = ((edge_colour_in_standard_hypercube +
     dimension_number) % total_colours) + 1;
    if desired_colour is feasible for the edge then
        edge colour = desired_colour;
for all edges which are left uncoloured do
    if feasible colour available for the edge then
        edge colour = first colour from feasible colours;
    else
        try_to_change_neighbours(edge);
        if feasible colour available for edge then
            edge colour = first colour from feasible colours;
```

Result: Worked around 85-90 % of times

3.6 Approach-6

The graph structures used from this approach are $(d-1)$-regular subgraphs of d-dimensional hypercube. This approach was based on proper edge colouring of the graph. The graph edges were coloured properly first and then tried to convert into acyclic colouring. The idea to convert into acyclic colouring is based on concept on matching. A properly edge coloured graph can have at most $n/2$ edges of any colour. Since, the graph is regular here, all the edges will be properly coloured with $d-1$ colours distributed equally. Hence, all the vertices will have each of the $d-1$ coloured edges incident on it. The very useful details for such graph is given in the Table 3.1. Some of these details are very well known, whereas some of them are presented by our analysis. If a graph is properly coloured, of course it contains bichromatic cycles. To break all those bichromatic cycles we need to recolour some of the edges using our last $(d+1)^{th}$ colour, and selection of such edges is done by matching. Since each edge participates in total $\Delta - 1$ bichromatic cycles, we need to select some edges for our matching, such that those edges will break all the cycles and the matching itself would not create a bichromatic cycle, with any colour

from $\{1, 2, ..., \Delta\}$. We tried to find a proper scheme for finding such matching, but did not get success 100% of the times. So, if some work is done for finding such matching, it could be very useful for acyclic edge colouring of this graph category.

Hypercube Dimension d	Regularity $\Delta = d$ - 1	$\|V(G)\|$ 2^d	$\|E(G)\|$ $\Delta \cdot 2^{d-1}$	Cycles per node $= \binom{\Delta}{2}$	Cycles per edge$= \Delta$-1	Max cycles
4	3	16	24	3	2	10
5	4	32	64	6	3	45
6	5	64	160	10	4	156
7	6	128	384	15	5	475
8	7	256	896	21	6	1338

Table 3.1: Useful details for $(d-1)$-regular subgraph of d-dimensional hypercube

3.7 Approach-7

In this approach, we have worked on restricted case of $(d-1)$-regular subgraph G, of d-dimensional hypercube. The idea was extended from [9]. We colour $(d-1)$-dimensional standard hypercube, according to this paper, and colour all the dimensional edges of G as per standard hypercube. Now, the inter-dimensional edges (cross edges) will be coloured in following manner. If cross edge is connected to other cross edge with distance of 1, both edges needs to be coloured differently, otherwise it will cause cycle of size 4. Then, we assign the last colour d to the cross edge wherever possible (taking care of previous condition) and one colour from 1, 2, ..., d-1, alternatively as per done in [9].
This approach has a limitation of ignoring the need to colour some pairs of edges that are far apart with different colours. Because of that, there will be bichromatic cycles at the end, and that also very long cycles. We initially worked on 3-regular subgraphs of 4-dimensional hypercube. The colouring was acyclic in every case, except when there were all 8 cross edges present in G. So, this is trivial case and without solving that, the colouring algorithm is still incomplete.

CHAPTER 4
Acyclic Edge Colouring of Arbitrary Subgraph of Hypercubes

The approach we are talking about here, works on arbitrary subgraphs of any dimensional hypercube. If dimension of the hypercube H is d, then it's arbitrary subgraph G can also have $\Delta(G)$ ranging from 1 to d.
Note that, $\Delta(G) = 1$ is a graph which is forest of only single edges, where none of the edge will have any adjacent edge and no longer single component. Acyclic edge colouring will be done using just one colour, hence $a'(G)$ will be 1.
$\Delta(G) = 2$ is the case where the graph can be consisting of one or more components. If the graph is single component then it can be either single cycle or a single path. Otherwise in case of multiple components, all components will be again either single cycle or single path. All such cases are very simple and acyclic edge colouring is straightforward for that. And acyclic edge colouring can be achieved by either $a'(G) = 3$ in case if there exists any single cycle or $a'(G) = 2$ in case of all components are path/s.
Similarly, $\Delta(G) = d$ is also trivial case because that means it is a hypercube of dimension d, but with some edges missing in such a way that maximum degree remains d. Hence acyclic edge colouring will take no efforts and directly can be done using acyclically edge coloured standard hypercube, as discussed already before. And yet, $a'(G)$ will be $d + 1$.
All the case above are not considered in this method, hence the goal of this approach is to achieve acyclic edge colouring of subgraphs with $\Delta(G) = \{3, 4, ..., \text{d-1}\}$ of d-dimensional hypercubes, with at most $\Delta(G) + 1$ colours.
If talking about state of art results, Basavaraju and Chandran shown in [4] that, graph with $\Delta(G) \leq 4$, with restriction of total number of edges $m \leq 2n - 1$, can be acyclic edge coloured using $a'(G) \leq \Delta(G) + 2$. If we go for higher values of $\Delta(G)$, the value of $a'(G)$ increases in that way. Whereas, our approach gives constant value of $a'(G) = \Delta(G) + 1$.

Conjecture 1. *Any subgraph G of hypercube can be coloured using, at most* $\Delta(G) + 1$ *colours, i.e.* $a'(G) \leq \Delta(G) + 1$

We are going to give various approaches, from now on, to acyclically edge colour the subgraphs of hypercubes, which satisfies the Conjecture 1.
We already discussed the complexity level of the colouring due to higher values of Δ and colour bound limited to $\Delta + 1$. We also shown the issue when attempting to recolour edges up to $N(e)$ and $N(N(e))$ for current edge e. This approach is not limited to such boundary. The flow of the approach with other important details is explained further.
All the vertices are equally likely to get degree between 1 and $d - 1$, while maintaining all the basic properties of any simple and finite graph in the graph theory. Also, graphs discussed here are always a single component. To get started, we try to find an edge which has one endpoint of degree 1, if no such edge is there we try to find an edge which has both endpoints of degree 2, else we go with any arbitrary edge.
Assign colour 1 to this initial edge. Now, we find all uncoloured neighbours of the current edge, sort them according to the number of feasible colours available to each neighbour edge and assign the colours in the same sorted order (so that the situation does not occur in which one edge which has more than 1 colour available as feasible colours is coloured first and due this another neighbour which had only 1 feasible colour available previously, has no colour available now). Then again pick one edge from these coloured neighbour edges as a current edge and look forward to colour their neighbours.
The issue arises when not a single feasible colour is available for the edge. Here consideration of neighbours comes to the picture. Initially we consider to recolour only the neighbour edges $N(e)$ of the current edge and call it *level 1*. Similarly, level 2 will be consisting the edges till $N(N(e))$, and so on. We repeatedly apply try_to_change_neighbours method, until we get a feasible colour for the current edge e, each time traversing one more level of neighbours.
Due to this traversal pattern, there exists some cases in which edges are left unvisited. Hence at the end of the colouring, we again check if there exists any such unvisited and uncoloured edges, and we again apply the same algorithm by selecting any one of these edges as a current edge e, until all edges are coloured.
The brief algorithm demonstrating this approach is given below:

Algorithm 5: Acyclic Edge Colouring of Arbitrary Subgraph of Hypercubes

```
initialization of first edge with colour 1 and as a current edge;
while current edge has uncoloured neighbours do
    uncoloured neighbours = sorted uncoloured neighbour edges based
     on number of feasible colours available to each edge;
    for edge in uncoloured neighbours do
        if feasible colour available for the edge then
            edge colour = first colour from feasible colours;
        else
            level = 1;
            while current edge is uncoloured do
                try_to_change_neighbours(edge, level);
                if feasible colour available for edge then
                    edge colour = first colour from feasible colours;
                level += 1;
for all edges which are left uncoloured do
    current edge = uncoloured edge;
    GOTO: step-2;
```

Result: Worked around 100 % of times for subgraph with any Δ of any dimensional hypercube

Here, we have not presented any mathematical proof for the approach, but talking about simulations, this approach works all the time and also achieved $a'(G) = \Delta(G) + 1$, for subgraph with any Δ, of hypercube with any dimension. The inner while loop (try_to_change_neighbours, level by level) runs mostly for 1 or 2 levels. Since, vertices in this range are equally likely of having degree between 1 and $d - 1$, we get our colour for the current edge e.

Note that, this approach works for arbitrary subgraphs of hypercube as discussed above, but not for regular subgraphs of the hypercube. The method for acyclic edge colouring of regular subgraphs of hypercube is discussed in further chapters.

CHAPTER 5
Acyclic Edge Colouring of Regular Subgraph of Hypercubes

In this chapter we are going to give some algorithms for several cases of acyclic edge colouring of regular subgraph of hypercubes. Section 5.1 and Section 5.2 covers the acyclic edge colouring of $(d-1)$-regular subgraph of d-dimensional hypercube with 2 cross edges and 4 cross edges respectively, where we cover the explanation of the approaches, with the proof of correctness.

5.1 Acyclic edge colouring of $(d-1)$-regular subgraph of d-dimensional hypercube with 2 cross edges

By now, we are familiar with construction of the hypercube. So, we know how d-dimensional hypercube is made using two $(d-1)$-dimensional sub-hypercubes and introducing new 2^{d-1} edges, each connecting the same vertex in both copies of sub-hypercubes. This new edges are called cross edges.
See Figure 5.1 for example, it shows the basic example of 3-regular subgraph of 4-dimensional hypercube with 2 cross edges. It is made of two 3-dimensional subgraphs and two cross edges, which are presented as dashed lines. If you see, because we introduced two cross edges, we also removed two edges, one from each copy (edge CG and edge $C'G'$).
When we introduce one cross edge, we need to remove one non-cross edge, incident on both endpoints of the cross edge, from both sub-hypercubes, as we did with the edge CG and $C'G'$ when introducing the cross edge CC'. This is done to maintain the regularity of the graph. Now, because we have removed one non-cross edge from each of the both sub-hypercubes, it would decrease the degree of the vertex which is another endpoint of removed non-cross edge, in this case the vertices G and G'. Hence, we need to introduce one more cross edge there also, which we did by introducing cross edge GG'. One may notice by now that cross

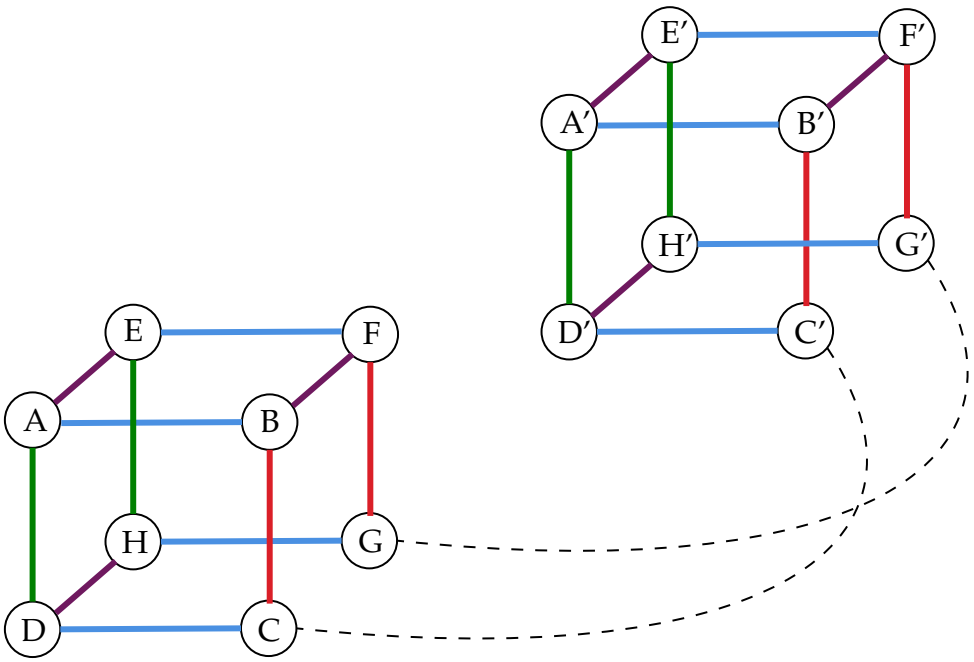

Figure 5.1: Example of 3-regular subgraph of 4-dimensional hypercube with 2 cross edges

edges are always introduced in pair of two. Consequently, non-cross edges are removed in pair of two also. If this pair of non-cross edges differs in both sub-hypercubes then more number (to be specific at least six) of cross-edges needs to be introduced to maintain regularity. Because of such constraint and structure, in case of 3-regular subgraph of 4-dimensional hypercube, there could be only 12 such configurations possible.

Theorem 1. *If we have* $(d-1)$*-regular subgraph of* d*-dimensional hypercube with 2 cross edges, the subgraph can be acyclic edge coloured with* d *colours.*

Proof. We have two $(d-1)$-regular subgraphs which are ultimately two $(d-1)$-dimensional hypercube with one edge missing in each copy.
We get our *standard hypercube* as acyclically edge coloured $(d-1)$-dimensional hypercube with d colours, as per [9].
Now, we can acyclically edge colour both copies using *standard hypercube*, by directly mapping the edge colour and ignoring the missing edge.
Note that, missing edge in both sub-hypercubes will have same colour in the *standard hypercube*. So, we use that colour to colour our two cross edges. This ensures that the edge colouring will be acyclic. Let's see how.
Assume, missing edge in first sub-hypercube is XY between vertices and X and

Y. Similarly in second sub-hypercube, the missing edge will be $X'Y'$ between vertices and X' and Y'. The missing edges' colour is α in *standard hypercube*.
Because these two non-cross edges are missing, our cross edges will be XX' and YY'. Now, consider the Figure 5.2, where first image shows two $d-1$-dimensional hypercubes and second image shows a single $d-1$-regular subgraph with two cross edges.

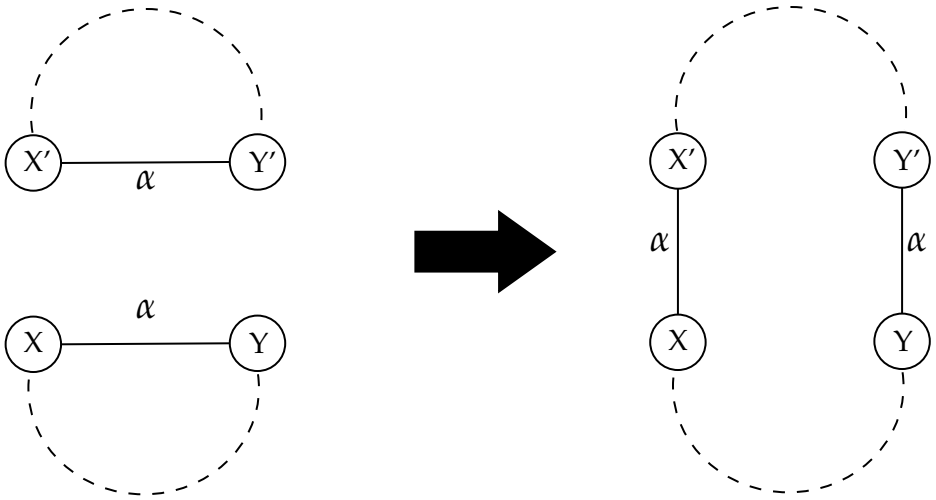

Figure 5.2: Example of cyclic edge colouring of $(d-1)$-regular subgraph of d-dimensional hypercube with 2 cross edges

If the edge XY in the first sub-hypercube is not part of any bichromatic cycle, then every possible path between vertices X and Y, after removing the edge XY, must contain at least two different colours other than α.
So, if any bichromatic cycle exists in the graph, then it must pass through both cross edges and both sub-hypercubes, i.e., cycle will be starting from first cross edge XX', then further going through some path between X' and Y' in second sub-hypercube, then going through second cross edge YY', then going through some path between X and Y in first sub-hypercube, and coming into first cross edge XX', hence creating a complete cycle.
However, both sub-hypercubes contains all the paths with at least two different colours than α, hence making the discussed cycle of at least three colours. Hence, the cycle will not be bichromatic and the edge colouring will be acyclic as intended.

□

5.2 Acyclic edge colouring of $(d-1)$-regular subgraph of d-dimensional hypercube with 4 cross edges

In previous section, we saw that how the $(d-1)$-regular subgraph of d-dimensional hypercube with 2 cross edges is acyclic edge coloured. In this section, we will increase the number of cross edges to 4. As we discussed, the cross edges are introduced in pair of two's, and how only 2 cross edges make two identical sub-hypercube of $d-1$-dimension with one non-cross edge missing in each. The similar case will happen with 4 cross edges. As we explained, if missing non-cross edges in both copy is not the same then it will cause at least 6 cross edges, to balance the regularity. Hence the graph with 4 cross edges will also make two identical sub-hypercube of $d-1$-dimension with two non-cross edge missing in each.

Theorem 2. *If we have $(d-1)$-regular subgraph of d-dimensional hypercube with 4 cross edges, the subgraph can be acyclic edge coloured with d colours.*

Proof. While colouring cross edges, we will take care of these two:

1. **One distance danger:** If two cross edges are connected by one non-cross edge between them in each sub-hypercube.

2. **Bichromatic path danger:** If two cross edges are connected with any bichromatic path (odd or even length) between them in each sub-hypercube.

We have two $(d-1)$-regular subgraphs which are ultimately two $(d-1)$-dimensional hypercube with two edge missing in each copy.
We get our *standard hypercube* as acyclically edge coloured $(d-1)$-dimensional hypercube with d colours, as per [9].
Note that, because of 4 cross edges, there are only three scenario possible:

1. Two cross edges are connected with other two cross edges with one common non-cross edge between them in each sub-hypercube.

2. One cross edge is connected with other cross edge with one common non-cross edge between them in each sub-hypercube.

3. No cross edge is in one distance danger with any other cross edge in each sub-hypercube.

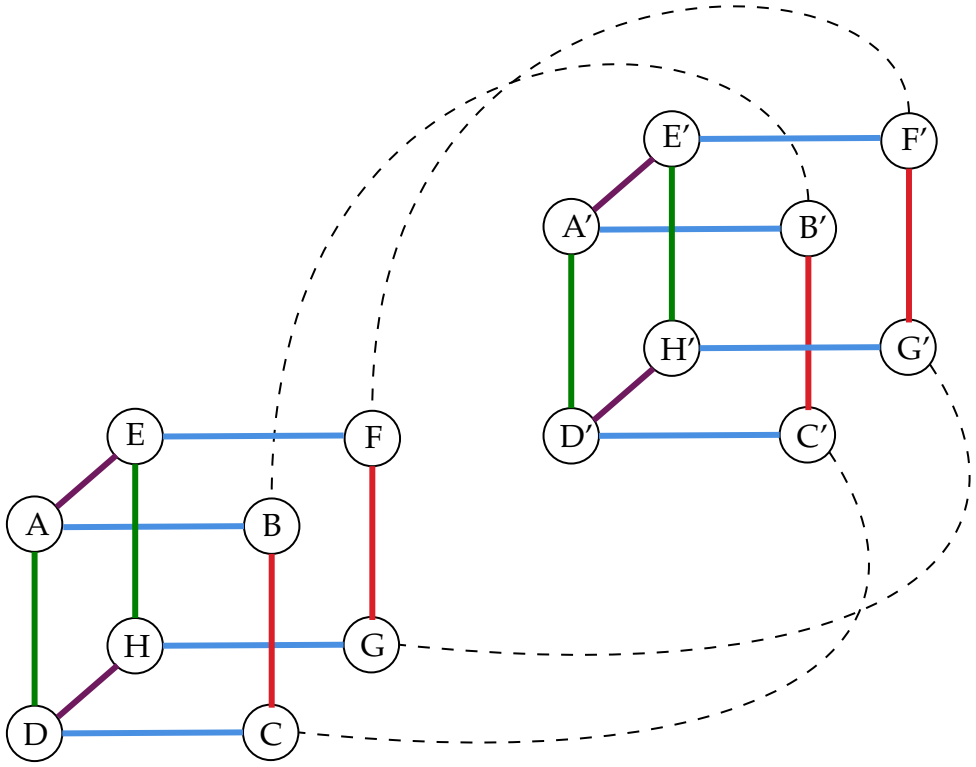

Figure 5.3: Example of cyclic edge colouring of $(d-1)$-regular subgraph of d-dimensional hypercube with 4 cross edges

See the Figure 5.3, the graph contains 4 cross edges: BB', CC', FF' and GG'. The cross edges BB' and CC' are connected with one common non-cross edge between them, i.e. one distance danger. Also they are at bichromatic path danger, by the path $B - A - D - C$ and $B' - A' - D' - C'$, in each sub-hypercube. The same thing goes with cross edges FF' and GG'. Note that, this is trivial case, where both issues of one distance danger and bichromatic path danger exists for single pair of cross edges, otherwise they also occurs individually with positive probability. Throughout this section, refer this example for explanation of any situation or scenario.

Let's first see issue of one distance danger. When two cross edges are connected with one common non-cross edge between them, and because both sub-hypercubes will also be identical in terms of structure and colouring, this will create a cycle of size 4. And if both edges are given the same colour, it will create bichromatic cycle. Hence, both such cross edges needs to be coloured differently.

Now, when two cross edges contains any bichromatic path between them, there are two possibilities of path, either even length or odd length. If such bichro-

matic danger path is of even length and we colour both cross edges with same colour from one of the colours used in bichromatic danger path, then it will create a bichromatic cycle. Similarly, if bichromatic danger path is of odd length and we colour both cross edges with both of the colours used in bichromatic danger path, then also it would create a bichromatic cycle. Hence this condition also needs to be taken care of.

If we take care of above mentioned two issues, then it will ensure acyclic edge colouring.

Note that, each cross edge here will have exactly two colours available to it, because the endpoint of the cross edge will have degree $\Delta = d - 1$ so exactly $d - 2$ would have been already used, so in remaining colours, one will the colour of removed non-cross edge and second will be due to uncoloured cross edge itself.

Solving the first issue is easy and straightforward, since both cross edges will have two colours available to it, even if in worst case both are same, then also both can be coloured using different colours.

To solve second problem, let's assume that bichromatic danger path is of colours α and β colours. In worst case both our cross edges are having α and β as available colours, which is impossible, since each cross edge is connected to an edge coloured with α or β, exactly one of them will be present at endpoint of the cross edge. Hence there exists at least one different colour from α and β, which is available to each cross edge. So assigning both cross edges a colour will be in such a way that, at most one of the cross edge gets the colour from α and β, while other cross edge will use one colour available to it other than α and β. So, the bichromatic danger issue is solved in this way.

Tricky scenario is where, both problems occurs, i.e. two cross edges are connected with one common non-cross edge, as well as bichromatic danger path. So assume, in worst case, both cross edges have α and γ as available colours, along with (α, β)- bichromatic danger path between them. To colour both cross edges, we have three options: (α, α), (α, γ) and (γ, γ). Since, to avoid the one distance danger problem, we can not consider (α, α) and (γ, γ). Hence, assignment of (α, γ) will solve the problem.

This way all the cases are covered and hence the edge colouring will be acyclic.

□

CHAPTER 6

Acyclic Edge Colouring of 3-Regular Subgraph of Hypercubes

In this chapter we are going to cover acyclic edge colouring of 3-regular subgraph of any dimensional hypercube, where the minimum dimension of hypercube will be 4. We propose two approaches for the same, with their explanation as well as algorithm. 3-regular graphs will have all the vertices of degree 3.

6.1 Acyclic edge colouring of 3-regular subgraph of hypercubes using Kempe chain

Before going further, let's understand what is the Kempe chain. **Kempe chain** [3] is a maximal two-coloured (α, β) path in the graph. Note that, when we say path, it will have endpoint vertices which will have one or more edges incident on it such that, one of the colours from α and β will be present and the other will be missing. We can use this property to find Kempe chains, for which the current vertex may be endpoint. Tracing for such present and absent colour pair, till we can, will give maximal two coloured alternating path which is the Kempe chain.

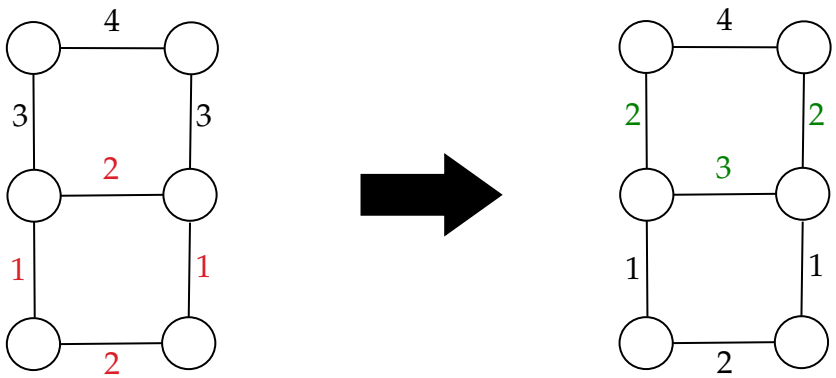

Figure 6.1: Example of Kempe chain, breaking bichromatic cycle

As one can see in the example Figure 6.1, first image shows how the graph contains bichromatic cycle of colours 1 and 2, marked by red. And the Second image shows how the Kempe chain of colours 2 and 3 swapped, marked by green.
Kempe chains are very useful in terms of graph colouring, and over the time it has been used for solving or proving well-known problems, such as four colour theorem. If used carefully, it can also be used to break bichromatic cycle in the graph.
Let's focus on the approach now.
As we discussed earlier, bipartite graphs can be properly edge coloured using Δ colours. Since, hypercube and its subgraphs are also bipartite, the same property of proper edge colouring will hold.
So, we proper edge colour our 3-regular subgraph G of any d-dimensional hypercube with at most 3 colours. The colouring will contain many bichromatic cycles since, every colour will be present at each node. Proper edge coloured 3-dimensional hypercube contains at most 6 cycles. 3-regular subgraph of 4-dimensional hypercube, which is made up of two 3-dimensional sub-hypercubes and some cross edges between both copies, will have at least 2 cross edges to maintain 3-regularity and to be connected as a single component. So introducing two cross edges will make two non-cross edges to be removed, and consequently 2 cycles to be broken in each copy of sub-hypercube. However, we did introduce two cross edges also, so there will be two bichromatic cycle going through these cross edges, increasing cycle count by two. Hence, 3-regular subgraph of 4-dimensional hypercube will have at most 18 cycles. Following by induction, we can say that, 3-regular subgraph of d-dimensional hypercube will have at most, $2 \times \big($maximum number of cycles in 3-regular subgraph of $(d-1)$-dimensional hypercube $-1\big)$.
Now, we have our graph G, properly edge coloured. All we need to do is, break all the bichromatic cycles using the last 4^{th} colour. This is done in two steps, first will be arbitrary and second will be systematically as per our algorithm. We also maintain a cycle list, to keep track of cycles.
The last colour is assigned arbitrarily to an edge e based on two conditions:

1. e should not have any other edge coloured with last colour, incident on it. (This condition maintains properness of colouring)

2. e should not be creating any additional bichromatic cycle of colour 4 and any other colour from $\{1,2,3\}$. (This condition maintains acyclicness of recolouring)

Now, arbitrary assignment of last colour is done in following way. For each cycle in proper edge coloured graph, we arbitrarily selects an edge of the cycle until, either the cycle gets broken or all the edges are visited. If any edge is found such that, recolouring it with last colour does not violates our above two conditions, then we recolour it. Now, because we have recoloured an edge, it will at least break one cycle (which is of course current cycle) or at most two cycles (since each edge will be part of two cycles exactly, when proper edge coloured). We check which other unbroken cycle did have this edge as a part of itself. If we found such cycle, which got broken due to this edge, we update our cycle list by removing them from the list (including current cycle).
Of course, by this stage, not all cycles would have been broken, because of our arbitrary selection. But the above step was to fasten the process. Now, we again find all unbroken cycles (if any), and perform the further steps. If no such unbroken cycle is left, then we have achieved the acyclic edge colouring.
Now, for each of the unbroken cycles, and for each edge edge of the current cycle, we do the following until the cycle is broken. We find the Kempe chain in which the current edge e is neither first nor last edge, i.e. trace the Kempe chain from middle, using all colours present at both endpoints of current edge e.

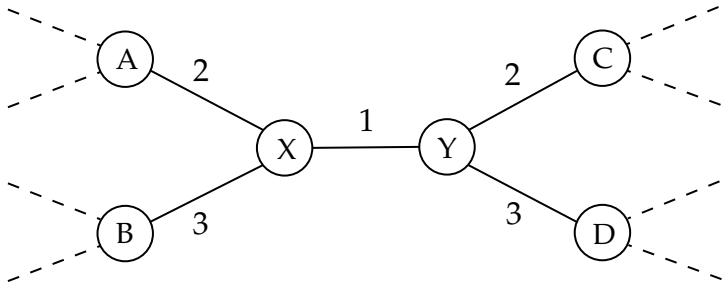

Figure 6.2: Tracing Kempe chain from the middle

For example, in the Figure 6.2, if the current edge is XY, then there will be two sub-traces of the same colour pair, one starting from X and another starting from Y. In this case, we can say that we have two sub-traces for colour pair of 1 and 2, which are $X - A - ...$ and $Y - C - ...$, starting from X and Y respectively.
If any of the two sub-traces, trace back to current edge, i.e. create a bichromatic cycle, we discard such sub-traces. Otherwise, merge them and create a complete Kempe chain, which in this case is $... - A - X - Y - C - ...$.
Now, check if swapping the Kempe chain reduces the number of the cycles compared to the before swap, then apply the swap and break the current cycle, and

update the cycle list based on which other cycles got broken along with current cycle, due to this Kempe chain swap. Repeat this step until there are no more cycles in the graph.
Although, we have not presented any mathematical proof for this approach, the algorithm simulation gave 100% results for acyclic edge colouring of 3-regular graphs of any dimensional hypercube.

6.2 Acyclic edge colouring of 3-regular subgraph of hypercubes in Hopcroft-Karp way

As we already seen how the graph can be properly edge coloured using Δ colours, recently in last Section 6.1. The approach demonstrated in this section is extended version of that approach. If one have noticed, when recolouring the edges to break the cycle in Kempe chain way in Section 6.1, There was positive probability that the last colour 4, might create new bichromatic cycle with any colour from $\{1, 2, 3\}$, if number of cycles after Kempe chain swap, is lesser than the number of cycles before Kempe chain swap. So there was a little overhead to remove such cycles also. Whereas, in this approach, no new cycle will be created using colour 4 and one colour from $\{1, 2, 3\}$, i.e. all the cycles introduced by proper edge colouring will be broken as well as no new cycle will be generated. Consequently, the total running time of the algorithm will be decreased by almost half. Before jumping into the algorithm, let us first see what is Hopcroft-Karp way, as mentioned in the title of the approach.
Hopcroft-Karp algorithm[7] takes bipartite graph as an input and generates maximum cardinality matching. The algorithm repeatedly increases the size of a partial matching, by finding augmenting paths. **Augmenting paths** are sequences of edges of the graph, which alternate between edges in the matching and edges out of the partial matching, and where the initial and final edge are not in the partial matching. Finding an augmenting path allows us to increment the size of the partial matching, by simply toggling the edges of the augmenting path (putting in the partial matching those that were not, and vice versa).
As one can see in the Figure 6.3, we are showing our bipartite graph example, to get it's maximum matching. The first iteration of Hopcroft-Karp algorithm adds the edge which is available first to the current vertex A, so edge AD is added to the matching. Now, vertex B has only one neighbour vertex D connected to it, but that is already taken for matching, so B has no option available. Similarly, vertex C has E connected to it, which is available and edge CE is not taken into matching.

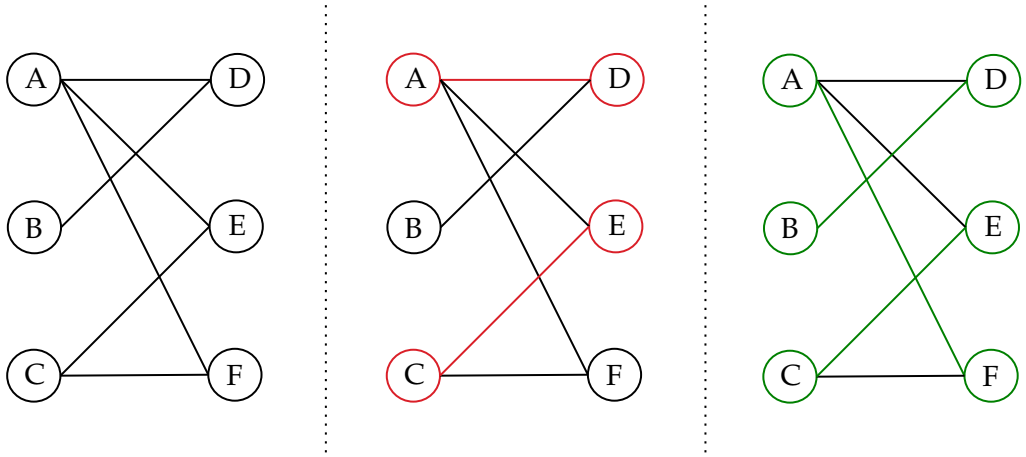

Figure 6.3: Example of Hopcroft-Karp algorithm to find maximum matching. First image shows our sample bipartite graph. Second image shows the intermediate matching after first iteration of algorithm. Third image shows maximum matching.

Hence, edge CF is added into matching.
But, we see that the vertex B and F are not matched to any edge, as shown in second image, matched edges and endpoint vertices of such edges, marked by red colour. So we find augmenting path starting from B and ending at F, which is $B - D - A - F$. Hence, by toggling this augmenting path, we get one more edge into matching set. So, we can see in third image that, all vertices are matched, hence it is maximum as well as perfect matching. We can see matching edges and endpoint vertices of such edges, marked by green colour. The characteristic of this algorithm is, in each iteration, we find one more edge for the matching, until maximum matching is found.
Let's see how this idea is useful acyclic edge colouring of of 3-regular subgraph G of any dimensional hypercubes.
We properly edge colour the G, using 3 colour ($\because \Delta = 3$), and we already know how it is done.
The properly edge coloured graph will of course contains cycles. We already shown the maximum number of cycles bounds, with explanation, in Section 6.1. Note, that we already told that no new bichromatic cycle is going to be introduced in this approach. So, we initially find list of all existing cycles, and sort them according to their length, in ascending order. So that, smaller cycles gets first priority to break themselves, because of lesser option of the edges.
For all cycles, create a Boolean array, indicating whether the cycle is broken or not. Initialize it to *false*. Here, *true* indicates that cycle is broken and *false* indicates that

cycle is not broken yet. So, anytime during the process, we can directly check if cycle is broken or not.
For each cycle, if it is not broken, then arbitrarily choose an edge of cycle, until either all edges are visited or cycle is broken.
The last colour is assigned arbitrarily to an edge e based on two conditions:

1. e should not have any other edge coloured with last colour, incident on it. (This condition maintains properness of colouring)

2. e should not be creating any additional bichromatic cycle of colour $\{4, \alpha\}$, where α is any colour from $\{1, 2, 3\}$. (This condition maintains acyclicness of recolouring)

If current edge, maintains both the above properties, then recolour the current edge with the colour 4.
Since, any edge will be part of exactly 2 cycles (broken or unbroken), find all such cycles and mark them as broken, including the current cycle.
At this stage, we does not guarantee that all the cycles would be broken, since edges were selected arbitrarily per cycle, but our simulation results shows that very high percentage of graphs becomes acyclic by now. The further steps in Hopcroft-Karp way are to ensure that, even if there is any cycle left unbroken, it would be broken after this step. We use the cycle list and Boolean array to get insights of unbroken cycles.
We create a bipartite graph for our understanding, it will be referred as BG. One partite will have all the cycles as vertices, and another partite will have all the edges as vertices. A cycle in first partite will be connected to all it's edges in second partite.
For each unbroken cycle, find all it's edges in BG. Now, we know that if none of the edges of the cycle is recoloured using colour 4, it may be because of only two reasons:

1. An edge is having an edge coloured with colour 4, incident on it.

2. Assigning an edge the colour 4 will make bichromatic cycle.

For each edge of the current unbroken cycle, find all the edges, which are at distance 0 or 1 (distance 0 edges will be edges incident on it) and coloured with colour 4. Call such edges the *blocked edges*.
Each blocker edge can be part of at least one or at most two cycles, and this will be called as *blocker cycle*. Find all such blocker cycles for blocker edges.

For each edge of unbroken cycle, we do following operations one by one. We find all the blocker edges and corresponding blocker cycles. We recolour these blocker edges with any colour from $\{1, 2, 3\}$, whichever is available to it (there will be only one such colour available to it). Hence, because we have removed colour 4 from all blocker edges, we have now colour 4, available for current edge of unbroken cycle.
Note that, because we have removed colour 4 from blocker edges, those edges *can* make blocker cycle to be bichromatic cycle of colours $\{\alpha, \beta\}$, where α and β are any two distinct colours from $\{1, 2, 3\}$, only if blocker cycle has no other edge, coloured with colour 4.
Hence, for blocker cycles, there are only two possibilities:

1. Apart from blocker edge, blocker cycle contains another edge coloured with colour 4.

2. Apart from blocker edge, blocker cycle *does not* contains any other edge coloured with colour 4.

For first case, there is no need to worry, because even after recolouring blocker edge with one of the colours from $\{1, 2, 3\}$, the blocker cycle will not become bichromatic cycle, because it has one edge coloured with colour 4.
But for case 2, we need to assign colour 4 to another edge of blocker cycle, apart from blocker edge. Since, any cycle will of length 4 at least, we will have still 3 other edges available to recolour using colour 4. So, here we again check our two primary condition to assign colour 4 to an edge, and if we find such edge, then blocker edge is also removed, blocker cycle is also taken care of and our unbroken cycle is also broken now.
To understand it in terms of BG, consider an unbroken cycle, from first partite. It will have edges associated with it in second partite. After finding blocker edges, we trace blocker cycle in BG. And as per done in Hopcroft-Karp algorithm to find augmenting path, we also find such path in BG, such that the current unbroken cycle finds all it's edges, then corresponding blocker edges, then blocker cycle and finally another edge available to recolour with colour 4. This way whole path is completed and all cycles get broken, thus achieving acyclic edge colouring.
Note that, although we have done this approach for specifically 3-regular subgraphs of hypercubes, this approach has very good scope for any bipartite graph with $\Delta = 3$, which is broader class of graphs as compared to done here. If one wants to work further on this approach, then they can go in that direction and get some very good results.

CHAPTER 7
Conclusions

In this section, we discuss some remarks about the thesis work we have shown till now. Let us see conclusive arguments and possible directions for future work. The research aimed to get acyclic edge colouring of subgraphs of hypercubes, using minimum number of colours. We have shown how the acyclic edge colouring is complex in structure; the total number of cycles to be taken care of, while recolouring a single edge.

We have shown some of the various approaches, we have tried. The approaches, which were lacking somewhere, are also brought to highlight. We mentioned in some of the approaches that, how the specific issue occurring and how can we possibly overcome them.

We presented the approaches, which always works using at most $\Delta + 1$ colours. We also gave conjecture that acyclic edge colouring the subgraphs of hypercubes can be done using at most $\Delta + 1$ colours.

We, also gave two theorems, which we proved in respective sections. Apart from these, we gave some insightful details which we found by analysis, such as maximum number of cycles in the properly edge coloured subgraph of hypercube.

The successful approaches arbitrary subgraphs of hypercubes, $(d-1)$-regular subgraphs of d-dimensional hypercubes and finally 3-regular subgraphs of hypercubes.

At the end of Section 6.2 on acyclic edge colouring 3-regular subgraphs of hypercube, we suggested how the future work is also possible for that approach, since the approach has high potentials and scope for all bipartite graphs of regularity equals to 3.

As a future work, apart from mentioned earlier, one can focus on acyclic edge colouring of $(d-2)$-regular, $(d-3)$-regular, and further categories of subgraphs of d-dimensional hypercubes. Also, one can try to get mathematical proofs of approaches we discussed above. We hope that future work on this will get some very good results.

Bibliography

[1] N. Alon, B. Sudakov, and A. Zaks. Acyclic edge colorings of graphs. *Journal of Graph Theory*, 37(3):157–167, 2001.

[2] N. Alon and A. Zaks. Algorithmic aspects of acyclic edge colorings. *Algorithmica*, 32:611–614, 04 2002.

[3] K. H. Appel and Wolfgang. Every Planar Map is Four-Colorable, Contemporary Mathematics. *98, With the collaboration of J. Koch., Providence, RI: American Mathematical Society, doi:10. 1090/conm/098, ISBN*, 0-8218-5103-9, 1989.

[4] M. Basavaraju and L. Chandran. Acyclic edge coloring of graphs with maximum degree 4. *Journal of Graph Theory*, 61:192 – 209, 07 2009.

[5] B. Grünbaum. Acyclic colorings of planar graphs. *Israel Journal of Mathematics*, 14(4):390–408, Dec 1973.

[6] F. Harary, J. P. Hayes, and H.-J. Wu. A survey of the theory of hypercube graphs. *Computers Mathematics with Applications*, 15(4):277 – 289, 1988.

[7] J. E. . K. Hopcroft and R. M. An n5/2 algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2:1137/0202019., 1971.

[8] A. Kotzig. Hamilton graphs and hamilton circuits, theory of graphs and its applications (proc. In *Sympos. Smolenice 1963), Nakl. CSAV*, page 62–82. Praha.

[9] R. Muthu, N. Narayanan, and C. Subramanian. Optimal acyclic edge colouring of grid like graphs. *Discrete Mathematics*, 310(21):2769 – 2775, 2010.

[10] J. Nešetřil and N. C. Wormald. The acyclic edge chromatic number of a random d-regular graph is d+1. *Journal of Graph Theory*, 49(1):69–74, 2005.

[11] S. Skulrattanakulchai. Acyclic colorings of subcubic graphs. *Information Processing Letters*, 92(4):161 – 167, 2004.

[12] V. Vizing. On an estimate of the chromatic class of a *p*-graph. *Russian) Diskret. Analiz*, (3):25–30.