Universidade Estadual de Campinas
Instituto de Computação

Celso Aimbiré Weffort Santos

# Proper gap-labellings: on the edge and vertex variants

# Rotulações próprias por *gap*: variantes de arestas e de vértices

CAMPINAS
2018

**Celso Aimbiré Weffort Santos**

# Proper gap-labellings: on the edge and vertex variants

# Rotulações próprias por *gap*: variantes de arestas e de vértices

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Thesis presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Computer Science.

**Supervisor/Orientadora: Profa. Dra. Christiane Neme Campos**
**Co-supervisor/Coorientador: Prof. Dr. Rafael Crivellari Saliba Schouery**

Este exemplar corresponde à versão final da Dissertação defendida por Celso Aimbiré Weffort Santos e orientada pela Profa. Dra. Christiane Neme Campos.

CAMPINAS
2018

**Agência(s) de fomento e nº(s) de processo(s):** CAPES
**ORCID:** https://orcid.org/0000-0001-6691-5316

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

Sa59p

Santos, Celso Aimbiré Weffort, 1990-
Proper gap-labellings : on the edge and vertex variants / Celso Aimbiré Weffort Santos. – Campinas, SP : [s.n.], 2018.

Orientador: Christiane Neme Campos.
Coorientador: Rafael Crivellari Saliba Schouery.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Rotulação de grafos. 2. Coloração de grafos. 3. Teoria de grafos. 4. Teoria da computação. I. Campos, Christiane Neme, 1972-. II. Schouery, Rafael Crivellari Saliba, 1986-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Rotulações próprias por gap : variantes de arestas e de vértices
**Palavras-chave em inglês:**
Graph labelings
Graph coloring
Graph theory
Theory of computing
**Área de concentração:** Ciência da Computação
**Titulação:** Mestre em Ciência da Computação
**Banca examinadora:**
Christiane Neme Campos [Orientador]
Simone Dantas de Souza
Flávio Keidi Miyazawa
Sheila Morais de Almeida
**Data de defesa:** 22-05-2018
**Programa de Pós-Graduação:** Ciência da Computação

Universidade Estadual de Campinas
Instituto de Computação

Celso Aimbiré Weffort Santos

# Proper gap-labellings: on the edge and vertex variants

# Rotulações próprias por *gap*: variantes de arestas e de vértices

**Banca Examinadora:**

- Profa. Dra. Christiane Neme Campos
  Universidade Estadual de Campinas
- Profa. Dra. Simone Dantas de Souza
  Universidade Federal Fluminense
- Prof. Dr. Flávio Keidi Miyazawa
  Universidade Estadual de Campinas
- Profa. Dra. Sheila Morais de Almeida
  Universidade Tecnológica Federal do Paraná

A ata da defesa com as respectivas assinaturas dos membros da banca encontra-se no processo de vida acadêmica do aluno.

Campinas, 22 de maio de 2018

# Agradecimentos

A conclusão deste trabalho não seria possível sem a ajuda indispensável de professores, familiares, amigos e colegas.

- Agradeço, em primeiro lugar, a Deus: pela capacitação e pela inteligência. Sem Ele, nada disso seria possível; a Ele, todo o mérito, todo o louvor, toda a honra e toda a glória.
- Agradeço à minha maravilhosa esposa, Bianca Cristine: pelo apoio e suporte, pela paciência, pelas discussões e correções, pela presença e companhia. Você é minha alegria, minha inspiração e o amor da minha vida.
- Agradeço, especialmente, à Profa. Christiane: por confiar no meu trabalho, por se dispor a me orientar e, principalmente, pela amizade e pelo carinho que você e sua família demonstram diariamente. Você me ensinou muito mais do que Teoria de Grafos e métodos de prova; você me ensinou como ser um pesquisador melhor e uma pessoa melhor.
- De semelhante modo, agradeço ao Prof. Rafael: pela contribuição importantíssima ao trabalho e à minha formação. Obrigado pelas discussões, ensinamentos e descontrações.
- Agradeço aos membros da banca: Profa. Sheila, Profa. Simone e Prof. Flávio. Pela cuidadosa leitura, correção e avaliação da dissertação.
- Agradeço aos meus pais: Cid e Almeriane. Pelo encorajamento, pelo apoio e, principalmente, pela amizade.
- Agradeço ao Prof. Lehilton por ter disponibilizado o tempo de seu aluno de doutorado para concluir os trabalhos pendentes do mestrado.
- Em particular, agradeço ao meu colega de orientação Atílio, pela contribuição que tornou possível a minha tão-desejada redução.
- Não por menos, agradeço também aos membros da Secretaria de Pós-Graduação do IC. Denise e Wilson, a ajuda e prontidão de vocês foi indispensável em todas as etapas deste mestrado.
- Agradeço às agências de fomento à pesquisa, CAPES e CNPq, pelo auxílio financeiro, tanto para os meus estudos quanto para as apresentações de trabalhos em congressos.
- Por fim, agradeço aos membros do LOCo pelo ambiente divertido de trabalho, pelas conversas, pelas críticas e pelos elogios.

# Resumo

Uma rotulação própria é uma atribuição de valores numéricos aos elementos de um grafo, que podem ser seus vértices, arestas ou ambos, de modo a obter – usando certas funções matemáticas sobre o conjunto de rótulos – uma coloração dos vértices do grafo tal que nenhum par de vértices adjacentes receba a mesma cor.

Este texto aborda o problema da rotulação própria por *gap* em suas versões de arestas e de vértices. Na versão de arestas, um vértice de grau pelo menos dois tem sua cor definida como a maior diferença, i.e. o maior *gap*, entre os rótulos de suas arestas incidentes; já na variante de vértices, o *gap* é definido pela maior diferença entre os rótulos dos seus vértices adjacentes. Para vértices de grau um, sua cor é dada pelo rótulo da sua aresta incidente, no caso da versão de arestas, e pelo rótulo de seu vértice adjacente, no caso da versão de vértices. Finalmente, vértices de grau zero recebem cor um. O menor número de rótulos para o qual um grafo admite uma rotulação própria por *gap* de arestas (vértices) é chamado *edge-gap* (*vertex-gap*) *number*.

Neste trabalho, apresentamos um breve histórico das rotulações próprias por *gap* e os resultados obtidos para as duas versões do problema. Estudamos o *edge-gap* e o *vertex-gap numbers* para as famílias de ciclos, coroas, rodas, grafos unicíclicos e algumas classes de *snarks*. Adicionalmente, na versão de vértices, investigamos a família de grafos cúbicos bipartidos hamiltonianos, desenvolvendo diversas técnicas de rotulação para grafos nesta classe.

Em uma abordagem relacionada, provamos resultados de complexidade para a família dos grafos subcúbicos bipartidos. Além disso, demonstramos propriedades estruturais destas rotulações de vértices por *gap* e estabelecemos limitantes inferiores e superiores justos para o *vertex-gap number* de grafos arbitrários. Mostramos, ainda, que nem todos os grafos admitem uma rotulação de vértices por *gap*, exibindo duas famílias infinitas que não admitem tal rotulação. A partir dessas classes, definimos um novo parâmetro chamado de *gap-strength*, referente ao menor número de arestas que precisam ser removidas de um grafo de modo a obter um novo grafo que é *gap*-vértice-rotulável. Estabelecemos um limitante superior para o *gap-strength* de grafos completos e apresentamos evidências de que este valor pode ser um limitante inferior.

# Abstract

A proper labelling is an assignment of numerical values to the elements of a graph, which can be vertices, edges or both, so as to obtain – through the use of mathematical functions over the set of labels – a vertex-colouring of the graph such that every pair of adjacent vertices receives different colours.

This text addresses the proper gap-labelling problem in its edge and vertex variants. In the former, a vertex of degree at least two has its colour defined by the largest difference, or gap, among the labels of its incident edges; in the vertex variant, the gap is defined by the largest difference among the labels of its adjacent vertices. For a degree-one vertex, its colour is defined by the label of its incident edge, in the edge version, and by the label of its adjacent vertex, in the vertex variant. Finally, degree-zero vertices receive colour one. The least number of labels for which a graph admits a proper gap-labelling of its edges (vertices) is called the edge-gap (vertex-gap) number.

In this work, we present a brief history of proper gap-labellings and our results for both versions of the problem. We study the edge-gap and vertex-gap numbers for the families of cycles, crowns, wheels, unicyclic graphs and some classes of snarks. Additionally, in the vertex version, we investigate the family of cubic bipartite hamiltonian graphs and develop several labelling techniques for graphs in this class.

In a related approach, we prove hardness results for the family of subcubic bipartite graphs. Also, we demonstrate structural properties of gap-vertex-labelable graphs and establish tight lower and upper bounds for the vertex-gap number of arbitrary graphs. We also show that not all graphs admit a proper gap-labelling, exhibiting two infinite families of graphs for which no such vertex-labelling exists. Thus, we define a new parameter called the gap-strength of graphs, which is the least number of edges that have to be removed from a graph so as to obtain a new, gap-vertex-labelable graph. We establish an upper bound for the gap-strength of complete graphs and argue that this value can also be used as a lower bound.

# Contents

# Chapter 1

# Introduction

Graph Theory is arguably one of the most important theoretical fields of study in Computer Science and its applications to day-to-day problems have attracted many researchers. The origins of Graph Theory can be traced back to 1852, when Francis Guthrie asked whether, given a map divided into regions and a set of colours, it would suffice to use only four of these colours to paint the regions of the map, such that no two neighbouring regions receive the same colour. Although the statement of this problem, which nowadays is referred to as the *Four-Colour Problem*, is fairly simple and intuitive, it remained unsolved for over one hundred years. In 1976, K. Appel and W. Haken [2] presented a controversial computer-aided proof to this problem. Almost twenty years later, N. Robertson et al. [23] presented a more simplified proof. However, no proof exists until this day to the Four-Colour Problem which does not require an extensive case-checking phase, that can only be done with the help of a computer.

The many attempts to prove (or disprove) the Four-Colour Problem originated and developed several fundamental areas in Graph Theory. In fact, many concepts in these areas are used in applications which are apparently unrelated to Graph Theory. As an example, consider the implementation and development of social networks, which are largely based on Graph Theory.

In this work, we study graph labellings, which is concerned with the assignment of numerical values to the elements of a graph, obeying some arithmetical properties. Moreover, we study the concepts of graph labellings intertwined with graph colourings, in an area of research called *Proper Graph Labellings*. This area originated in the 1960s when A. Rosa [24] proposed labellings of graphs using numerical values that, through some mathematical function over the set of labelled elements, create a colouring of the graph. In particular, this thesis presents progress on two types of proper labellings: the edge and the vertex versions of gap-labellings.

The remainder of this chapter is divided as follows. We begin by presenting some fundamental concepts of Computer Science in Section 1.1. In Section 1.2, we introduce basic concepts, definitions and terminology used in Graph Theory. Finally, Section 1.3 provides a short history of graph labellings and an overview of proper gap-labellings.

## 1.1 Computational complexity

A (*computational*) *problem* is a general question accompanied with some parameters, referred to as *input*, for which one desires to obtain a specific answer, called *output*. A set of input data to a specific computational problem is called an *instance* of the problem. The *size* $n$ of an instance is the value which reflects the amount of data that is required to describe such instance. The *statement* informs the desired relationship between input and output. As an example, consider the *Primality* problem.

PRIMALITY
**Instance:** An integer $k$.
**Question:** Is $k$ a prime number?

In this case, $k$ is the input data and each distinct value of $k$ is a different instance of PRIMALITY. The value of $k$ in this problem, however, does not necessarily reflect the size of the instance. Given a binary representation of $k$, it requires $\lceil \log_2 k \rceil$ bits to describe each instance. Hence, in this example, the size of the input is $n = \lceil \log_2 k \rceil$.

Note that the output of PRIMALITY is a simple[1] "yes" or "no" answer. A problem whose output is a yes or no answer is called a *decision problem*. An instance of a decision problem $\mathcal{P}$ whose answer is "yes" is referred to as a *yes instance of* $\mathcal{P}$. Analogously, a *no instance* of $\mathcal{P}$ is one whose answer is "no".

There are other types of problems which require an answer that is more complex than just a yes/no. For example, consider a problem which asks for an ordering of a given set of integers $\{n_1, n_2, \ldots, n_m\}$. This example perfectly distinguishes the concepts of input and instance. For this problem, the input is always a set and each distinct set is a different instance. The output for this problem is an $m$-tuple $(n'_1, n'_2, \ldots, n'_m)$, which is a permutation of the input data, where $n'_1 \leq n'_2 \leq \ldots \leq n'_m$.

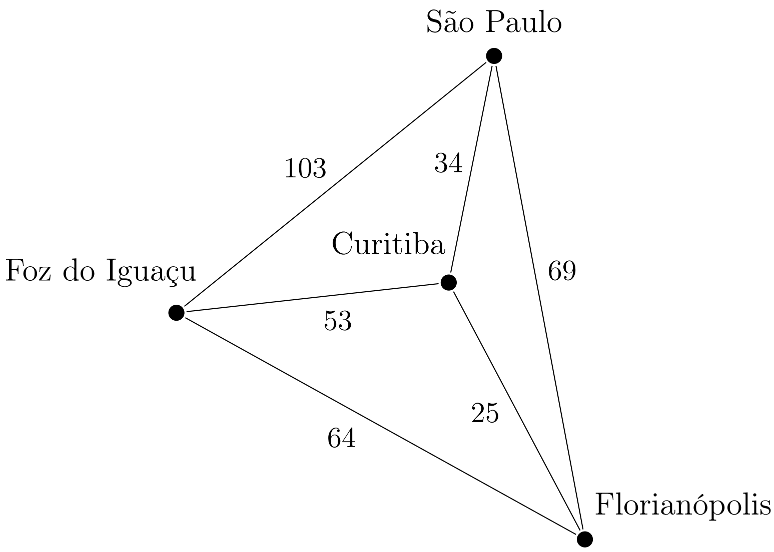
Another type of problems are those called *optimization* problems. In this case, we wish to find, for a given instance, a solution which is *optimal* according to some criteria that is specified in the statement. For example, consider one of the most notorious problems in Theoretical Computer Science, the TRAVELLING SALESMAN PROBLEM, stated as follows.

TRAVELLING SALESMAN PROBLEM (TSP)
**Instance:** A set of cities and distances between every pair of them.
**Question:** What is the shortest possible route that visits each city exactly once and returns to the first city?

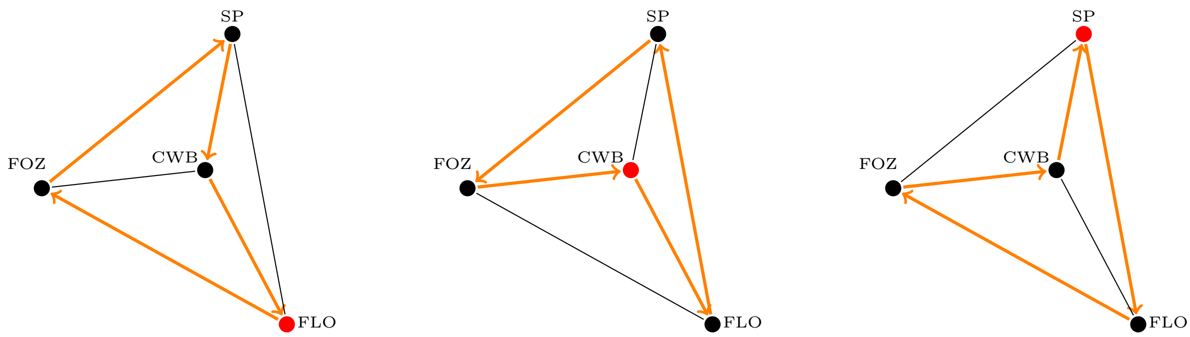
As an example, consider Figure 1.1(a), which depicts a set of four cities. Figure 1.1(b) presents some of the possible routes, each of which visits every city and returns to the first city, highlighted in red. For this particular instance, an optimal route travels a distance of 220, starting at São Paulo, for example, then visiting Florianópolis, Foz do Iguaçu and Curitiba, in sequence.

An *algorithm* is a finite set of rules which provides a sequence of operations that resolve a specific computational problem. According to D. E. Knuth [19], every algorithm has

[1] The word simple, here, is used as an abuse of notation. We are not implying that solving this problem is simple in any way; only that the answer is just a yes or no.




(a) An instance of TSP.




(b) Three possible routes, with travelled distances 226, 250 and 220, respectively. The rightmost is an optimal solution for this instance of TSP.

Figure 1.1: Example of the Travelling Salesman Problem.

five important characteristics: (i) it always ends, that is, it *executes* in a finite amount of time; (ii) each step of the algorithm is rigorously defined, allowing no ambiguities or doubts as to which operation should be executed in each step; (iii) it has input; and (iv) output data; and, finally, (v) an algorithm must be feasible, that is, the operations in an algorithm must be sufficiently basic such that any person will be able to perform them.

The area of *Computational Complexity* consists of determining the amount of resources that are required to execute an algorithm. These *resources* encompass memory usage, communication bandwidth, power consumption, amount of hardware required, and execution time. The latter is the main focus of our study. Let us, then, define $T(n)$ as the maximum *running time* of an algorithm that solves a given problem $\mathcal{P}$, for instances of size $n$. An algorithm is said to be *efficient* if $T(n)$ is bound by some polynomial $f(n)$, for a sufficiently large $n$.

Consider an arbitrary computational problem $\mathcal{P}$ and let $\mathscr{A}^{\mathcal{P}}$ be the collection of all (known and unknown) algorithms that solve $\mathcal{P}$. If there exists an efficient algorithm in $\mathcal{A}^{\mathcal{P}}$, then $\mathcal{P}$ is said to be *tractable*, or *polynomial-time solvable*. However, if no such algorithm exists, and $\mathcal{A}^{\mathcal{P}}$ is a nonempty collection, then $\mathcal{P}$ is said to be *intractable*. Class $\mathsf{P}$ comprises all tractable decision problems, that is, problems for which the yes/no answer can be determined in polynomial time.

Instances of intractable problems can be, at most, *verified* in polynomial time. This procedure is done by a *verification algorithm*, which receives as input two objects: an instance of the problem and a set of arguments related to that instance; these arguments are called a *certificate*. The output of a verification algorithm is either *yes* or *no*. In case of the former, we say that the verification algorithm *accepts* the certificate. According to P. Feofiloff [9], a polynomial-time verification algorithm for a decision problem $\mathcal{P}$ is such that: (i) for every *yes* instance of $\mathcal{P}$, there exists a certificate which the algorithm accepts in polynomial time (in the size of the instance of $\mathcal{P}$); and (ii) for every *no* instance, there is no acceptable certificate.

As an illustration of a verifying algorithm, consider the decision version of the Travelling Salesman Problem, TSP-Decision. In this variant, we ask whether there exists a route that visits each city exactly once, returns to the first city and has length no more than a parameter $k \in \mathbb{Z}_{\geq 0}$, instead of asking for an optimal solution/route. A possible verifying algorithm for this problem receives as input an arbitrary instance of TSP-Decision and a sequence of cities $c_1, c_2, \ldots, c_n, c_1$. By following this sequence, the algorithm sums the distances between consecutive cities. At the end, it checks if the total sum is less than or equal to the input parameter $k$. If so, the algorithm answers *yes*. In this case, sequence $c_1, c_2, \ldots, c_n, c_1$ is the certificate and, moreover, the verification is done in polynomial time.

With that in mind, let us define $\mathsf{NP}$ as the class that comprises all the decision problems whose *yes* instances can be verified in polynomial time. Note that a problem $\mathcal{P}$ which belongs to $\mathsf{P}$ also belongs to class $\mathsf{NP}$ – one needs only use, as a verifying algorithm for $\mathcal{P}$, the existing efficient algorithm that solves the problem. Therefore, $\mathsf{P} \subseteq \mathsf{NP}$.

Now, consider two decision problems $\mathcal{P}_1$ and $\mathcal{P}_2$. Let $I_1$ be any instance of $\mathcal{P}_1$ whose answer is $R_1$. Let $f$ be an algorithm which transforms $I_1$ into an instance $I_2$ of $\mathcal{P}_2$, whose answer is $R_2$. If answer $R_1$ is *yes* if and only if answer $R_2$ is also *yes*, then $f$ is a *reduction* from $\mathcal{P}_1$ to $\mathcal{P}_2$. Additionally, if $f$ executes in polynomial time, we say that $\mathcal{P}_1$ is *polynomial-time reducible* to $\mathcal{P}_2$ and denote this relationship by $\mathcal{P}_1 \preceq_P \mathcal{P}_2$.

Polynomial-time reducibility between problems $\mathcal{P}_1$ and $\mathcal{P}_2$ implies two fundamental consequences. First, suppose there exists an efficient algorithm $A_2$ that solves $\mathcal{P}_2$, i.e. $A_2 \in \mathsf{P}$. Then, there exists an algorithm $A_1$ that: transforms $I_1$ into $I_2$ using $f$; solves $I_2$ using $A_2$; and answers $R_1 = R_2$. Thus, $\mathcal{P}_1$ is also in $\mathsf{P}$ since $A_1$ also executes in polynomial time. In this case, we say that $\mathcal{P}_1$ is *as easy as* $\mathcal{P}_2$. The second consequence follows in the other direction and requires further attention; this is done next.

Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two problems. If $\mathcal{P}_1 \preceq_P \mathcal{P}_2$, we know that solving an instance of $\mathcal{P}_2$ also solves an (equivalent) instance of $\mathcal{P}_1$. Therefore, we conclude that solving problem $\mathcal{P}_2$ is at least *as hard as* solving problem $\mathcal{P}_1$. Note that if $\mathcal{P}_1$ belongs to a certain class of problems – say $\mathsf{NP}$, for example – then problem $\mathcal{P}_2$ is at least as hard as an $\mathsf{NP}$ problem.

Then, we can define $\mathsf{NP}$-*hard* as the class of problems $\mathcal{P}$ such that every problem $\mathcal{P}' \in \mathsf{NP}$ can be reduced to $\mathcal{P}$ in polynomial time. Finally, class $\mathsf{NP}$-*complete* is defined as the class of decision problems $\mathcal{P}$ such that: (i) $\mathcal{P} \in \mathsf{NP}$; and (ii) $\mathcal{P}$ is $\mathsf{NP}$-hard. We abuse notation and say that a problem $\mathcal{P}$ *is* NP-complete when $\mathcal{P}$ belongs to this class. The aforementioned problem TSP-Decision is an example of an NP-complete problem. This means that every other decision problem in NP is polynomial-time reducible to TSP-Decision. Therefore, if a polynomial-time algorithm is discovered for this problem (or any other NP-complete problem), then the entire NP class collapses into P, implying that P = NP. In this context, NP-complete problems are considered the "hardest" problems in NP and comprise many of the day-to-day problems we face. Moreover, an efficient algorithm for any NP-complete problem has yet to be discovered. In fact, the question "is P = NP?" remains open, with many researchers devoting their efforts to settle the question.

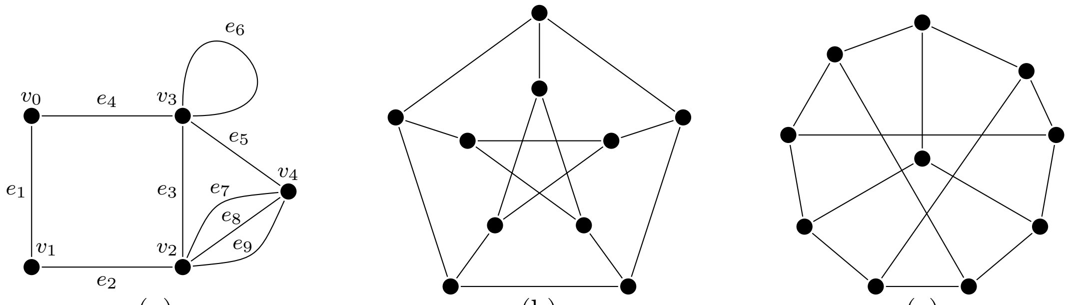
In this work, we focus on certain problems in Graph Theory, many of which have been proven to be NP-complete. Before we present these problems, it is necessary to introduce some concepts and the basic terminology used throughout this thesis.

## 1.2 Graph theory

A *graph* $G = (V(G), E(G))$ is an ordered pair consisting of a nonempty finite set $V(G)$ of *vertices*, a finite set $E(G)$ of *edges*, disjoint from $V(G)$, together with an *incidence function*, $\psi_G$, that associates each edge $e \in E(G)$ with an unordered pair of (not necessarily distinct) vertices of $V(G)$. The *elements* of a graph are its vertices and its edges. The number $|V(G)|$ denotes the *order* of a graph and $|E(G)|$ denotes its *size*. If $\psi_G(e) = \{u, v\}$ for an edge $e$ of $E(G)$, we say that $u$ and $v$ are the *ends* of $e$, or, equivalently, that $u$ and $v$ are its *endpoints*. Whenever there is no ambiguity, $V(G)$ may be denoted simply by $V$, $E(G)$, by $E$ and $\psi_G$, by $\psi$.

A graphical representation of a graph $G$ in the plane is called a *drawing* of $G$. In this work, all drawings of graphs have vertices represented as different points on the plane, and each edge is represented by a simple curve joining its ends. Also, a point corresponding to a vertex $v \in V(G)$ and a curve corresponding to an edge $e \in E(G)$ intersect in a drawing of $G$ if and only if $v$ is an endpoint of $e$. We say that $G$ is *planar* if there exists a drawing of the graph in the plane such that no two edges of $G$ intersect, except at its endpoints. Figure 1.2 shows drawings of some graphs. Observe that in Figure 1.2(a), the incidence function is represented implicitly by the curves connecting vertices; as examples: $\psi(e_1) = \{v_0, v_1\}$, $\psi(e_6) = \{v_3\}$ and $\psi(e_7) = \psi(e_8) = \psi(e_9) = \{v_2, v_4\}$. Also, note that the graph in Figure 1.2(a) is planar. Figures 1.2(b) and 1.2(c) illustrate two different drawings of the well known *Petersen Graph* without naming its vertices and edges.

A graph with a single vertex and no edges is called a *trivial graph*; a graph with no edges is called an *empty graph*. We say that an edge $e = \{u, v\}$ *links* vertices $u$ and $v$. If an edge of $G$ has a single vertex as both its ends, that is, edge $e$ has $\psi_G(e) = \{u, u\} = \{u\}$, for some $u \in V(G)$, then $e$ is called a *loop*. If there are two edges $e, f \in E(G)$ such that $\psi(e) = \psi(f)$, then $e$ and $f$ are called *parallel* or *multiple* edges. A *simple graph* is a graph without loops and parallel edges. In this work, all graphs considered are simple. Therefore, each edge $e$ of $G$, such that $\psi(e) = \{u, v\}$, is unique and can be denoted simply by $e = uv$. Thus, we can omit the incidence function $\psi$ since it is implicitly defined by




(a) (b) (c)

Figure 1.2: Drawings of graphs.

the ends of the edges. In Figure 1.2(a), $e_6$ is a loop and $e_7, e_8$ and $e_9$ are parallel edges.

*Adjacency* is a relation between elements of the same set. Two vertices $u, v$ in a graph $G$ are *adjacent* if edge $uv$ exists in $E(G)$. Similarly, two edges are adjacent if they share a common endpoint. Additionally, we define *incidence* as a relation between elements of different sets, that is, between an edge and a vertex. We say that an edge $e$ is *incident with* a vertex $v$ (and vice-versa) if $e$ has $v$ as one of its endpoints.

The *neighbourhood* $N(v) \subseteq V(G)$ of a vertex $v \in V(G)$ is the set of vertices that are adjacent to $v$. The *closed neighbourhood* of a vertex $v$, denoted by $N[v]$, is defined as $N[v] = N(v) \cup \{v\}$. If $N[v] = V(G)$, then $v$ is called a *universal vertex*. The set of edges incident with a vertex $v$ is denoted by $E(v)$. In Figure 1.2(a), for instance, $E(v_1) = \{e_1, e_2\}$, $E(v_4) = \{e_5, e_7, e_8, e_9\}$, and $N(v_2) = \{v_1, v_3, v_4\}$, $N[v_2] = \{v_1, v_2, v_3, v_4\}$, and $N(v_3) = N[v_3] = \{v_0, v_2, v_3, v_4\}$.

Let $v$ be a vertex of $V(G)$. The *degree* of $v$ in $G$, denoted by $d_G(v)$, is the number of times $v$ is an endpoint of edges in $G$. For example, vertices $v_3$ and $v_4$ in Figure 1.2(a) have $d_G(v_3) = 5$ and $d_G(v_4) = 4$. If there is no ambiguity, $d_G(v)$ is denoted in the text simply by $d(v)$. The *maximum degree* of $G$ is defined as $\Delta(G) = \max\{d(v) : v \in V(G)\}$; similarly, the *minimum degree* of $G$ is defined as $\delta(G) = \min\{d(v) : v \in V(G)\}$. If $d(v) = k$ for every $v \in V(G)$, then $G$ is said to be *$k$-regular*. In particular, when $k = 3$, $G$ is called a *cubic* graph. The Petersen Graph, illustrated in Figure 1.2(b), is an example of a cubic graph. For the purposes of this work, we also define a *subcubic* graph, in which $d(v) \leq 3$ for every vertex $v$.

A graph $H$ is a *subgraph* of $G$, denoted by $H \subseteq G$, if $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$ and $\psi_H$ is a restriction of $\psi_G$ to $E(H)$. Note that every edge $e = uv$ in $E(H)$ has its ends $u, v$ in $V(H)$. If $H \subseteq G$, we say that $G$ *contains* graph $H$ or, equivalently, that $G$ *has* $H$ (as a subgraph). Additionally, $H$ is said to *be contained* in $G$. A graph with vertex set $X \subseteq V(G)$ and edge set composed of every edge of $G$ with both ends in $X$ is an *induced subgraph* of $G$ and is denoted by $G[X]$. Figure 1.3 illustrates a graph $G$, a subgraph $H \subseteq G$ and an induced subgraph $H' \subseteq G$.

Let $G$ and $H$ be two graphs. An *isomorphism* from $G$ to $H$ is a pair $(\phi, \theta)$ where $\phi : V(G) \to V(H)$ and $\theta : E(G) \to E(H)$ are two bijections with the property that $\psi_G(e) = \{u, v\}$ if and only if $\psi_H(\theta(e)) = \{\phi(u), \phi(v)\}$. In this case, we say that $G$ and $H$ *are isomorphic* and denote this relation by $G \cong H$. Note that the graphs in Figure 1.2(b)

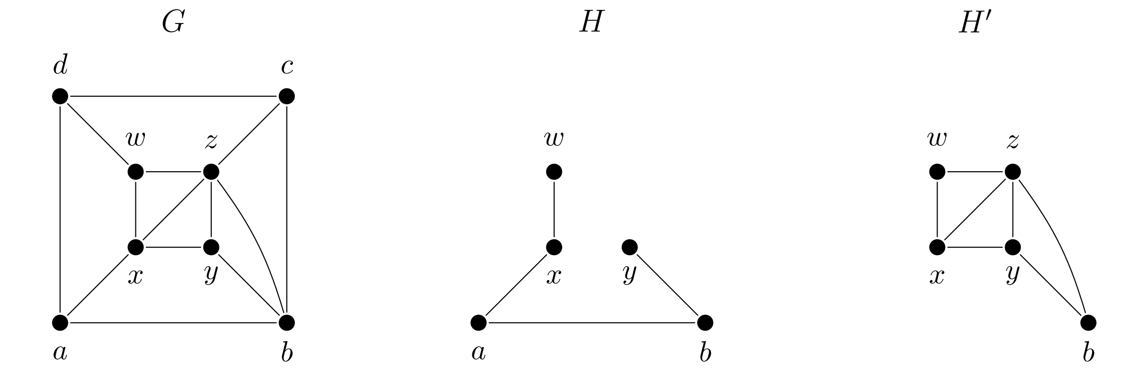
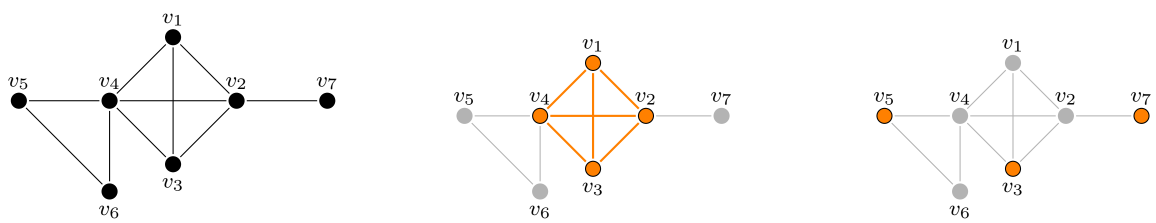



Figure 1.3: In (a), a graph $G$; in (b), a subgraph of $H$ of $G$; and in (c), an induced subgraph $H' = G[\{b, x, y, z, w\}]$. Note that $H$ is not an induced subgraph since edge $xy \notin E(H)$.

and 1.2(c) are isomorphic since they are different drawings of the Petersen Graph.

A *clique* in a graph is a set of mutually adjacent vertices. On the other hand, a set of vertices that is pairwise nonadjacent is an *independent set*. Figure 1.4 illustrates these concepts. A *matching* $M$ of a graph $G$ is a set of pairwise nonadjacent edges; such a set is also called an *independent set* of edges of $E(G)$. If a vertex $v$ of $G$ is incident with an edge $e \in M$, we say that $v$ is *saturated* by $M$; otherwise, $v$ is *unsaturated* by $M$. A matching $M$ is said to be *maximal* if there is no other matching $M'$ in $G$ such that $M \subset M'$. If every $v \in V(G)$ is saturated by a matching $M$, we say that $M$ is a *perfect matching* of $G$. In Figure 1.5, we provide some examples of matchings of graphs.




(a) Graph $G$. (b) A clique in $G$. (c) An independent set.

Figure 1.4: A graph $G$ and illustrations of cliques and independents sets in $G$.

Many times, it is necessary to perform modifications to the structure of a graph $G$ through some operations in the elements of $G$. Next, we define some of these that are important in this work.

Let $G$ be a graph, $e \in E(G)$ and $v \in V(G)$. The graph $G - v$ is obtained by *removing vertex* $v$ from $V(G)$. Therefore, $G - v$ has vertex set $V(G)\backslash\{v\}$ and edge set $E(G)\backslash\{uv : uv \in E(G)\}$. Similarly, graph $G - e$ is obtained by *removing edge* $e$ from $E(G)$. In this case, the vertex set remains unchanged, while $E(G - e) = E(G)\backslash\{e\}$. For a given set of elements $X \subseteq V(G)$ or $X \subseteq E(G)$, the *removal of* $X$ from $G$, denoted by $G\backslash X$, is defined as the removal, in any order, of each element $x \in X$ from $G$, according to the previous operations.

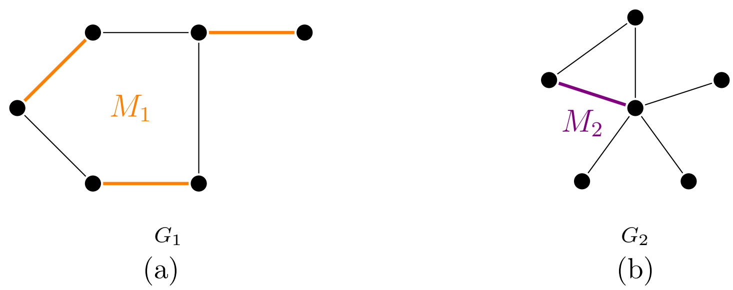



Figure 1.5: Examples of matchings.

Now, let $u, w \in V(G)$ be two distinct vertices of a simple graph $G$. The *identification* of $u$ and $w$ is the operation defined by: (i) adding a new vertex $v_{uw}$ to $G$; (ii) removing vertices $u$ and $w$ from $G$; (iii) for every $xy \in E(G)$, $x \in \{u, w\}$, add edge $v_{uw}y$ to the new graph; and (iv) removing any parallel edges and loops that may have been created in step (iii). The graph resulting from identifying vertices $u, w$ is denoted by $G_{uw}$. Figure 1.6 illustrates this operation.

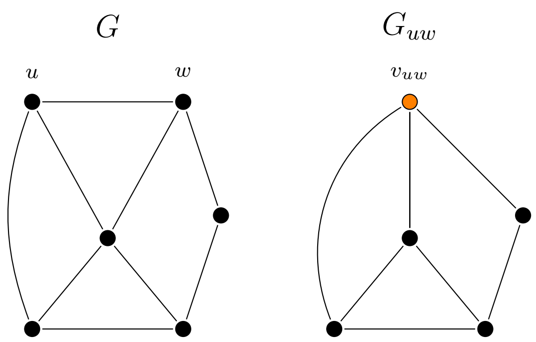



Figure 1.6: The identification of vertices $u$ and $w$.

A *walk* in a graph $G$ is an alternating sequence of vertices and edges $P = v_0e_1v_1 \dots e_lv_l$ such that $e_i \in E(G)$, $v_i \in V(G)$ and $e_i = v_{i-1}v_i$. If there is no repetition of vertices in $P$, it is called a *path* between $v_0$ and $v_l$. In case of simple graphs, we omit the edges in $P$ since every edge $v_{i-1}v_i$ is uniquely determined. The number $l$ of edges in a walk is its *length* and is denoted by $|P|$. If there exists a path between $u, v$ in $G$, then $u$ and $v$ are *connected* and the *distance* between them is $\text{dist}(u, v) = \min\{|P| : P \text{ is a path between } u \text{ and } v\}$; if $u$ and $v$ are not connected, we define $\text{dist}(u, v) = \infty$. For example, there are several paths between vertices $a$ and $z$ in Figure 1.3(a): $P_1 = abcdwz$, $P_2 = adwxybz$, $P_3 = axwdcz$. However, the shortest paths are $P_4 = abz$ and $P_5 = axz$, both of length 2. Therefore, $\text{dist}(a, z) = 2$.

A graph is *connected* if every pair $u, v$ of vertices of $G$ is connected. A maximal subgraph of $G$ that satisfies this property is called a *connected component* of $G$. A graph that has no cycle is *acyclic*. A *tree* is a connected acyclic graph and is usually denoted by $T$. Connectedness plays an essential role in Graph Theory. For instance, when considering planarity, we can restrict our attention to connected graphs because a graph is planar if and only if each of its connected components is planar.

Now, let $c(G)$ denote the number of connected components of a graph $G$. For any edge $e$ of $G$, either $c(G - e) = c(G)$ or $c(G - e) = c(G) + 1$. In the latter case, we say that $e$ is a *cut edge* or, equivalently, a *bridge*. Thus, removing $e$ from $G$ increases the number of connected components of $G$. A connected graph is said to be *$l$-edge-connected*, $l \geq 1$, if it requires the removal of $l$ or more edges in order to disconnect $G$. For example, graph $G$ in Figure 1.7 is a 2-edge-connected graph since: there are no cut edges in $G$; and removing edges $e_1$ and $e_2$ disconnects $G$.

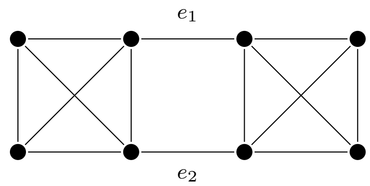



Figure 1.7: A 2-edge-connected graph $G$.

A similar definition is used for vertices. A subset $V'$ of $V(G)$ of a graph $G$ is a *vertex cut* of $G$ if $c(G - V') > c(G)$. If $V' = \{v\}$, vertex $v$ is a *cut vertex* of $G$. A connected graph is *$l$-connected*, $l \geq 1$, if there does not exist a vertex cut $V'$ in $G$ with $|V'| < l$. Note that graph $G$ in Figure 1.7 is also 2-connected since there are no cut vertices in $G$.

We close this section defining some traditional *families* of graphs, which are collections of graphs. By studying a family $\mathcal{F}$ of graphs, it is possible to extend results obtained for a given graph $G \in \mathcal{F}$, which shares some structural property with every other graph $H \in \mathcal{F}$. Other families and their properties are defined in further chapters of the text.

First, a *complete graph*, $K_n$, is a simple graph of order $n$ for which every pair of distinct vertices is adjacent. Figure 1.8(a) shows the complete graph with five vertices. A *cycle* $C_n$ of order $n \geq 3$ is a simple graph with vertices $V = \{v_0, v_1, \ldots, v_{n-1}\}$ and edge set $E = \{v_0v_1, v_1v_2, \ldots, v_{n-2}v_{n-1}, v_{n-1}v_0\}$. If the order of a cycle $C_n$ is even (odd), we say that $C_n$ is an *even* (*odd*) *cycle*. Figure 1.8(b) exemplifies cycle $C_7$.

A *bipartite graph* is a graph whose vertex set $V$ can be partitioned into two subsets, $X$ and $Y$, such that every edge $e \in E(G)$ has one end in $X$ and the other, in $Y$. Such a partition $\{X, Y\}$ of the vertices of $V(G)$ is called a *bipartition* of $G$. If $G$ is a bipartite graph with $\{X, Y\}$ one of its bipartitions, it is also denoted by $G[X, Y]$. If each vertex $x \in X$ is linked to every vertex $y \in Y$, then the graph is called a *complete bipartite graph* and is denoted by $K_{r,s}$ with $r = |X|$ and $s = |Y|$. Figure 1.8(c) exemplifies $K_{3,5}$. In particular, the complete bipartite graph $K_{1,n}$ is called a *star* and is denoted by $S_n$. In this text, $V(S_n) = \{v_0, v_1, \ldots, v_n\}$, where $v_n \in X$ is the *central vertex*. Notice that for star graph $S_n$, $|X| = 1$ and $n$ denotes the size of part $Y$.

Finally, a *hypergraph* $\mathscr{H} = (V, \mathscr{E})$ is defined similarly to graphs, where $V$ is a nonempty finite set of vertices and $\mathscr{E}$ is a set of nonempty subsets of $V$ called *hyperedges*. Observe that if every hyperedge has exactly two distinct vertices, then $\mathscr{H}$ can be viewed as a simple graph. If every hyperedge in $\mathscr{H}$ contains the same number $k$ of vertices, then $\mathscr{H}$ is said to be *$k$-uniform*. Moreover, if every vertex in $V$ belongs to $k$ hyperedges, then $\mathscr{H}$ is *$k$-regular*.

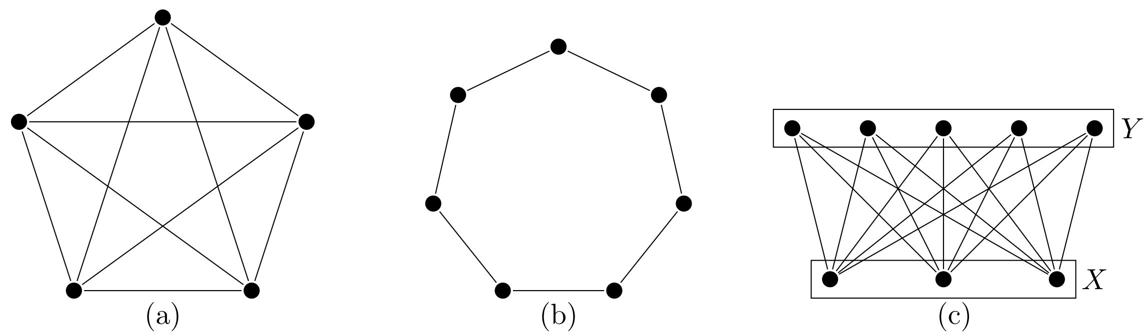


Figure 1.8: In (a), complete graph $K_5$; in (b), cycle $C_7$; and in (c), complete bipartite graph $K_{3,5}$.

### 1.2.1 Graph Colourings

In Graph Theory, the area of *Graph Colourings* studies the assignment of *colours* to the elements of a graph $G$; these elements can be its vertices, edges or both. A *vertex-colouring* is an assignment of colours to the vertices of $G$. Similarly, an *edge-colouring* is an assignment of colours to the edges of $G$. Finally, a *total-colouring* is an assignment of colours to both the edges and the vertices. A colouring of a graph is said to be *proper* if no two adjacent/incident elements receive the same colour.

Let $u, v \in V(G)$ be two adjacent vertices in a graph $G$. If $u$ and $v$ receive the same colour in a vertex-colouring $c$ of $G$, we say that there exists a *conflict* in $c$. Equivalently, $u$ and $v$ are said to be *conflicting* vertices, or that they have *conflicting colours*.

If a proper vertex-colouring of $G$ uses $k$ different colours, we say that $G$ *admits* a *$k$-(vertex-)colouring*. Equivalently, we say that $G$ is *$k$-colourable*. The least $k$ for which $G$ admits a proper vertex-colouring is called the *chromatic number of* $G$ and is denoted by $\chi(G)$. Additionally, a *proper edge-colouring* of a graph $G$ is an assignment of colours to the edges of $G$ such that no two adjacent edges receive the same colour. An edge-colouring of a graph which uses $k$ distinct colours is called a *$k$-edge-colouring*. Similarly, the least $k$ for which a graph admits a proper $k$-edge-colouring is called the *chromatic index of* $G$, and is denoted by $\chi'(G)$. In this work, we are interested in vertex-colourings only.

Note that any graph can be coloured with $k \geq \chi(G)$ colours, as illustrated in Figure 1.9 for the Petersen Graph. In fact, any graph can be properly coloured by assigning a different colour to each vertex in $V(G)$. Therefore, $\chi(G) \leq |V(G)|$. In 1941, R. L. Brooks [5] demonstrated that $\chi(G) \leq \Delta(G)$ for every graph that is not a complete graph or an odd cycle; these graphs have $\chi(K_n) = \Delta(K_n) + 1$ and $\chi(C_{2k+1}) = \Delta(C_{2k+1}) + 1$, respectively. On the other hand, if set $E(G)$ is nonempty, then there are at least two adjacent vertices in $G$. This implies that any proper colouring of $G$ needs at least two colours, which establishes a lower bound for $\chi(G)$. In particular, observe that the existence of a proper 2-colouring of a graph with at least two vertices is an alternative definition for a bipartite graph, as stated in the following theorem.

**Theorem 1.1.** *A simple graph with at least two vertices is bipartite if and only if it is 2-colourable.*

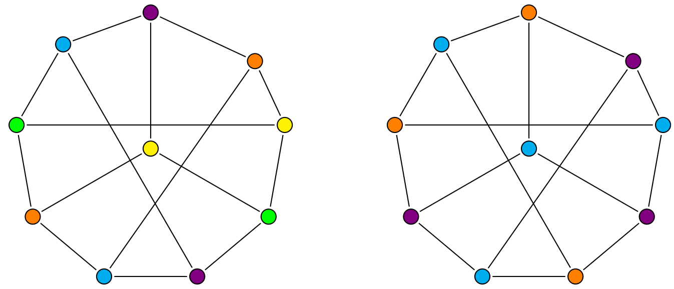


Figure 1.9: A 5-colouring and a 3-colouring of the Petersen Graph. The chromatic number of the Petersen Graph is 3.

*Proof.* Let $G$ be a simple graph. The result follows from the fact that, in any 2-colouring of $G$, vertices with the same assigned colour form an independent set and each part of any bipartition of $G$ is also an independent set. $\square$

Now, consider the family of cycles. Even-length cycles are bipartite graphs and are 2-colourable by Theorem 1.1. Also, by the same theorem, odd cycles do not admit 2-colourings. In fact, $\chi(C_{2k+1}) = 3$. This can be observed by assigning a colour to an arbitrary vertex of $C_{2k+1}$ and, then, alternating two new colours along the remaining vertices of the cycle.

**Theorem 1.2.** *Let $G \cong C_n$. Then, $\chi(G) = 2$ if $n$ is even, and $\chi(G) = 3$, otherwise.* $\square$

It is important to remark that determining $\chi(G)$ can be very difficult. In fact, deciding whether a graph admits a $k$-colouring, for any $k \geq 3$, is an $\mathsf{NP}$-complete problem. For $k = 2$, however, this decision problem can be solved in polynomial time since there exists a polynomial-time algorithm that decides whether a graph is bipartite.

As stated in the preamble of this chapter, graph colourings have been studied since the 18th century, with discoveries and properties shaping the very basis of Graph Theory. Each colouring of a graph can be seen as an assignment of labels, or colours, to elements of the graph subject to certain constraints. In the 1960s, A. Rosa [24] defined a new type of graph labellings, which we present in detail in the following section.

## 1.3 Graph labellings

Many authors trace the origins of *graph labellings* to Rosa [24] who proposed, in 1967, the assignment of (numerical) *labels* to the elements of the graph, rather than simply colours. In his article, Rosa defined a *$\beta$-valuation* $f$ of a graph $G$ with $m$ edges as an injection $f : V(G) \to \{0, 1, \ldots, m\}$ such that $f$ *induces* another injection $g : E(G) \to \{1, \ldots, m\}$, for which each edge $e = uv$ is assigned label $g(e) = |f(u) - f(v)|$. This labelling was later renamed by S. W. Golomb [16] as *graceful labelling*; graphs that admit such a labelling are also called *graceful*. Complete graph $K_4$ is an example of a graceful graph, as illustrated by Figure 1.10.

Since Rosa's [24] original paper, different types of graph labellings have been proposed, making use of mathematical properties of the labels. As examples, we cite: irregular assignments, harmonious labellings, AVD-colourings, magic and anti-magic labellings. For

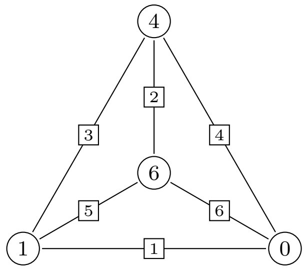



Figure 1.10: A graceful labelling of complete graph $K_4$. The number inside each vertex is its label $f(v)$ and every edge $e$ is assigned the absolute difference of the labels of its endpoints.

a more detailed survey on these labellings, we refer the reader to: B. D. Aharya et al.'s [1] "Labelings of Discrete Structures and its Applications"; A. M. Marr and W. D. Wall's [21] "Magic Graphs"; P. Zhang's [32] "Color-Induced Graph Colorings"; J. Gallian's [11] "Dynamic survey on graph labellings"; or S. C. López and F. A. Muntaner-Batle's [20] "Graceful, Harmonious and Magic Type Labelings - Relations and Techniques".

In 1986, G. Chartrand et al. [6] proposed an assignment of labels $\{1, 2, \ldots, k\}$ to the edges of a graph $G$, such that every vertex $v \in V(G)$ receives a unique colour, computed as the sum of the labels of the edges incident with $v$. This labelling is called an *irregular assigment* and has several applications for nonsimple graphs. Based on their work, in 2004, M. Karoński et al. [18] presented a labelling in which the induced colouring is just a proper vertex-colouring of the graph, rather than a colouring in which every vertex receives a distinct colour. They prove that every 3-colourable graph admits such a labelling using label set $\{1, 2, 3\}$ and posed the *1-2-3 Conjecture*, which states that every graph with no connected component isomorphic to $K_2$ admits such a labelling. In their article, Karoński et al. [18] use the term *proper edge-colouring* to indicate a labelling of a graph that – in this case, via the sum of edge labels – induces a proper vertex-colouring.

The notation of labellings and colourings is not standardized in the literature. In several books, articles and papers, labels/colours are sometimes referred to as "weights", weights become colours, and even the words "labels" and "colours" are interchanged. In order to avoid any ambiguity, we formally define a *proper labelling* of a graph $G$ as a pair $(\pi, c_\pi)$, where $\pi : S \to \{1, \ldots, k\}$ is a labelling of a set $S$ of elements of $G$ and $c_\pi$ is a proper vertex-colouring of $G$ such that $c_\pi(v)$ depends on $\pi$ for every $v \in V(G)$. Labelling $\pi$ is said to *induce* colouring $c_\pi$ and we say that colouring $c_\pi$ is *induced* by $\pi$. Proper labellings are said to be *neighbour-distinguishing* since induced colouring $c_\pi$ is a proper vertex-colouring. In particular, if $c_\pi$ induces a distinct colour for each vertex $v \in V(G)$, we say that $(\pi, c_\pi)$ is also *vertex-distinguishing*. When $S = E(G)$, $(\pi, c_\pi)$ is a *proper edge-labelling*. On the other hand, if $S = V(G)$, then $(\pi, c_\pi)$ is a *proper vertex-labelling*. We state the 1-2-3 Conjecture as an example of our notation.

**Conjecture 1.3** (1-2-3 Conjecture). *Let $G$ be a graph with no component isomorphic to $K_2$. Then, $G$ admits a neighbour-distinguishing proper edge-labelling $(\pi, c_\pi)$, such that $\pi : E(G) \to \{1, 2, 3\}$ and $c_\pi(v) = \sum_{e \in E(v)} \pi(e)$ for every vertex $v \in V(G)$.*

The irregular assignment of graphs inspired many other proper labelling problems, that assign labels to different elements or use new mathematical functions to induce colouring $c_\pi$. In this text, we are interested in two specific proper labellings: the edge and vertex versions of the *proper gap-labelling* of graphs. Each version of this labelling is defined in detail in Chapters 2 and 3, respectively. Here, we introduce only the basic concept in which both labellings are based: inducing colours by "gaps".

Let $S' \subseteq S$ be a subset of some elements $S$ of $G$. These elements can be the vertices or the edges of $G$. Also, let $\pi : S \to \{1, 2, \ldots, k\}$ be a labelling of $S$. We define $\Pi_{S'}$ as the set comprising the labels assigned to the elements of $S'$ in $\pi$. Formally, $\Pi_{S'} = \{\pi(s) : s \in S'\}$. To exemplify, consider Figure 1.11, where $S'_1 = N(v)$, $S'_2 = E(u)$ and $S'_3 = N[w]$. Then, we have $\Pi_{S'_1} = \{1, 2\}$, $\Pi_{S'_2} = \{1, 2, 4\}$ and $\Pi_{S'_3} = \{2, 3, 5\}$.

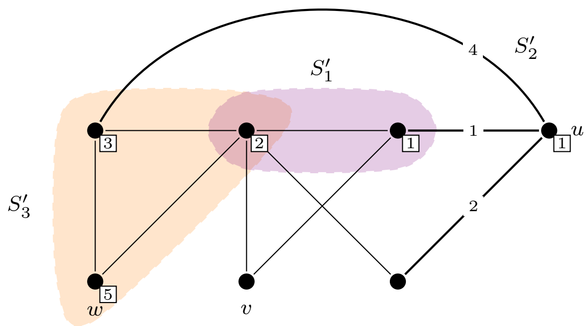



Figure 1.11: A simple graph $G$ with some elements labelled. The numbers inside the white boxes represent the labels of vertices.

In proper gap-labellings, the colour $c_\pi(v)$ of a vertex $v$ of degree at least two is induced by the maximum difference among the labels in set $\Pi_{S'}$ for a specific set $S'$: in the edge version, $S' = E(v)$ and in the vertex version, $S' = N(v)$. We refer to this computed value as the largest *gap* in $\Pi_{S'}$. Isolated vertices in $G$ and vertices with degree one are treated separately in proper gap-labellings and are discussed in detail in the following chapters.

We close this chapter with an outline of this text. In Chapter 2, we present the definition, history and our results for the edge version of the gap-labelling problem. Chapter 3 presents the definition and history of the vertex version of proper gap-labellings, as well as results we obtained during our research. We establish the first lower bound for the vertex-gap number of graphs, a parameter associated with this labelling, and determine it for some traditional classes of graphs. In Chapter 4, we present a different approach to gap-$[k]$-vertex-labellings and define a new parameter called the gap-strength of graphs. Chapter 5 presents concluding remarks.

# Chapter 2

# Gap-$[k]$-edge-labellings

We begin our study of proper gap-labellings by investigating the first version of this problem, which was introduced by M. Tahraoui et al. [27] in 2012. This type of proper labelling assigns numerical values to the edges of a graph so as to induce a proper vertex-colouring. In this work, we refer to this labelling as the gap-$[k]$-edge-labelling of a graph and it is formally defined in the next section.

## 2.1 Preliminaries

In the previous chapter, we mentioned that many researchers proposed different types of proper labellings since A. Rosa's [24] seminal paper. In 2012, M. Tahraoui et al. [27] introduced a new type of proper labelling called gap-$k$-colouring. A *gap-$k$-colouring* of a graph $G = (V, E)$ is defined as a pair $(\pi, c_\pi)$ where $\pi : E \to \{1, 2, \ldots, k\}$ is an edge-labelling of $G$ and $c_\pi : V \to \mathcal{C}$ is a vertex-colouring of $G$ for which every vertex $v \in V$ has a distinct colour defined by:

$$c_\pi(v) = \begin{cases} \max\limits_{e \in E(v)} \{\pi(e)\} - \min\limits_{e \in E(v)} \{\pi(e)\}, & \text{if } d(v) \geq 2; \\ \pi(e)_{e \in E(v)}, & \text{if } d(v) = 1; \\ 1, & \text{if } d(v) = 0. \end{cases} \tag{2.1}$$

We remind the reader that $E(v)$ denotes the set of edges incident with a vertex $v \in V$, as defined in Chapter 1. When $d(v) \geq 2$, $c_\pi(v)$ is induced by the largest difference, i.e. the largest *gap*, among the labels of its incident edges. As an example, Figure 2.1 exemplifies a gap-5-colouring. Note that each edge has been assigned a label between 1 and 5 and the colour of every vertex is unique.

Tahraoui et al. [27] defined the least $k$ for which a graph $G$ admits a gap-$k$-colouring as the *gap chromatic number of* $G$; they denote this parameter by $\text{gap}(G)$. In their article, the authors show that every graph $G$ with no connected components isomorphic to $K_1$ or $K_2$ (also referred to as *isolated edges*) admits a gap-$k$-colouring, for some $k \in \mathbb{N}$. In fact, Tahraoui et al. [27] showed that $\text{gap}(G) \leq 2^{|E|-1}$. They also established the gap chromatic number of paths, cycles, some families of trees and all $l$-connected graphs, for $l \geq 2$. Based

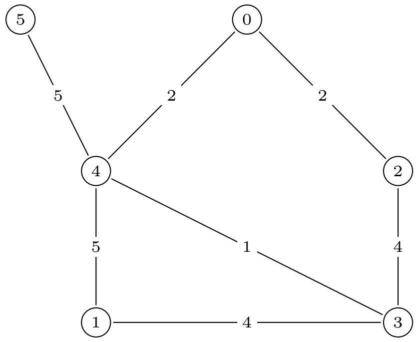



Figure 2.1: A gap-5-colouring of a graph. The number inside each vertex $v$ is its induced colour $c_\pi(v)$.

on their results, the authors conjectured that $\text{gap}(G) \in \{|V|-1, |V|, |V|+1\}$, for every graph $G$. In 2014, R. Scheidweiler and E. Triesch [25] showed that $\text{gap}(G) \leq |V|+7$ for all graphs $G$ with 2-edge connected components. They also improved the upper bound for the gap chromatic number of arbitrary graphs, proving that $\text{gap}(G) \leq |V|+9$. This is the best known bound for arbitrary graphs. Lastly, the authors disproved Tahraoui et al.'s [27] Conjecture by exhibiting a class of graphs for which $\text{gap}(G) = |V|+2$.

In the finishing comments of Tahraoui et al.'s article, they propose that it would be interesting to investigate a version of gap-$k$-colourings in which induced colouring $c_\pi$ is just a proper vertex-colouring. In Figure 2.2, we show that the graph from Figure 2.1 admits an edge-labelling $\pi$ using only labels 1, 2 and 3 such that $c_\pi$ is a proper vertex-colouring of $G$. The colour of each vertex is defined exactly as it is in a gap-$k$-colouring.

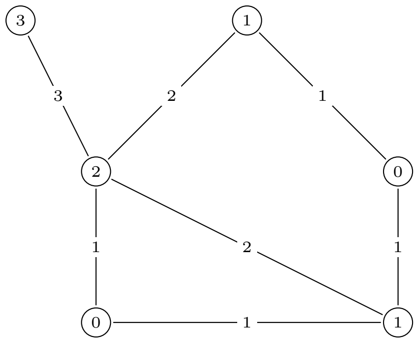



Figure 2.2: An edge-labelling $\pi$ which induces a proper vertex-colouring $c_\pi$.

In 2013, A. Dehghan et al. [8] formally defined this new version of gap-$k$-colourings. An *edge-labelling by gap* of a graph $G = (V, E)$ is a proper labelling $(\pi, c_\pi)$ in which $\pi : E \to \{1, 2, \ldots, k\}$ is an edge-labelling and $c_\pi$, a proper vertex-colouring such that, for every $v \in V$, its colour is defined by equation (2.1).

Note that every gap-$k$-colouring is an edge-labelling by gap. In fact, Tahraoui et al.'s [27] proof on the existence of the first labelling can be used to determine whether a graph admits the latter. Dehghan et al. [8] also proved that every complete graph $K_n$, $n \geq 3$, admits an edge-labelling by gap using label set $\{1, 2, \ldots, n+1\}$. With these results in mind, the authors questioned whether every graph admits an edge-labelling by gap using label set $\{1, 2, \ldots, \chi(G)+1\}$.

The focus of Dehghan et al.'s [8] article, however, is on determining the algorithmic complexity of proper labelling problems, such as the edge-labelling by gap. They showed that deciding whether a graph $G$ admits an edge-labelling by gap using label set $\{1, 2, \ldots, k\}$ is $\mathsf{NP}$-complete when $k \geq 3$. For $k = 2$, the authors proved that deciding whether a planar bipartite graph with minimum degree two admits an edge-labelling by gap can be solved in polynomial time, and that by admitting degree-one[1] vertices in these graphs, the problem becomes $\mathsf{NP}$-complete.

In 2015, Scheidweiler and Triesch [26] also studied edge-labellings by gap and defined the *gap-adjacent-chromatic number of* $G$, $\mathrm{gap}_{ad}(G)$, as the least $k$ for which a graph $G$ admits an edge-labelling by gap using label set $\{1, 2, \ldots, k\}$. In their article, the authors establish bounds for $\mathrm{gap}_{ad}(G)$ for bipartite graphs and 3-colourable graphs and prove that $\chi(G) - 1 \leq \mathrm{gap}_{ad}(G) \leq \chi(G) + 5$ for arbitrary graphs.

Later, in 2016, A. Brandt et al. [4] proposed the *local gap $k$-colouring* of a graph without isolated vertices as a slightly different version of edge-labellings by gap, in which every vertex, regardless of degree, has its colour induced by the largest gap among the labels of its incident edges. Note that vertices $v$ with $d(v) = 1$ always have induced colour 0 in the local version. The authors use the local gap $k$-colouring to improve the bounds set by Scheidweiler and Triesch [26], as stated in the following theorem. We remark that Brandt et al.'s result shows that Scheidweiler and Triesch's lower bound is tight for stars.

**Theorem 2.1** (Brandt et al.)**.** *If $G$ is a graph without isolated edges, then $gap_{ad}(G) \in \{\chi(G), \chi(G) + 1\}$ unless $G$ is a star, in which case $gap_{ad}(G) = 1 = \chi(G) - 1$.* □

The best known bounds for the gap-adjacent-chromatic number of graphs are the ones established in Theorem 2.1. In the concluding remarks of their article, Brandt et al. [4] also determine the gap-adjacent-chromatic number for cycles and give a simpler labelling for complete graphs (the one proposed by Dehghan et al. [8] is recursive).

## Notation

As we mention in Chapter 1, there is no standard notation for proper labellings of graphs. Moreover, some of the names used in the literature are misleading and do not accurately express which elements are being labelled and/or coloured. Therefore, in this text, we rename the concepts with the purpose of establishing a notation that properly reflects these differences.

We define a *gap-$[k]$-edge-labelling* of a graph $G = (V, E)$ as an ordered pair $(\pi, c_\pi)$ where $\pi : E \to \{1, 2, \ldots, k\}$ is an edge-labelling of $G$ and $c_\pi : V \to \mathcal{C}$ is a proper vertex-colouring of $G$. Set $\mathcal{C}$ is the set of induced colours. For every vertex $v \in V$, its colour is defined as

$$c_\pi(v) = \begin{cases} \max\limits_{e \in E(v)} \{\pi(e)\} - \min\limits_{e \in E(v)} \{\pi(e)\}, & \text{if } d(v) \geq 2; \\ \pi(e)_{e \in E(v)}, & \text{if } d(v) = 1; \\ 1, & \text{if } d(v) = 0. \end{cases}$$

[1]A *degree-one* vertex $v$ is a vertex with $d(v) = 1$.

The least $k$ for which a graph $G$ admits a gap-[$k$]-edge-labelling is called the *edge-gap number* of $G$ and is denoted by $\chi^{\mathrm{g}}_{E}(G)$. Note the three components of $\chi^{\mathrm{g}}_{E}(G)$: $\chi$ indicates that we are interested in a proper colouring – in this case, of the vertices – of $G$; the superscript $g$ indicates we use gaps to induce the colour in each vertex; and the subscript $E$ is used to imply that the labels are assigned to the edges of $G$. Observe that $\chi^{\mathrm{g}}_{E}(G) = \mathrm{gap}_{ad}(G)$. Thus, we rewrite Brandt et al.'s [4] theorem as follows.

**Theorem 2.1** (Brandt et al.)**.** *If $G$ is a graph without isolated edges, then $\chi^{g}_{E}(G) \in \{\chi(G), \chi(G)+1\}$ unless $G$ is a star, in which case $\chi^{g}_{E}(G) = 1 = \chi(G) - 1$.* □

In Figure 2.3, we exhibit a gap-[3]-edge-labelling of the *Heawood Graph.* Unless otherwise stated, the notation for the labels and colours displayed in this image is used throughout the entirety of this chapter. As an example, consider the topmost vertex $v$ in the image. The edges incident with $v$ have received labels 1, 1 and 3, which induces $c_\pi(v) = 2$.

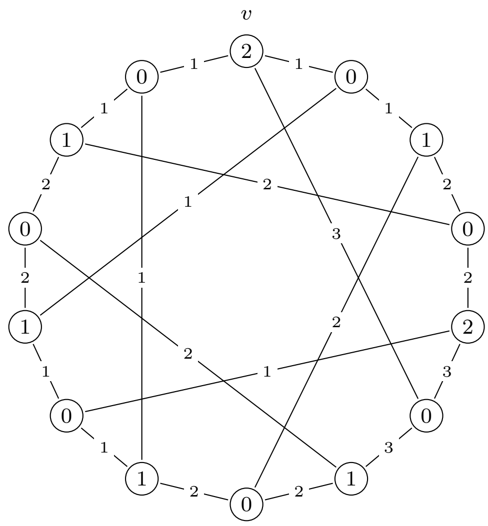



Figure 2.3: A gap-[3]-edge-labelling of the Heawood Graph. The number in each edge corresponds to its label and the number in each vertex, to its induced colour.

We close this section defining the decision problem associated with the gap-[$k$]-edge-labellings of graphs.

Gap-[$k$]-edge-labelling [**Gkel**]

**Instance:** A graph $G = (V, E)$ and a parameter $k$.
**Question:** Does $G$ admit a gap-[$k$]-edge-labelling?

When considering a specific value of $k$, we denote **Gkel** by replacing **k** with its value. For example, we can rewrite the results by Dehghan et al. [8] as:

- **G2el** is NP-complete for planar bipartite graphs;
- **G2el** for planar bipartite graphs $G$ with $\delta(G) \geq 2$ is in P; and
- **Gkel** is NP-complete for $k \geq 3$;

## 2.2 The edge-gap number for classes of graphs

In this section, we present results on the edge-gap number for some classes of graphs. Initially, we determine $\chi_E^{g}(G)$ for cycles, crowns and wheels. These results, along with others presented in Chapter 3, were accepted and presented at the *XXXVII Congresso da Sociedade Brasileira de Computação - 2º Encontro de Teoria da Computação*, July 2017. Then, we investigate and establish the edge-gap number for unicyclic graphs with odd cycles. We close the chapter determining the edge-gap number of some classes of snarks.

### 2.2.1 Cycles

The family of cycles is introduced in Chapter 1. To recall, cycle $C_n$ is a 2-regular graph with vertex set $V(C_n) = \{v_0, v_1, \ldots, v_{n-1}\}$ and edge set $E(C_n) = \{v_0v_1, v_1v_2, \ldots, v_{n-1}v_0\}$. The length of a cycle is the size of its edge set. Theorem 2.2 establishes the edge-gap number for cycles $C_n$, $n \geq 4$. In particular for cycle $C_3$, which is isomorphic to $K_3$, Dehghan et al. [8] established that $\chi_E^{g}(K_3) = 4$. In 2016, Brandt et al. [4], in an independent work, also determined $\chi_E^{g}(C_n)$.

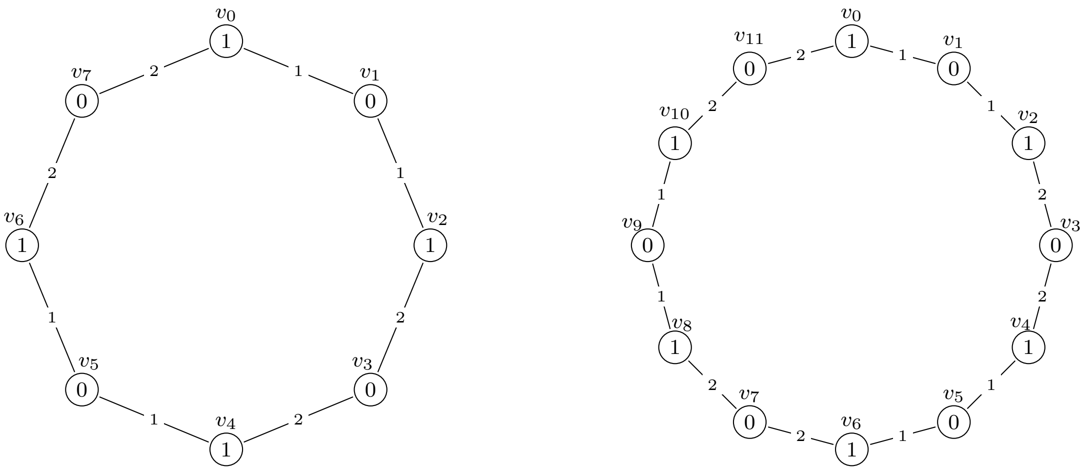
**Theorem 2.2.** *Let $G \cong C_n$, $n \geq 4$. Then, $\chi_E^{g}(G) = 2$ if $n \equiv 0 \pmod 4$, and $\chi_E^{g}(G) = 3$, otherwise.*

*Proof.* Let $G = C_n$, $n \geq 4$ and let $e_i = v_iv_{i+1}$. As stated in Chapter 1, the chromatic number of cycles is $\chi(C_n) = 2$ when $n$ is even, and $\chi(C_n) = 3$, otherwise. Therefore, by Theorem 2.1, in order to prove the result, we have to show that: (i) $G$ admits a gap-[2]-edge-labelling when $n \equiv 0 \pmod 4$; (ii) there is no gap-[2]-edge-labelling of $G$ when $n \equiv 2 \pmod 4$; and (iii) the remaining cases admit a gap-[3]-edge-labelling. Operations on the indices of the vertices are taken modulo $n$.

We prove (i) by showing a gap-[2]-edge-labelling of $G$ when $n \equiv 0 \pmod 4$. Define labelling $\pi$ of $E(G)$ as follows: for every $e_i$, assign $\pi(e_i) = 1$ if $i \equiv 0, 1 \pmod 4$; and $\pi(e_i) = 2$, otherwise. Define colouring $c_\pi$ as usual. In order to prove that $(\pi, c_\pi)$ is a gap-[2]-edge-labelling of $G$, it suffices to show that $c_\pi$ is a proper colouring of $G$.

Let $v_i$ and $v_j$ denote vertices with $i$ odd and $j$ even. Every vertex $v_i$ has $\Pi_{E(v_i)} = \{a\}$, $a \in \{1, 2\}$. Therefore, $c_\pi(v_i) = 0$. For vertices $v_j$, we have $\Pi_{E(v_j)} = \{1, 2\}$ which, in turn, induces $c_\pi(v_j) = 1$. Therefore, $c_\pi(v_l) = (l + 1) \bmod 2$ for every vertex $v_l \in V(G)$, and we conclude that $c_\pi$ is a proper colouring of $G$. Figure 2.4 exemplifies this labelling for cycles $C_8$ and $C_{12}$.

Next, we consider the case $n \equiv 2 \pmod 4$. Suppose $G$ admits a gap-[2]-edge-labelling $(\pi, c_\pi)$. Then, since $G$ is bipartite, we know that colours 0 and 1 alternate on the vertices of $G$. Adjust notation so that $c_\pi(v_i) = i \pmod 2$. Observe that in order to induce colour 0 on vertex $v_0$, $\pi(e_{n-1}) = \pi(e_0) = a$, for some $a \in \{1, 2\}$. Since $c_\pi(v_1) = 1$ and $E(v_1) = \{e_0, e_1\}$, this implies that $\pi(e_0) \neq \pi(e_1)$ and, therefore, $\pi(e_1) = b$, for $b \in \{1, 2\}$, $b \neq a$. For vertex $v_2$, we have $c_\pi(v_2) = 0$ and $E(v_2) = \{e_1, e_2\}$, which implies $\pi(e_2) = \pi(e_1) = b$. Following the vertices in cyclic order, we observe that the sequence $(a, a, b, b)$ repeats itself on every group of four edges $(e_{i-1}, e_i, e_{i+1}, e_{i+2})$, for $i$ even. As $\pi(e_0) = a$ and since $n \equiv 2 \pmod 4$, we have $\pi(e_{n-2}) = \pi(e_{n-1}) = \pi(e_0) = a$ which, in




(a) (b)

Figure 2.4: The gap-[2]-edge-labellings of cycles $C_8$ and $C_{12}$ in (a) and (b), respectively.

turn, induces $c_\pi(v_{n-1}) = 0$. Then $c_\pi(v_{n-1}) = c_\pi(v_0)$, which contradicts the fact that $c_\pi$ is a proper colouring of $G$. This contradiction is illustrated in Figure 2.5. We conclude that $G$ does not admit a gap-[2]-edge-labelling when $n \equiv 2 \pmod 4$.

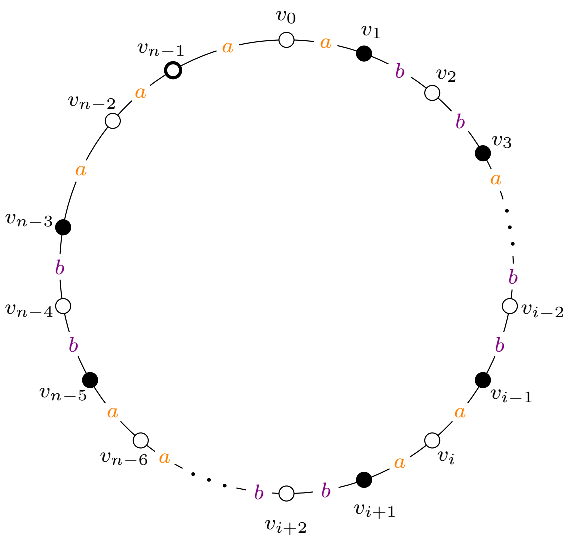



Figure 2.5: Supposing cycle $C_n$ admits a gap-[2]-edge-labelling when $n \equiv 2 \pmod 4$. Vertices coloured in white have $c_\pi(v) = 0$ and in black, $c_\pi(v) = 1$.

In order to complete the proof, it suffices to show gap-[3]-edge-labellings of $G$ for the remaining cases $n \equiv 1, 2, 3 \pmod 4$. Define labelling $\pi$ as follows. First, assign $\pi(e_{n-1}) = 3$. Next, assign $\pi(e_{n-2}) = 2$ if $n \equiv 1 \pmod 4$, and $\pi(e_{n-2}) = 3$, otherwise. Finally, for $0 \leq i \leq n-3$, let $\pi(e_i) = 1$ if $i \equiv 0, 1 \pmod 4$, and $\pi(e_i) = 2$, otherwise. Define colouring $c_\pi$ as usual. Figure 2.6 illustrates $(\pi, c_\pi)$ for cycles $C_5$, $C_6$ and $C_7$, cases where $n \equiv 1, 2, 3 \pmod 4$, respectively.

In order to prove that $(\pi, c_\pi)$ is a gap-[3]-edge-labelling of $G$, it suffices to show that $c_\pi$ is a proper colouring of $G$. First, observe that in all cases, $\Pi_{E(v_0)} = \{1, 3\}$, which induces $c_\pi(v_0) = 2$. For every $1 \leq i \leq n-3$, $i$ odd, $\pi(e_{i-1}) = \pi(e_i)$, inducing $c_\pi(v_i) = 0$.

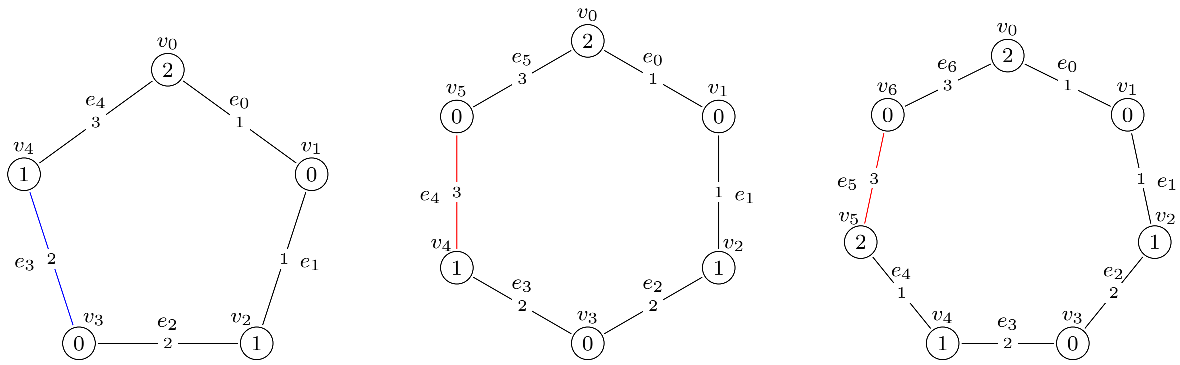



Figure 2.6: The gap-[3]-edge-labellings of cycles $C_5$, $C_6$ and $C_7$ in (a), (b) and (c), respectively. Edge $e_{n-2}$ has been highlighted so as to show the difference between the cases $n \equiv 1 \pmod 4$ (in blue) and $n \equiv 2, 3 \pmod 4$ (in red).

Alternately, for $2 \leq j \leq n-3$, $j$ even, $\Pi_{E(v_j)} = \{1, 2\}$, which induces $c_\pi(v_j) = 1$. For the remaining vertices, $v_{n-2}$ and $v_{n-1}$, we have

$$c_\pi(v_{n-2}) = \begin{cases} 0, & \text{if } n \equiv 1 \pmod 4; \\ 1, & \text{if } n \equiv 2 \pmod 4; \\ 2, & \text{if } n \equiv 3 \pmod 4; \end{cases} \quad \text{and} \quad c_\pi(v_{n-1}) = \begin{cases} 1, & \text{if } n \equiv 1 \pmod 4; \\ 0, & \text{if } n \equiv 2, 3 \pmod 4). \end{cases}$$

By inspection, we conclude that $c_\pi(v_{n-2}) \neq c_\pi(v_{n-1})$. Also, since $c_\pi(v_0) = 2$, $c_\pi(v_0) \neq c_\pi(v_{n-1})$. Now, if $n$ is odd, then $n-3$ is even and we know that $c_\pi(v_{n-3}) = 1$. Moreover, $c_\pi(v_{n-2}) = a$, $a \in \{0, 2\}$. On the other hand, if $n$ is even and, therefore, $n \equiv 2 \pmod 4$, $c_\pi(v_{n-3}) = 0$, and $c_\pi(v_{n-2}) = 1$. In both cases, colouring $c_\pi$ has no two adjacent vertices with the same induced colour and, thus, is a proper colouring of $G$. $\square$

As we have mentioned, in 2016, Brandt et al. [4] also determined $\chi_E^g(C_n)$, constructing the same labelling for cycles. The proof presented in their article uses concepts of the local gap-$k$-colouring of graphs which, for graphs $G$ with $\delta(G) \geq 2$, coincides with gap-$[k]$-edge-labellings.

The case $n \equiv 2 \pmod 4$ shows that the edge-gap number is not always equal to the chromatic number of a graph. Since $d(v) = 2$ for every vertex in $C_n$, we wanted to better understand how vertices of degree one influence the edge-gap number of graphs. For this reason, the next class considered is the family of crown graphs, defined in the next section.

### 2.2.2 Crowns

A *crown* $R_n$ is the graph constructed by taking cycle $C_n$, $n$ copies of the complete graph $K_2$ and identifying each vertex of the cycle with a vertex of a different copy of $K_2$. This construction yields a graph with $2n$ vertices: $n$ vertices of degree 1; and $n$ vertices of degree 3. Let $V(R_n) = \{v_0, \ldots, v_{n-1}\} \cup \{u_0, \ldots, u_{n-1}\}$, where $d(v_i) = 3$ and $d(u_i) = 1$. Figure 2.7 illustrates crown $R_8$. Observe that $\chi(R_n) = \chi(C_n)$ since $C_n \subseteq R_n$ and, thus, one can extend a proper vertex-colouring of cycle $C_n$ to a vertex-colouring of $R_n$ without the use of any additional colours. For this family, the edge-gap number is established in Theorem 2.3.

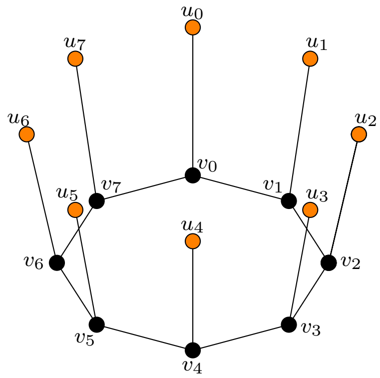



Figure 2.7: Crown $R_8$.

**Theorem 2.3.** *Let $G \cong R_n$, $n \geq 3$. Then, $\chi^g_E(G) = 2$ if $n$ is even, and $\chi^g_E(G) = 3$, otherwise.*

*Proof.* Let $G = R_n$. Since $\chi(R_n) = \chi(C_n)$, in order to prove the result, it suffices to show that crowns admit a gap-[2]-edge-labelling when $n$ is even, and a gap-[3]-edge-labelling, otherwise. Define labelling $\pi$ of $E(G)$ as follows: $\pi(v_iv_{i+1}) = 1, 0 \leq i < n$; $\pi(v_iu_i) = 1 + i \bmod 2$, $0 \leq i \leq n-2$; $\pi(v_{n-1}u_{n-1}) = \chi(R_n)$. Define colouring $c_\pi$ as usual. These labellings are exemplified in Figure 2.8 for crowns $R_8$ and $R_9$. Note that $\pi$ uses label set $\{1, 2\}$ when $n$ is even, and $\{1, 2, 3\}$ when $n$ is odd. Therefore, it remains to show that $c_\pi$ is a proper colouring of $G$.

First, consider vertices $v_0, \ldots, v_{n-2}$ and note that $\Pi_{E(v_i)} = \{1, \pi(v_iu_i)\}$. Therefore, $c_\pi(v_i) = \pi(v_iu_i) - 1$. Since the labels of edges $v_iu_i$ alternate between 1 and 2, with $\pi(v_0u_0) = 1$, we conclude that $c_\pi(v_i)$ alternates between colours 0 and 1, with $c_\pi(v_0) = 0$. Furthermore, since $c_\pi(u_i) = \pi(v_iu_i)$, $c_\pi(u_i)$ alternates between colours 1 and 2, with $c_\pi(u_0) = 1$. We conclude that $c_\pi(v_i) \neq c_\pi(u_i)$ for all $0 \leq i \leq n-2$. Finally, vertices $v_{n-1}$ and $u_{n-1}$ have, respectively, $\Pi_{E(v_{n-1})} = \{1, \chi(R_n)\}$ and $\Pi_{E(u_{n-1})} = \{\chi(R_n)\}$. This, in turn, implies $c_\pi(v_{n-1}) = \chi(R_n) - 1$ and $c_\pi(u_{n-1}) = \chi(R_n)$. Therefore, we have $c_\pi(v_{n-1}) = 1$ and $c_\pi(v_{u-1}) = 2$ when $n$ is even, and $c_\pi(v_{n-1}) = 2$ and $c_\pi(v_{u-1}) = 3$, otherwise. We conclude that $c_\pi$ is a proper vertex-colouring of $G$. □

Recall that cycles $C_n$, when $n \equiv 2 \pmod 4$, do not admit a gap-[2]-edge-labelling. Here, notice that the existence of degree one vertices in the graph enables us to properly label this inner cycle of size $n \equiv 2 \pmod 4$ using only labels 1 and 2. However, the labelling is possible not only because vertices $u_i$ have $d(u_i) = 1$, but because vertices $v_i$ have an extra incident edge (when comparing to cycles), thus allowing the incorporation of another label to $\Pi_{E(v_i)}$. This "extra label" is, in fact, the reason why every crown admits a gap-[$\chi(R_n)$]-edge-labelling, regardless of $n$.

Continuing the previous observations, a natural step is to consider graphs which have universal vertices. Since the degree of the universal vertex can be arbitrarily large, it

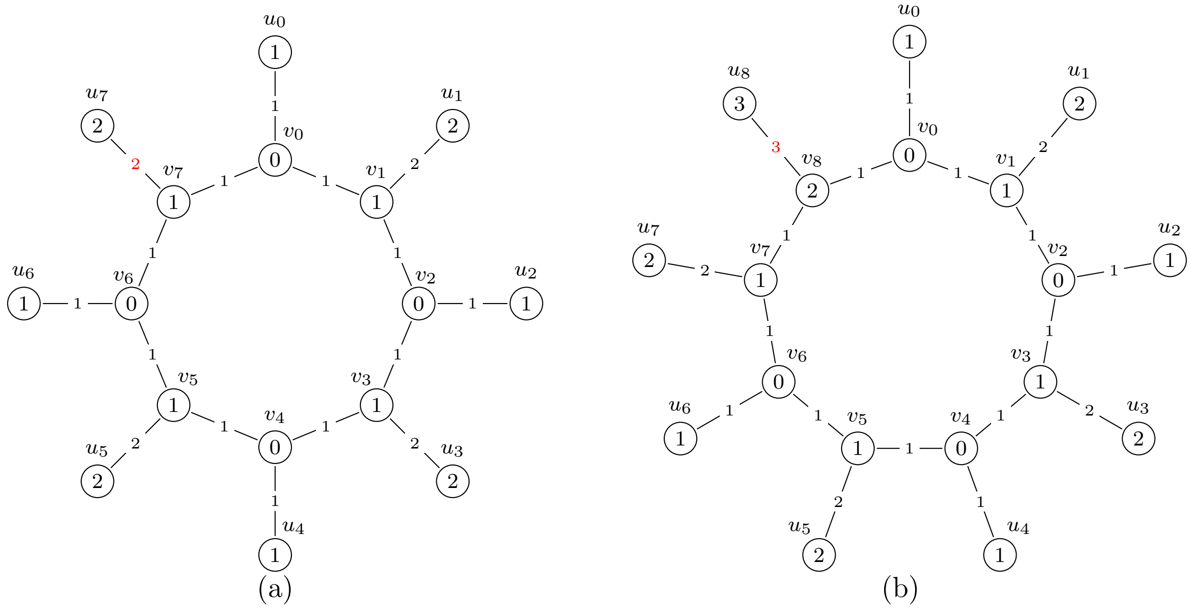



Figure 2.8: The gap-$[\chi(R_n)]$-edge-labellings of crowns: $R_8$ in (a); and $R_9$ in (b).

brings a new perspective to our investigations that, so far, considered only graphs with vertices of low degree. By identifying vertices $u_i$ in crown $R_n$, we obtain a universal vertex. The resulting graph after this operation is the wheel graph, $W_n$, which is defined in the next section.

### 2.2.3 Wheels

As stated in the previous section, *wheel* $W_n$, $n \geq 3$, is the graph obtained by identifying all degree-one vertices $u_i$ in crown $R_n$. This new vertex is called the *central vertex* and is denoted by $v_n$. Figure 2.13(a) illustrates wheel $W_6$.

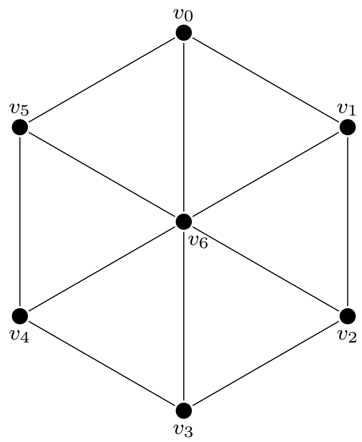


Figure 2.9: Wheel $W_6$ and the notation for the vertex set.

The cycle of length $n$ in wheel $W_n$ is its *rim*. Observe that $\chi(W_n) = \chi(C_n)+1$ since the universal vertex must have a colour different from any other vertex of the rim; assigning a proper vertex-colouring of cycle $C_n$ to the rim of $W_n$ and a new colour to the central vertex yields a proper colouring of $W_n$. Therefore, it follows that $\chi(W_n) = 3$ when $n$ is even, and $\chi(W_n) = 4$, otherwise.

We remark that wheel $W_3$ is isomorphic to complete graph $K_4$, for which Brandt et al. [4] established that $\chi^{\mathrm{g}}_{E}(K_4) = 4$. We exhibit a gap-[4]-edge-labelling of $W_3$ in Figure 2.10. For $n \geq 4$, Theorem 2.4 establishes the edge-gap number for this class.

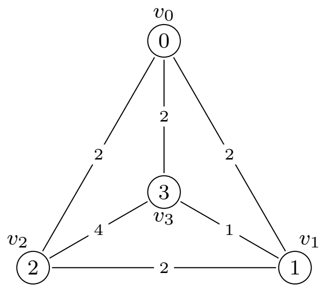



Figure 2.10: The gap-[4]-edge-labelling of $W_3$.

**Theorem 2.4.** *Let $G \cong W_n$, $n \geq 4$. Then, $\chi^{g}_{E}(G) = \chi(G)$.*

*Proof.* Let $G = W_n$, $n \geq 4$, with $V(G) = \{v_0, \ldots, v_n\}$ and $v_n$, the central vertex. Recall that $\chi(G) = 3$ when $n$ is even, and $\chi(G) = 4$, otherwise. Therefore, by Theorem 2.1, it suffices to show a gap-[$\chi(G)$]-edge-labelling of $G$.

We begin considering $n \geq 5$ and odd. We define labelling $\pi$ of $E(G)$ as follows: $\pi(v_iv_{i+1}) = 3 - i \bmod 2$, $1 \leq i \leq n-3$; $\pi(v_iv_n) = 1 + i \bmod 2$, $0 \leq i \leq n-3$; the remaining edges, $v_{n-2}v_{n-1}, v_{n-1}v_0, v_0v_1, v_{n-2}v_n, v_{n-1}v_n$, receive labels $3, 1, 1, 4, 1$, respectively. Define colouring $c_\pi$ as usual. This gap-[4]-edge-labelling $(\pi, c_\pi)$ is presented for wheels $W_5$ and $W_7$ in Figure 2.11.

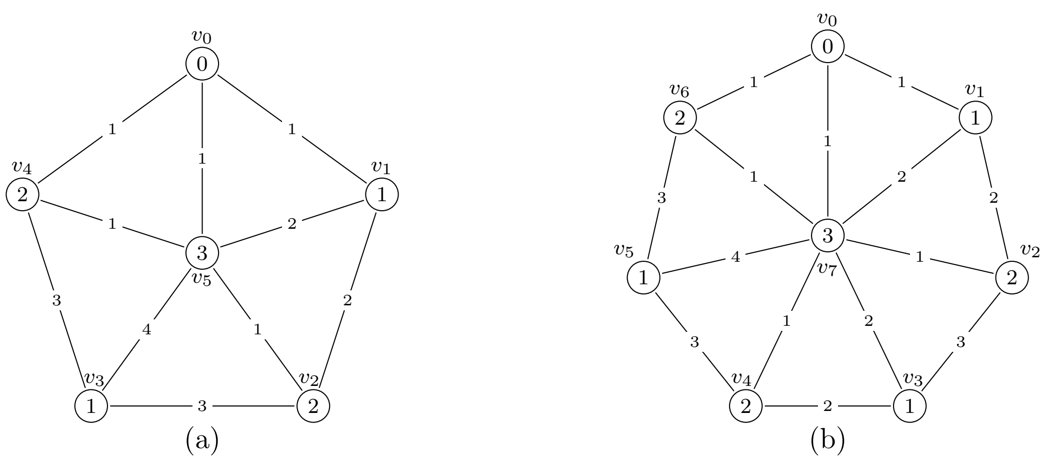



Figure 2.11: The gap-[4]-edge-labellings of wheels $W_5$ and $W_7$ in (a) and (b), respectively.

First, observe that labelling $\pi$ uses label set $\{1, 2, 3, 4\}$. Also, note that $\{1, 4\} \subset \Pi_{E(v_n)}$. This implies that $c_\pi(v_n) = 3$. Next, consider vertices $v_i$, $1 \leq i \leq n-3$. Observe that $\Pi_{E(v_1)} = \{1, 2\}$, $\Pi_{E(v_i)} = \{2, 3\}$ when $i$ is odd, and $\Pi_{E(v_i)} = \{1, 2, 3\}$, otherwise. This implies that $c_\pi(v_i) = 2 - i \bmod 2$, $0 \leq i \leq n-3$. For the remaining vertices $v_0$, $v_{n-2}$ and $v_{n-1}$, we have $\Pi_{E(v_0)} = \{1\}$, $\Pi_{E(v_{n-1})} = \{3, 4\}$ and $\Pi_{E(v_{n-2})} = \{1, 3\}$. This induces colours $c_\pi(v_0) = 0$, $c_\pi(v_{n-1}) = 1$ and $c_\pi(v_{n-1}) = 2$, respectively. We conclude that $(\pi, c_\pi)$ is a gap-[4]-edge-labelling of $G$ in this case.

It remains to consider the case where $n$ is even, for which it suffices to show that $G$ admits a gap-[3]-edge-labelling. For $W_4$, Figure 2.12 exhibits a gap-[3]-edge-labelling. By inspection, we conclude that $c_\pi$ is a proper colouring of that graph.

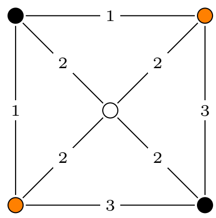
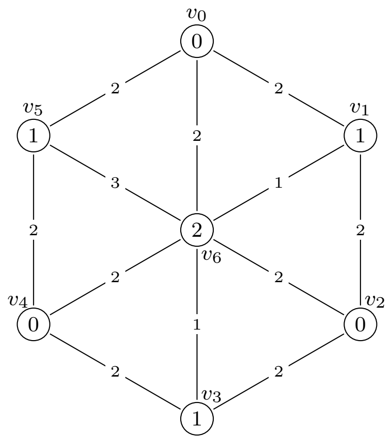
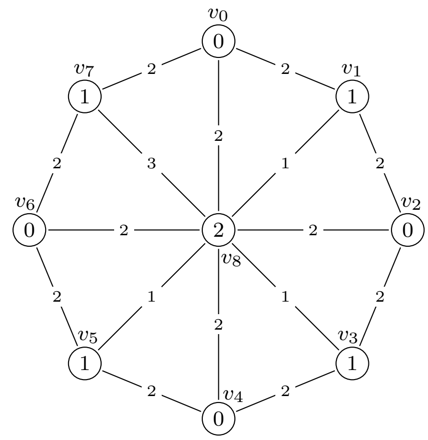



Figure 2.12: The gap-[3]-edge-labelling of wheel $W_4$. Vertices in black have induced colour 1, in orange, colour 2 and the central vertex in white, colour 0.

For $n \geq 6$ and even, define labelling $\pi$ as follows: $\pi(v_iv_{i+1}) = 2, 0 \leq i \leq n-2$; $\pi(v_iv_n) = 2 - i \bmod 2$, $0 \leq i \leq n-2$; $\pi(v_{n-1}v_n) = 3$. Colouring $c_\pi$ is defined as usual. Figure 2.13 illustrates the gap-[3]-edge-labelling of wheels $W_6$ and $W_8$.




(a)




(b)

Figure 2.13: The gap-[3]-edge-labellings of wheels $W_6$ and $W_8$ in (a) and (b), respectively.

Since labelling $\pi$ uses label set $\{1, 2, 3\}$, it suffices to show that $c_\pi$ is a proper colouring of $G$ in this case. First, observe that $v_{n-1}$ has $\Pi_{E(v_{n-1})} = \{2, 3\}$. Therefore, $c_\pi(v_{n-1}) = 1$. Now, for $0 \leq i \leq n-2$ and even, note that $\Pi_{E(v_i)} = \{2\}$, which implies $c_\pi(v_i) = 0$. On the other hand, for $1 \leq i \leq n-3$ and odd, we have $\Pi_{E(v_i)} = \{1, 2\}$, which induces $c_\pi(v_i) = 1$. Finally, since central vertex $v_n$ has $\{1, 3\} \subset \Pi_{E(v_n)}$, $c_\pi(v_n) = 2$. Therefore, the central vertex has colour 2 and vertices $v_i$ in $G$ alternate colours 0, 1 along the rim. We conclude that $c_\pi$ is a proper colouring of $G$, and the result follows. $\square$

As mentioned in Section 2.1, both Scheidweiler and Triesch [26] and Brandt et al. [4] studied versions of this proper labelling for the family of trees. Inspired by their results – and motivated by our work on cycles and crowns – we investigate the edge-gap number for the family of unicyclic graphs, which is defined in the next section.

### 2.2.4 Unicyclic graphs

A *unicyclic graph* is a connected simple graph $G = (V, E)$ with $|V| = |E|$. Note that $G$ contains a single cycle. This family includes the family of cycles and crown graphs. However, instead of having vertices of degree one adjacent to the vertices of the cycle, which is the case of crowns, a unicyclic graph allows the existence of an entire tree rooted at each vertex $v_i$ of the cycle. Figure 2.14 illustrates a unicyclic graph. In this example, the cycle has red edges and its topmost vertices are roots of two nontrivial trees.

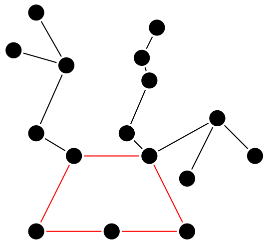


Figure 2.14: An example of a unicyclic graph.

We denote the vertices of the (single) cycle, $C_p$, of $G$ by $v_0, \ldots, v_{p-1}$. We denote $T_i$ the tree *rooted* at $v_i$ with $E(T_i) \cap E(C_p) = \emptyset$. Now, let $v_i$ be an arbitrary vertex of cycle $C_p$. A *leaf* of $T_i$ is a vertex $w \in V(T_i)$ such that $d(w) = 1$. An *internal vertex* of tree $T_i$ is a node that is neither the root nor a leaf of $T_i$. For every leaf $w_j \in V(T_i)$, $w_j \neq v_i$, the *branch* $B_i^{w_j}$ is defined by the path $v_i, \ldots, w_j$. The length of this path is denoted by $\text{dist}(B_i^{w_j})$. Figure 2.15 illustrates this notation for a vertex $v_i$ in $G$. In this example, $\text{dist}(B_i^{w_1}) = \text{dist}(B_i^{w_4}) = 2$, while $\text{dist}(B_i^{w_2}) = 1$ and $\text{dist}(B_i^{w_3}) = 3$.

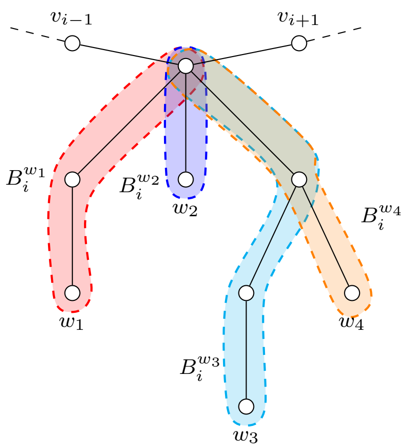



Figure 2.15: A tree $T_i$ from a unicyclic graph $G$ with 4 branches.

Observe that a unicyclic graph $G$ is bipartite if and only if cycle $C_p$ has even size. Therefore, by Theorem 2.1, we know that $\chi_E^{\mathrm{g}}(G) \in \{2, 3\}$ when $p$ is even, and $\chi_E^{\mathrm{g}}(G) \in \{3, 4\}$, otherwise. We determine the edge-gap number of unicyclic graphs with $p$ odd in Theorem 2.5.

**Theorem 2.5.** *Let $G$ be a unicyclic graph and $p$, the size of the cycle in $G$. If $p$ is odd, then $\chi_E^{g}(G) = 3$.*

*Proof.* Let $G = (V, E)$ be a unicyclic graph with a cycle of odd size $p$. Let $v_0, v_1, \ldots, v_{p-1}$ denote the vertices of the cycle, and $T_0, T_1, \ldots, T_{p-1}$ their respective disjoint rooted trees. Note that if $T_i$ is a trivial graph for every $0 \leq i < p$, then $G \cong C_p$, for which the edge-gap number is established in Theorem 2.2. Therefore, for the remainder of the proof, we can safely assume that there exists at least one tree $T_i$ with $|V(T_i)| \geq 2$. Adjust notation so that $v_0$ is the root of a nontrivial tree.

In order to prove the result, it is sufficient to show that $G$ admits a gap-[3]-edge-labelling $(\pi, c_\pi)$ since $\chi(G) = 3$. Define labelling $\pi$ as follows. For every $v_i \in V(C_p)$, let

$$\pi(v_i v_{i+1}) = \begin{cases} 3, & \text{if } i = p-1; \\ 2, & \text{if } i \equiv 0, 1 \pmod 4; \\ 1, & \text{otherwise.} \end{cases}$$

If $p \equiv 1 \pmod 4$, assign label 2 to edges $v_0u \in E(T_0)$ and label 1 to every edge $v_{n-1}u \in E(T_{n-1})$, when they exist. Otherwise, if $p \equiv 3 \pmod 4$, assign $\pi(v_0u) = 1$ and $\pi(v_{n-1}u) = 2$. For the remaining edges $v_iu \in E(T_i)$, when they exist, assign label 1 if $i \equiv 3 \pmod 4$, and label 2, otherwise. This labelling is sketched in Figure 2.16 for unicyclic graphs with cycles $C_7$ and $C_9$ as subgraphs.

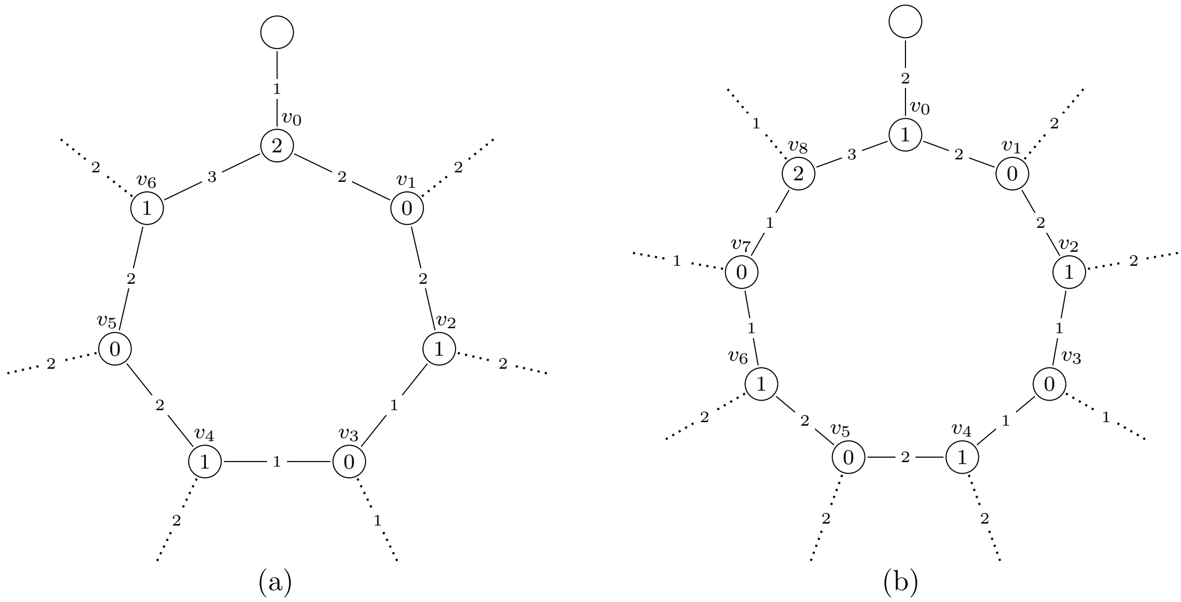


Figure 2.16: Partial labellings for cycles of unicyclic graphs: $C_7$ in (a); and $C_9$ in (b). The dotted edges sketch edges $v_iu \in E(T_i)$, that may not exist.

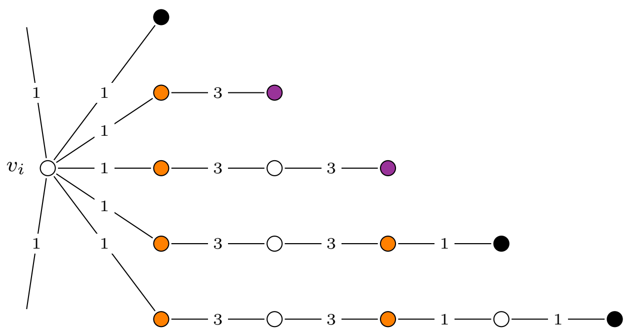
Define colouring $c_\pi$ for the vertices of $C_p$ as usual since all the edges incident with vertices $v_i \in V(C_p)$ have already been assigned a label. Note that every vertex $v_i \in V(C_p)$, $1 \leq i \leq p-2$ and odd, has $\Pi_{E(v_i)} = \{a\}$, for $a \in \{1, 2\}$. This induces colour $c_\pi(v_i) = 0$ in these vertices. Furthermore, every $v_j \in V(C_p)$, $2 \leq j \leq p-3$ and even, has $\Pi_{E(v_j)} = \{1, 2\}$, thus, inducing colour $c_\pi(v_j) = 1$. For vertex $v_{p-1}$, note that $\Pi_{E(v_{p-1})} = \{1, 3\}$ if $p \equiv 1 \pmod 4$ and $\Pi_{E(v_{p-1})} = \{2, 3\}$, otherwise. This induces colours $c_\pi(v_{p-1}) = 2$ and $c_\pi(v_{p-1}) = 1$, respectively. Finally, the edges incident with vertex $v_0$ have been labelled such that $\Pi_{E(v_0)} = \{2, 3\}$ when $p \equiv 1 \pmod 4$, and $\Pi_{E(v_0)} = \{1, 2, 3\}$, otherwise, inducing colours $c_\pi(v_0) = 1$ and $c_\pi(v_0) = 2$, respectively. Since no two adjacent vertices in the cycle have the same induced colour, we conclude that $c_\pi$ is a proper colouring of $C_p \subset G$.

Next, we assign label to trees $T_i$ of $G$. For every vertex $v_i$ in $C_p$, $B_i^w = v_i, u_1, u_2, \ldots, w$ denotes the branches connecting $v_i$ and leaves $w \in V(T_i)$. Also, denote $u_0 = v_i$ and $u_{\mathrm{dist}(B_i^w)} = w$. We label the remaining edges of trees $T_i$ depending on the induced colour of vertices $v_i$.

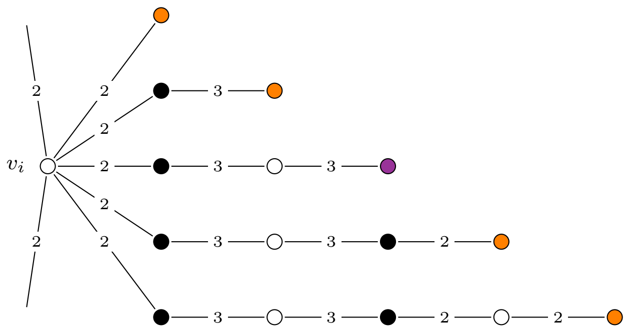
*Case 1.* $c_\pi(v_i) = 0$
In this case, observe that $\Pi_{E(v_i)} = \{a\}$, $a \in \{1, 2\}$. For every edge $u_j u_{j+1}$, $1 \leq j < \mathrm{dist}(B_i^w)$, let:

$$\pi(u_j u_{j+1}) = \begin{cases} a, & \text{if } j \equiv 0, 3 \pmod 4; \\ 3, & \text{otherwise.} \end{cases}$$

This labelling and its induced colouring are illustrated in figures 2.17(a) and 2.17(b) for cases $a = 1$ and $a = 2$, respectively.




(a) Labelling of tree $T_i$, when $a = 1$.




(b) Labelling of tree $T_i$, when $a = 2$.

Figure 2.17: Partial labellings of trees $T_i$ when $c_\pi(v_i) = 0$. White, black, orange and violet vertices have induced colours 0, 1, 2 and 3, respectively.

First, note that $v_i = u_0$ has its (previously defined) colour preserved since the labels assigned to $E(B_i^w)$ do not alter set $\Pi_{E(v_i)}$. Next, consider internal vertices $u_i$, $1 \leq i <$ dist$(B_i^w)$. Note that $\Pi_{E(u_i)} = \{a, 3\}$ if $i$ is odd. This implies that $c_\pi(u_i) = 2$ if $a = 1$, and $c_\pi(u_i) = 1$, otherwise. Now, if $i$ is even, then $\Pi_{E(u_i)} = \{b\}$, $b \in \{a, 3\}$, which implies that $c_\pi(u_i) = 0$. Therefore, colours $3 - a$ and 0 alternate in the internal vertices of every branch $B_i^w$, starting with $c_\pi(u_1) = 3 - a$. Since $a \in \{1, 2\}$, $3 - a \neq 0$.

We conclude that there are no adjacent internal vertices with the same colour. Moreover, $c_\pi(u_0) = 0 \neq c_\pi(u_1)$.

Next, we consider the leaves of $T_i$. Recall that a leaf $w$ has its colour induced by the label of its incident edge. Therefore, $c_\pi(w) \in \{3, a\}$. Let $u$ be the neighbour of $w$. As previously defined, $c_\pi(u) \in \{0, 3-a\}$. This implies that $c_\pi(w) \neq c_\pi(u)$ since $a \neq 0$ and $a \neq 3-a$. We conclude that $c_\pi$ is a proper vertex-colouring of the tree.

*Case 2.* $c_\pi(v_i) = 1$
In this case, note that $c_\pi(v_i)$ is induced by $\Pi_{E(v_i)} = \{2, a\}$, where $a \in \{1, 3\}$ is the label assigned to an edge of cycle $C_p$. Also, recall that every edge $v_i u \in E(T_i)$ receives label 2. Assign labels to every edge $u_j u_{j+1}$ in branch $B_i^w$ of $T_i$, $1 \leq j < \text{dist}(B_i^w)$, as follows:

$$\pi(u_j u_{j+1}) = \begin{cases} 2, & \text{if } j \equiv 0, 1 \pmod 4; \\ 3, & \text{otherwise.} \end{cases}$$

Figure 2.18 illustrates this case. Note that $\pi(v_i u_1) = 2$ and, therefore, if $u_1$ is a leaf of $T_i$, then $c_\pi(v_i) \neq c_\pi(u_i)$. Next, observe that odd-index internal vertices $u_j$, $1 \leq j < \text{dist}(B_i^w)$, have $\Pi_{E(u_j)} = \{a\}$, $a \in \{2, 3\}$, while when $j$ is even, $\Pi_{E(u_j)} = \{2, 3\}$. This implies that induced colours 0 and 1 alternate along the internal vertices of every branch $B_i^w$ of $T_i$, with $c_\pi(u_1) = 0$. Furthermore, since only labels 2 and 3 are assigned to edges in $T_i$, $c_\pi(w) \in \{2, 3\}$ for every leaf $w \in V(T_i)$. We conclude that there are no adjacent vertices in $T_i$ with conflicting colours.

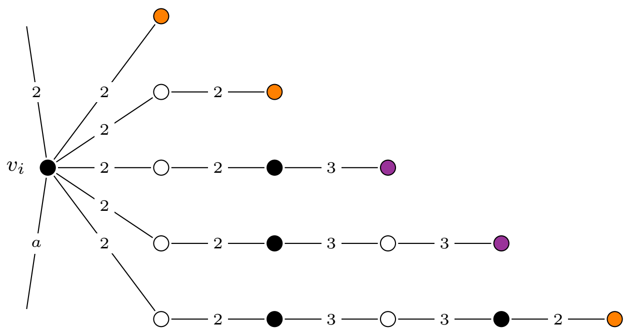



Figure 2.18: Partial labelling of $T_i$ when $c_\pi(v_i) = 1$. Note that label $a \in \{1, 3\}$ assigned to the edge of $C_p$ induces colour 1 in $v_i$. White, black, orange and violet vertices have induced colours 0, 1, 2 and 3, respectively.

*Case 3.* $c_\pi(v_i) = 2$
This case only occurs on vertex $v_{n-1}$ when $n \equiv 1 \pmod 4$, and on vertex $v_0$, when $n \equiv 3 \pmod 4$. It is important to remark that every edge $v_i u \in E(T_i)$ receives label 1. Once again, we assign labels to edges $u_j u_{j+1}$, $1 \leq j < \text{dist}(B_i^w)$, of each branch $B_i^w$ as follows:

$$\pi(u_j u_{j+1}) = \begin{cases} 2, & \text{if } j \equiv 1, 2 \pmod 4; \\ 3, & \text{otherwise.} \end{cases}$$

Figure 2.19 illustrates this case. Consider vertices $u_1$ in $T_i$. If $u_1$ is a leaf, then $c_\pi(u_1) = 1 \neq c_\pi(v_i)$. Otherwise, $\Pi_{E(u_1)} = \{1, 2\}$, which also induces colour 1. Now, consider internal vertices $u_j$, $2 \leq j < \mathrm{dist}(B_i^w)$. If $j$ is even, then $\Pi_{E(u_j)} = \{a\}$, $a \in \{2,3\}$. This induces $c_\pi(u_j) = 0$. On the other hand, if $j$ is odd, then $\Pi_{E(u_j)} = \{2,3\}$, inducing colour 1. We conclude that every internal vertex $u_j$ has $c_\pi(u_j) = j \bmod 2$.

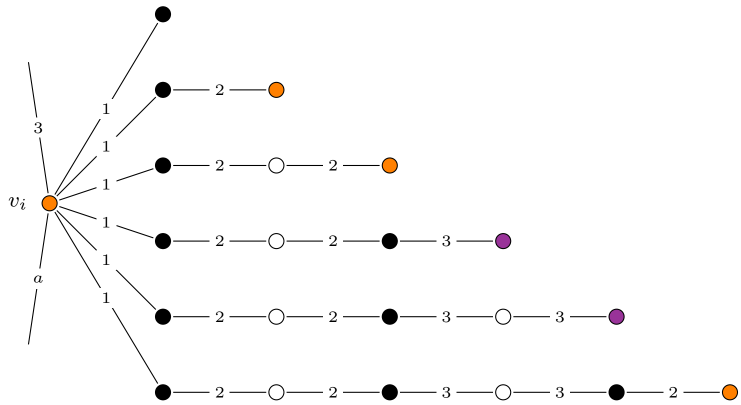


Figure 2.19: Partial labelling of tree $T_i$ when $c_\pi(v_i) = 2$. Note that label $a \in \{1,2\}$, assigned to the edge of $C_p$, does not influence the induced colour of $v_i$. White, black, orange and violet vertices have induced colours 0, 1, 2 and 3, respectively.

In order to conclude this case – and, hence, the proof – observe that the leaves have incident edges labelled with either 2 or 3; this implies that $c_\pi(w) \in \{2,3\}$ for every leaf $w \in T_i$. Therefore, no adjacent vertices have the same induced colour, and $c_\pi$ is a proper colouring of tree $T_i$ in this case. □

The different labellings of trees $T_i$ in the proof of Theorem 2.5 are inspired by the gap-[3]-edge-labellings of trees designed by Scheidweiler and Triesch [26]. In their article, they investigate bounds of the edge-gap number for several families, including trees. Moreover, they showed that there are trees that do not admit gap-[2]-edge-labellings by presenting a counterexample, replicated in Figure 2.20.

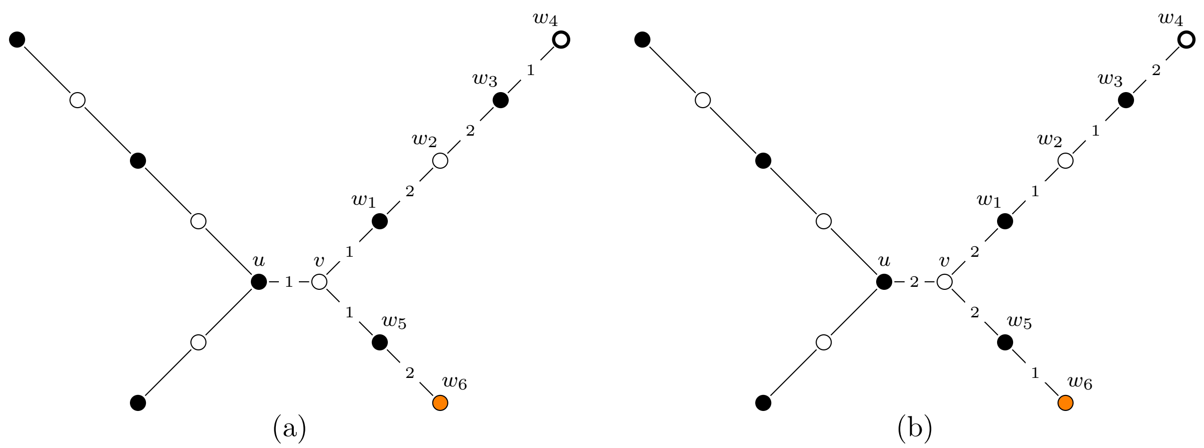


Figure 2.20: The tree presented by Scheidweiler and Triesch [26].

Let us explain Scheidweiler and Triesch's counterexample. Suppose this graph admits a gap-[2]-edge-labelling $(\pi, c_\pi)$. Since every internal vertex $x$ of the tree has $d(x) \geq 2$, we know that $c_\pi(x) \in \{0,1\}$. Thus, we can assume, by the symmetries of the tree, that one end of edge $uv$ has induced colour 0. Furthermore, we know that colours 0 and 1 alternate in the internal vertices of the rightmost branches of figures 2.20(a) and 2.20(b).

Let $c_\pi(v) = 0$. This implies that every edge incident with $v$ receives the same label $a \in \{1, 2\}$. First, suppose $a = 1$, as illustrated in Figure 2.20(a). Since $c_\pi(w_1) = 1$, we know edge $w_1w_2$ is labelled with 2. This, in turn, implies that $\pi(w_2w_3) = 2$ since $c_\pi(w_2) = 0$. Lastly, given that $c_\pi(w_3) = 1$, edge $w_3w_4$ is labelled with 1. However, since $d(w_4) = 1$, $c_\pi(w_4)$ is defined by the label of its incident edge, which received label 1. Then, $c_\pi(w_4) = 1 = c_\pi(w_3)$, which is a contradiction.

Thus, we conclude that $a = 2$, as can be seen in Figure 2.20(b). However, if this is the case, note that an analogous reasoning can be applied to the branch containing vertices $w_5$ and $w_6$. In order to induce $c_\pi(w_5) = 1$, we have $\pi(w_5w_6) = 1$ since $\pi(vw_5) = 2$. This also induces colour 1 on vertex $w_6$, which is impossible. Therefore, there is no gap-[2]-edge-labelling for this graph.

The counterexample presented by Scheidweiler and Triesch [26] shows that there are trees that do not admit gap-[2]-edge-labellings. As an extension of Scheidweiler and Triesch's result, we show that there also exists bipartite unicyclic graphs which do not admit a gap-[2]-edge-labelling. Consider unicyclic graphs $G_1$ and $G_2$ in Figure 2.21. Both graphs have even cycles and, consequently, $\chi(G_1) = \chi(G_2) = 2$. In Figure 2.21(a), we present a gap-[2]-edge-labelling of $G_1$.

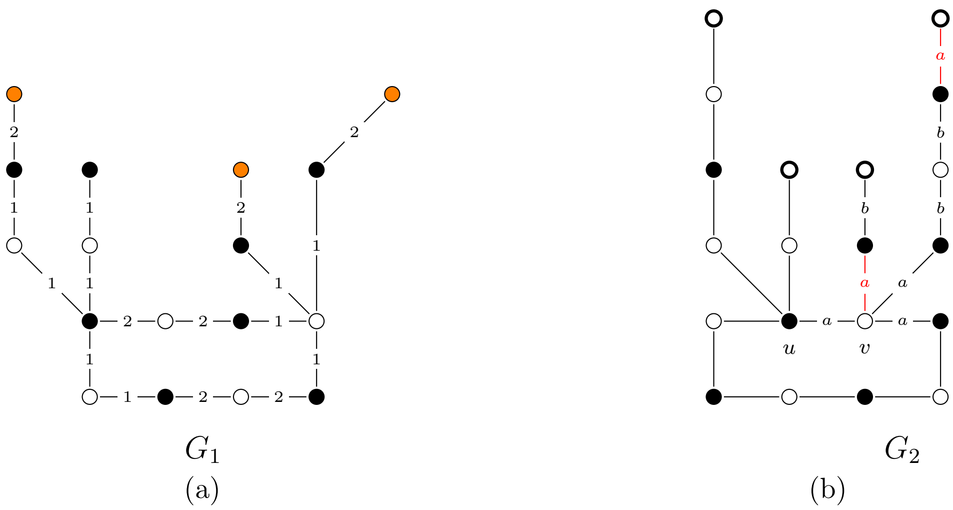



Figure 2.21: Two unicyclic graphs $G_1$ and $G_2$ in (a) and (b), respectively.

Consider graph $G_2$ in Figure 2.21(b). Suppose this graph admits a gap-[2]-edge-labelling. Then, since $G_2$ is bipartite and every vertex $v_i$ in cycle $C_p \subset G$ has $d(v_i) \geq 2$, we know that colours 0 and 1 alternate in the vertices of the cycle. Then, one of trees $T_u, T_v$ has its root coloured with 0 and the other, with 1. In Figure 2.21(b), $c_\pi(u) = 1$ and we present a sketch of the labels assigned to the edges of $T_v$. By inspecting the drawing, it is possible to conclude that the same reasoning used by Scheidweiler and Triesch [26] can be extended to this graph, which leads us to conclude that $G_2$ does not admit a gap-[2]-edge-labelling.

There is still work to be done regarding the gap-[$k$]-edge-labellings of unicyclic graphs with $p \equiv 0 \pmod 2$. In particular, characterizing which unicyclic graphs with even cycles admit a gap-[2]-edge-labelling is an interesting open problem

**Problem 2.6.** *Determine the edge-gap number for unicyclic graphs with a cycle of even size.*

We close this chapter presenting our results for gap-$[k]$-edge-labellings of some families of snarks, which are defined in the following section.

### 2.2.5 Snarks

A *snark* is a bridgeless, cubic graph with chromatic index four without parallel edges or cycles of length three. The search for such a graph was motivated by the Four-Colour Problem, described in Chapter 1. To recall, this problem states that every planar map admits a colouring of its regions such that no two neighbouring regions receive the same colour. Out of the many attempts to solve this problem, P. G. Tait [28, 29] showed, in 1880, that it could be reduced to an edge-colouring problem. He remarked that if a bridgeless cubic graph with chromatic index four was discovered, with the additional property of being planar, then the answer to the Four-Colour Problem would be "no". On the other hand, a proof that every such graph is not planar would result in a positive answer. Therefore, his work provided another way to approach the Four-Colour Problem and motivated the search for non-3-edge-colourable bridgeless cubic graphs. The first discoveries of these graphs, however, were very sporadic and became a challenge for researchers.

In light of this, M. Gardner [12] proposed to call these graphs "snarks" in 1976. He was inspired by the poem *The Hunting of the Snark*, which describes a crew's struggled journey in search of a fantastic, rare creature named Snark. The first snark was discovered by Petersen in 1898 [22] and is known as the Petersen Graph. We exhibit in Figure 2.22 four different representations of the Petersen Graph, the most common of which is the one in Figure 2.22(b).

In this section, we establish the edge-gap number for the families of Blanuša, Flower, Goldberg and Twisted Goldberg snarks. Although the labelling presented for each of these families is distinct, they are all based on the same idea. Each family of infinite snarks is constructed by using subgraphs as "building blocks", which are connected by edges. Our technique for establishing the edge-gap number for these graphs is to assign labels to the edges in each block and to the edges that connect them, such that the labelling of the whole graph induces a proper colouring. This same framework was also used for the vertex version of this labelling, which is presented in Section 3.3.6.

#### Blanuša Snarks

The first infinite family of snarks we present are the Blanuša Snarks. The *First Blanuša Snark*, denoted by $B_1^1$, was discovered by Blanuša [3] in 1964. It has 18 vertices and two of its drawings are presented in Figure 2.23. This graph is obtained from two copies of the Petersen Graph, an observation made quite clear when observing Figure 2.23(b). Later, a different modification in the copies of the Petersen Graph yielded the *Second Blanuša Snark*, denoted by $B_1^2$. An illustration of this graph is presented in Figure 2.24.

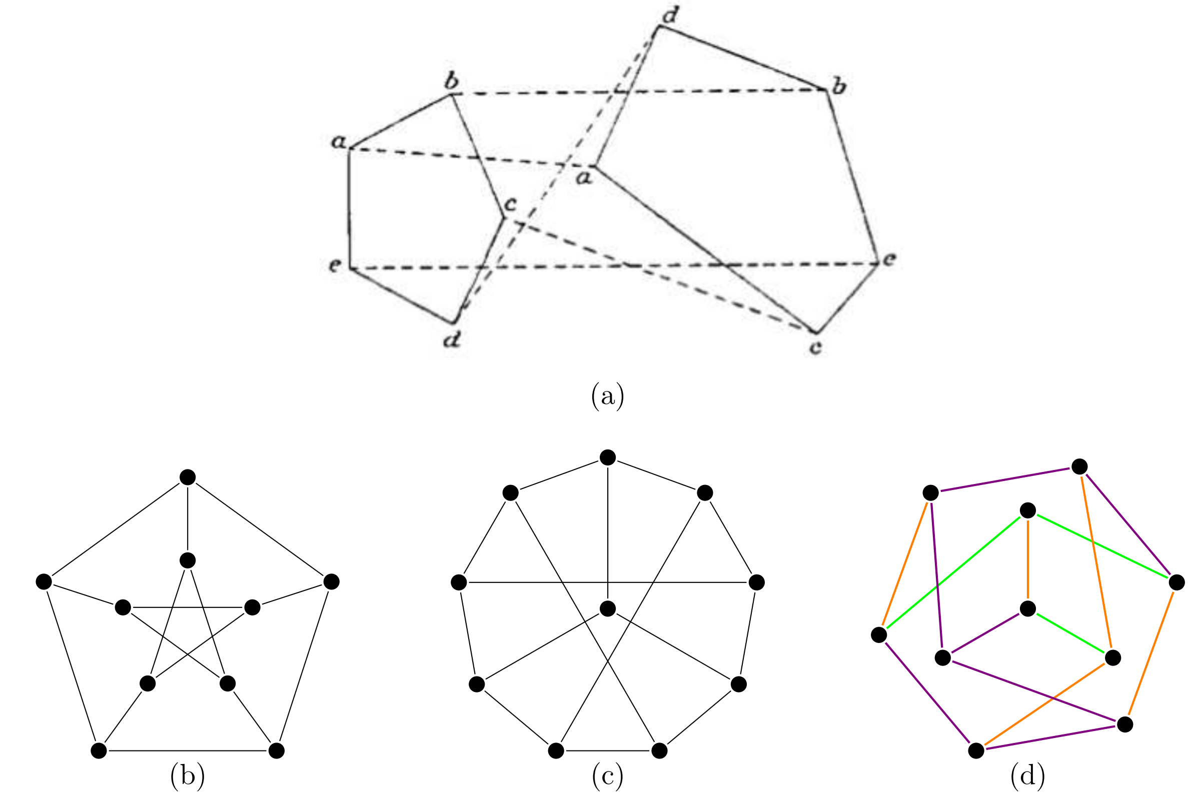



Figure 2.22: Representations of the Petersen Graph. In (a), the original drawing from Petersen's notes. In (d) is illustrated a proper 4-edge-colouring.

J. J. Watkins [30, 31] generalised the construction of First and Second Blanuša Snarks, defining two infinite families, which are referred to as *Generalised Blanuša Snarks*. Let $\mathcal{B}^1 = \{B_1^1, B_2^1, B_3^1, \ldots\}$ denote the family of Generalised First Blanuša Snarks. In what follows, we describe the construction of graph $B_i^1$. The construction of Second Blanuša Snarks is described further in the section.

Let $B_0^1$ and $B$ be the graphs in Figures 2.25(a) and 2.25(b), respectively. We refer to these graphs as *blocks*. The Generalised First Blanuša Snark, $B_i^1$, uses a copy of $B_0^1$ and $i \geq 1$ copies of graph $B$. Let $B_j$ denoted the $j$-th copy of block $B$ in $B_i^1$. These components are connected by, first, adding edges $u_0w_1$ and $v_0z_1$, thus connecting blocks $B_0^1$ and $B_1$. Next, we connect block $B_j$ to $B_{j+1}$ by adding edges $t_jw_{j+1}$ and $r_jz_{j+1}$, for $1 \leq j \leq i-1$. Finally, edges $t_ix_0$ and $r_iy_0$ are added. This construction is presented for the Generalised

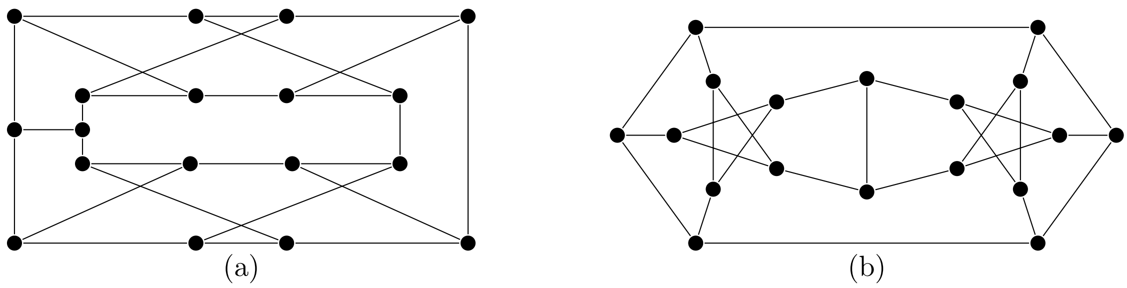



Figure 2.23: Drawings of the First Blanuša Snark.

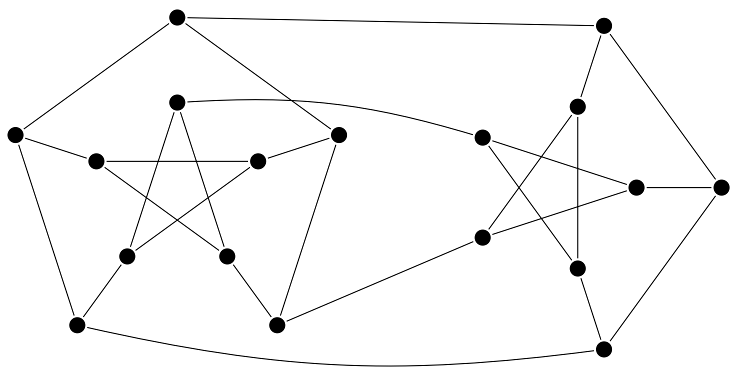


Figure 2.24: Second Blanuša Snark $B_1^2$.

First Blanuša Snark $B_i^1$ in Figure 2.26.

When the family of Generalised First Blanuša Snarks was introduced, Watkins [30, 31] demonstrated that the chromatic number of each graph in this family is $\chi(B_i^1) = 3$. With this result in mind, we present the edge-gap number for this first family in Theorem 2.7.

**Theorem 2.7.** *Let $G$ be a Generalised First Blanuša Snark. Then, $\chi_E^g(G) = 3$.*

*Proof.* Let $G \cong B_i^1$, with $B_0^1$ and $B$, the blocks used in its construction. In order to prove the result, by Theorem 2.1, it is sufficient to show that $G$ admits a gap-[3]-edge-labelling since $\chi(G) = 3$.

Define labelling $\pi$ of $G$ as follows: for block $B_0^1$, assign labels to the edges according to Figure 2.27(a); and for every block $B_j$, $1 \leq j \leq i$, label $E(B_j)$ according to Figure 2.27(b). For the edges connecting adjacent blocks, let $\pi(v_0 z_1) = 1$ and assign label 2 to every remaining edge. Colouring $c_\pi$ is defined as usual.

In order to complete the proof, we show that $c_\pi$ is a proper colouring of $G$. First, consider block $B_0^1$, starting with vertex $y_0$. Observe that $\{1, 3\} \subset \Pi_{E(y_0)}$ and, therefore, $c_\pi(y_0) = 2$. Next, observe that vertices $x_0$ and $u_0$ have $\Pi_{E(x_0)} = \{1, 2\}$ and $\Pi_{E(u_0)} = \{2, 3\}$.

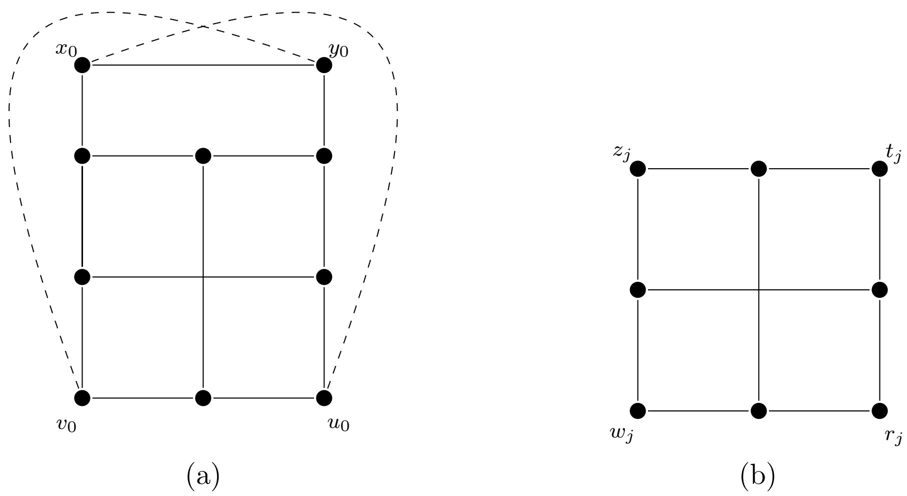



Figure 2.25: In (a), the first block, $B_0^1$, used in the construction $B_i^1$; and in (b), the iterating block $B$.

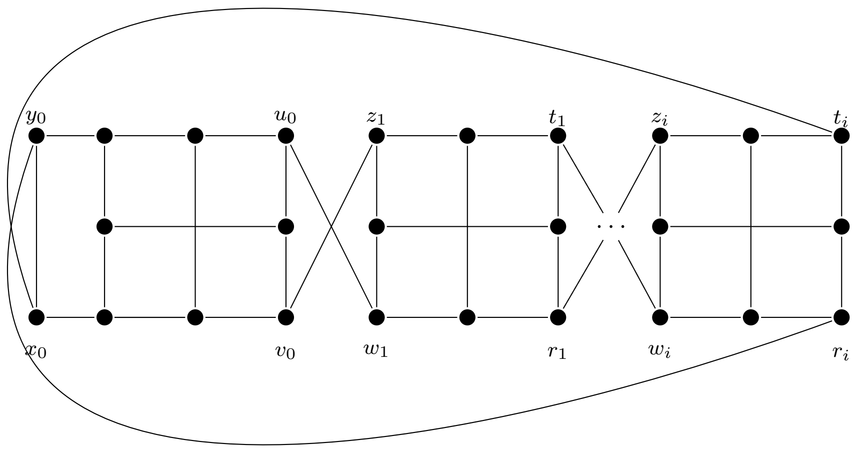


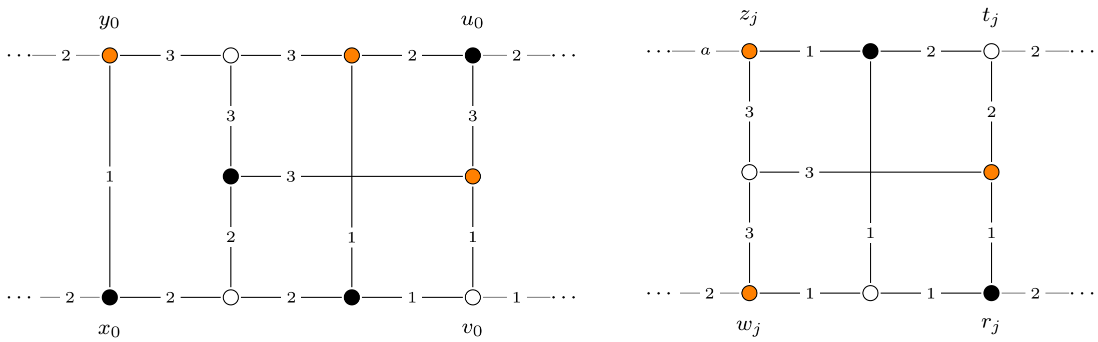

Figure 2.26: A sketch of the construction of $B_i^1$, $i \geq 3$.

This implies that $c_\pi(x_0) = c_\pi(u_0) = 1$. For vertex $v_0$, recall that edge $v_0z_1$ receives label 1. Therefore, $\Pi_{E(v_0)} = \{1\}$, which induces $c_\pi(v_0) = 0$. The remaining *internal* vertices of block $B_0^1$ have their respective induced colours exhibited in Figure 2.27(a). By inspection, we conclude that labelling $\pi$ induces a proper colouring of $B_0^1$.

It remains to consider the induced colouring of blocks $B_j$, $1 \leq j \leq i$. We start by analysing vertices $z_j$ and $w_j$. Note that both vertices have incident edges which receive labels 1 and 3. This implies that $z_j$ and $w_j$ have induced colour 2. Next, observe that $\Pi_{E(t_j)} = \{2\}$ and $\Pi_{E(r_j)} = \{1, 2\}$, inducing colours 0 and 1 in vertices $t_j$ and $r_j$, respectively. Furthermore, note that $c_\pi(t_j) \neq c_\pi(w_j)$ and $c_\pi(r_j) \neq c_\pi(z_j)$, which implies that distinct blocks $B_j$ do not have adjacent vertices with conflicting colours. Finally, note that $c_\pi(y_0) \neq c_\pi(r_i)$ and $c_\pi(x_0) \neq c_\pi(t_i)$. Therefore, $c_\pi$ is a proper colouring of $G$, and the result follows. In Figure 2.28, we illustrate $(\pi, c_\pi)$ for $B_3^1$. $\square$

As previously mentioned, the family of Generalised Second Blanuša Snarks $\mathcal{B}^2 = \{B_1^2, B_2^2, B_3^2, \ldots\}$ is created by replacing block $B_0^1$ with block $B_0^2$, presented in Figure 2.29.

Graph $B_i^2$ from the family of Generalised Second Blanuša Snarks is constructed by




(a) Labelling $\pi$ of block $B_0^1$.

(b) Labelling $\pi$ of blocks $B_j$.

Figure 2.27: The labellings of blocks $B_0^1$ and $B_j$. The edges that connect $B_0^1$ and $B_j$ to their neighbours are represented in gray, with their respective labels. In particular, the edge incident with $z_j$ in (b) is labelled with $a = 1$ when $j = 1$ and $a = 2$, otherwise. Note, however, that this does not alter the induced colour of $z_j$.

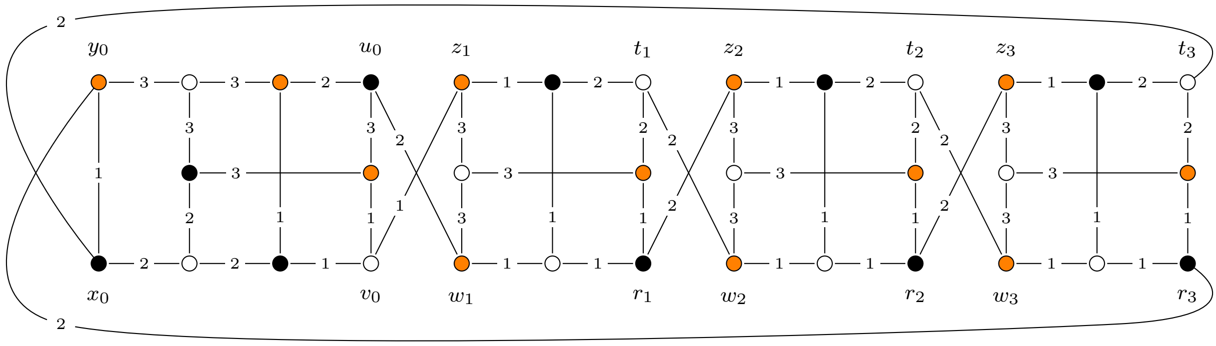



Figure 2.28: The gap-[3]-edge-labelling $(\pi, c_\pi)$ of Generalised First Blanuša Snark $B_3^1$.

connecting block $B_0^2$ and $i$ copies of block $B$. Once again, we denote by $B_j$ the $j$-th copy of block $B$. The connection is done as follows: we add edges $u_0w_1$ and $v_0z_1$ between $B_0^2$ and $B_1$; then, we connect $B_j$ and $B_{j+1}$ by adding edges $t_jw_{j+1}$ and $r_jz_{j+1}$, for $1 \leq j \leq i-1$; finally, we connect block $B_i$ to $B_0^2$ with edges $t_ix_0$ and $r_iy_0$. A sketch of the graph obtained by this construction is presented in Figure 2.30.

In Theorem 2.8, we establish the edge-gap number for the family of Generalised Second Blanuša Snarks.

**Theorem 2.8.** *Let $\mathcal{B}^2 = \{B_1^2, B_2^2, \ldots\}$ be the family of Generalised Second Blanuša Snarks. For $G \cong B_i^2$, $\chi_E^g(G) = 3$.*

*Proof.* Let $G$ be the Generalised Second Blanuša Snark $B_i^2$. Similar to the proof of First Blanuša Snarks, we demonstrate that $G$ admits a gap-[3]-edge-labelling $(\pi, c_\pi)$, thus proving that $\chi_E^g(G) = 3$ since $\chi(G) = 3$.

Define labelling $\pi$ of $G$ as follows. For blocks $B_j$, $1 \leq j \leq i$, we assign edge-labels exactly as we did in the case of Generalised First Blanuša Snarks. We recall this labelling in Figure 2.31(b). For the initial block $B_0^2$, assign labels according to Figure 2.31(a). Finally, let $\pi(v_0z_1) = \pi(u_0w_1) = 1$, and $\pi(e) = 2$ to every other edge $e$ connecting adjacent blocks. Define colouring $c_\pi$ as usual.

In order to prove the result, we show that $c_\pi$ is a proper colouring of $G$. First, consider blocks $B_j$, $1 \leq j \leq i$. Since the labelling of these blocks is essentially the same (note that

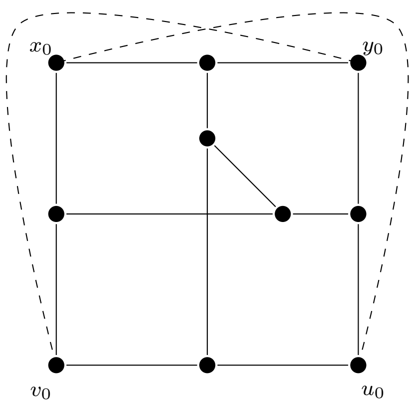



Figure 2.29: Block $B_0^2$ used in the construction of Generalised Second Blanuša Snarks.

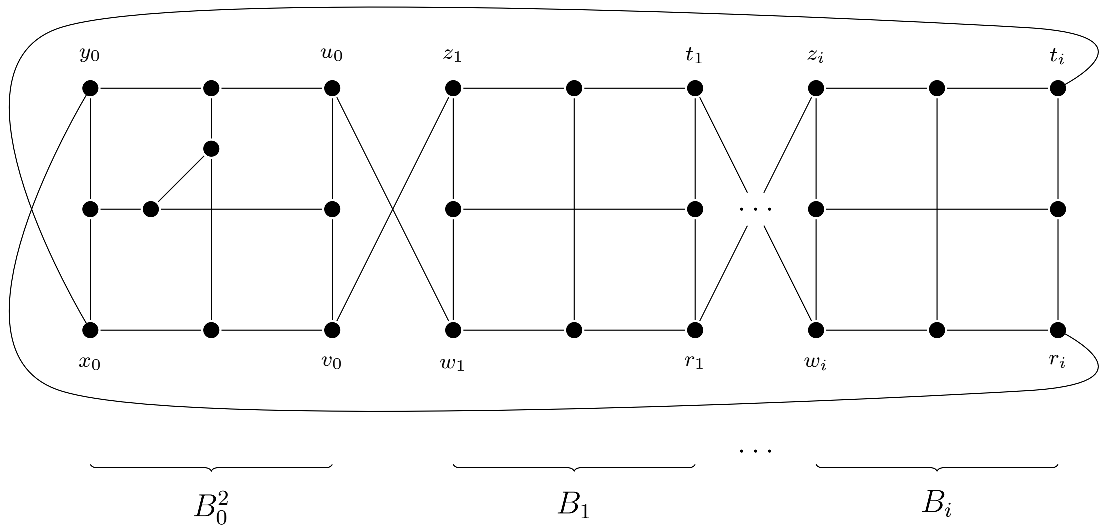



Figure 2.30: The construction of Generalised Second Blanuša Snark $B_i^2$.

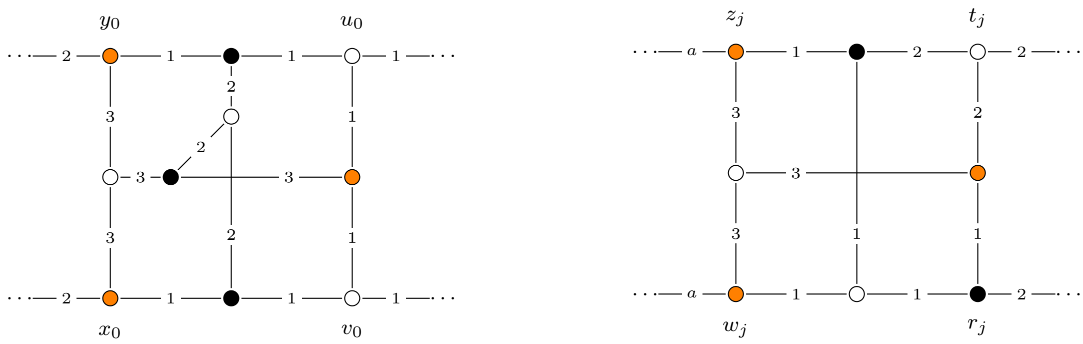
$\{1,3\} \subseteq \Pi_{E(w_j)})$ as in the proof of Theorem 2.7, it follows that $c_\pi$ is a proper colouring of $V(B_j)$. Also, since $c_\pi(z_j) \neq c_\pi(r_j)$ and $c_\pi(w_j) \neq c_\pi(t_j)$, we conclude that blocks $B_j$ and $B_{j+1}$ are connected by vertices with different induced colours.

For the remaining block $B_0^2$, by inspecting Figure 2.31(a), we observe that there are no two adjacent vertices with the same induced colour. Furthermore, $c_\pi(u_0) \neq c_\pi(w_1)$, $c_\pi(v_0) \neq c_\pi(z_1)$, $c_\pi(y_0) \neq c_\pi(r_i)$ and $c_\pi(x_0) \neq c_\pi(t_i)$. Therefore, $c_\pi$ is a proper colouring of $G$, which completes the proof. In Figure 2.32, we illustrate $(\pi, c_\pi)$ for $B_2^2$. $\square$

The next family of snarks considered is the that of Flower Snarks, which is described in the next section.

### Flower Snarks

In 1975, R. Isaacs [17] described an infinite family of snarks named *Flower Snarks*, which are defined as follows. Let $T \cong S_3$ be the star with vertices $v, x, y, z$, where $v$ is the central vertex. Graph $T$ is illustrated in Figure 2.33(a). For an odd integer $l$, $l \geq 3$, Flower Snark $J_l$ is constructed by using $l$ copies of $T$, $T_0, T_1, \ldots, T_{l-1}$. We denote the




(a) Labelling $\pi$ of block $B_0^2$.

(b) Labelling $\pi$ of blocks $B_j$.

Figure 2.31: The labellings of blocks $B_0^2$ and $B_j$ in (a) and (b), respectively.

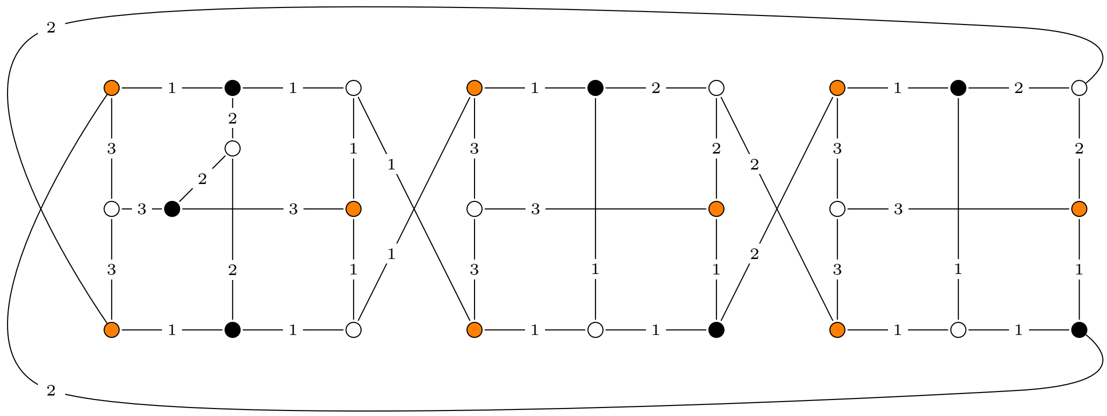


Figure 2.32: The gap-[3]-edge-labelling of Second Blanuša Snark $B_2^2$.

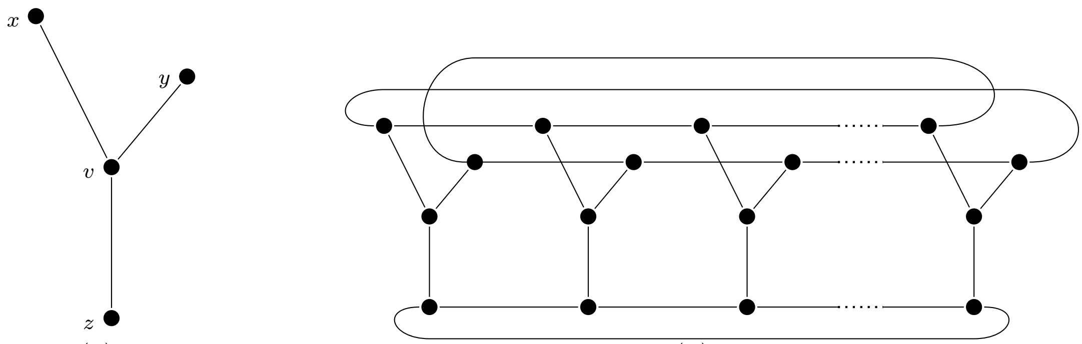
vertices of each $T_i$ as $v_i, x_i, y_i$ and $z_i$. Graphs $T_0, \ldots, T_{l-1}$ are connected by two cycles: $C_0 = \{z_0, z_1, \ldots, z_{l-1}\}$, and $C_1 = \{x_0, x_1, \ldots, x_{l-1}, y_0, y_1 \ldots, y_{l-1}\}$. This construction is illustrated in Figure 2.33(b). A more common visual representation of Flower Snarks is exemplified for $J_5$ in Figure 2.34, which perfectly depicts why snarks in this construction were named "flowers".

For the family of Flower Snarks, denoted by $\mathcal{J} = \{J_3, J_5, \ldots\}$, we establish the edge-gap number in Theorem 2.9.

**Theorem 2.9.** *Let $G$ be a Flower Snark. Then, $\chi_E^g(G) = 3$.*

*Proof.* Let $G$ be a Flower Snark constructed from $l$ copies of $T$, as defined in the text. For each copy of $T_i$, its vertex set is denoted by $V(T_i) = \{v_i, x_i, y_i, z_i\}$. In order to prove the result, by Theorem 2.1, it suffices to show that $G$ admits a gap-[3]-edge-labelling $(\pi, c_\pi)$.

Similarly to the construction of Blanuša Snarks in the previous section, we assign labels to each $T_i$, $0 \leq i < l$, such that colouring $c_\pi$ induced in $G$ is a proper vertex-colouring. In the case of Flower Snarks, however, we define labellings of $T_i$ depending on the value of $i$ as follows. For the edges of $T_i$, assign:




(a) (b)

Figure 2.33: In (a), graph $T$ with its vertices and their names; and in (b), the construction of $J_l$ using $l$ copies of $T$.

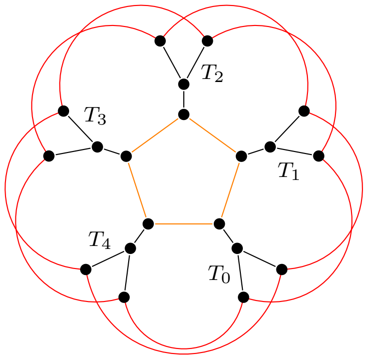
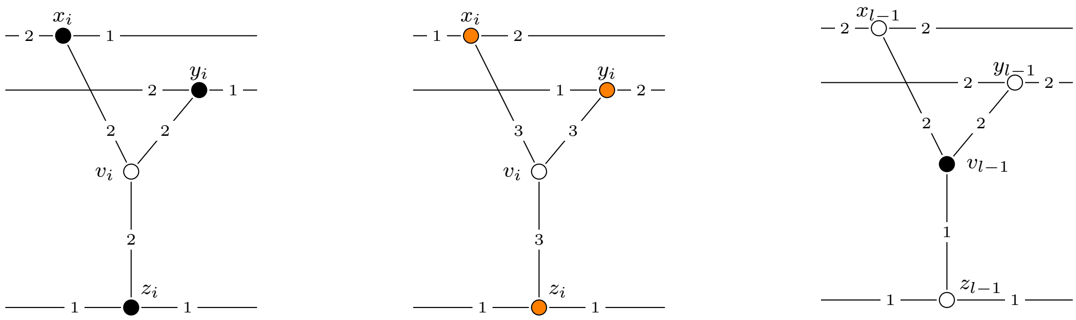



Figure 2.34: Flower Snark $J_5$. Cycles $C_0$ and $C_1$ are highlighted in orange and red, respectively. The bottommost edges in the image are edges $x_4y_0$ and $y_4x_0$.

$$\pi(v_ix_i) = \pi(v_iy_i) = \begin{cases} 2, & \text{if } i \text{ is even;} \\ 3, & \text{otherwise.} \end{cases} \qquad \pi(v_iz_i) = \begin{cases} 1, & \text{if } i = l-1; \\ 2, & \text{if } i \text{ is even, } i \neq l-1; \\ 3, & \text{otherwise.} \end{cases}$$

Next, assign label 1 to every edge $e \in E(C_0)$, that is, edges connecting vertices $z_iz_{(i+1) \bmod l}$, $0 \leq i < l$. It remains to assign labels to edges in $E(C_1)$, that is, the cycle defined by edges $x_0x_1, x_1x_2, \ldots, x_{l-1}y_0, y_0y_1, \ldots, y_{l-1}x_0$. Let $\pi(x_{l-1}y_0) = \pi(y_{l-1}x_0) = 2$, and $\pi(x_ix_{i+1}) = \pi(y_iy_{i+1}) = 1 + (i \bmod 2)$. Define colouring $c_\pi$ as usual. Observe that this labelling produces three distinct colourings for $T_i$, depending on the value of $i$, which are represented in Figure 2.35.




(a) $i$ even. (b) $i$ odd. (c) $i = l-1$.

Figure 2.35: Labellings $\pi$ of each $T_i$ of $G$. White vertices have induced colour 0, black vertices, colour 1, and orange vertices, colour 2.

In order to conclude the proof, it suffices to show that $c_\pi$ is a proper colouring of $G$. First, consider $T_{l-1}$ and note that $\Pi_{E(v_{l-1})} = \{1, 2\}$, inducing colour 1. By the definition, both edges incident with vertices $x_{l-1}$ and $y_{l-1}$ in $C_1$ receive label 2. This implies that $c_\pi(x_{l-1}) = c_\pi(y_{l-1}) = 0$. For $z_{l-1}$, we have $\Pi_{E(z_{l-1})} = \{1\}$ since every edge in $C_0$ is assigned label 1. This also induces colour 0 in $z_{l-1}$. Figure 2.35(c) exhibits this colouring.

Next, we consider $T_i$, $0 \leq i < l-1$, starting with vertices $v_i$. The edges incident with $v_i$ were labelled such that $\Pi_{E(v_i)} = \{2 + (i \bmod 2)\}$, which induces $c_\pi(v_i) = 0$. For vertices $x_i$ and $y_i$, note that their incident edges in $C_1$ receive labels 1 and 2. Therefore, $\Pi_{E(x_i)} = \{1, 2, \pi(v_i x_i)\}$, and an analogous reasoning holds for vertex $y_i$. Now, since $\pi(v_i x_i) = \pi(v_i y_i)$ alternates between labels 2 and 3, with $\pi(v_0 x_0) = 2$, we conclude that $c_\pi(x_i) = c_\pi(y_i) = 1 + (i \bmod 2)$ for every $T_i$, $i < l - 1$. Also, note that $c_\pi(v_i) \neq c_\pi(x_i)$ and $c_\pi(v_i) \neq c_\pi(y_i)$. For the remaining vertices $z_i$, note that $\Pi_{E(z_i)} = \{1, \pi(v_i z_i)\}$. Labelling $\pi$ alternates labels 2 and 3 in graphs $T_i$, $0 \leq i < l - 1$, with $\pi(v_0 z_0) = 2$, which also induces $c_\pi(z_i) = 1 + (i \bmod 2)$. The colourings for these cases are depicted in figures 2.35(a) and 2.35(b).

In order to conclude the proof, note that $c_\pi(x_{l-1}) = c_\pi(y_{l-1}) = c_\pi(z_{l-1}) = 0$ and $c_\pi(x_i) = c_\pi(y_i) = c_\pi(z_i) = 1 + (i \bmod 2)$. Therefore, cycles $C_0$ and $C_1$ are coloured as illustrated in Figure 2.36 and we conclude that $c_\pi$ is a proper colouring of $G$. □

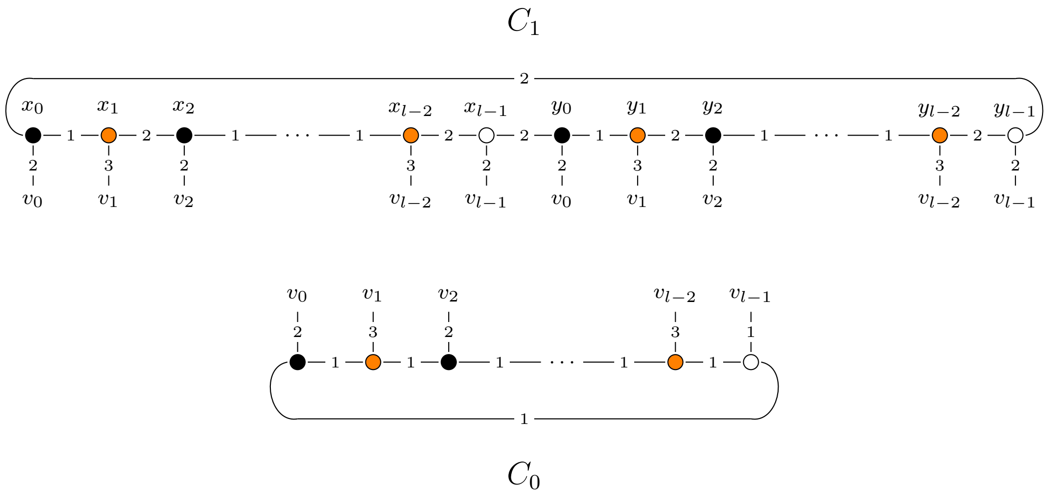



Figure 2.36: The induced colourings of cycles $C_1$ (above) and $C_0$ (below). Vertices in white, black and orange have induced colours 0, 1 and 2, respectively.

An interesting observation is that identifying vertices $v_0, v_1$ and $v_2$ in Flower Snark $J_3$ produces the Petersen Graph, the smallest known snark. In fact, the labelling presented in the proof of Theorem 2.9 is also a gap-[3]-edge-labelling of the Petersen Graph.

We remark that the labelling of Flower Snarks is different from that of Blanuša Snarks since each adjacent block of $J_l$ was assigned a different labelling, whereas this is not the case for Blanuša Snarks. The next family of snarks considered, the Goldberg Snarks, uses a labelling technique similar to that of Flower Snarks.

### Goldberg and Twisted Goldberg Snarks

The family of *Goldberg Snarks* $\mathcal{G} = \{G_3, G_5, \ldots\}$ was introduced in 1981 by M. K. Goldberg [15], who described a method for constructing graphs $G$ with $\chi'(G) = 4$ and maximum degree three. His technique can be used to obtain several different families of snarks – for example, Flower Snarks $J_l$ described in the previous section. The details of Goldberg's method, however, is beyond the scope of this text. Here, we describe only the families of Goldberg and Twisted Goldberg Snarks.

For each $l$, $l \geq 5$ and odd, Goldberg Snark $G_l$ is constructed using crown $R_l$ and $l$ copies of block $B$, represented in Figure 2.37. For every block $B_j$, its vertex set is denoted by $V(B_j) = \{u_j, y_j, r_j, w_j, t_j, v_j, x_j\}$. Here, we rename the vertex set of crown $R_l$ as $V(R_l) = \{s_0, \dots, s_{l-1}\} \cup \{z_0, \dots, z_{l-1}\}$, with $d(s_j) = 3$ and $d(z_j) = 1$ for all $j < l$. This is done to avoid ambiguity with vertices $v_j$ and $u_j$ from blocks $B_j$.

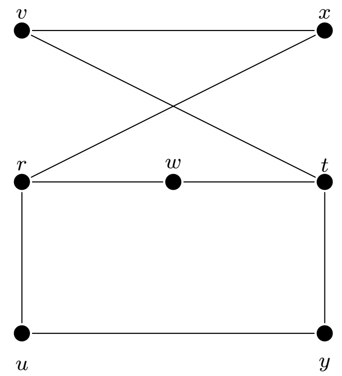



Figure 2.37: Block $B$ used in the construction of the Goldberg Snark $G_l$.

In order to construct $G_l$, we cyclically connect blocks $B_j$ and $B_{j+1}$ by adding edges $\{x_j v_{j+1}, y_j u_{j+1}\}$ for all $0 \leq j < l$. Also, we identify vertices $z_j$ and $w_j$ from crown $R_l$ and block $B_l$, respectively. A general representation of this construction is presented in Figure 2.38. Particularly for $l = 3$, the construction is done by using star $S_3$ instead of crown $R_3$. Goldberg Snark $G_3$ is presented, together with a gap-[3]-edge-labelling of the graph, in Figure 2.39.

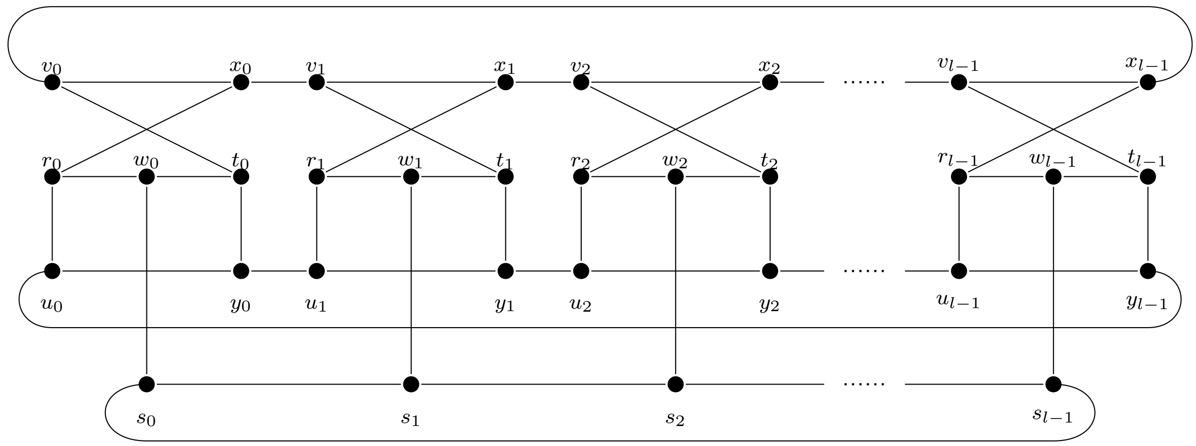



Figure 2.38: The construction of Goldberg snark $G_l$, $l \geq 5$.

Regarding Goldberg Snarks, some authors define the operation of *twisting* edges in $G_l$ as the removal of edges $x_j v_{j+1}$ and $y_j u_{j+1}$ from the graph and, then, adding edges $x_j u_{j+1}$ and $y_j v_{j+1}$. Figure 2.40 exemplifies this operation for edges connecting blocks $B_{j-1}$ and $B_j$.

In 2007, M. Ghebleh [14] defined the *Twisted Goldberg Snark* $TG_l$, $l \geq 3$ and odd, as the graph obtained by twisting edges connecting two adjacent blocks in Goldberg Snark $G_l$. The author stated that applying more twists to $TG_l$ does not produce any new graphs. For example, consider Figure 2.41, which illustrates two pairs of twisted edges

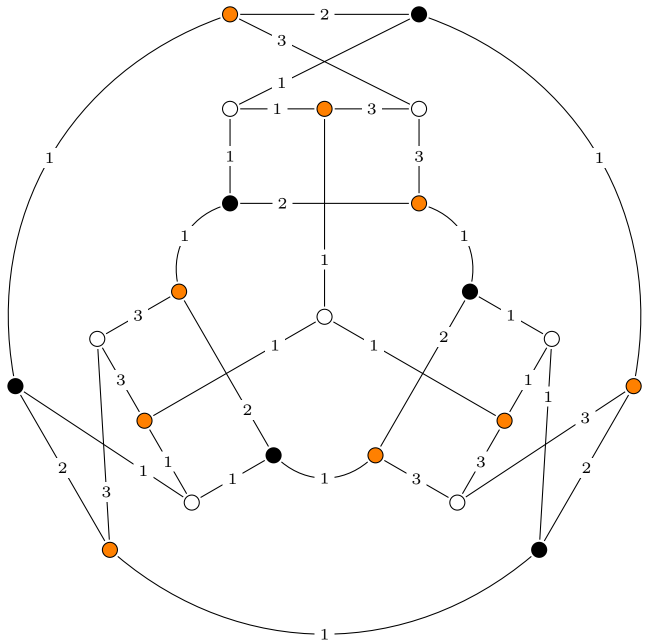


Figure 2.39: The gap-[3]-edge-labelling of $G_3$. Vertices in white have induced colour 0, in black, colour 1, and in orange, colour 2.

in a Goldberg Snark $G_l$. By renaming vertices $(u_j, y_j, r_j, t_j, v_j, x_j)$ as $(v_j, x_j, t_j, r_j, u_j, y_j)$, we conclude that the graph $G'$ resulting from this operation is $G' \cong G_l$. In fact, Ghebleh remarks that applying any even number of twists to Goldberg Snark $G_l$ yields $G_l$ itself. Otherwise, if an odd number of twists is applied, then the resulting graph is $TG_l$.

Twisted Goldberg Snark $TG_3$ is defined from Goldberg Snarks $G_3$ and, therefore, also uses star $S_3$ in its construction. We illustrate $TG_3$ in Figure 2.42, together with a gap-[3]-edge-labelling for it. Observe the twisted edges connecting the bottommost blocks in the image.

We establish the edge-gap number for both Goldberg and Twisted Goldberg Snarks in Theorem 2.10.

**Theorem 2.10.** *Let $G$ be a (Twisted) Goldberg Snark. Then,* $\chi^g_E(G) = 3$.

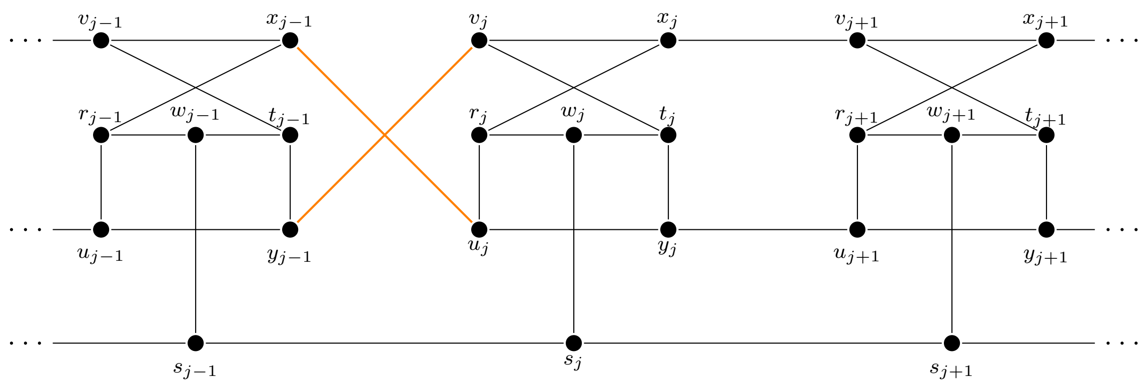



Figure 2.40: A twisted edge in Goldberg Snark $G_l$.

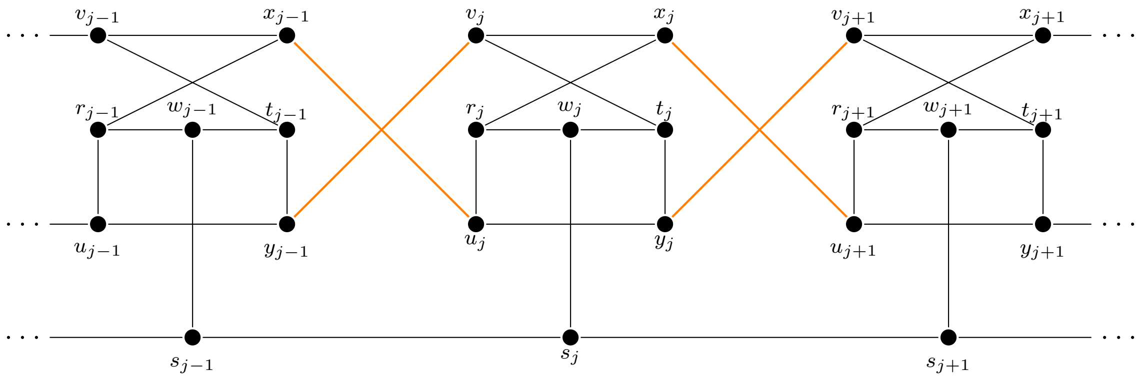



Figure 2.41: Two twisted edges in Goldberg Snark $G_l$.

*Proof.* Let $G \cong G_l$, $G' \cong TG_l$ and $l \geq 3$ and odd. It is well known that $\chi(G) = \chi(G') = 3$. Therefore, by Theorem 2.1, showing that $G$ and $G'$ admit gap-[3]-edge-labellings proves the result.

Figures 2.39 and 2.42 respectively show gap-[3]-edge-labellings for $G_3$ and $TG_3$. Next, consider $l \geq 5$ and odd. For every $e = uv$, $u \in \{x_i, y_i\}$ and $v \in \{u_{i+1}, v_{i+1}\}$, assign $\pi(e) = 1$ if $i$ is even, and $\pi(e) = 3$, otherwise. Now, for every block $B_i$, assign labels to $E(B_i)$ according to Figure 2.43. For the remaining edges, assign labels: $\pi(s_0w_0) = 3$; $\pi(s_iw_i) = 1 + (i \bmod 2)$, $1 \leq i < l$; and $\pi(s_is_{(i+1) \bmod l}) = 1$ for $0 \leq i < l$. Colouring $c_\pi$ is defined as usual. Figure 2.44 illustrates $(\pi, c_\pi)$ for snarks $G_5$ and $TG_5$. In the latter, the edges connecting blocks $B_3$ and $B_4$ have been twisted. We remark that both edges connecting adjacent blocks $B_j$ and $B_{j+1}$ always receive the same label. Therefore, colouring $c_\pi$ is induced in the same manner regardless of whether these edges are twisted.

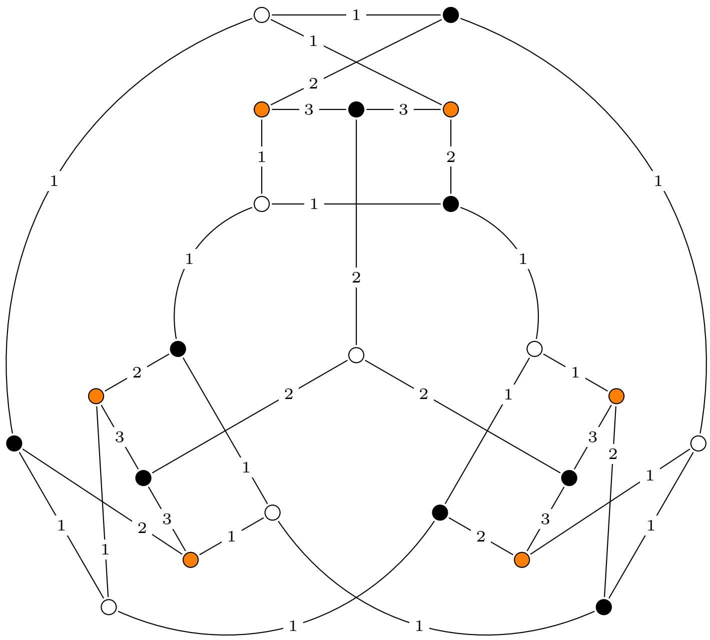
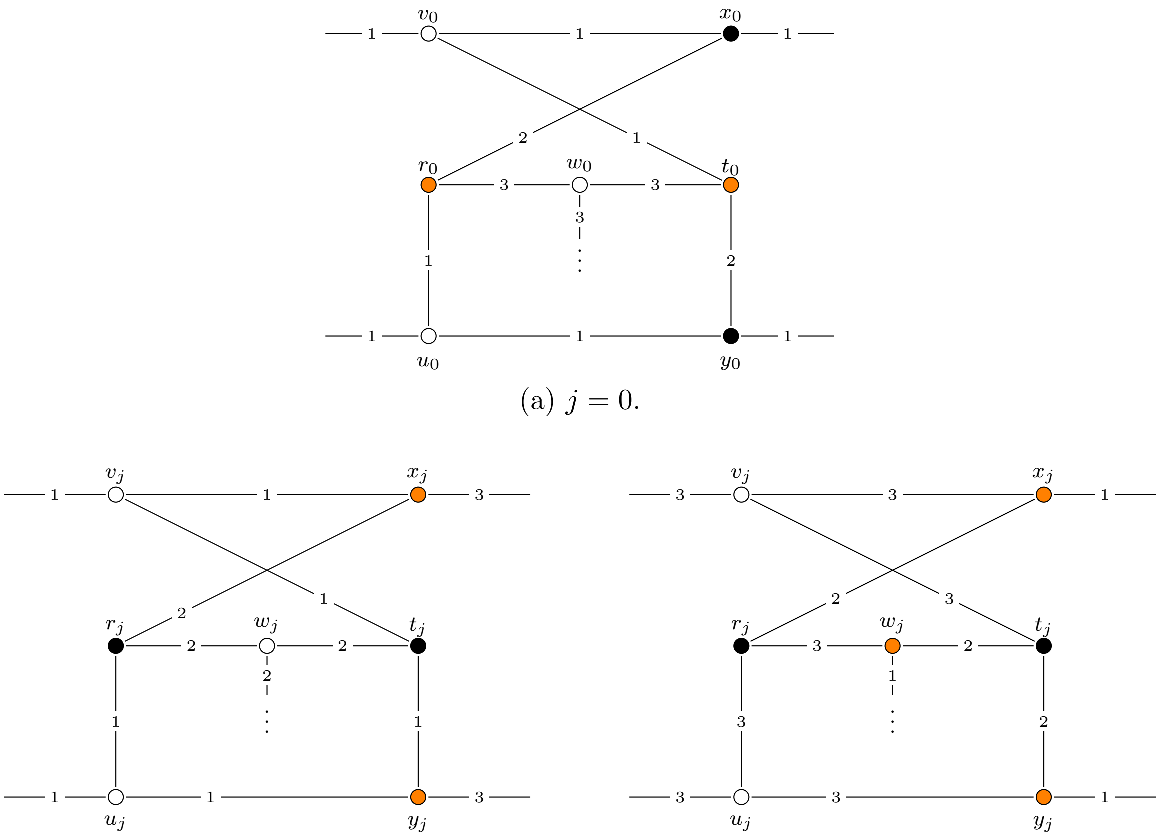


Figure 2.42: The gap-[3]-vertex-labelling of Twisted Goldberg Snark $TG_3$. Vertices filled in white, black and orange have induced colours 0, 1 and 2, respectively.




(a) $j = 0$.

(b) $1 \leq j \leq l-2$ and odd.

(c) $2 \leq j \leq l-1$ and even.

Figure 2.43: Labelling $\pi$ and induced colouring $c_\pi$ of blocks $B_i$. Vertices in white, black and orange have induced colours 0, 1 and 2, respectively.

In order to prove the result, it suffices to show that $c_\pi$ is a proper vertex-colouring of the graphs. First, consider block $B_0$. Since $l$ is odd, we know that every edge connecting $B_0$ to $B_1$ and to $B_{l-1}$ receives label 1. Also, we have $\pi(s_0w_0) = 3$. By inspecting Figure 2.43(a), which depicts this labelling, we conclude that $c_\pi$ is a proper colouring of $V(B_0)$. Now, for $1 \leq i \leq l-2$ and odd, we have $\pi(e) = 1$ for edges $e$ connecting $B_i$ to $B_{i-1}$, and $\pi(e') = 3$, connecting $B_i$ to $B_{i+1}$. Also, $\pi(s_iw_i) = 2$ in this case. Then, by inspecting Figure 2.43(b), we conclude that these blocks are also properly coloured. The same reasoning applied to blocks $B_j$, $2 \leq j \leq l-1$ and even, illustrated in Figure 2.43(c) leads us to the conclusion that there are no conflicting internal vertices in blocks $B_j$ of $G$ $(G')$.

Next, consider the labelling and induced colouring of crown $R_l$, which is sketched in Figure 2.45. Since every edge $s_js_{(j+1) \bmod l}$ receives label 1, we have $\Pi_{E(s_j)} = \{1, \pi(s_jw_j)\}$. This implies that $c_\pi(s_0) = 2$ and $c_\pi(s_j) = j \bmod 2$ for $1 \leq j \leq l-1$. This is a proper colouring of crown $R_l$.

Thus, it remains to prove that there are no conflicting vertices connecting adjacent blocks in $G$. First, note that $c_\pi(u_j) = c_\pi(v_j) = 0$ for all blocks $B_j$. Also, every $B_j$, $0 \leq j \leq l-1$, has $c_\pi(x_j) = c_\pi(y_j) \neq 0$. Then, every edge $uv$, $u \in \{x_j, y_j\}$, $v \in \{u_{j+1}, v_{j+1}\}$, connecting blocks $B_j$ and $B_{j+1}$ has one end with induced colour 0 and the other, with some colour $c \in \{1, 2\}$. Thus, we conclude that $c_\pi$ is a proper vertex-colouring for $G$

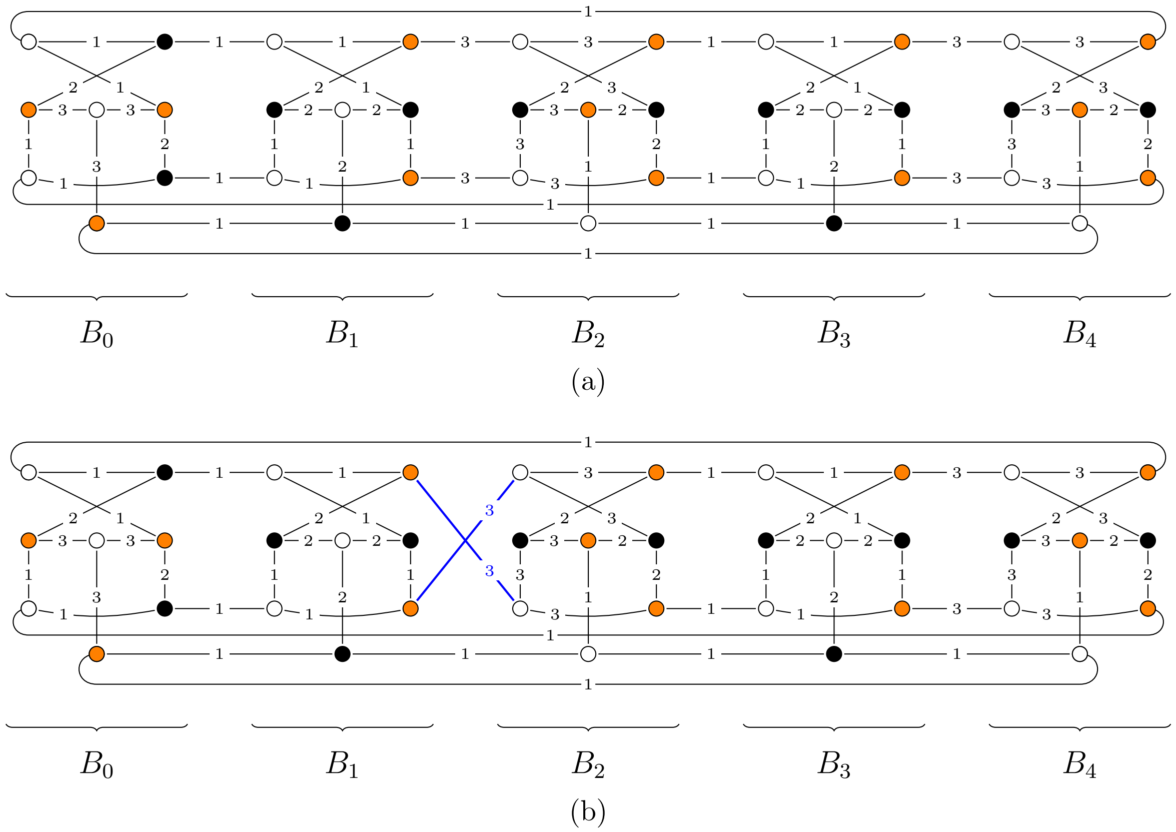



Figure 2.44: Gap-[3]-edge-labellings of $G_5$ and $TG_5$ in (a) and (b), respectively. Vertices in white have induced colour 0, in black, colour 1, and in orange, colour 2.

and $G'$, and the result follows. □

This completes our study of gap-[$k$]-edge-labellings for classes of graphs. In the next chapter, we introduce and study the vertex variant of this labelling, which was formally defined by A. Dehghan et al. [8] in 2013.

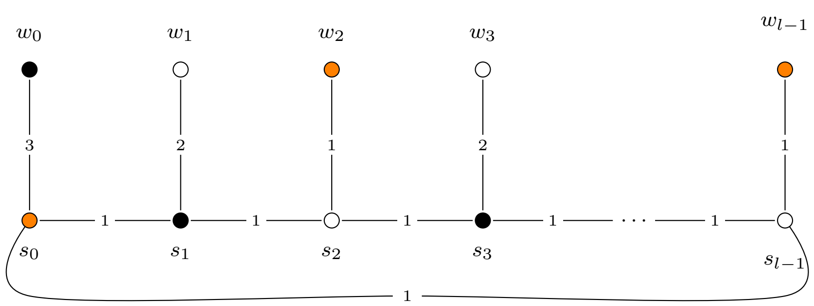



Figure 2.45: The labelling and induced colouring of crown $R_l$. White vertices have induced colour 0, black vertices, colour 1, and orange vertices, colour 2.

# Chapter 3

# Gap-$[k]$-vertex-labellings

In the previous chapter, we discussed the gap-$[k]$-edge-labelling problem, **Gkel**, and established the edge-gap number, $\chi^{\mathrm{g}}_{E}$, for some classes of graphs. The next proper labelling problem we address also uses the concept of inducing a proper vertex-colouring in a graph by the largest gap between labels. In this version, however, the labels are assigned to its vertices. This proper labelling was introduced by A. Dehghan et al. in 2013 [8], under the name *vertex-labelling by gap*.

We mention in Chapter 2 that the notation used in the literature for proper labellings of graphs is often misleading. Therefore, as we did for gap-$[k]$-edge-labellings, we rename Dehghan et al.'s labelling as a gap-$[k]$-vertex-labelling of a graph $G$. It is defined as an assignment of labels to the vertices (rather than to the edges) of a graph $G$ such that the colour of every vertex $v$ is computed as the maximum difference among the labels of its neighbours (cases where $d(v) = 0$ and $d(v) = 1$ are treated separately and are defined in detail below). In this chapter, we advance the computational complexity analysis of this problem, which began with Dehghan et al. [8], and prove hardness results for problems associated with this labelling. Also, we investigate the gap-$[k]$-vertex-labelling for some classes of graphs and discuss properties of this labelling, establishing bounds for the minimum $k$ for which an arbitrary graph admits a gap-$[k]$-vertex-labelling. We remark that an upper bound for this parameter is established in Chapter 4, where properties of another decision problem associated with this labelling are investigated.

## 3.1 Preliminaries

A *gap-$[k]$-vertex-labelling* of a simple graph $G = (V, E)$ is a proper labelling defined by a pair $(\pi, c_\pi)$, where $\pi : V \to \{1, 2, \ldots, k\}$ is a labelling of the vertices of $G$ and $c_\pi$ is a proper vertex-colouring of $G$ such that, for every vertex $v \in V$, its induced colour is:

$$c_\pi(v) = \begin{cases} \max\limits_{u \in N(v)} \{\pi(u)\} - \min\limits_{u \in N(v)} \{\pi(u)\}, & \text{if } d(v) \geq 2; \\ \pi(u)_{u \in N(v)}, & \text{if } d(v) = 1; \\ 1, & \text{if } d(v) = 0. \end{cases}$$

Similar to the definition of gap-$[k]$-edge-labellings in Chapter 2, the colour of vertices

$v \in V(G)$ with $d(v) \geq 2$ are induced by the largest difference among the labels of its adjacent vertices. Hence, $c_\pi(v)$ is induced by the maximum *gap* in $\Pi_{N(v)}$. Figure 3.1 illustrates a gap-[3]-vertex-labelling of the Petersen Graph. Since both the labels and the colours are assigned to the vertices of the graph, when necessary, we distinguish these values by representing the label assigned to a vertex in a box next to its lower right corner. The number inside each vertex corresponds to its induced colour. For example, vertices $v_0$, $v_7$ and $v_9$ in Figure 3.1 have labels $\pi(v_0) = 3$, $\pi(v_7) = 2$ and $\pi(v_9) = 1$, and their induced colours are $c_\pi(v_0) = c_\pi(v_7) = 0$ and $c_\pi(v_9) = 1$. The notation used in this figure to denote labelling $\pi$ is used throughout the chapter.

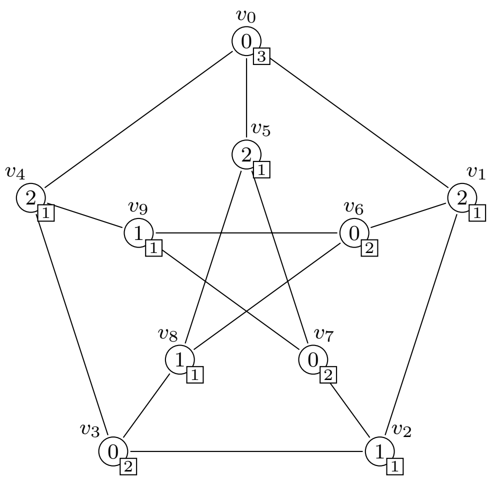



Figure 3.1: A gap-[3]-vertex-labelling of the Petersen Graph.

Whenever a new proper labelling is introduced, it is customary to investigate the least number $k$ of labels that is required to properly label an arbitrary graph. For this labelling, we define the minimum number $k$ for which a graph $G$ admits a gap-[$k$]-vertex-labelling as the *vertex-gap number of* $G$ and we denote this parameter by $\chi^{\mathrm{g}}_{V}(G)$. This is done so as to maintain the pattern of the notation defined in Chapter 2. Once again, observe the three components of $\chi^{\mathrm{g}}_{V}(G)$: $\chi$ indicates we are interested in a proper colouring, in this case, of the vertices of $G$; the superscript $g$ indicates we are using gaps to induce the colour of each vertex; and, finally, the subscript $V$ indicates we assign labels to the vertices of $G$.

Gap-[$k$]-vertex-labellings of graphs were introduced by A. Dehghan et al. [8] in 2013, under the name *vertex-labelling by gap*. In their article, they prove that every tree $T$ admits a gap-[2]-vertex-labelling, thus establishing[1] that $\chi^{\mathrm{g}}_{V}(T) = 2$. An example of the labelling presented in their article is illustrated in Figure 3.2. The authors also determined that $r$-regular bipartite graphs $G$, with $r \geq 4$, have $\chi^{\mathrm{g}}_{V}(G) = 2$. The proof of this result is based on the fact that every $k$-regular $k$-uniform hypergraph $\mathscr{H}$ admits a 2-colouring when $k \geq 4$; the authors used this result to create a gap-[2]-vertex-labelling of the $r$-regular bipartite graphs, $r \geq 4$. Figure 3.3 illustrates a gap-[2]-vertex-labelling of a 5-regular bipartite graph $G$, constructed from a 5-regular 5-uniform hypergraph $\mathscr{H}$.

Although Dehghan et al. [8] established the vertex-gap number for these two families of graphs, the focus of their article was on determining the algorithmic complexity of

[1]Our proof of this result is presented in Section 3.3.4, Lemma 3.14.

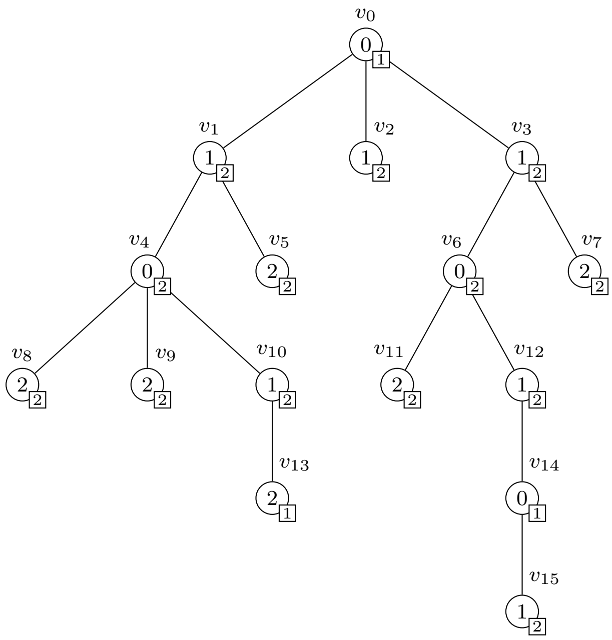



Figure 3.2: A gap-[2]-vertex-labelling of a tree $T$.

decision problems associated with several proper labellings. For the purposes of this work, the statement of the decision problem associated with the gap-$[k]$-vertex-labelling of graphs is presented below.

Gap-$[k]$-vertex-labelling [**Gkvl**]

**Instance:** A graph $G = (V, E)$ and an integer $k \geq 1$.
**Question:** Does $G$ admit a gap-$[k]$-vertex-labelling?

When considering a specific value of $k$, we denote **Gkvl** by replacing **k** with its value. Dehghan et al. [8] proved that **Gkvl** is NP-complete for arbitrary graphs when $k \geq 3$. However, for $k = 2$, the problem is polynomial-time solvable for some classes of graphs and remains NP-complete for others. The authors determined the complexity of the following problems:

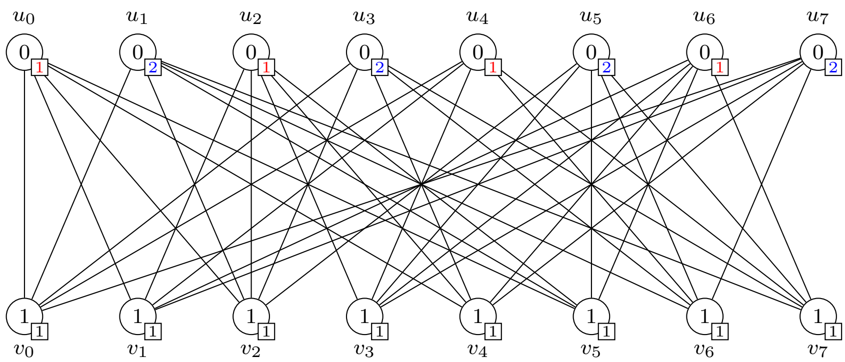



Figure 3.3: A gap-[2]-vertex-labelling of a 5-regular bipartite graph $G$. The corresponding 5-regular 5-uniform hypergraph $\mathscr{H}$ has vertex set $X = \{v_0, \ldots, v_7\}$ and hyperedge set $Y = \{N(u_0), \ldots, N(u_7)\}$.

(i) **G2VL** is NP-complete for bipartite graphs and for planar 3-colourable graphs;

(ii) **G2VL** is in P for planar bipartite and for $r$-regular bipartite, $r \geq 4$, graphs.

Dehghan et al. [8] showed that it is easy[2] to solve **G2VL** when the given bipartite graph is planar, whereas if the graph is not planar, the problem is NP-complete. The authors remarked that planarity of graphs could be a facilitating factor. However, in 2016, Dehghan [7] proved that deciding whether a planar bipartite graph $G$ admits a gap-[2]-vertex-labelling $(\pi, c_\pi)$ such that $c_\pi$ is a 2-colouring of $G$ is also NP-complete. This result also shows that **G2VL** for bipartite graphs is, in fact, a problem with interesting properties which demands further research.

Our approach to study the gap-[$k$]-vertex-labellings of graphs is divided into three fronts: determining $\chi^{\mathrm{g}}_{V}(G)$ for families of graphs; establishing bounds for the vertex-gap number of arbitrary graphs; and studying the computational complexity of **G2VL** for cubic bipartite graphs.

In the first front, we investigated the vertex-gap number for the same classes addressed in the edge version, namely cycles, crowns, wheels, unicyclic graphs and some families of snarks. In addition, motivated by a question posed by Dehghan et al. [8], we also considered the family of cubic bipartite graphs. For this class, we designed several labelling techniques and algorithms, which are presented in detail in Section 3.3.5. Our findings for these graphs led us to conjecture that, with the exception of the Heawood Graph, $\chi^{\mathrm{g}}_{V}(G) = 2$ for every hamiltonian cubic bipartite graph $G$.

The second front was to establish bounds for the vertex-gap number of arbitrary graphs. In Section 3.3, we prove that the lower bound for $\chi^{\mathrm{g}}_{V}(G)$ is the same as the one for the edge version. As previously stated, an upper bound for the parameter is presented in Chapter 4, where we discuss further structural properties regarding the gap-[$k$]-vertex-labelling of graphs.

Third, we investigate the computational complexity of **G2VL** for cubic bipartite graphs, an approach also motivated by Dehghan et al.'s work. We know that this problem is in NP since one can verify (in polynomial time) whether a labelling $\pi : V(G) \to \{1, 2\}$ induces a proper vertex-colouring of the graph. However, it is unclear if the problem is also NP-complete. In order to obtain advances in this front, we decided to broaden our set of instances to subcubic bipartite graphs. Upon such consideration, we proved that **G2VL** for subcubic bipartite graphs remains NP-complete. To simplify, we refer to **G2VL** for this class of graphs as **G2VL (ScB)**.

**Theorem 3.1. G2VL (ScB)** *is* NP*-complete.* □

The proof of this result is presented in Section 3.2, where we reduce the Monochromatic Triangle problem to **G2VL (ScB)** in polynomial time. The statement of Monochromatic Triangle [13] is presented below.

---

[2]Here, we use the term *easy* to indicate that there exists a polynomial-time algorithm that decides this problem. The algorithm is described in A. Dehghan et al.'s article [8].

<u>MONOCHROMATIC TRIANGLE (**MT**)</u>

**Instance:** A graph $G = (V, E)$.

**Question:** Is there a partition of $E$ into two disjoint sets $E_1, E_2$ such that neither $G_1 = (V, E_1)$ nor $G_2 = (V, E_2)$ contains a triangle?

This problem can also be stated as an edge-colouring problem, where the question is whether $G$ admits a colouring of its edges in two colours, namely *red* and *blue*, such that every triangle in $G$ has at least one blue edge and one red edge; thus, no triangle is monochromatic. This problem was proved to be NP-complete by Burr in 1976, but this result was only published by Garey & Johnson [13] in 1979. The MONOCHROMATIC TRIANGLE problem is closely related to a branch of mathematics known as *Ramsey Theory*, which is beyond the scope of this text.

The remainder of this chapter is divided as follows. The next section presents the proof of Theorem 3.1. In the beginning of Section 3.3, we establish a lower bound for the vertex-gap number of arbitrary graphs. After this, still in Section 3.3, we present our results for $\chi^{g}_{V}$ for some well-known classes of graphs: cycles, crowns, wheels, unicyclic graphs, families of cubic bipartite hamiltonian graphs and families of snarks.

## 3.2 G2VL (SCB) is NP-complete

We reduce an instance of the MONOCHROMATIC TRIANGLE problem, a graph $G = (V, E)$, to a subcubic bipartite graph $G' = (V', E')$ such that $G$ admits a 2-edge-colouring with no monochromatic triangles if and only if the constructed graph $G'$ admits a gap-[2]-vertex-labelling. The reduction is accomplished with the aid of two gadgets: a triangle gadget and a negation gadget. The first gadget represents each triangle $t_i$ in $G$ as a group of vertices in $G'$. These vertices are labelled and coloured by a gap-[2]-vertex-labelling $(\pi, c_\pi)$ of $G'$, when one exists. The negation gadget provides further structural properties in $G'$.

### 3.2.1 Triangle gadget

The *triangle gadget* $G^{\triangle}$ is an auxiliary simple bipartite graph with 19 vertices, 20 edges and is defined as follows. Let $e_x, e_y, e_z$, be the edges of a triangle $t$ of $G$. We abuse notation and say that $t = \{e_x, e_y, e_z\}$. The gadget, $G^{\triangle}$, has a vertex $u$ that represents $t$ in $G'$. For each edge $e_j$ in $t$, $j \in \{x, y, z\}$, the gadget has two adjacent vertices $v_j$ and $w_j$. Each vertex $v_j$ is also adjacent to $u$. There is also a copy of path $P_{12} = \{q_0, q_1, \ldots, q_{11}\}$ and edges $w_x q_0$, $w_y q_4$ and $w_z q_8$. Figure 3.4(a) illustrates the triangle gadget for a triangle $t$ of $G$ with edges $e_1, e_4, e_7$. Since the triangle gadget does not contain any odd cycles, it is bipartite. Also, no vertex has degree greater than 3, thus $G^{\triangle}$ is subcubic. A simplified representation of the triangle gadget is illustrated in Figure 3.4(b), in which we omit some of the vertices and edges so as to simplify the visualization of larger constructions further in this section.

**Property 3.2.** *Let $G^{\triangle}$ be a triangle gadget. If $G^{\triangle}$ admits a gap-[2]-vertex-labelling $(\pi, c_\pi)$ such that $c_\pi(u) = 1$, then $\pi(q_0) = \pi(q_4) = \pi(q_8)$.*

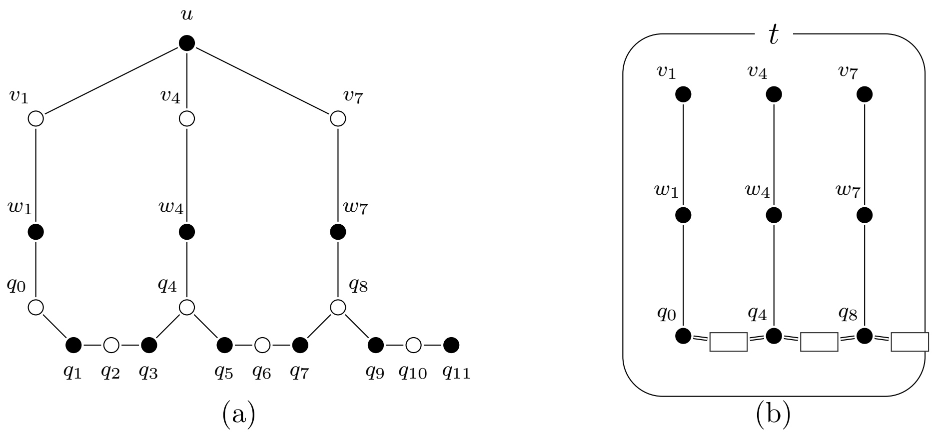



Figure 3.4: In (a), the triangle gadget $G^{\triangle}$ for a triangle $t = \{e_1, e_4, e_7\}$, with its vertex set partitioned into sets $A$ (in white) and $B$ (in black); and in (b), its simplified representation. The white rectangles and doubled lines connecting vertices $q_0$, $q_4$ and $q_8$ omit some vertices of path $P_{12}$.

*Proof.* Let $G^{\triangle} = (V, E)$ be a triangle gadget representing triangle $\{e_x, e_y, e_z\}$ and suppose $G^{\triangle}$ admits a gap-[2]-vertex-labelling $(\pi, c_\pi)$. Let $\{A, B\}$ be a bipartition of $G^{\triangle}$. Note that one part, say $A$, comprises vertices $v_x$, $v_y$, $v_z$, and every $q_i$, $i$ even; and the other part, $B$, comprises the remaining vertices. Bipartition $\{A, B\}$ is illustrated in Figure 3.4(a). Also, note that the colours of vertices in $V(G^{\triangle}) \setminus q_{11} \in \{0, 1\}$ since their degrees are greater than one. Colour of vertex $q_{11} \in \{1, 2\}$, depending on the label of $q_{10}$.

Suppose $c_\pi(u) = 1$. Since every $v \in A$ has $d(v) \geq 2$, we conclude that these vertices have induced colour 0. Now, consider vertex $q_{11}$, for which we know that $c_\pi(q_{11}) = \pi(q_{10})$. Let $a \in \{1, 2\}$ be the label assigned to $q_{10}$. Then, since $N(q_9) = \{q_8, q_{10}\}$ and $c_\pi(q_9) = 1$, we have $\pi(q_8) \neq \pi(q_{10})$. Thus, we conclude that $\pi(q_8) = b$, $b \in \{1, 2\}$ and $b \neq a$. Following this reasoning, and analysing vertices $q_7$, $q_5$, $q_3$ and $q_1$ in sequence, we obtain $\pi(q_0) = \pi(q_4) = \pi(q_8) = b$ and $\pi(q_2) = \pi(q_6) = \pi(q_{10}) = a$, as illustrated in Figure 3.5. This concludes the proof. □

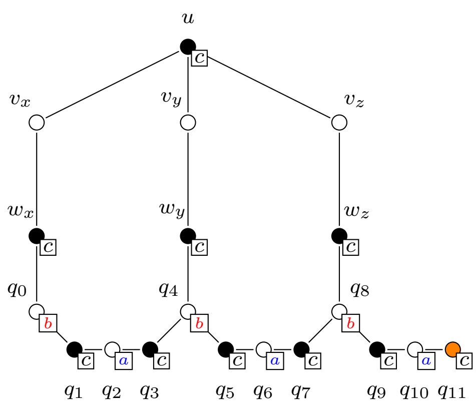



Figure 3.5: A (partial) labelling of $G^{\triangle}$ when $c_\pi(u) = 1$, with $c \in \{1, 2\}$. Vertices with induced colour 1 are filled in black, and vertices with colour 0, in white. Vertex $q_{11}$ is coloured in orange, implying $c_\pi(q_{11}) \in \{1, 2\}$.

Observe that no implication is made for the labels of vertices $v_x$, $v_y$ and $v_z$. In order to properly label these vertices so as to induce colouring $c_\pi$, we require the use of another gadget.

### 3.2.2 Negation gadget

The *negation gadget* $G^\neg$ is an auxiliary simple bipartite graph obtained by removing an edge $e$ from the Heawood Graph $G$, which is the cubic bipartite graph presented in Figure 2.3, and linking two new vertices, $v_{\text{in}}$ and $w$, to the ends of $e$. Let $V(G) = \{s_0, \ldots, s_{13}\}$ and $e = s_0s_9$. We also refer to vertex $s_9$ as $v_{\text{out}}$. This construction of $G^\neg$ yields a graph with 16 vertices and 22 edges. This gadget is illustrated in Figure 3.6(a).

The negation gadget is only used to connect vertices $v_i$ and $w_i$ that belong to triangle gadgets. These two vertices are identified with $v_{\text{in}}$ and $w$ from the negation gadget, respectively. Therefore, upon performing this operation, vertices $v_i$ and $w_i$ in the corresponding triangle gadgets have degree 3. Observe that the negation gadget contains no odd-length cycles and, therefore, is bipartite. We also remark that vertices $v_{\text{in}}$ and $v_{\text{out}}$ belong to the same part of any bipartition of $G^\neg$, as depicted in Figure 3.6(a). In order to simplify larger images further in this section, the negation gadget is illustrated by the symbol "$\neg$" in a box incident with doubled lines (not to be confused with parallel edges), linking vertices $v_{\text{in}}$ and $v_{\text{out}}$ with the box, as illustrated in 3.6(b).

**Property 3.3.** *Let $G$ be a subcubic bipartite graph with $G^\neg \subseteq G$ and $d(v_{\text{in}}) = d(w) = 3$. If $G$ admits a gap-[2]-vertex-labelling and $c_\pi(v_{\text{in}}) = c_\pi(v_{\text{out}}) = 0$, then $\pi(v_{\text{in}}) \neq \pi(v_{\text{out}})$.*

*Proof.* Let $G$ be a graph as stated in the hypothesis and $G^\neg = (V, E)$ be a negation gadget in $G$, with vertex set $V = \{v_{\text{in}}, w, s_0, \ldots, s_{13}\}$. Recall that vertex $s_9$ is also called $v_{\text{out}}$. Let $\{A, B\}$ be a bipartition of the negation gadget where one part, say $A$, comprises vertices $v_{\text{in}}$, $v_{\text{out}}$ and every $s_i$, $i$ odd, and the other, $B$, consists of the remaining vertices. Suppose $G$ admits a gap-[2]-vertex-labelling $(\pi, c_\pi)$. Then, since $d(v) = 3$, we have $c_\pi(v) \in \{0, 1\}$ for every vertex $v \in V(G^\neg)$.

Suppose $c_\pi(v_{\text{in}}) = 0$. Since $G^\neg$ is connected, all vertices in $A$ have colour 0. This implies that all vertices in $B$ receive the same label $c \in \{1, 2\}$. It remains to consider

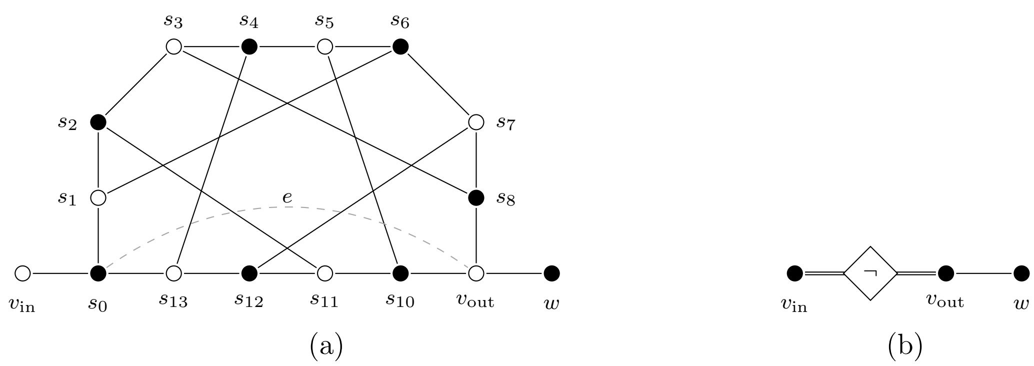



Figure 3.6: In (a), negation gadget $G^\neg$ and its vertex set partitioned into sets $A$ (in white) and $B$ (in black); and in (b), the representation of the gadget connecting vertices $v_{\text{in}}$ and $w$. The dashed edge in (a) represents the removed edge.

the labels of vertices in part $A$. Since $s_0 \in B$ and $c_\pi$ is a proper colouring of $G^\neg$ in colours $\{0,1\}$, we know that $c_\pi(s_0) = 1$. Recall that $N(s_0) = \{v_{\text{in}}, s_1, s_{13}\}$. Let $a, b \in \{1,2\}$, $a \neq b$, be the possible labels. Then, we have $\Pi_{N(s_0)} = \{a,b\}$.

Suppose $\pi(s_1) \neq \pi(s_{13})$, so that colour 1 is induced in $s_0$ regardless of the label assigned to $v_{\text{in}}$. Without loss of generality, let $\pi(s_1) = a$. Consider vertex $s_2$, and recall that $N(s_2) = \{s_1, s_3, s_{11}\}$. Since $s_2 \in B$, $c_\pi(s_2) = 1$, which also implies that $\{\pi(s_1), \pi(s_3), \pi(s_{11})\} = \{a,b\}$. This opens two possibilities for the label of vertices $s_3$ and $s_{11}$.

Suppose $\pi(s_3) = b$. In this case, knowing that $s_3, s_{13} \in N(s_4)$ and $\pi(s_3) = \pi(s_{13}) = b$, we conclude that $\pi(s_5) = a$ since $\pi(s_5) = b$ would induce $c_\pi(s_4) = 0$. Figure 3.7(a) illustrates this case. Now, since $s_1, s_5 \in N(s_6)$ and $c_\pi(s_6) = 1$, a similar reasoning allows us to conclude that $\pi(s_7) = b$, as illustrated in Figure 3.7(b). Analogously, we have $\pi(s_7) = \pi(s_{13}) = b$, $s_7, s_{13} \in N(s_{12})$ and $c_\pi(s_{12}) = 1$, which implies $\pi(s_{11}) = a$, as illustrated in Figure 3.7(c). The only remaining vertex to be considered is $v_{\text{out}}$. However, if $\pi(v_{\text{out}}) = a$, then $\pi(v_{\text{out}}) = \pi(s_5) = \pi(s_{11})$ and, since $\{v_{\text{out}}, s_5, s_{11}\} = N(s_{10})$, we have $c_\pi(s_{10}) = 0$, which is a contradiction. Otherwise, if $\pi(v_{\text{out}}) = b$, then, by a similar argument, $c_\pi(s_8) = 0$, which is also a contradiction. These analyses are illustrated in Figures 3.7(a), 3.7(b) and 3.7(c).

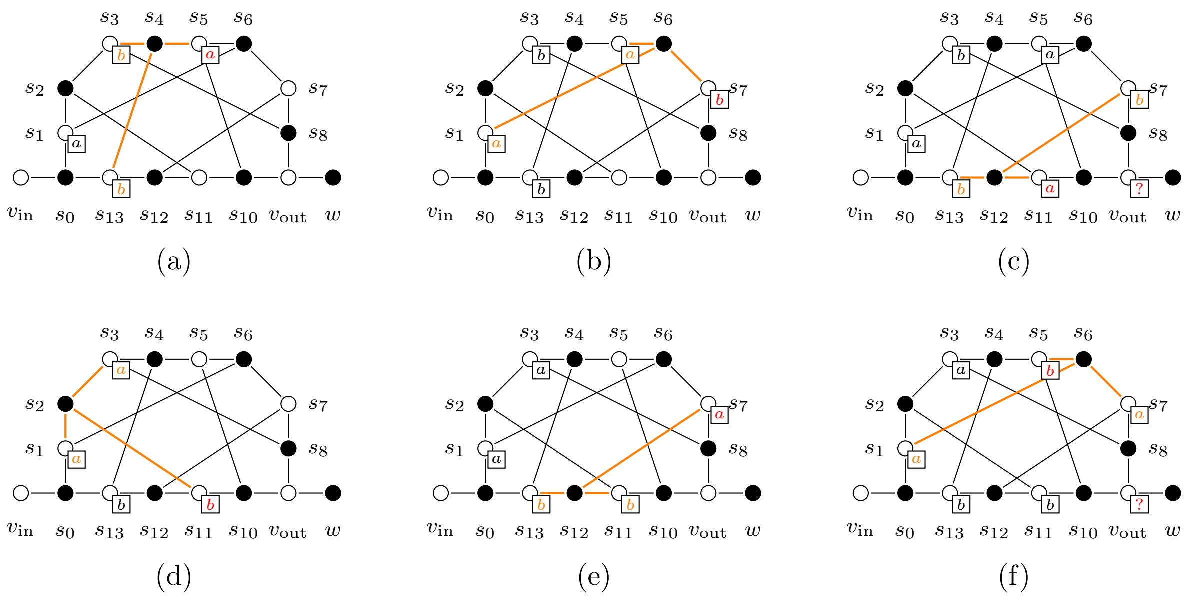



Figure 3.7: Case $\pi(s_1) \neq \pi(s_{13})$. In (a), (b), and(c), vertex $s_3$ has been assigned label $b$, which determines the assignment of labels to vertices $s_5$, $s_7$ and $s_{11}$, respectively; and in (d), (e) and (f), $s_3$ has label $a$ and the determined labels of vertices $s_{11}$, $s_7$ and $s_5$ are respectively illustrated. In all figures, edges and labels highlighted in orange are those that force labels, which are highlighted in red. Also, we denote black vertices as those with induced colour 1 and white vertices, colour 0.

Since considering $\pi(s_3) = b$ leads us to a contradiction, we conclude that $\pi(s_3) = a$. Then, by an analogous argument, we can determine the assignment of labels $\pi(s_{11}) = b$, $\pi(s_7) = a$, $\pi(s_5) = b$ in sequence, as represented in Figures 3.7(d), 3.7(e) and 3.7(f), respectively. Again, we have a contradiction in determining the label of $v_{\text{out}}$ and we conclude that $\pi(s_1) = \pi(s_{13})$.

Now, let $\pi(s_1) = \pi(s_{13}) = b$ and, consequently, $\pi(v_{\text{in}}) = a$. For the sake of contradiction, suppose $\pi(v_{\text{in}}) = \pi(v_{\text{out}})$ and consider vertex $s_{11}$, which belongs to $N(s_{12}) \cap N(s_{10})$. We have two possible labels for $s_{11}$: $a$ or $b$. First, suppose $\pi(s_{11}) = a$. Since $N(s_{10}) = \{s_5, s_{11}, v_{\text{out}}\}$ and $\pi(s_{11}) = \pi(v_{\text{out}}) = a$, in order to induce $c_\pi(s_{10}) = 1$, $\pi(s_5) = b$ as illustrated in Figure 3.8(a). However, $\pi(s_1) = \pi(s_5) = b$ and, since $N(s_6) = \{s_1, s_5, s_7\}$ and $c_\pi(s_6) = 1$, we conclude that $\pi(s_7) = a$. Figure 3.8(b) illustrates this case. The only remaining vertex to be considered is $s_3$. However, note that if label $a$ is assigned to $s_3$, then vertex $s_8$ would be coloured with 0 since $N(s_8) = \{s_3, s_7, v_{\text{out}}\}$ and, by our reasoning, $\pi(s_3) = \pi(s_7) = \pi(v_{\text{out}}) = a$. Otherwise, if $\pi(s_3) = b$, we have $\pi(s_3) = \pi(s_{13}) = \pi(s_5) = b$ which induces $c_\pi(s_4) = 0$ since $N(s_4) = \{s_3, s_5, s_{13}\}$. Both cases contradict the fact that $c_\pi$ is a proper colouring of $G^\neg$. We conclude that $\pi(s_{11}) \neq a$.

Thus, we can assume $\pi(s_{11}) = b$. However, since $\pi(s_{13}) = \pi(s_{11})$, by a similar argument, label $a$ is assigned to vertex $s_7$. Once again, we reach a contradiction upon determining $\pi(s_3)$: the assignment of label $b$ would induce $c_\pi(s_2) = 0$ and, if $\pi(s_3) = a$, then $c_\pi(s_8) = 0$. Both cases contradict the fact that $c_\pi$ is a proper colouring of the gadget, as illustrated in Figure 3.8(c). We conclude that $\pi(v_{\text{in}}) \neq \pi(v_{\text{out}})$. □

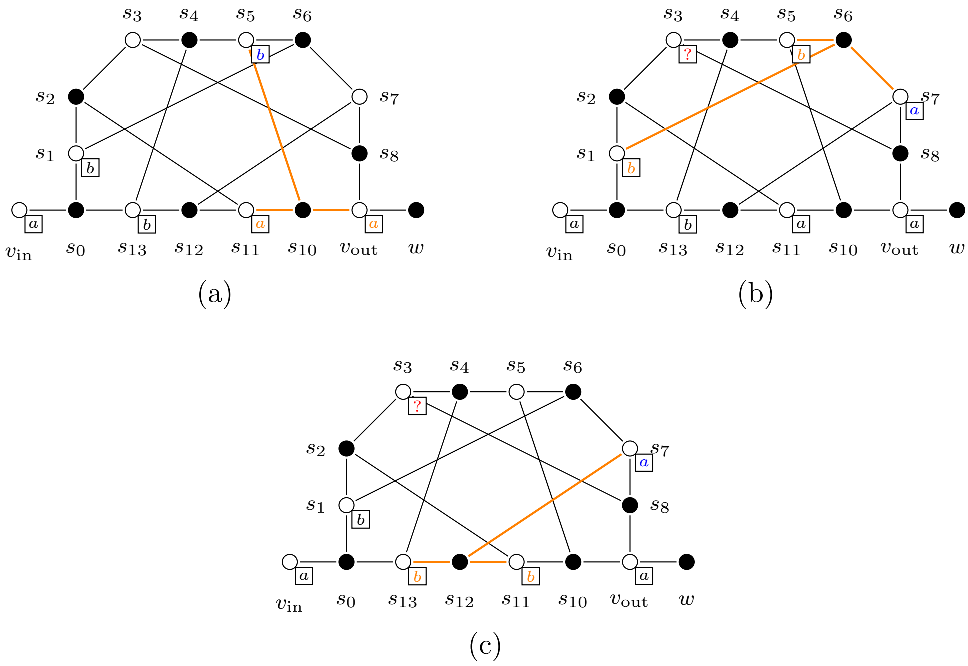



Figure 3.8: Case $\pi(v_{\text{in}}) = \pi(v_{\text{out}})$. In (a) and (b), labels for vertices $s_5$ and $s_7$ are determined by the established labels, while, in (c), the only determined label is $s_7$. In both cases, there is no label that can be assigned to vertex $s_3$ so as to induce a proper colouring of $G^\neg$. We denote vertices with induced colour 1 as those filled with black, and vertices coloured with 0, in white. Forced labels are highlighted in blue.

We are ready to reduce an instance of MONOCHROMATIC TRIANGLE, a known NP-complete problem, to **G2VL (SCB)**, in polynomial time. The details of the reduction are presented in the next section.

### 3.2.3 The reduction

Let $G = (V, E)$ be an instance of **MT**. We construct a bipartite graph $G' = (V', E')$, with $d(v) \leq 3$ for every $v \in V'$, from $G$ in polynomial time. We prove that $G$ admits a 2-edge-colouring without monochromatic triangles if and only if $G'$ admits a gap-[2]-vertex-labelling.

Let $p$ be the number of (not necessarily disjoint) triangles in $G$. We remark that it is possible to determine $p$ in $\mathcal{O}(n^3)$-time – one needs only check every possible combination of three distinct vertices in $V$, that is, $p \leq \binom{n}{3} = \mathcal{O}(n^3)$. Let $\mathcal{T} = \{t_1, t_2, \ldots, t_p\}$ be the set of the $p$ triangles in $G$. For every triangle $t_i \in \mathcal{T}$, $t_i = \{e_x, e_y, e_z\}$, add a new triangle gadget $G_i^{\triangle}$ to $G'$. Denote the vertices of $G_i^{\triangle}$ as $\{u^i, v_x^i, w_x^i, v_y^i, \ldots, q_0^i, \ldots, q_{11}^i\}$; observe that every edge $e_j$ of $t_i$, $j \in \{x, y, z\}$, has a corresponding vertex $v_j^i$. We refer to vertices $u^i$ as *triangle-vertices*, to vertices $v_x^i$, $v_y^i$ and $v_z^i$, as *e-vertices* and to vertices $w_x^i$, $w_y^i$ and $w_z^i$, as their respective *correspondents*. For every $1 \leq i \leq p-1$, connect vertices $q_{11}^i$ and $q_0^{i+1}$ with an edge. Also, add a copy of cycle $C_6 = \{c_0, \ldots, c_5\}$ to $G'$ and connect vertices $q_0^1$ and $c_0$. We exemplify this construction for a graph $G$ in Figure 3.9(a). This graph has $p = 3$ triangles: $t_1 = \{e_1, e_4, e_5\}$ (violet), $t_2 = \{e_2, e_3, e_5\}$ (orange) and $t_3 = \{e_5, e_6, e_7\}$ (cyan).

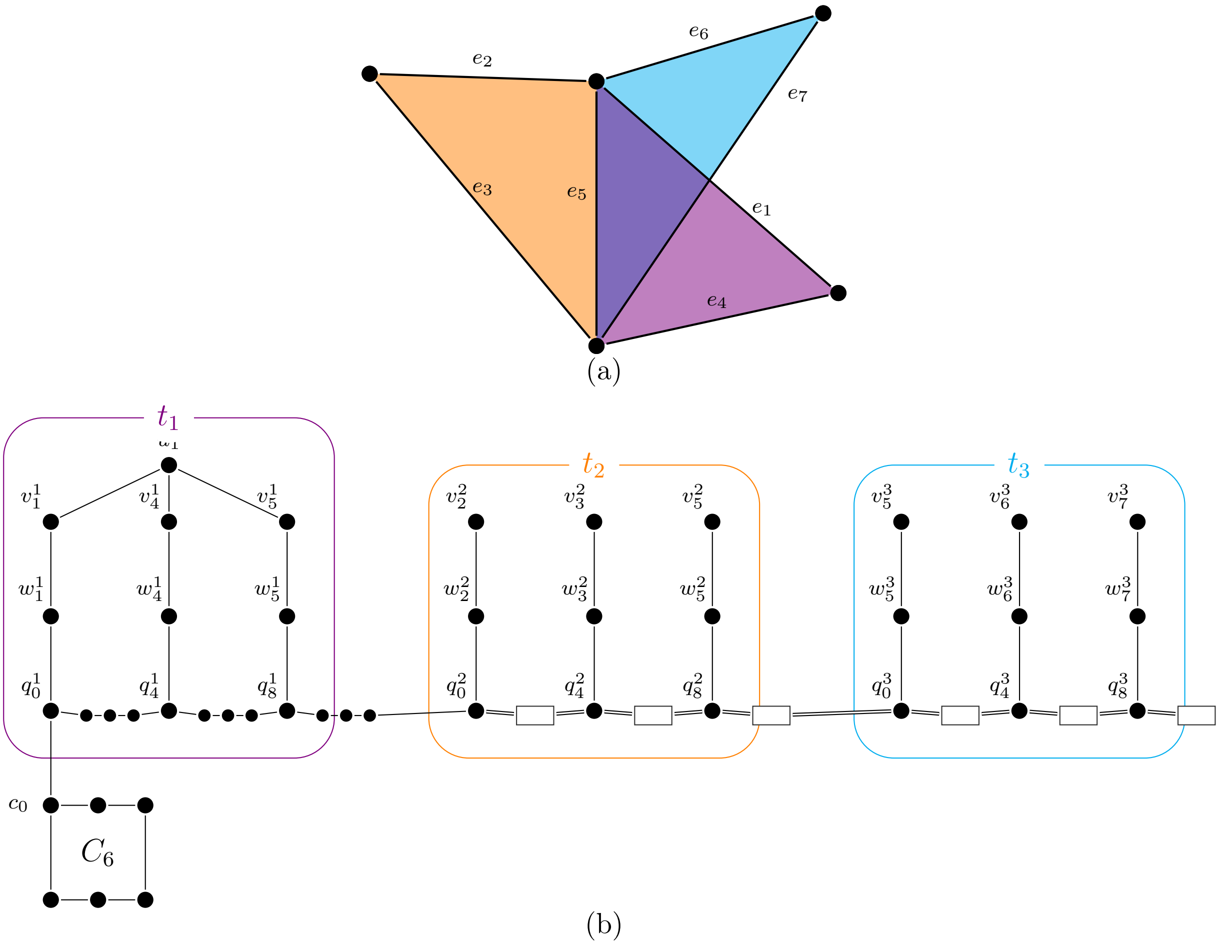



Figure 3.9: In (a), a graph $G$ with three triangles. In (b), the (initial) construction of graph $G'$.

Observe that this initial construction yields $d(u^i) = 3$ for every triangle-vertex, and $d(v^i) = d(w^i) = 2$ for every $e$-vertex and its correspondent. Also, with the exception of $d(q_{11}^p) = 1$, note that, for all paths $P_{12}$, $d(q_0^i) = d(q_4^i) = d(q_8^i) = 3$, and $d(q_j) = 2$ for every remaining vertex $q_j$. For the attached cycle, we have $d(c_0) = 3$, while every remaining vertex of $C_6$ has degree 2. Thus, this initial construction yields a graph with maximum degree three.

We complete the construction of $G'$ by connecting some vertices using negation gadgets. Every edge $e_x \in E$ belongs to, at most, $p_x \leq p$ triangles in $G$. Let $\mathcal{T}_x \subseteq \mathcal{T}$ be the set of triangles to which edge $e_x$ belongs to in $G$, and let $(t_1^x, t_2^x, \ldots, t_{p_x}^x)$ be an order of the elements of $\mathcal{T}_x$. Then, following this order of $\mathcal{T}_x$ cyclically, connect vertices $v_x^i$ and $w_x^{i+1}$ with a negation gadget, for every pair of consecutive triangle gadgets $G_i^\triangle$ and $G_{i+1}^\triangle$. This connection is done by identifying $v_x^i$ with $v_{\text{in}}$ and $w_x^{i+1}$ with $w$, respectively. Note that this operation adds exactly one edge to each $e$-vertex $v_x$ and to its correspondent $w_x$ in every triangle gadget, which yields $d(v_x) = d(w_x) = 3$. Since every vertex in $G^\neg$ has degree three, our construction of $G'$ yields a subcubic graph. Also, observe that the connections between triangle gadgets and negation gadgets do not create any odd-length cycles. Therefore, $G'$ is also bipartite. Figure 3.10 exemplifies this reduction process for a graph $G$, depicting the resulting subcubic bipartite graph $G'$.

In order to prove that **G2vL (ScB)** is NP-complete, we prove the following statement.

**Proposition 3.4.** *$G$ admits an edge-colouring $c : E \to \{\text{red}, \text{blue}\}$ such that no triangle is monochromatic if and only if $G'$ admits a gap-[2]-vertex-labelling.*

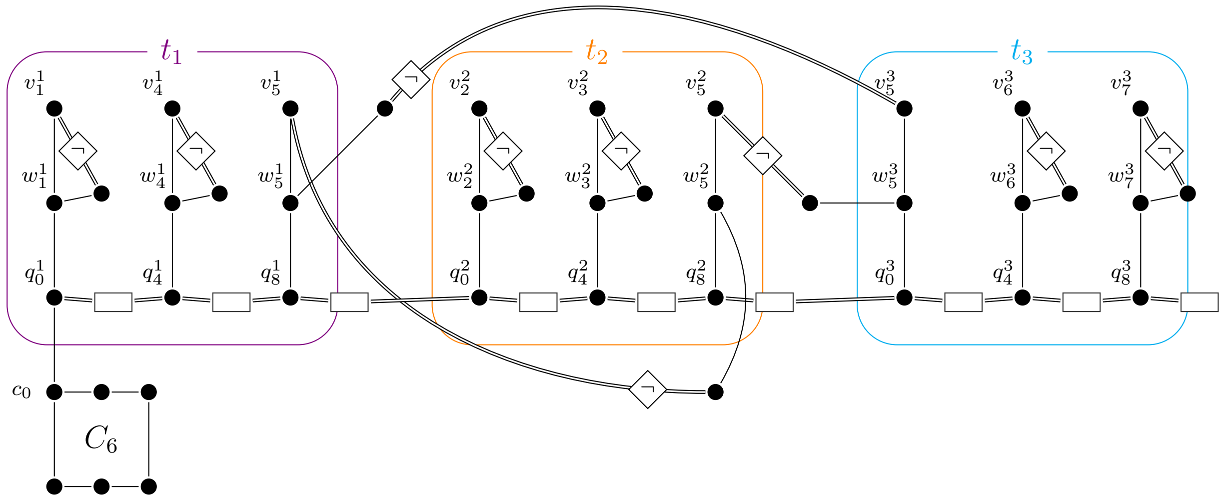



Figure 3.10: Graph $G'$. Observe the connections using negation gadgets exemplified by edge $e_5$, which belongs to all three triangles in $G$.

*Proof.* ($\Rightarrow$) Suppose $G$ admits an edge colouring $c : E \to \{\mathit{red}, \mathit{blue}\}$ such that there are no monochromatic triangles in $G$. Let $\{E_R, E_B\}$ be a partition of $E$, such that $E_R$ and $E_B$ are the sets of edges coloured with *red* and *blue*, respectively. We define a labelling $\pi : V \to \{1, 2\}$ of $G'$ as follows.

- For each triangle gadget $G_i^{\triangle}$ in $G'$, $1 \le i \le p$:
  - For every $e$-vertex $v_x^i$, assign $\pi(v_x^i) = 2$ if and only if $e_x \in E_B$;
  - assign label 2 to vertices $q_2$, $q_6$, $q_{10}$; and
  - label the remaining vertices in $G_i^{\triangle}$ with 1.

  Figure 3.11 illustrates this labelling for one of the triangle gadgets of $G'$ from Figure 3.10; note that $c_\pi(u_2) = 1$.

- For each negation gadget $G^{\neg}$, connecting vertices $v_i^j$ and $w_i^k$ of two triangle gadgets $G_j^{\triangle}$ and $G_k^{\triangle}$, label vertices $(s_0, \ldots, s_{13})$ as illustrated in Figure 3.12:
  - If $\pi(v_i^j) = 1$, assign labels $(1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2)$; and
  - if $\pi(v_i^j) = 2$, assign labels $(1, 1, 1, 2, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1)$.

- For cycle $C_6$, assign labels $(1, 2, 1, 1, 1, 2)$ to vertices $(c_0, c_1, c_2, c_3, c_4, c_5)$, respectively.

In order to prove that $(\pi, c_\pi)$ is a gap-[2]-vertex-labelling of $G'$, it suffices to show that $c_\pi$ is a proper colouring of $G'$. First, consider the attached cycle and observe that vertices $c_0, \ldots, c_5$ have induced colours alternating between 1 and 0, as depicted in Figure 3.13. For the negation gadgets, by inspection of Figure 3.12, we observe that both labellings induce a 2-colouring of every vertex $s_i \in V(G^{\neg})$. However, in order to determine the colours of vertices $v_{\text{in}}$ and $w$ of each negation gadget, we have to analyse the labellings of the triangle gadgets.

We start by considering paths $P_{12}$ in each $G_i^{\triangle}$. Except for $q_{11}^p$, which has induced colour 2, every $q_l^i$ with odd index $l$, $1 \le l \le 9$, has $\{\pi(q_{l-1}^i), \pi(q_{l+1}^i)\} = \{1, 2\}$; recall that $N(q_l^i) = \{q_{l-1}^i, q_{l+1}^i\}$. This implies that these vertices have $c_\pi(q_l^i) = 1$. For vertices $q_{11}^i$, with $1 \le i < p$, their neighbourhoods comprise vertices $q_{10}^i$ and $q_0^{i+1}$, which receive labels 2 and 1, respectively. Therefore, they also have induced colour 1. Finally, observe that all vertices $q_l^i$ with odd $l$ receive label 1, as well as every correspondent vertex $w_x^i$. Therefore, every even-index vertex $q_l^i$ has all neighbours labelled with 1, which implies that $c_\pi(q_l^i) = 0$ for every even $l$. We conclude that $c_\pi(q_l^i) = l \bmod 2$ for every vertex $q_l^i \neq q_{11}^p$.

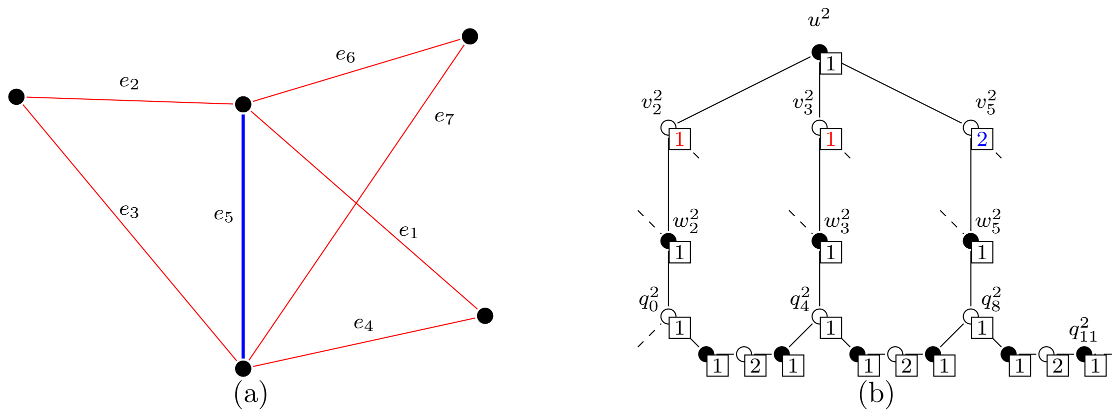



Figure 3.11: In (a), a 2-edge-colouring of $G$ without monochromatic triangles; and in (b), the corresponding labelling of the triangle gadget $G_2^{\triangle}$, constructed from triangle $t_2$.

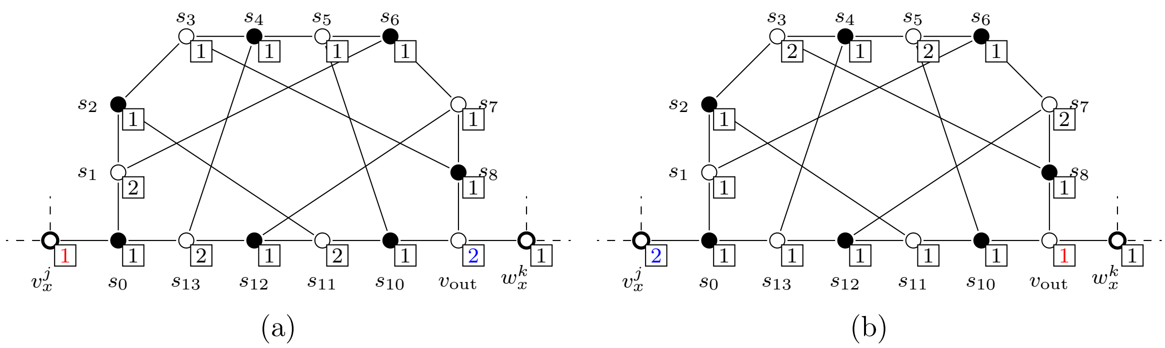



Figure 3.12: The labellings of negation gadget $G^{\neg}$ connecting vertices $v_x^j$ and $w_x^k$. In (a), $\pi(v_x^j) = 1$; and in (b), $\pi(v_x^j) = 2$. Recall that $v_{\text{in}}$ is identified with vertices $v_x^j$ from triangle gadgets. Vertices with induced colour 1 are depicted in black, and with colour 0, in white.

Next, consider the $e$-vertices, $v_x^i$, in gadgets $G_i^{\triangle}$. Recall that $N(v_x^i) = \{u^i, w_x^i, s_0\}$, where $s_0$ corresponds to a vertex in a negation gadget. Since labelling $\pi$ assigned $\pi(u^i) = \pi(w_x^i) = \pi(s_0) = 1$ for all gadgets, then $c_\pi(v_x^i) = 0$ for every $e$-vertex. Regarding the triangle vertices, $u^i$, since the triangle $t = \{e_x, e_y, e_z\}$ is not monochromatic in $G$, $\{\pi(v_x^i), \pi(v_y^i), \pi(v_z^i)\} = \{1, 2\}$ in every $G_i^{\triangle}$, which induces $c_\pi(u^i) = 1$.

For the triangle gadgets, it remains to consider the induced colour of every correspondent vertex $w_x^i$. Recall that $N(w_x^i) = \{v_x^i, v_{\text{out}}, q_l^i\}$, where $v_{\text{out}}$ is a vertex from a negation gadget, and $q_l^i$ is a vertex from $P_{12}$, with $l \equiv 0 \pmod 4$. For all edges $e_x \in G$, the corresponding $e$-vertices in each of the $p_x$ triangle gadgets $G_i^{\triangle}$ received label 1 if $e_x \in E_R$, and 2, otherwise. Also, since we have established that $c_\pi(v_x^i) = 0$ in every triangle gadget $G_i^{\triangle}$, we know that $\pi(v_x^i) \neq \pi(v_{\text{out}})$ by Property 3.3. Given that every correspondent vertex $w_x^i$ is adjacent to a vertex $v_{\text{out}}$ in a negation gadget which connects $w_x^i$ to some $e$-vertex $v_x^j \in G_j^{\triangle}$, we conclude that set $\Pi_{N(w_x^i)}$ contains $\{\pi(v_x^i), \pi(v_{\text{out}})\} = \{1, 2\}$. This implies that $c_\pi(w_x^i) = 1$ in every $G_i^{\triangle}$. We conclude that cycle $C_6$, every triangle gadget and every negation gadget has been properly coloured, and, thus, that $(\pi, c_\pi)$ is a gap-[2]-vertex-labelling of $G'$.

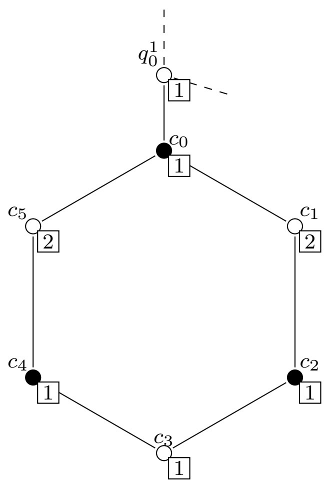


Figure 3.13: The labelling $\pi$ and induced colouring $c_\pi$ of cycle $C_6$.

(⇐) Conversely, suppose $G'$ admits a gap-[2]-vertex-labelling $(\pi, c_\pi)$. We prove that the original graph $G$ admits a 2-edge-colouring such that $G$ has no monochromatic triangles. Recall that $G'$ is bipartite and let $\{V_0, V_1\}$ be a partition of $V(G')$ such that, for every vertex $v \in V' \backslash \{q^p_{11}\}$, $v \in V_i$ if and only if $c_\pi(v) = i$. Since $G'$ is connected and every vertex $v \in V_0$ has $c_\pi(v) = 0$, it follows that all vertices in $V_1$ are assigned the same label $c \in \{1, 2\}$. Therefore, for the remaining figures in this proof, we omit the labels of the black vertices, that is, vertices that belong to $V_1$. This is done for the sake of clarity of the drawings. Now, we analyse the vertices of $V_0$.

First, consider cycle $C_6$ attached to triangle gadget $G^\triangle_1$ and suppose $c_3 \in V_1$, that is, $c_\pi(c_3) = 1$. (Recall that $c_0$ is adjacent to $q^1_0$, as sketched in Figure 3.10.) Observe that this implies that $c_\pi(c_1) = c_\pi(c_5) = 1$ because they belong to the same part as $c_3$. Additionally, $N(c_3) = \{c_2, c_4\}$, which implies $\{\pi(c_2), \pi(c_4)\} = \{1, 2\}$. Observe that, by symmetry of the cycle, we can assume, without loss of generality, that $\pi(c_2) = 1$ and $\pi(c_4) = 2$. This implies that $\pi(c_0) = 2$ and, considering $N(c_5)$, we conclude that $\pi(c_4) = 1$. This is a contradiction since we have already established $\pi(c_4) = 2$, as illustrated in Figure 3.14(a).

We conclude that $c_3 \notin V_1$ and, therefore, that $c_3 \in V_0$. We remark that $\{V_0, V_1 \cup \{q^p_{11}\}\}$ is a bipartition of $G'$: part $V_0$ comprises all $e$-vertices $v^i_x$ and every $q^i_l$ with $l$ even, for every triangle gadget $G^\triangle_i$, and every $s_j$, $j$ odd, in every negation gadget $G^\neg$; the other part, $V_1$, comprises every triangle-vertex $u^i$, every correspondent vertex $w^i_x$, every $q^i_l$, $l$ odd, and every $s_j$, with $j$ even. We return to our analysis of cycle $C_6$.

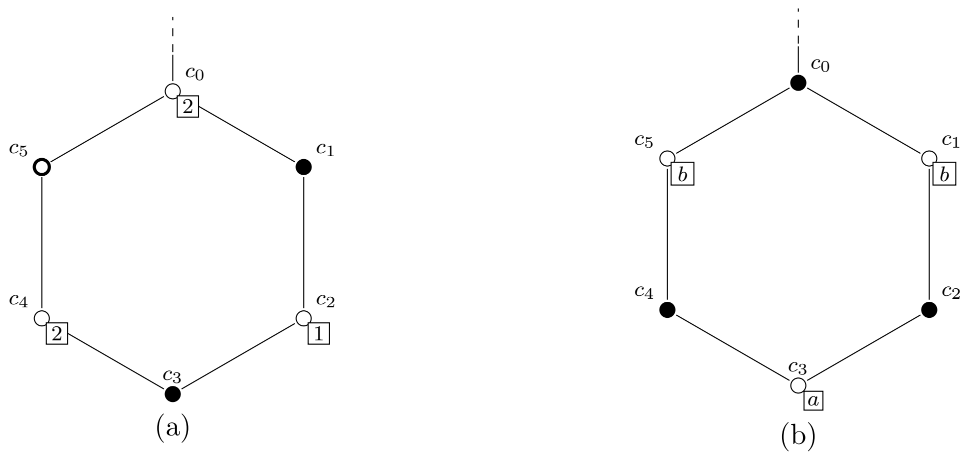
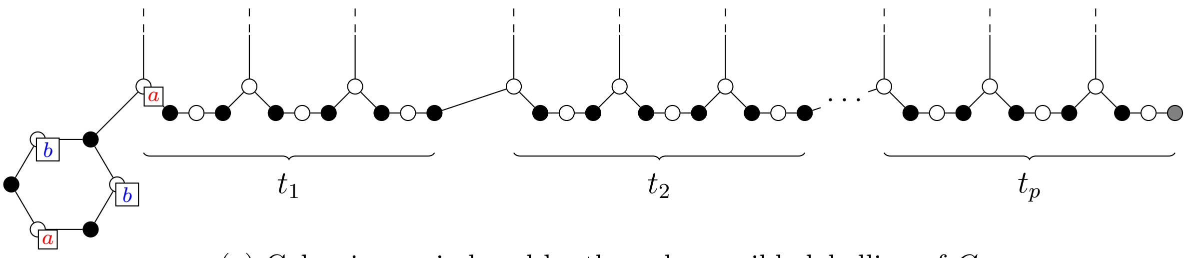


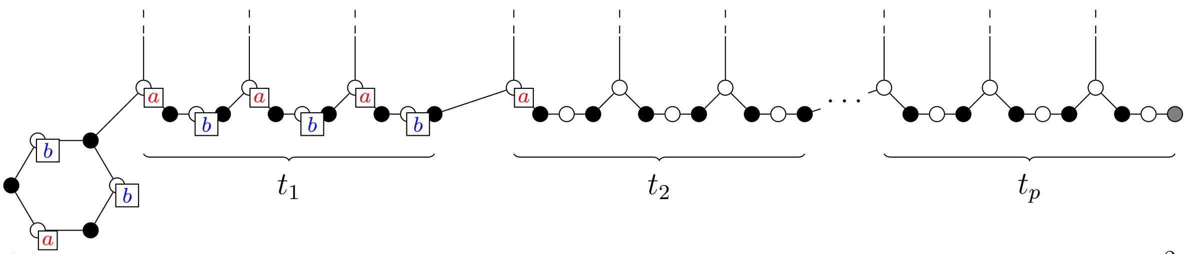

Figure 3.14: In (a), $c_3 \in V_1$; and in (b), $c_3 \in V_0$. Black vertices have induced colour 1 and white vertices, colour 0.

By the previous reasoning, $\pi(c_3) = a$, $a \in \{1, 2\}$, as depicted in Figure 3.14(b). Since $N(c_2) = \{c_1, c_3\}$, $\pi(c_1) = b$, $b \in \{1, 2\}$ and $b \neq a$. If we consider vertex $c_4$, by a similar argument, we have $\pi(c_5) = b$. Finally, since $\{c_1, c_5\} \subset N(c_0)$, $\pi(c_1) = \pi(c_5)$ and $c_\pi(c_0) = 1$, we conclude that $\pi(q^1_0) = a$. Recall that $q^1_0$ is the first vertex in path $P_{12}$ in the first triangle gadget of $G'$.

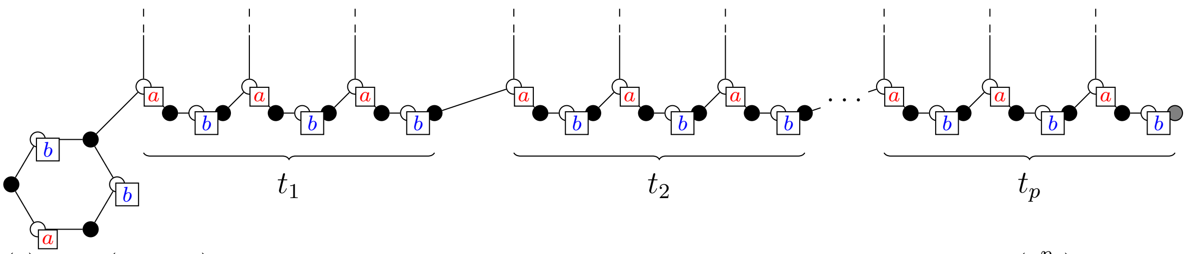
Consider the first triangle gadget $G^\triangle_1$ and observe path $P_{12}$ in it. Also, recall that $c_\pi(q^i_l) = l \bmod 2$, $1 \leq l \leq p$, except, perhaps, for $q^p_{11}$ since $d(q^p_{11}) = 1$. By Property 3.2, we know that $\pi(q^1_0) = \pi(q^1_4) = \pi(q^1_8) = a$. This implies that $\pi(q^1_{10}) = b$ since $c_\pi(q^1_9) = 1$. Now, consider vertex $q^1_{11}$. Since $N(q^1_{11}) = \{q^1_{10}, q^2_0\}$ and $c_\pi(q^1_{11}) = 1$, $\pi(q^2_0) \neq \pi(q^1_{10})$ and, therefore, $\pi(q_0^2) = a$. Upon following the order of triangle gadgets, by an analogous reasoning, we conclude that $\pi(q_l^i) = a$ for every vertex $q_l^i$ of triangle gadgets $G_i^\triangle$, when $l \equiv 0 \pmod 4$. This conclusion can be observed in Figure 3.15.




(a) Colouring $c_\pi$ induced by the only possible labelling of $C_6$.




(b) Property 3.2 applied in path $P_{12}$ in the first triangle gadget and the label $a$ applied to $q_0^2$ so as to induce $c_\pi(q_{11}^1) = 1$.




(c) The (partial) labelling $\pi$ of paths $P_{12}$ in all triangle gadgets. Observe that $c_\pi(q_{11}^p) = b$. Since $b \in \{1, 2\}$, $c_\pi(q_{10}^p) \neq c_\pi(q_{11}^p)$.

Figure 3.15: Illustrating the labels of vertices in paths $P_{12}$.

We have, thus far, established the label of every vertex in $V_1$ and of the vertices of cycle $C_6$ and paths $P_{12}$. It remains to consider the labels of the $e$-vertices $v_x^i$ of each $G_i^\triangle$ and of vertices $s_l \in V_0$ of each negation gadget $G^\neg$. In order to do this, consider the correspondent vertices, $w_x^i$, in triangle gadgets $G_i^\triangle$. Recall that $N(w_x^i) = \{v_x^i, v_{\text{out}}, q_l^i\}$, with $l \equiv 0 \pmod 4$; also, the labelling of paths $P_{12}$ implies that $\pi(q_l^i) = a$ for these vertices. Therefore, every correspondent vertex is adjacent to a vertex labelled with $a$. Since $(\pi, c_\pi)$ is a gap-[2]-vertex-labelling of $G'$, we know that $b \in \{\pi(v_{\text{out}}), \pi(v_x^i)\}$. Now, we take into consideration the number of triangles $p_x$ to which edge $e_x$ belongs to in the original graph $G$.

We recall some definitions used in the construction of $G'$. An edge $e_x$ belongs to $p_x \leq p$ triangles in $G$, and $\mathcal{T}_x \subseteq \mathcal{T}$ is the subset of triangles of $G$ to which edge $e_x$ belongs to. Also, $(t_1^x, \ldots, t_{p_x}^x)$ is an ordering of $\mathcal{T}_x$ used to connect the corresponding triangle gadgets in $G'$; this connection is done by identifying vertices $v_{\text{in}}$ and $w$ in each negation gadget with vertices $v_x^j$ and $w_x^{j+1}$, following the cyclic order of $\mathcal{T}_x$.

Consider the case where edge $e_x$ belongs to a single triangle in $G$, that is, $p_x = 1$. Then, the negation gadget connects vertices $v_x^i$ and $w_x^i$ in the same triangle gadget $G_i^{\triangle}$. Since $v_x^i, v_{\text{out}} \in V_0$, by Property 3.3, we have $\pi(v_{\text{in}}) \neq \pi(v_{\text{out}})$. Therefore, we know that $\{\pi(v_x^i), \pi(v_{\text{out}}), \pi(q_l^i)\} = \{1, 2\}$, which implies that $c_\pi(w_x^i) = 1$. Figure 3.16 illustrates a labelling of these vertices where $v_x^i$ received label $a$. Notice that if this were not the case, that is, if $\pi(v_x^i) = b$, the same result would follow since this would imply that $\pi(v_{\text{out}}) = a$. Thus, it remains to consider when an edge $e_x$ belongs to more than one triangle in $G$, that is, $p_x \geq 2$. In order to do this, we prove the following claim.

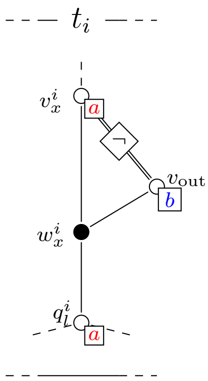



Figure 3.16: The (partial) labelling of the neighbours of a correspondent vertex $w_x^i$ in a triangle gadget $G_i^{\triangle}$ of $G'$. Black vertices represent vertices with induced colour 1, and white vertices, colour 0.

**Claim 3.5.** *Let $e_x$ be an edge of $G$ that belongs to $p_x \geq 2$ triangles in $G$. Let $\mathcal{T}_x$ be the order of these triangles used in the construction of $G'$, with $t_j^x$ denoting the $j$-th triangle in $\mathcal{T}_x$. If $G_i^{\triangle}$ corresponds to triangle $t_j^x$ rename: $G_i^{\triangle}$ to $G_{x,j}^{\triangle}$, and $v_x^i$, to $v^{x,j}$. Then, $\pi(v^{x,j}) = \pi(v^{x,j+1})$ for every pair of triangle gadgets $G_{x,j}^{\triangle}, G_{x,j+1}^{\triangle}$ representing consecutive triangles $t_j^x, t_{j+1}^x$ in $\mathcal{T}_x$.*

*Proof.* Let $G$, $G'$, $(\pi, c_\pi)$ and $e_x$ as stated in the hypothesis. Consider the $p_x$ triangle gadgets, $G_{x,j}^{\triangle}$, which have vertices $v^{x,j}$ corresponding to $e_x$. Then, every $e$-vertex $v^{x,j}$ is connected to $w^{x,j+1}$ through the use of a negation gadget, and its correspondent vertex $w^{x,j}$, to $v^{x,j-1}$. Also, recall that: every correspondent vertex $w^{x,j}$ is adjacent to vertices $v_{\text{out}}$, $v^{x,j}$ and $q_l^{x,j}$, with $l \equiv 0 \pmod 4$, and these $q_l^{x,j}$ have been labelled with the same $a \in \{1, 2\}$.

Suppose that there exist $v^{x,j}$ and $v^{x,j+1}$ for which $\{\pi(v^{x,j}), \pi(v^{x,j+1}\} = \{a, b\}$. Adjust notation so that $\pi(v^{x,j}) = b$. By Property 3.3, $\pi(v^{x,j}) = \pi(v_{\text{in}}^{x,j}) = b \neq \pi(v_{\text{out}})$ for $v_{\text{out}}$ connecting $v^{x,j}$ and $w^{x,j+1}$. Therefore, $\Pi_{N(w^{x,j+1})} = \{a\}$, which induces $c_\pi(w^{x,j+1}) = 0$. This is a contradiction since $w^{x,j+1} \in V_1$. Figure 3.17 depicts this analysis for an arbitrary $e$-vertex $v_x$. We conclude that in a gap-[2]-vertex-labelling of $G'$, every $e$-vertex $v^{x,j}$ corresponding to an edge $e_x \in E(G)$ has received the same label, and the result follows. $\square$

Therefore, every $e$-vertex $v_x$, which corresponds to an edge $e_x$ in a triangle $t$ of $G$, has received the same label – either $a$ or $b$ – regardless of how many triangles it belongs to in the original graph. Moreover, since $(\pi, c_\pi)$ is a gap-[2]-vertex-labelling of $G$ and $c_\pi(v_x) = 0$

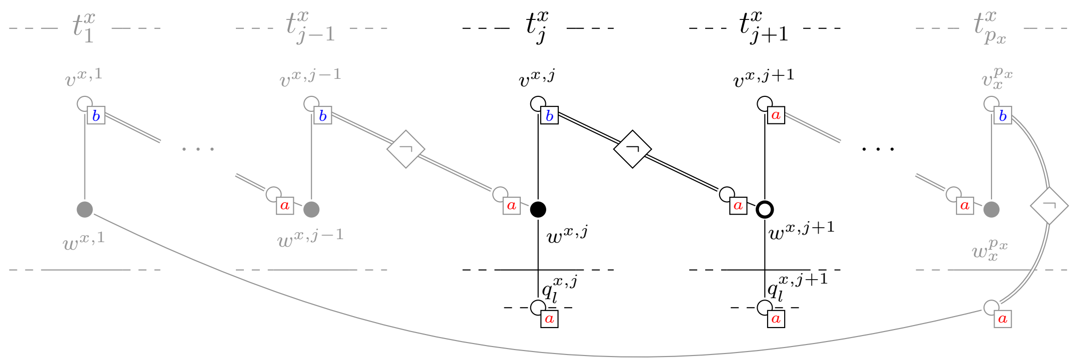



Figure 3.17: Two $e$-vertices $v^{x,j}$ and $v^{x,j+1}$ labelled with $b$ and $a$, respectively. The contradiction is reached when observing the correspondent vertex $w^{x,j+1}$ which, in this labelling, would have induced colour 0.

for every $e$-vertex $v_x$, we conclude that $c_\pi(u^i) = 1$ for every triangle gadget $G_i^\triangle$. This implies that in every $G_i^\triangle$, $\{\pi(v_x^i), \pi(v_y^i), \pi(v_z^i)\} = \{a, b\}$.

Define an edge-colouring $c : E \to \{red, blue\}$ of $G$ as follows. Assign colour *red* to edge $e_x$ if the corresponding $e$-vertices $v_x$ are labelled with $a$; assign colour *blue* to the remaining edges. We remark that no edge $e_x$ of $G$ was assigned two colours since Claim 3.5 ensures that every $e$-vertex $v_x$ received the same label in $G'$. Furthermore, since $\{\pi(v_x^i), \pi(v_y^i), \pi(v_z^i)\} = \{a, b\}$ for every triangle gadget $G_i^\triangle$, then, for every triangle $t_i$ of $G$, $\{c(e_x), c(e_y), c(e_z)\} = \{red, blue\}$. Hence, no triangle is monochromatic. This completes the proof. $\square$

Observe that the only vertex in $G'$ with degree one is $q_{11}^p$, i.e. the last vertex of path $P_{12}$ in the last triangle gadget $G_p^\triangle$. If we remove vertices $q_9^p$, $q_{10}^p$ and $q_{11}^p$, the resulting graph $G'$ would have $\delta(G') = 2$. Moreover, note that this modification to the constructed graph does not alter any structural properties of the gadgets. Therefore, it is possible to adapt the demonstration of Theorem 3.1 to this new graph, which unfolds in a second result.

**Corollary 3.6.** **G2VL** *remains* NP*-complete when restricted to bipartite graphs $G$ with $\delta(G) = 2$ and $\Delta(G) = 3$.* $\square$

Corollary 3.6 indicates that degree-one vertices seem to have no impact on the hardness of **G2VL**, that is, their existence in a graph neither facilitates nor hinders the existence of a gap-[2]-vertex-labelling. This result, however, opposes the intuition we gained upon studying the gap-[$k$]-vertex-labelling of some families of graphs. For the classes we addressed, the presence of low-degree vertices seemed to facilitate the labelling. On the other hand, in the edge-version of gap labellings, the role of degree-one vertices seems to be in the opposite direction. For instance, deciding whether a planar bipartite graph $G$ with $\delta(G) \geq 2$ admits a gap-[2]-edge-labelling can be solved in polynomial time but if the existence of degree-one vertices is allowed, the problem becomes NP-complete.

## 3.3 The vertex-gap number, $\chi_V^{\mathrm{g}}$, for classes of graphs

The gap-$[k]$-vertex-labelling problem is relatively new in the field of proper labellings and, with the exception of trees and $r$-regular bipartite graphs, $r \geq 4$, there are no known results for the vertex-gap number, even for classic families of graphs. Therefore, in this section, we determine this parameter for cycles, crowns, wheels, unicyclic graphs and some families of snarks. We also present some progress in the establishment of the vertex-gap number of cubic bipartite hamiltonian graphs.

Initially, we establish a lower bound for the vertex-gap number of arbitrary graphs. As previously mentioned, an upper bound for this parameter is presented in Chapter 4.

**Theorem 3.7.** *Let $G$ be a connected simple graph. If $G$ admits a gap-$[k]$-vertex-labelling, $k \in \mathbb{N}$, then $\chi_V^g(G) \geq \chi(G)$, unless $G \cong S_n$, $n \geq 2$, in which case $\chi_V^g(G) = \chi(G) - 1 = 1$.*

*Proof.* Let $G$ be a connected simple graph that admits a gap-$[k]$-vertex-labelling, $k \in \mathbb{N}$. First, consider the case $G \cong S_n$, $n \geq 2$. Recall that $V(S_n) = \{v_0, v_1, \ldots, v_n\}$, where $v_n$ is the central vertex. Therefore, $d(v_i) = 1$ for every $i < n$, $d(v_n) = n \geq 2$. Define a labelling $\pi$ of $G$ by assigning 1 to every vertex of $G$ and define colouring $c_\pi$ as usual. This induces $c_\pi(v_i) = \pi(v_n) = 1$ for $i < n$, and $c_\pi(v_n) = 0$, as illustrated in Figure 3.18. We conclude that $(\pi, c_\pi)$ is a gap-[1]-vertex-labelling of $G$ and, thus, $\chi_V^g(G) = 1 = \chi(G) - 1$.

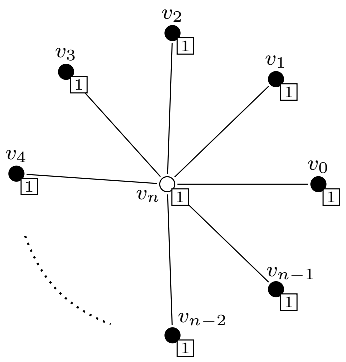



Figure 3.18: The gap-[1]-vertex-labelling of star $S_n$. The central vertex, $v_n$, has induced colour 0 (in white), and the remaining vertices, colour 1 (in black).

Next, consider the cases of $G \cong S_0$ and $G \cong S_1$. Graph $S_0$ is a trivial graph, in which case assigning 1 to its vertex induces a proper colouring of $G$ using $\chi(G) = 1$ labels. For $n = 1$, note that $G \cong K_2$. Since both vertices, $v_0$ and $v_1$, have degree one, their induced colours are equal to the label of their respective neighbours. By assigning label 1 to both vertices, the induced colouring is not proper. Thus, $\chi_V^{\mathrm{g}}(G) > 1$ and assigning labels 1 and 2 to $v_0$ and $v_1$, in any order, induces a proper colouring of $G$. Therefore, $\chi_V^{\mathrm{g}}(G) = \chi(G)$ in cases $G \cong S_0$ and $G \cong S_1$.

It remains to consider the case $G \not\cong S_n$. Let $(\pi, c_\pi)$ be a gap-$[\chi_V^{\mathrm{g}}(G)]$-vertex-labelling of $G$ and let $\mathcal{C}$ be the set of induced colours of $c_\pi$. Since $c_\pi$ is a vertex-colouring of $G$, we know that $|\mathcal{C}| \geq \chi(G)$.

*Case 1. There exists $v \in V(G)$ such that $c_\pi(v) \geq \chi(G)$.*

First, suppose $d(v) = 1$ and let $N(v) = \{u\}$. Then, we have $c_\pi(v) = \pi(u)$, which implies that $\chi^{\mathrm{g}}_V(G) \geq \pi(u) \geq \chi(G)$, and the result follows. Otherwise, that is, if $d(v) \geq 2$, then $c_\pi(v) = \pi(u) - \pi(w)$, where $\pi(u) = \max\{\pi(x) : x \in N(v)\}$ and, analogously, $\pi(w) = \min\{\pi(x) : x \in N(v)\}$. By our hypothesis, $c_\pi(v) \geq \chi(G)$ and, therefore, $\pi(u) \geq \chi(G) + \pi(w) \geq \chi(G) + 1$. We conclude that $\chi^{\mathrm{g}}_V(G) \geq \chi(G)$.

*Case 2. For every $v \in V(G)$, $c_\pi(v) < \chi(G)$.*

In this case, $\mathcal{C} = \{0, 1, \ldots, \chi(G) - 1\}$ since it is not possible to have a proper vertex-colouring of $G$ with less than $\chi(G)$ colours. Let $L$ be the set of vertices $v \in V(G)$ with induced colour $c_\pi(v) = \chi(G) - 1$.

Suppose there exists $v \in L$ with $d(v) \geq 2$. Then, $c_\pi(v) = \pi(u) - \pi(w)$, where $\pi(u)$ and $\pi(w)$ are defined as in the previous case. Since $c_\pi(v) = \chi(G) - 1$ and $\pi(w) \geq 1$, we conclude that $\pi(u) \geq \chi(G)$. Therefore, $\chi^{\mathrm{g}}_V(G) \geq \pi(u) \geq \chi(G)$.

Now, we can assume that every vertex $v \in L$ has $d(v) = 1$. Let $G' = G - L$ and let $c_{\pi'}$ be the restriction of $c_\pi$ to $V(G')$. Note that $c_{\pi'}$ is a proper colouring of $G'$ since $G' \subseteq G$. Furthermore, there is no vertex in $G'$ with colour $\chi(G) - 1$. Hence, $c_{\pi'}$ is a proper $(\chi(G) - 1)$-colouring of $G'$.

Since $G \ncong S_n$, $|V(G')| \geq 2$. Also, by hypothesis, $G$ is connected. Moreover, $G'$ is obtained from $G$ by removing only degree-one vertices, which implies that $G'$ is also connected. Hence, we know that $\chi(G') \geq 2$. Now, observe that colouring $c_{\pi'}$ can be expanded to $G$ using the same set of colours, assigning to each vertex $v \in L$ a different colour from that of its neighbour in $G$. This implies that $G$ is $(\chi(G) - 1)$-colourable – a contradiction. This completes the proof. □

With Theorem 3.7 established, the following corollary naturally holds.

**Corollary 3.8.** *Let $G \ncong S_n$, $n \geq 2$, be a simple graph. If $G$ admits a gap-$[\chi(G)]$-vertex-labelling, then $\chi^{g}_V(G) = \chi(G)$.* □

In the following sections we present the results obtained from our study of the vertex-gap number for some traditional families of graphs. Our method for determining $\chi^{\mathrm{g}}_V(G)$ for these graphs was: first, verify if and when graphs belonging to the studied family admit gap-$[\chi(G)]$-vertex-labellings; if this approach fails, we search for properties and characteristics of each family that interfere with the existence of labellings using at most $\chi(G)$ labels. The first family of graphs presented, cycles, exemplifies the fact that the vertex-gap number is not always equal to the chromatic number of the graph.

### 3.3.1 Cycles

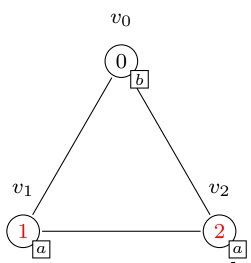
The family of cycles is introduced in Chapter 1. It is a well known result[3] that $\chi(C_n) = 2$ when $n$ is even, and $\chi(C_n) = 3$, otherwise. Considering the lower bound for the vertex-gap number, we ask whether even cycles admit a gap-[2]-vertex-labelling and odd cycles,

[3] The proof of this result is presented in Chapter 1, Theorem 1.2.

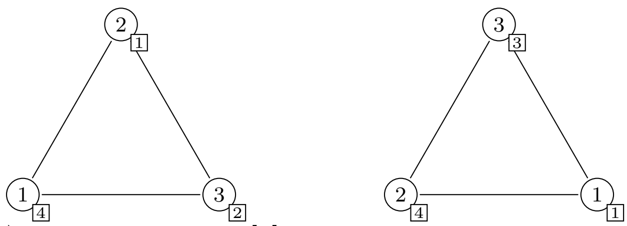
a gap-[3]-vertex-labelling. The answer to the latter is negative: odd cycle $C_3$ does not admit a gap-[3]-vertex-labelling.

**Property 3.9.** *Let $G \cong C_3$. Then, $\chi^g_V(G) = 4$.*

*Proof.* Let $G = C_3$ and let $v_0, v_1$ and $v_2$ be its vertices. Suppose $G$ admits a gap-[3]-vertex-labelling. Since $d(v_i) = 2$ for every $v_i$ and the set of labels is $\{1, 2, 3\}$, there is no way to induce colour 3 in any vertex of $G$. Therefore, $c_\pi : V(G) \to \{0, 1, 2\}$, which implies that, without loss of generality, vertices $v_0$, $v_1$ and $v_2$ are coloured with 0, 1 and 2. respectively. This configuration is illustrated in Figure 3.19(a). Observe that the only way to induce $c_\pi(v_0) = 0$ would be to assign both its neighbours the same label $a \in \{1, 2, 3\}$. However, for any label $b \in \{1, 2, 3\}$ assigned to vertex $v_0$, we would have $c_\pi(v_1) = |a - b|$ and $c_\pi(v_2) = |a - b|$, contradicting the fact that $c_\pi(v_1) \neq c_\pi(v_2)$ in any proper colouring of $C_3$. Therefore, no such gap-[3]-vertex-labelling of this cycle exists and, thus, $\chi^g_V(C_3) \geq 4$. We conclude the proof exhibiting two gap-[4]-vertex-labellings for $C_3$ in Figure 3.19(b). $\square$



(a) Cycle $C_3$ does not admit a gap-[3]-vertex-labelling.



(b) Two distinct gap-[4]-vertex-labellings of cycle $C_3$.

Figure 3.19: The labellings of cycle $C_3$ for $k = 3$ and $k = 4$.

For larger values of $n$, the vertex-gap number of $C_n$ is established in Theorem 3.10. Note that the result is, in fact, different from our first conjecture.

**Theorem 3.10.** *Let $G \cong C_n$, $n \geq 4$. Then, $\chi^g_V(G) = 2$, if $n \equiv 0 \pmod 4$, and $\chi^g_V(G) = 3$, otherwise.*

*Proof.* Let $G \cong C_n$, $n \geq 4$, with $V(G) = \{v_0, v_1, \ldots, v_{n-1}\}$. We remark that operations on the indices of the vertices are taken modulo $n$. As previously stated, $\chi(C_n) = 2$ when $n$ is even, and $\chi(C_n) = 3$, otherwise. Therefore, by Corollary 3.8, in order to prove the result, we have to show that: (i) there exists a gap-[2]-vertex-labelling of $C_n$ when $n \equiv 0 \pmod 4$; (ii) if the length of the cycle is $n \equiv 2 \pmod 4$, then there is no gap-[2]-vertex-labelling of $C_n$; and (iii) $C_n$ admits a gap-[3]-vertex-labelling for all $n \geq 4$.

We prove item (i) by providing a gap-[2]-vertex-labelling of cycle $C_n$, when $n \equiv 0 \pmod 4$. Define labelling $\pi$ of the vertices of $G$ as follows:

$$\pi(v_i) = \begin{cases} 2, & \text{if } i \equiv 0 \pmod 4; \\ 1, & \text{otherwise.} \end{cases}$$

Define colouring $c_\pi$ as usual. This labelling is exemplified for cycles $C_8$ and $C_{12}$ in Figure 3.20. Consider $i$ odd and $j$ even such that $0 \leq i, j < n$. Observe that every vertex $v_i$ receives label 1. Therefore, every vertex $v_j$ has $\Pi_{N(v_j)} = \{1\}$, which induces $c_\pi(v_j) = 0$. Moreover, following the cyclic order of the graph starting at vertex $v_0$, the labels of vertices with even index alternate between 2 and 1. Furthermore, since $n \equiv 0 \pmod 4$, every vertex $v_i$ has $\Pi_{N(v_i)} = \{1, 2\}$, which induces $c_\pi(v_i) = 1$. Therefore, we conclude that $c_\pi(v_l) = l \bmod 2$ for every vertex $v_l \in G$. Consequently, $(\pi, c_\pi)$ is a gap-[2]-vertex-labelling of $G$ in this case.

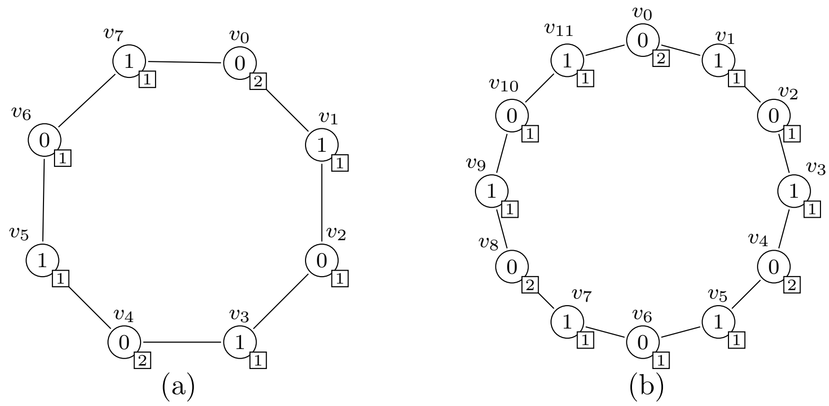


Figure 3.20: The gap-[2]-vertex-labelling of cycles $C_8$ and $C_{12}$ in (a) and (b), respectively.

For item (ii), we prove that cycle $C_n$, $n \equiv 2 \pmod 4$, does not admit a gap-[2]-vertex-labelling $(\pi, c_\pi)$. Suppose, by contradiction, it does. Since the labelling uses only labels 1 and 2 and there are no vertices of degree 1, the only induced colours of the vertices of $C_n$ are 0 and 1. Moreover, since $C_n$ is bipartite, these colours alternate along the vertices of the cycle. Adjust notation so that $c_\pi(v_l) = l \bmod 2$, for $0 \leq l < n$.

Once again, consider $i$ odd and $j$ even such that $0 \leq i, j < n$. Since $c_\pi(v_j) = 0$, $\pi(v_{j-1}) = \pi(v_{j+1}) = a$, for $a \in \{1, 2\}$. Moreover, since $N(v_j) \cap N(v_{j+2}) = \{v_{j+1}\}$ and $c_\pi(v_{j+2}) = 0$, we conclude that $\pi(v_{j+3}) = \pi(v_{j+1}) = a$. By following the cyclic order of the vertices with even index in $G$, we conclude that every vertex $v_i$ has the same label $a \in \{1, 2\}$. It remains to consider the labels of vertices $v_j$.

For every vertex $v_i$, we have $\Pi_{N(v_i)} = \{1, 2\}$ since $c_\pi(v_i) = 1$. Once again, considering the intersecting neighbourhoods between two consecutive vertices $v_i$ and $v_{i+2}$, we conclude that the labels of vertices with even index alternate between 1 and 2 along the cycle. This implies that every sequence of four vertices $(v_{i-1}, v_i, v_{i+1}, v_{i+2})$, starting with some odd $i$, is labelled with either $(a, a, b, a)$ or $(b, a, a, a)$, for $\{a, b\} \in \{1, 2\}$ and $a \neq b$. Moreover, the distance between any two consecutive vertices $u, v \in V(C_n)$ with label $b$ is exactly four. Suppose sequence $(a, a, b, a)$ starts at $v_0$ and repeats itself along the cycle.

Since $n \equiv 2 \pmod 4$, $\pi(v_0) = \pi(v_{n-2}) = a$ because $0 \equiv 0 \pmod 4$ and $n - 2 \equiv 0 \pmod 4$. Also, $\pi(v_{n-1}) = \pi(v_1) = a$. Therefore, $c_\pi(v_0) = c_\pi(v_{n-1}) = 0$, which contradicts the fact that $c_\pi$ is a proper colouring of $G$. This implication is illustrated in Figure 3.21.

We conclude that there is no gap-[2]-vertex-labelling of $C_n$ in this case. If the sequence is $(b, a, a, a)$, we reach a similar contradiction with the same reasoning.

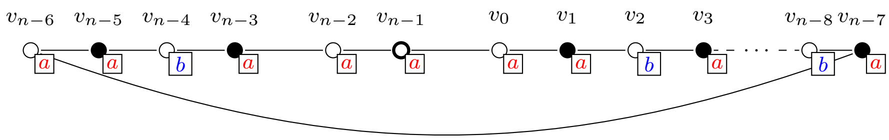



Figure 3.21: The gap-[2]-vertex-labelling of $C_n$, when $n \equiv 2 \pmod 4$, as described in the text. Observe the conflicting colours of vertices $v_{n-1}$ and $v_0$.

Finally, we prove item (iii) showing that when $n \geq 4$, $G$ admits a gap-[3]-vertex-labelling. In order to do this, we prove the following (stronger) statement: if $n \geq 4$, then $G$ admits a gap-[3]-vertex-labelling with labels $(\pi(v_{n-2}), \pi(v_{n-1}), \pi(v_0))$ and colours $(c_\pi(v_{n-2}), c_\pi(v_{n-1}), c_\pi(v_0))$ being equal to one of the following alternatives:

(I) $(1, 2, 1)$ and $(1, 0, 1)$; or

(II) $(2, 3, 2)$ and $(2, 0, 2)$; or

(III) $(3, 1, 3)$ and $(1, 0, 1)$; or

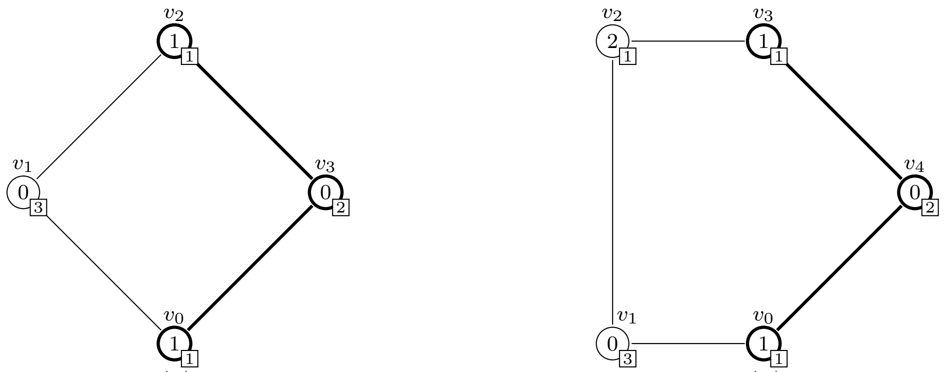
(IV) $(1, 1, 1)$ and $(2, 0, 2)$.

We prove this statement by induction on $n$. For cycles $C_4$ and $C_5$, assign labels $(1, 3, 1, 2)$ and $(1, 3, 1, 1, 2)$ to vertices $(v_0, \ldots, v_{n-1})$, as illustrated in Figure 3.22. Observe that both labellings satisfy (I). Now, suppose there exists a gap-[3]-vertex-labelling $(\pi, c_\pi)$ for cycle $C_n$, $n \geq 4$, satisfying one of the above conditions. Consider cycle $C_{n+2}$, with $V(C_{n+2}) = \{v_0, v_1, \ldots, v_{n+1}\}$. Define a new labelling $\pi' : V(C_{n+2}) \to \{1, 2, 3\}$ from labelling $\pi$, such that:

$$\pi'(v_i) = \begin{cases} \pi(v_i), & \text{if } 0 \leq i \leq n-2; \\ \pi(v_{n-1}), & \text{if } i \in \{n-1, n+1\}; \\ a, & \text{if } i = n; \end{cases} \quad \text{where } \; a = \begin{cases} 3, & \text{if (I) is satisfied;} \\ 1, & \text{if (II) or (III) is satisfied;} \\ 2, & \text{if (IV) is satisfied.} \end{cases}$$

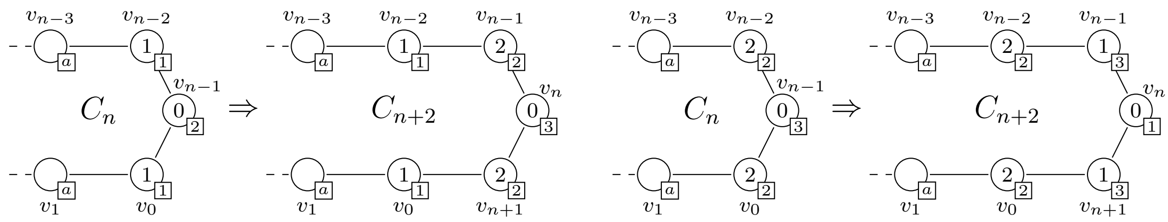
Define colouring $c_{\pi'}$ as usual. First, we show that $(\pi', c_{\pi'})$ is a gap-[3]-vertex-labelling of $C_{n+2}$. Let $w \in V(C_{n+2}) \backslash (N(v_{n-1}) \cup N(v_n) \cup N(v_{n+1}))$. Since $N_{C_{n+2}}(w) = N_{C_n}(w)$ and $\pi'$ preserves the labels from $\pi$ in these vertices, we conclude that $c_{\pi'}(w) = c_\pi(w)$. This implies that for every $v_i, v_{i+1} \in V(C_{n+2}) \backslash N(v_{n-1}) \cup N(v_n) \cup N(v_{n+1})$, we have $c_{\pi'}(v_i) \neq c_{\pi'}(v_{i+1})$. Now consider $w \in N(v_{n-1}) \cup N(v_n) \cup N(v_{n+1})$. We depict these vertices, their respective labels and induced colours in Figure 3.23. By inspection, one can see that colour $c_{\pi'}(w)$ is different from the colour of each of its neighbours.

In order to conclude this proof, observe that new labelling $(\pi', c_{\pi'})$ of $C_{n+2}$ satisfies one of (I), (II), (III), (IV) after renaming the vertices of $C_{n+2}$ so that $v_i \leftarrow v_{(i+1) \bmod n+2}$. That is:



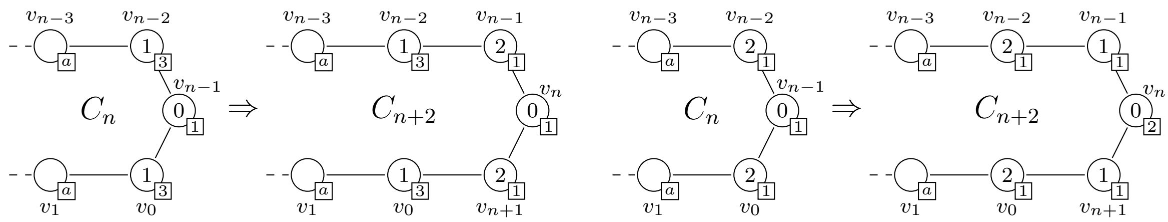

(a) (b)

Figure 3.22: The gap-[3]-vertex-labelling of cycles $C_4$ and $C_5$, the basis of our induction. Observe that the highlighted elements satisfy (I).




(a) Labelling $\pi'$ when $\pi$ satisfies (I).

(b) Labelling $\pi'$ when $\pi$ satisfies (II).




(c) Labelling $\pi'$ when $\pi$ satisfies (III).

(d) Labelling $\pi'$ when $\pi$ satisfies (IV).

Figure 3.23: Labellings for $C_{n+2}$.

(a) If $C_n$ satisfies (I), then $C_{n+2}$ satisfies (II);

(b) If $C_n$ satisfies (II), then $C_{n+2}$ satisfies (III);

(c) If $C_n$ satisfies (III), then $C_{n+2}$ satisfies (IV);

(d) If $C_n$ satisfies (IV), then $C_{n+2}$ satisfies (I). $\square$

An alternative proof for cases of $n \equiv 1, 2, 3 \pmod 4$ of Theorem 3.10 was proposed by a reviewer when this result was submitted to a conference in 2017[4]. He proposed a different labelling for these cycles, which is presented below.

[4] This result, along with others in this section, was accepted and presented at the 2º ETC, a conference held in São Paulo in July, 2017.

*Alternative proof of Theorem 3.10, item (iii).* Let $G \cong C_n$, $n \geq 4$, with vertex set $V = \{v_0, v_1, \ldots, v_{n-1}\}$. For the cases where $n \equiv 1, 2, 3 \pmod 4$, it suffices to show that $G$ admits a gap-[3]-vertex-labelling. Define a labelling $\pi$ of $G$ as follows. For $0 \leq i < n$, assign

$$\pi(v_i) = \begin{cases} 3, & \text{if } i = n-1; \\ 1, & \text{if } i \equiv 0, 1, 2 \pmod 4; \\ 2, & \text{otherwise.} \end{cases}$$

Figure 3.24 illustrates this labelling for cycles $C_9$, $C_{10}$ and $C_{11}$. Define colouring $c_\pi$ as usual. In order to prove that $(\pi, c_\pi)$ is a gap-[3]-vertex-labelling of $G$, we have to show that $c_\pi$ is, in fact, a proper colouring of $G$.

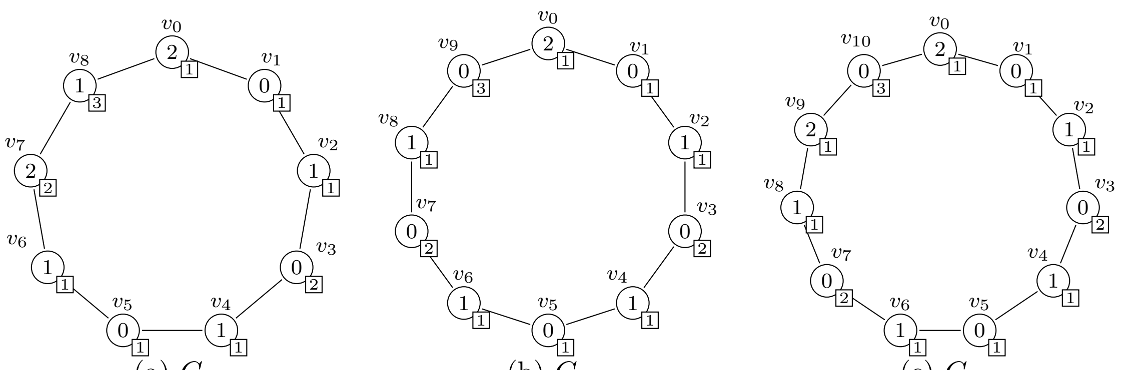

(a) $C_9$. (b) $C_{10}$. (c) $C_{11}$.

Figure 3.24: Examples of gap-[3]-vertex-labellings of cycles.

Observe that $\Pi_{N(v_0)} = \{1, 3\}$ in all cases, which induces $c_\pi(v_0) = 2$. Also, for $1 \leq i \leq n-3$, $i$ odd, we have $\Pi_{N(v_i)} = \{1\}$, which induces $c_\pi(v_i) = 0$. On the other hand, for even $i$, $2 \leq i \leq n-3$, we have $\Pi_{N(v_i)} = \{1, 2\}$, inducing $c_\pi(v_i) = 1$.

In order to conclude the proof, we analyse the colours of vertices $v_{n-2}$ and $v_{n-1}$. For $n$ odd, $\Pi_{N(v_{n-2})} = \{1, 3\}$, which induces $c_\pi(v_{n-2}) = 2$, while $\Pi_{N(v_{n-2})} = \{2, 3\}$ for $n \equiv 2 \pmod 4$, which induces $c_\pi(v_{n-2}) = 1$. Moreover, for $n \equiv 1 \pmod 4$, we have $\Pi_{N(v_{n-1})} = \{1, 2\}$ and, for $n \equiv 2, 3 \pmod 4$, $\pi(v_{n-2}) = \pi(v_0) = 1$, which colours vertex $v_{n-1}$ with 1 and 0, respectively. Thus, $c_\pi$ is a proper colouring of $G$, which completes the proof. □

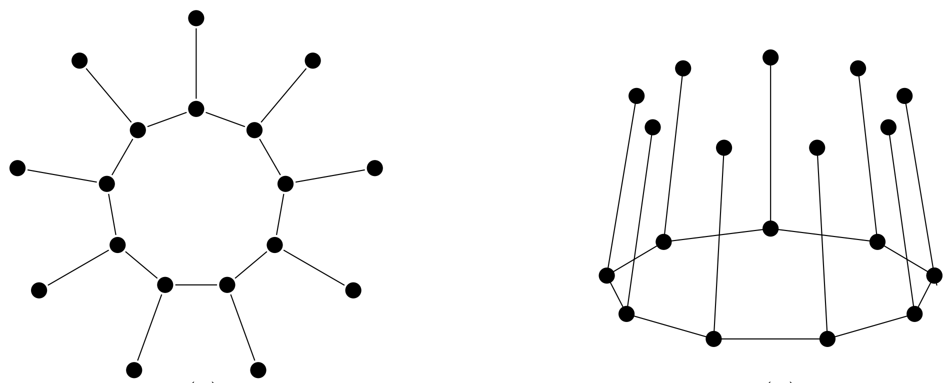
An immediate implication of Theorem 3.10 is that the decision problem **G2VL** when restricted to cycles, which are 2-regular connected graphs, can be solved (in polynomial time) simply by knowing the order of the cycle. Therefore, the following corollary holds.

**Corollary 3.11.** **G2VL** *is in* P *when restricted to 2-regular connected graphs.* □

Although studying the gap-[$k$]-vertex-labelling for this family of graphs enabled us to better understand some restrictions when trying to determine the vertex-gap number, as in the case of $n \equiv 2 \pmod 4$, we wanted to deepen our comprehension of having vertices with degree one in the graph and of how the colouring induced by these vertices influences the vertex-gap number. Recall that as a corollary of Theorem 3.1, when considering subcubic bipartite graphs, degree-one vertices had no effect on the hardness of determining whether these graphs admit gap-[2]-vertex-labellings. In order to further investigate these implications, the next class of graphs we address is the family of crown graphs.

### 3.3.2 Crowns

The family of crown graphs was defined in Chapter 2. To recall, a *crown* $R_n$ is the graph constructed by taking cycle $C_n$, $n$ copies of the complete graph $K_2$ and identifying each vertex of the cycle with a vertex of a different copy of $K_2$. This construction yields a graph with $2n$ vertices: $n$ vertices of degree 1 and $n$ vertices of degree 3. Also, recall that $\chi(R_n) = \chi(C_n)$. Figure 3.25 presents two drawings of crown $R_9$.



(a) (b)

Figure 3.25: Two representations of crown $R_9$.

For this class of graphs, we establish the vertex-gap number in the following theorem.

**Theorem 3.12.** *Let $G \cong R_n$, $n \geq 3$. Then, $\chi_V^g(G) = \chi(G)$.*

*Proof.* Let $G = R_n$, with $V(G) = \{v_0, \ldots, v_{n-1}\} \cup \{u_0, \ldots, u_{n-1}\}$, $d(v_i) = 3$ and $d(u_i) = 1$, $0 \leq i < n$. By Corollary 3.8, in order to prove the result, it suffices to show that every crown $R_n$ admits a gap-[$\chi(R_n)$]-vertex-labelling. Therefore, we show that crowns with even cycles, which have $\chi(G) = 2$, admit a gap-[2]-vertex-labelling and, the others, with $\chi(G) = 3$, admit a gap-[3]-vertex-labelling.

Define a labelling $\pi$ of $G$ as follows. Let $\pi(v_i) = \chi(R_n)$, $0 \leq i < n$, and $\pi(u_i) = 1 + (i \bmod 2)$, $0 \leq i < n - (n \bmod 2)$. If $n$ is odd, let $\pi(v_{n-1}) = 3$. Define colouring $c_\pi$ as usual. This assignment $(\pi, c_\pi)$ for crowns $R_7$ and $R_8$ are exhibited in Figure 3.26.

Observe that every degree-one vertex is adjacent to a vertex labelled with $\chi(R_n)$; therefore, $c_\pi(u_i) = \chi(R_n)$ for all $u_i \in V(G)$. For vertices $v_i$, which have degree three, observe that two of their neighbours, $v_{i-1}$ and $v_{i+1}$, are also labelled with $\chi(R_n)$. Therefore, the colour induced in each of these vertices is $c_\pi(v_i) = \chi(R_n) - \pi(u_i)$. Except for $v_{n-1}$, with $n$ odd, $\pi$ alternates labels 1 and 2 along the degree-one vertices, with $\pi(v_0) = 1$. Therefore, $c_\pi(v_i)$ alternates between colours 2 and 1 when $n$ is odd, and 1 and 0, when $n$ is even. We conclude that $c_\pi(v_i) = \chi(R_n) - (1 + i \bmod 2)$ for all $0 \leq i < n - (n \bmod 2)$. Finally, for the case $n$ odd, vertex $u_{n-1}$ was labelled with $\chi(R_n) = 3$. Therefore, vertex $v_{n-1}$ was uniquely coloured with 0 when $n$ is odd. We conclude that $c_\pi$ is a proper colouring of $G$, which completes the proof. $\square$

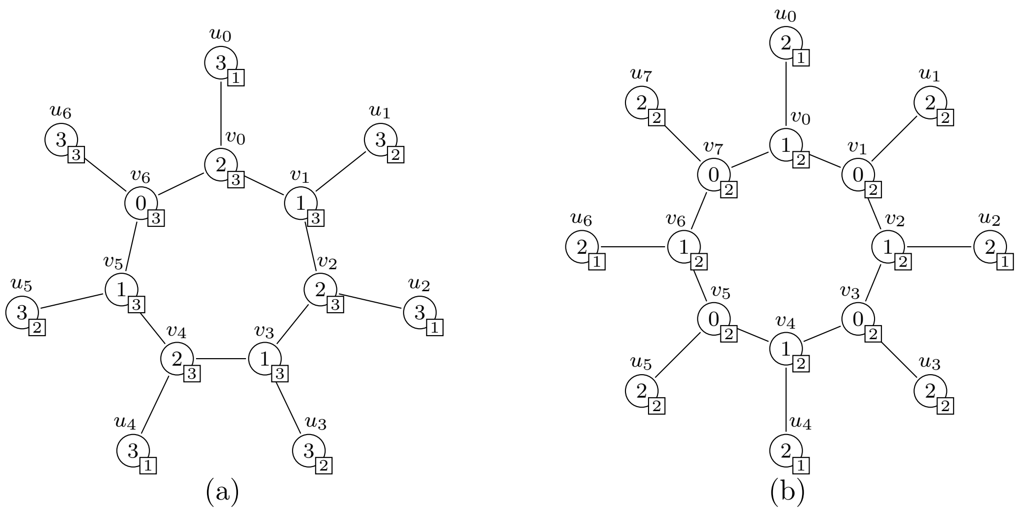



Figure 3.26: In (a) and (b), the gap-[2]-vertex-labelling and gap-[3]-vertex-labelling of crowns $R_7$ and $R_8$, respectively.

By studying crowns graphs, we better understand the effect of degree-one vertices in the induced colouring obtained by a gap-[$k$]-vertex-labelling of graphs. In fact, for the case $n \equiv 2 \pmod 4$, the existence of these vertices enabled the graph to admit a gap-[2]-vertex-labelling – which was impossible for cycles.

The next class of graphs considered unfolds from the study of crowns. Recall that crown $R_n$ is obtained by identifying vertices from cycle $C_n$ to vertices of complete graphs $K_2$. If we choose to identify all degree-one vertices in $R_n$, we obtain the wheel $W_n$: a graph which has no degree-one vertices, but, on the other hand, has a universal vertex. By this construction, the universal vertex may have arbitrarily large degree and, thus, provides us the opportunity for studying the vertex-gap number for graphs with large degrees.

### 3.3.3 Wheels

The family of wheels is formally introduced in Chapter 2, but as mentioned in the previous section, it can be obtained by identifying all degree-one vertices from crown $R_n$. Observe that wheel $W_n$ has $n+1$ vertices: vertices $v_0, \ldots, v_{n-1}$ have degree one and vertex $v_n$, degree $n$. Recall that the center of the wheel is vertex $v_n$, and the cycle of order $n$, surrounding the central vertex, is the rim. Also, $\chi(W_n) = \chi(C_n) + 1$, which implies $\chi(W_n) = 3$ when $n$ is even, and $\chi(W_n) = 4$, otherwise. Figure 3.27 exemplifies wheels of odd and even length.

This family of graphs is the first to present some interesting properties when attempting to establish its vertex-gap number. In the edge version of this labelling, discussed in Chapter 2, the label assigned to an edge in $G$ only affects the two vertices incident with that edge. Here, however, a label assigned to a vertex affects its entire neighbourhood. Therefore, in this family, the label assigned to the universal vertex has a high impact on the induced colours of all remaining vertices in the wheel; the assignment of a very large (small) value would define the maximum (minimum) for all vertices in the rim, making it difficult to properly label the graph. Moreover, as established in Theorem 3.13, wheels $W_4$

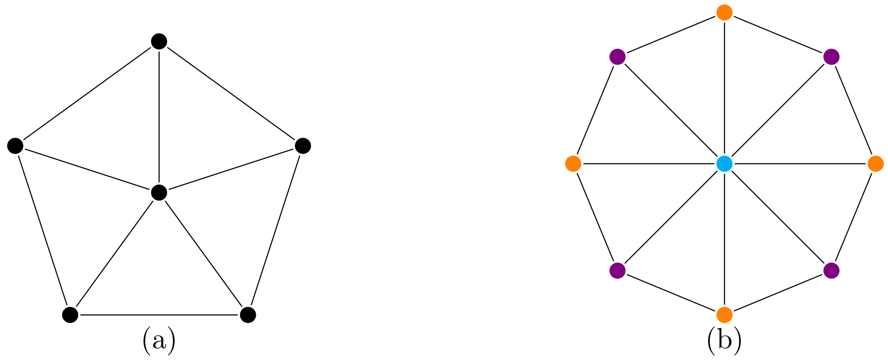


Figure 3.27: In (a), wheel $W_5$; and in (b), wheel $W_8$, which has been properly coloured using 3 colours.

and $W_6$ do not admit a gap-[3]-vertex-labelling, which seems to indicate that the size of the rim also has an impact in this result. Wheel $W_3$ does not admit a gap-[$k$]-vertex-labelling for any $k$ since it is isomorphic to complete graph $K_4$. The non-existence of gap-[$k$]-vertex-labellings is discussed in Chapter 4.

For $W_n$, $n \geq 4$, the vertex-gap number is established in Theorem 3.13.

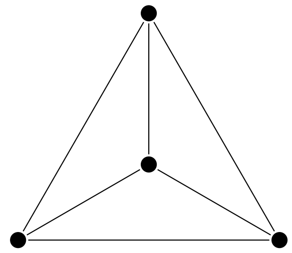


Figure 3.28: Wheel $W_3$, for which there is no gap-[$k$]-vertex-labelling, for $k \in \mathbb{N}$.

**Theorem 3.13.** *Let $G \cong W_n$, $n \geq 4$. Then, $\chi_V^g(G) = 3$ if $n \geq 8$ and even, and $\chi_V^g(G) = 4$, otherwise.*

*Proof.* Let $G \cong W_n$, with vertex set $V = \{v_0, v_1, \ldots, v_n\}$ and $d(v_n) = n$. Recall that $\chi(W_n) = 3$ when $n$ is even, and $\chi(W_n) = 4$, otherwise. Then, by Corollary 3.8, in order to prove the result, we show that: (i) wheels $W_4$ and $W_6$ do not admit a gap-[3]-vertex-labelling; (ii) wheels $W_n$, $n \geq 8$ and even, admit a gap-[3]-vertex-labelling; and (iii) wheels $W_n$, $n \geq 4$, admit a gap-[4]-vertex-labelling.

First, we prove item (i) by contradiction, starting with $G \cong W_4$. Suppose $G$ admits a gap-[3]-vertex-labelling $(\pi, c_\pi)$. Since there is no vertex in $G$ with degree one, we know that colour 3 cannot be induced in any vertex of $G$. This implies that $c_\pi : V(G) \to \{0, 1, 2\}$. Since $v_n$ is adjacent to all other vertices in $G$, then its colour is unique. This opens three possibilities for the colour of the central vertex, one for each colour in $c_\pi$. First, suppose $c_\pi(v_n) = 0$. This implies that all the labels of vertices in the rim of $G$ are the same $a \in \{1, 2, 3\}$. Let $\pi(v_n) = b \in \{1, 2, 3\}$. If this is the case, then $c_\pi(v_i) = |a - b|$ for all $0 \leq i \leq 3$, contradicting the fact that $c_\pi$ is a proper colouring of $G$, as illustrated by Figure 3.29(a).

We conclude that $c_\pi(v_0) \neq 0$. Consequently, we now know that colour 0 alternates in the vertices of the rim. Adjust notation so that $c_\pi(v_0) = 0$. This directly implies that $\Pi_{N(v_0)} = \{a\}$, for $a \in \{1, 2, 3\}$, that is, $\pi(v_1) = \pi(v_3) = \pi(v_4) = a$. Furthermore, observe that $\{v_1, v_3, v_4\} = N(v_2)$, which implies that $c_\pi(v_2) = 0$. However, consider vertex $v_1$, and let $b, c \in \{1, 2, 3\}$ be the labels assigned to vertices $v_0$ and $v_2$, respectively, as illustrated in Figure 3.29(b). Observe that $\Pi_{N(v_1)} = \{a, b, c\} = \Pi_{N(v_3)}$, which implies that $c_\pi(v_1) = c_\pi(v_3)$. However, $\Pi_{N(v_4)} = \{a, b, c\}$ as well, which implies $c_\pi(v_4) = c_\pi(v_1)$, contradicting the fact that $c_\pi$ is a proper vertex-colouring. Since we have exhausted every possible colour for the central vertex, we conclude that there is no gap-[3]-vertex-labelling for wheel $W_4$.

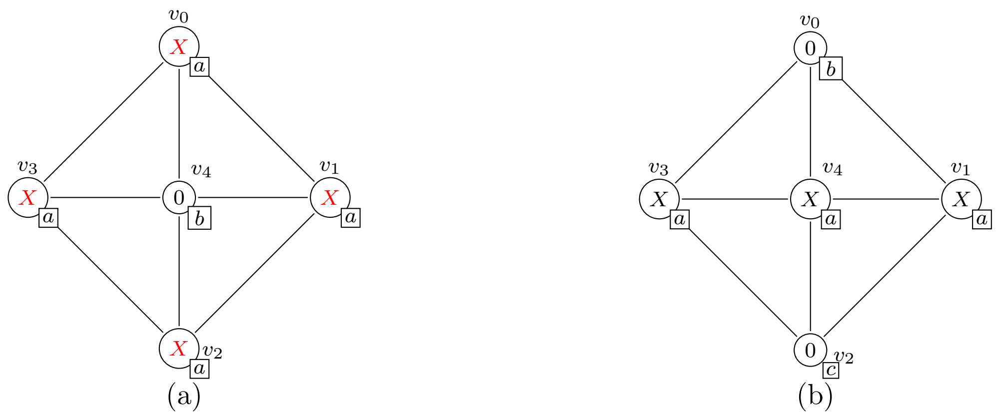



Figure 3.29: The hypothesis that wheel $W_4$ admits a gap-[3]-vertex-labelling. In (a), colour 0 is induced in the central vertex, $v_4$. In (b), colour 0 alternates along the rim. Both cases reach a contradiction.

By a similar reasoning, we conclude, by contradiction, that $W_6$ does not admit a gap-[3]-vertex-labelling. Analogously to $W_4$, $c_\pi(v_6) \neq 0$. Therefore, colours 0 and $X$, $X \in \{1, 2\}$, alternate along the vertices of the rim. Adjust notation so that $c_\pi(v_0) = c_\pi(v_2) = c_\pi(v_4) = 0$ and $c_\pi(v_1) = c_\pi(v_3) = c_\pi(v_5) = X$. Moreover, $\pi(v_1) = \pi(v_3) = \pi(v_5) = \pi(v_6) = a$, $a \in \{1, 2, 3\}$.

Suppose, first, $X = 1$. In this case, $c_\pi(v_6) = 2$. Therefore, $\{1, 3\} \subseteq \Pi_{N(v_6)}$. Moreover, $a \notin \{1, 3\}$ since $a = 1$ or $a = 3$ would induce a vertex with colour 2 in the rim. Therefore, $X = 2$, implying $c_\pi(v_6) = 1$ and $\{1, 3\} \not\subseteq \Pi_{N(v_6)}$. We conclude that $\{\pi(v_0), \ldots, \pi(v_5)\} = \{1, 2\}$ or $\{\pi(v_0), \ldots, \pi(v_5)\} = \{2, 3\}$. In the first case, $\pi(v_6) = 3$ and in the second case, $\pi(v_6) = 1$. In both cases, $\pi(v_6) \neq \pi(v_1)$, which is a contradiction.

The main issue for wheels $W_4$ and $W_6$ that prevents them from admitting a gap-[3]-vertex-labelling is, in other words, an insufficient number of vertices in the rim. For wheels $W_n$, $n \geq 8$ and even, however, this ceases to be a problem. We define a gap-[3]-vertex-labelling $(\pi, c_\pi)$ for these graphs based on the insight gained by proving item (iii).

Assign labels 2,1 alternately to vertices $v_i$, $0 \leq i \leq n - 6$, starting with $\pi(v_0) = 2$. For $v_{n-3}$, let $\pi(v_{n-3}) = 3$ and to the remaining vertices, namely $v_{n-5}$, $v_{n-4}$, $v_{n-2}$, $v_{n-1}$ and $v_n$, assign label 2. This labelling is depicted for wheel $W_{14}$ in Figure 3.31. Define colouring $c_\pi$ as usual.

Note that every vertex $v_i$, $i$ even, as well as the central vertex has been assigned

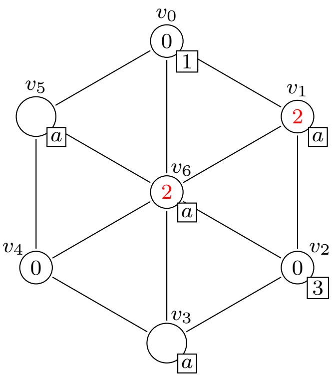



Figure 3.30: Supposing wheel $W_6$ admits a gap-[3]-vertex-labelling. Note that assigning label 2 to $v_4$ would induce colour 1 for both $v_3$ and $v_5$.

the same label 2. This implies that $\Pi_{N(v_j)} = \{2\}$ for every $j$ odd, and, consequently, $c_\pi(v_j) = 0$. For the central vertex $v_n$, we have $\Pi_{N(v_n)} = \{1, 2, 3\}$, which induces $c_\pi(v_n) = 2$. Now, consider vertices $v_{n-2}$ and $v_{n-4}$, which have $\Pi_{N(v_{n-2})} = \Pi_{N(v_{n-4})} = \{2, 3\}$, inducing colour 1. Finally, the remaining even-index vertices $v_i$ have $\Pi_{N(v_i)} = \{1, 2\}$, also inducing $c_\pi(v_i) = 1$. We conclude that $c_\pi(v_n) = 2$ and $c_\pi(v_i) = (i + 1) \bmod 2$, for every $0 \leq i < n$. Therefore, $(\pi, c_\pi)$ is a gap-[3]-vertex-labelling of $G$. This completes the proof of item (i).

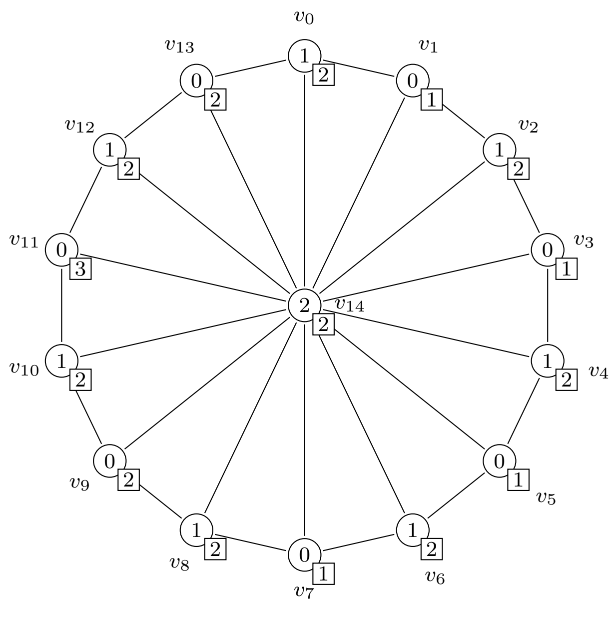


Figure 3.31: The gap-[3]-vertex-labelling of wheel $W_{14}$ described in the text.

It remains to consider item (iii), where we prove that every wheel $W_n$, with $n \geq 4$, admits a gap-[4]-vertex-labelling. For wheel $W_4$, assign labels $(4, 1, 4, 1, 3)$ to vertices $(v_0, v_1, v_2, v_3, v_4)$, and define colouring $c_\pi$ as usual. By inspecting Figure 3.32, which depicts this labelling, we observe that all adjacent vertices have distinct induced colours and, therefore, $(\pi, c_\pi)$ is a gap-[4]-vertex-labelling of $W_4$. In the following, we define the labelling for all wheels $W_n$, $n \geq 5$.

Assign label 2 to vertices $v_0$, $v_1$, $v_2$ and $v_n$, and labels $4, 1$, alternately, to the remaining

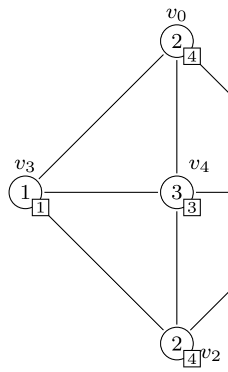



Figure 3.32: The gap-[4]-vertex-labelling of $W_4$ described in the text.

vertices $v_i$, $3 \leq i < n$, starting with $\pi(v_3) = 4$. Define colouring $c_\pi$ as usual. This labelling is depicted in figures 3.33(a) and 3.33(b) for wheels $W_9$ and $W_{10}$, examples of odd and even length, respectively. Note that, for every vertex $v_i$, $2 \leq i < n$ and even, $N(v_i) = \{v_{i-1}, v_{i+1}, v_n\}$, and we have $\Pi_{N(v_i)} = \{2, 4\}$, which yields $c_\pi(v_i) = 2$. For $v_j$, with $j$ odd and $3 \leq j < n$, a similar analysis shows that $\Pi_{N(v_j)} = \{1, 2\}$, which implies $c_\pi(v_j) = 1$. Therefore, $c_\pi(v_i) = 2 - (i \bmod 2)$ for all $2 \leq i < n$. Also, observe that $c_\pi(v_0) = 2 - (n \bmod 2)$ since $\Pi_{N(v_0)} = \{1, 2\}$ when $n$ is odd, and $\Pi_{N(v_0)} = \{2, 4\}$, otherwise. Finally, we have $\Pi_{N(v_1)} = \{2\}$, inducing $c_\pi(v_1) = 0$, and $\{1, 4\} \subset \Pi_{N(v_n)}$, inducing $c_\pi(v_n) = 3$. We conclude that $(\pi, c_\pi)$ is a gap-[4]-vertex-labelling of $G$. This completes the proof of Theorem 3.13. $\square$

Recall that in 2013, Dehghan et al. [8] proposed a labelling for trees. With this in mind, similar to Chapter 2, the next step of our research on gap-[$k$]-vertex-labellings addresses the family of Unicyclic graphs, where we use Dehghan et al.'s labelling together with our established labellings for cycles and crowns.

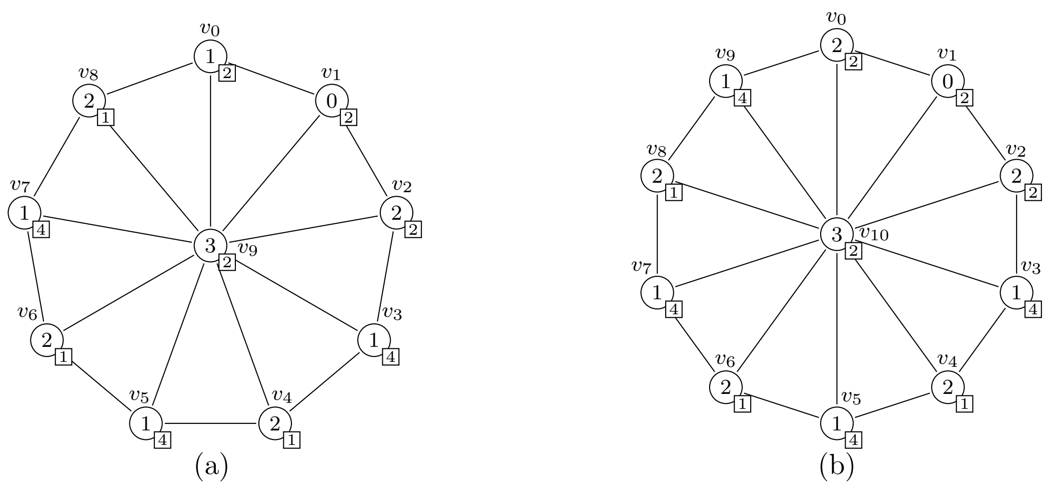



Figure 3.33: The gap-[4]-vertex-labelling for wheels $W_9$ and $W_{10}$ in (a) and (b), respectively, as described in the text.

### 3.3.4 Unicyclic graphs

The family of unicyclic graphs is defined in Section 2.2.4. Recall that this class comprises the connected graphs $G = (V, E)$ with $|V| = |E|$. Figure 3.34 illustrates a unicyclic graph. We establish the vertex-gap number of unicyclic graphs by using the gap-[2]-vertex-labelling of trees defined by Dehghan et al. [8]. Since their article does not present a formal proof that $\chi_V^g(T) = 2$ for every tree $T$, we demonstrate this result in Lemma 3.14.

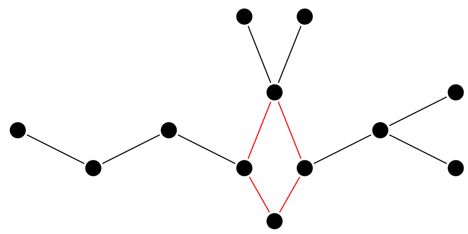


Figure 3.34: An example of a unicyclic graph $G$, with $|V| = |E| = 12$. In red, the edges of cycle $C_4$ – the only one in $G$.

**Lemma 3.14** (Dehghan et al.)**.** *Let $G \cong T$ be a nontrivial tree not isomorphic to a star. Then, $\chi_V^g(G) = 2$.*

*Proof.* Let $G = (V, E)$ be a nontrivial rooted tree that is not a star, and let $u$ be its root. Since $G$ is not a star, by Lemma 3.7, we know that $\chi_V^g(G) \geq \chi(G)$. It is well-known that $\chi(G) = 2$ for every nontrivial tree. Therefore, it suffices to exhibit a gap-[2]-vertex-labelling of $G$. Define a gap-[2]-vertex-labelling $(\pi, c_\pi)$ of $G$ as follows. For every vertex $v \in V(G)\backslash\{u\}$, assign

$$\pi(v) = \begin{cases} 1, & \text{if } \operatorname{dist}(u, v) \equiv 0 \pmod 4\text{; and} \\ 2, & \text{otherwise.} \end{cases}$$

Recall that $\operatorname{dist}(u, v)$ is the minimum distance between vertices $u$ and $v$ in $G$. Therefore, every vertex at odd distance from $u$ was labelled with 2, while vertices at even distance alternate their labels between 1 and 2. Note that since $G$ is a tree, vertices at even distance of $u$ are adjacent exclusively to vertices at odd distance and vice-versa. Define colouring $c_\pi$ as usual. Figure 3.2, presented in the beginning of this chapter and replicated below, is an example of this labelling.

First, observe that the root of the tree has $\Pi_{N(u)} = \{2\}$, which induces $c_\pi(u) = 0$. Next, we draw the reader's attention to the internal vertices of $G$, that is, vertices $v$ with $d(v) \geq 2$. For vertices at even distance from root $u$, we have $\Pi_{N(v)} = \{2\}$, while vertices at odd distance have $\Pi_{N(v)} = \{1, 2\}$. Therefore, $c_\pi(v) = \operatorname{dist}(u, v) \bmod 2$ for every internal vertex $v$. It remains to consider the induced colours of the leaves of $G$.

Let $w$ be an arbitrary leaf of $G$, adjacent to a vertex $v$. If $\operatorname{dist}(u, w) \equiv 0, 2 \pmod 4$, then its neighbour has $\operatorname{dist}(u, v) \equiv 3, 1 \pmod 4$ since $\operatorname{dist}(u, w) = \operatorname{dist}(u, v) - 1$. Therefore, $\pi(v) = 2$. Since $w$ is a leaf, its colour is determined by the label of its neighbour, that is, $c_\pi(w) = 2$. Furthermore, as stated in the previous paragraph, $c_\pi(v) = 1$. Similarly, consider $\operatorname{dist}(u, w) \equiv 1, 3 \pmod 4$. In this case, $\pi(v) = 1$ if $\operatorname{dist}(u, v) \equiv 0 \pmod 4$, and

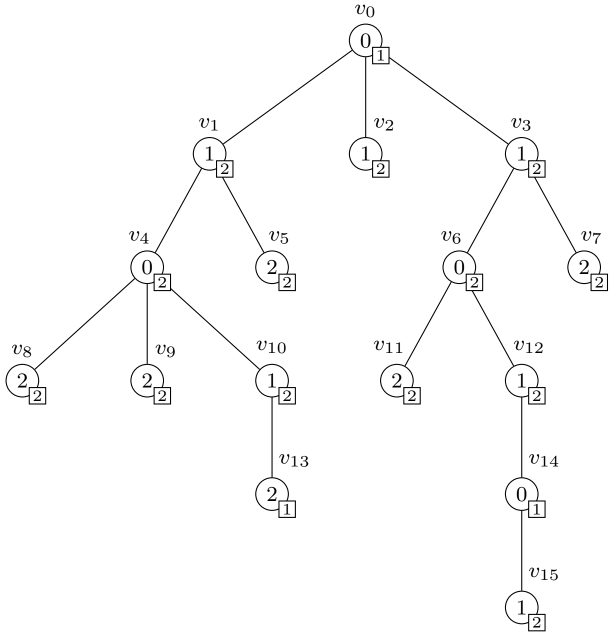



Figure 3.35: The gap-[2]-vertex-labelling of a nontrivial tree $T$, as defined by Dehghan et al. [8].

$\pi(v) = 2$, otherwise. Since $c_\pi(v) = 0$, we conclude that $c_\pi(w) \neq c_\pi(v)$ in both cases, and $c_\pi$ is a proper colouring of $G$. $\square$

The labelling presented in the proof of Lemma 3.14 is used, albeit with some modifications, to establish the vertex-gap number of unicyclic graphs, which is presented in the next theorem.

**Theorem 3.15.** *Let $G = (V, E)$ be a unicyclic graph and let $p$ be the size of the cycle in $G$. Then, $\chi^g_V(G) = 2$, if $p$ is even and $G \ncong C_p$, and $\chi^g_V(G) = 3$, otherwise.*

*Proof.* Let $G = (V, E)$ be a unicyclic graph, with $v_0, \ldots, v_{p-1}$ denoting the vertices of cycle $C_p$. Also, let $T_0, T_1, \ldots, T_{p-1}$, be the $p$ trees rooted at vertices $v_0, v_1, \ldots, v_{p-1}$, respectively, as defined in Section 2.2.4. In Figure 3.36, we exemplify a unicyclic graph with $p = 7$ and three nontrivial graphs $T_0$, $T_1$ and $T_4$.

We remark that $T_i$ cannot be a trivial graph for every $0 \leq i < p$ since $G \ncong C_p$. Therefore, for the remainder of the proof, we can assume that there exists at least one tree $T_i$ with $|V(T_i)| \geq 2$. By Lemma 3.7, we know that $\chi^{\mathrm{g}}_V(G) \geq \chi(G)$. Therefore, it suffices to show that $G$ admits a gap-[2]-vertex-labelling when $p$ is even and a gap-[3]-vertex-labelling, otherwise.

First, it is necessary to introduce a notation which is used throughout the proof. Let $v_i$ be an arbitrary vertex of the cycle in $G$. Define $L^j_i \subset V(T_i)$ as the set of vertices of $T_i$ that are at distance $j$ from $v_i$, that is, $L^j_i = \{v \in V(T_i) : \mathrm{dist}(v_i, v) = j\}$. We refer to $L^j_i$ as the $j$-th *level* of tree $T_i$. Figure 3.37 exhibits a tree $T_i$ of a unicyclic graph $G$, rooted at $v_i$, highlighting its four levels. Since set $L^0_i$ contains only vertex $v_i$, it will be omitted in the remaining figures.

With the notation set for the levels of trees $T_i$, we are ready to prove that $\chi^{\mathrm{g}}_V(G) = \chi(G)$. First, we consider the case when $p$ is even. Adjust notation so that $v_0$ is the root of a

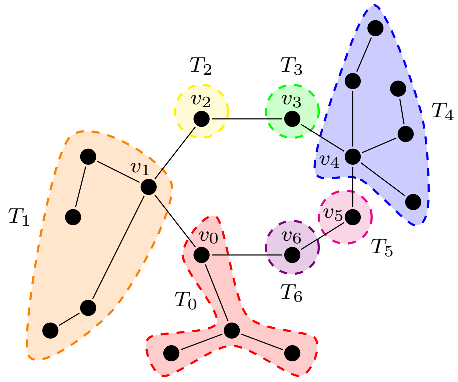



Figure 3.36: An example of the notation defined in the text for a unicyclic graph $G$, with $n = 7$. Observe that in this graph, tree $T_2$ is a trivial graph, while tree $T_4$ is nontrivial and rooted in $v_4$, with $d_{T_4}(v_4) = 3$.

nontrivial tree. Define labelling $\pi : V(G) \to \{1, 2\}$ of $G$ as follows. For every vertex $v_i \in V(C_p)$, assign

$$\pi(v_i) = \begin{cases} 1, & \text{if } i \equiv 3 \pmod 4; \text{ and} \\ 2, & \text{otherwise.} \end{cases}$$

Next, assign labels to the vertices in the first level of trees $T_i$, $0 \leq i < p$, when they exist. For every $u \in V(L_i^1)$, assign $\pi(u) = 1 + (i \bmod 2)$. Since all vertices in the cycle and their neighbours have been assigned labels, define colouring $c_\pi$ as usual for vertices in $C_p$. This partial labelling is exemplified for unicyclic graphs with $p = 6$ and $p = 8$ in Figure 3.38.

Note that, for every $v_i \in V(C_p)$, $i$ odd, $\Pi_{N(v_i)} = \{2\}$, inducing $c_\pi(v_i) = 0$. As for vertices $v_j \in V(C_p)$, $j$ even, we have $\Pi_{N(v_j)} = \{1, 2\}$. In particular, note that $v_0$ is adjacent to at least one vertex labelled with 1 in $T_0$. Therefore, $c_\pi(v_j) = 1$, and we conclude that $c_\pi(v_l) = (l + 1) \bmod 2$, for every $v_l \in V(C_p)$.

Since no label has been assigned to the other vertices of trees $T_i$, apart from the vertices in $L_0^1$, it remains to label these vertices. By inspecting Figure 3.38(b), the (partial)

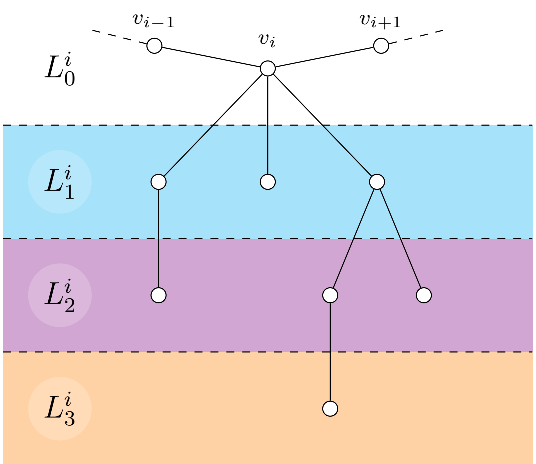
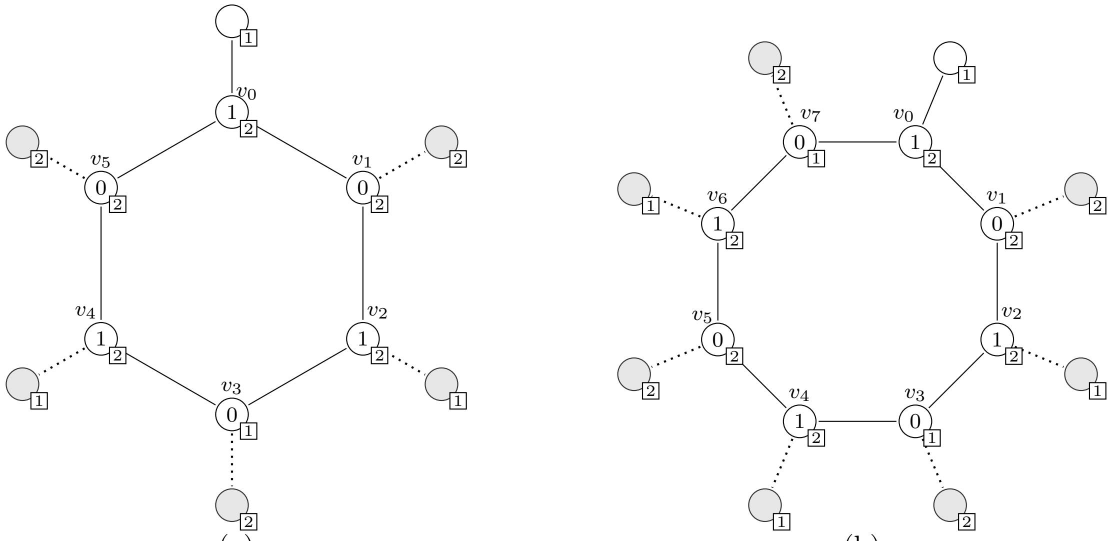



Figure 3.37: A tree $T_i$ from a unicyclic graph $G$ with four levels.




(a) (b)

Figure 3.38: Partial labellings of two unicyclic graphs. The gray vertices indicate vertices in $L_i^1$, which may or may not exist. Note that $T_0$ is always a nontrivial tree.

gap-[2]-vertex-labelling $(\pi, c_\pi)$ has created three possibilities for each pair $(\pi(v_i), c_\pi(v_i))$, $v_i \in V(C_p)$:

(i) $(\pi(v_i), c_\pi(v_i)) = (1, 0)$; or

(ii) $(\pi(v_i), c_\pi(v_i)) = (2, 0)$; or

(iii) $(\pi(v_i), c_\pi(v_i)) = (2, 1)$.

We continue to label the vertices in each $T_i$ of $G$ depending on which case, (i), (ii) or (iii), vertex $v_i$ corresponds to.

*Case 1. $v_i$ corresponds to (i).*

For every $u \in V(T_i)$, assign

$$\pi(u) = \begin{cases} 1, & \text{if } u \in L_i^j, j \equiv 0 \pmod 4; \text{ and} \\ 2, & \text{otherwise.} \end{cases}$$

Figure 3.39(a) illustrates the first five levels of a tree $T_i$ in which its root, $v_i$, satisfies this case. First, consider an internal vertex $u$ of tree $T_i$, observing that $\Pi_{N(u)} = \{1, 2\}$ if $j$ is odd, and $\Pi_{N(u)} = \{2\}$, otherwise. This implies that $c_\pi(u) = j \bmod 2$ for these vertices.

The leaves $w$ of $T_i$ have their colours induced by the label of their neighbour, say $v \in N(w)$. Since $\pi(v) = 2$ for vertices $v \in L_i^j$, $j \equiv 1, 2, 3 \pmod 4$, then $c_\pi(w) = 2$ for leaves $w \in L_i^{j+1}$. Otherwise, that is, if $j \equiv 0 \pmod 4$, then $\pi(v) = 1$, inducing $c_\pi(w) = 1$. In this case, however, $j - 1$ is odd and, thus, $c_\pi(v) = 0$ since $v$ is an internal vertex of $T_i$. Therefore, there are no conflicting vertices in this case.

*Case 2. $v_i$ corresponds to item (ii).*

For vertices $u \in V(T_i)$, assign

$$\pi(u) = \begin{cases} 1, & \text{if } u \in L_i^j, j \equiv 2 \pmod 4; \text{ and} \\ 2, & \text{otherwise.} \end{cases}$$

Figure 3.39(b) illustrates labelling $\pi$ in this case. Similar to Case 1, internal vertices $u \in L_i^j$, also have $\Pi_{N(u)} = \{1,2\}$ when $j$ is odd and $\Pi_{N(u)} = \{2\}$ when $j$ is even. Therefore, $c_\pi(u) = j \pmod 2$ for every internal node $u \in L_i^j$.

For the leaves $w$ of $T_i$, their neighbours $v \in N(w)$ have labels $\pi(v) = 2$, when $j \equiv 0,1,3 \pmod 4$, which induces colour $c_\pi(w) = 2$ for leaves $w \in L_i^{j+1}$. Finally, when $v \in L_i^j$, $j \equiv 2 \pmod 4$, vertex $w$ has induced colour 1 while $v$ has induced colour $c_\pi(v) = 0$. Therefore, $c_\pi$ is a proper colouring of $T_i$.

*Case 3. $v_i$ corresponds to item (iii).*

Finally, for vertices $u$ in $V(T_i)$ in this case, assign

$$\pi(u) = \begin{cases} 1, & \text{if } u \in L_i^j, j \equiv 1 \pmod 4; \text{ or} \\ 2, & \text{otherwise.} \end{cases}$$

This last case is illustrated in Figure 3.39(c). Here, internal vertices $u \in L_i^j$ have $\Pi_{N(u)} = \{2\}$ when $j$ is odd, and $\Pi_{N(u)} = \{1,2\}$, otherwise. This labelling induces $c_\pi(u) = (j+1) \bmod 2$. For leaves $w$ of $T_i$, $c_\pi(w) = 2$ when $w \in L_i^{j+1}$, $j \equiv 0,2,3 \pmod 4$ and $c_\pi(w) = 1$, otherwise; note that in this last case, its neighbour $v$ has colour 2.

We conclude that $c_\pi$ is a proper colouring of each tree $T_i$ of $G$. Consequently, $(\pi, c_\pi)$ is a gap-[2]-vertex-labelling of $G$, and the result follows for graphs with even $p$.

Next, we consider unicyclic graphs $G$ with $p$ odd. We use a similar approach to the case $p$ even: first, we assign labels to vertices $v \in V(C_p) \cup L_{p-1}^1$ that induce a proper colouring (restricted to the cycle); then, we assign labels to the remaining vertices in each tree $T_i$ accordingly.

First, adjust notation so that $v_{p-1}$ is the root of a nontrivial tree $T_{p-1}$. Define a labelling $\pi : V(G) \to \{1,2,3\}$ as follows. If $p = 3$, assign labels $3, 2, 3$ to vertices $v_0$, $v_1$ and $v_2$, respectively. This case is illustrated in Figure 3.40. Otherwise, if $p \geq 5$, assign:

$$\pi(v_i) = \begin{cases} 1, & \text{if } i = 2; \\ 2, & \text{if } i \equiv 0 \pmod 4 \text{ and } i \neq 0; \\ 3, & \text{if } i = 0 \text{ or } (i \equiv 1,2,3 \pmod 4 \text{ and } i \neq 2). \end{cases}$$

Finally, when $p \equiv 3 \pmod 4$, assign label 1 to every vertex $u \in L_{p-1}^1$, that is, the first level of (the nontrivial) tree $T_{p-1}$. The other cases are defined later. This labelling is depicted in Figure 3.41 for graphs with $p = 13$ and $p = 15$.

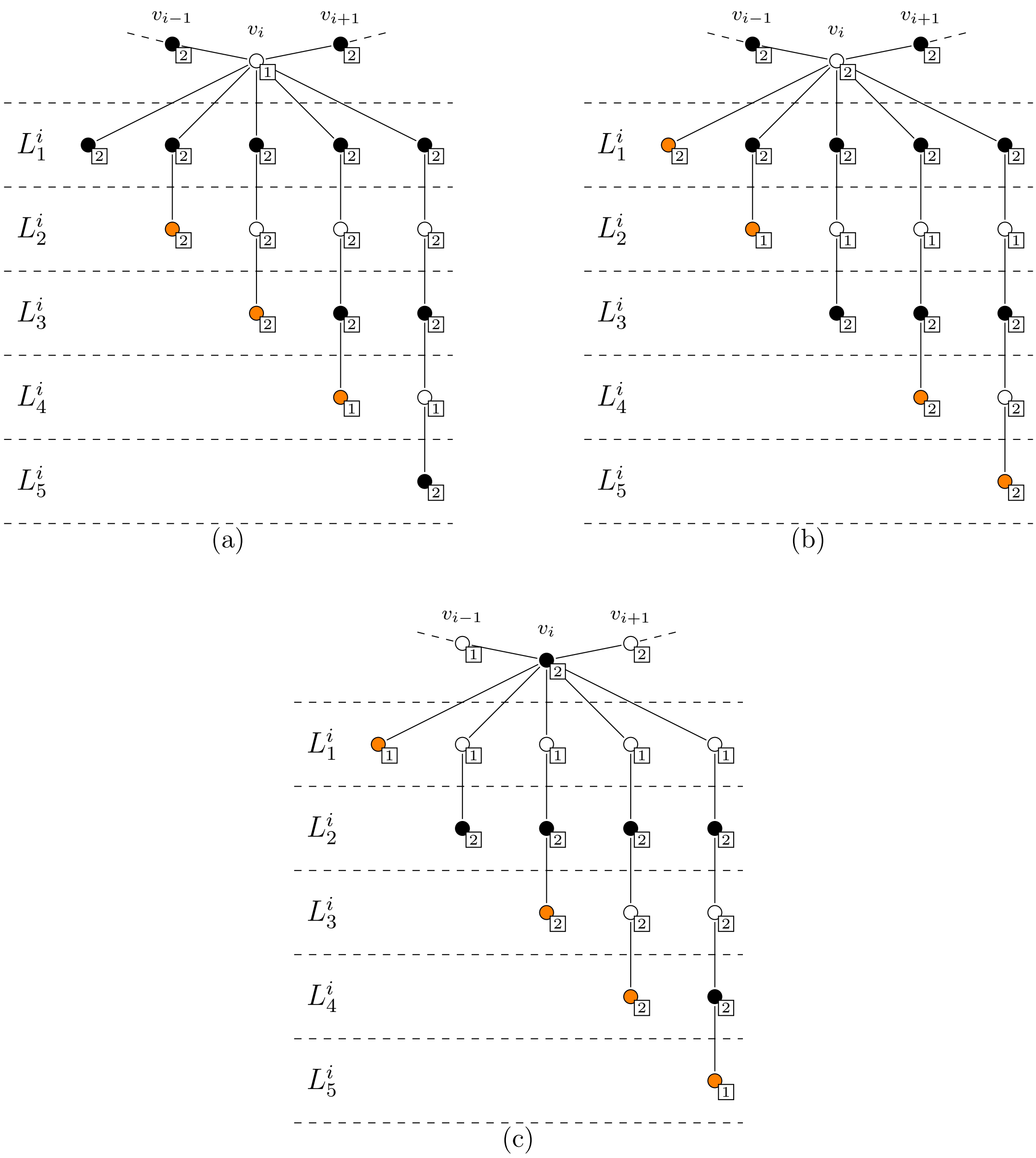



Figure 3.39: The labellings of vertices in $T_i$ for cases (i), (ii) and (iii) in (a), (b) and (c), respectively. Vertices filled with: white have induced colour 0; black, colour 1; and orange, colour 2.

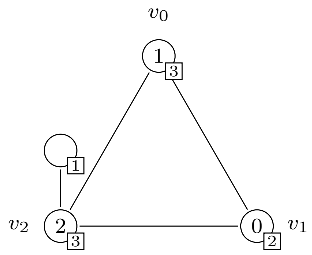
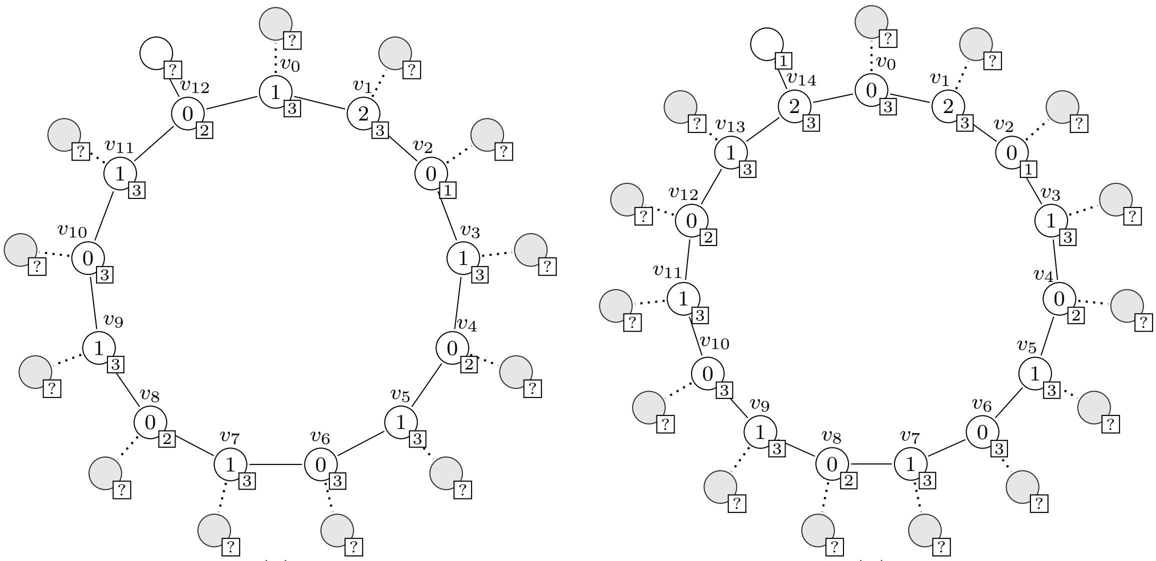



Figure 3.40: The gap-[3]-vertex-labelling of $C_p$ when $p = 3$.

Note that this partial labelling assigns labels only to the vertices of cycle $C_p$ and to the vertices of the first level of at most one tree, $T_{p-1}$. Now, for every other tree $T_i$, $0 \leq i \leq p-2$, if $|V(T_i)| > 1$, then $\Pi_{N(v_i)}$ cannot be fully determined since there are unlabeled vertices in $N(v_i)$. However, we compute the colour of each vertex $v_i \in V(C_p)$ considering only this partial labelling. With these observations, in the following analysis of vertices $v_i \in V(C_p)$, when we say $\Pi_{N(v_i)}$ is equal to a set of labels, we refer to the labels assigned only to the aforementioned vertices. Furthermore, when defining the labels of the remaining vertices of $G$, we guarantee that the labels assigned to vertices in the first level of each $T_i$ do not alter set $\Pi_{N(v_i)}$ and, consequently, create no conflicts to the colours of the vertices of the cycle.

In order to prove that this labelling induces a proper colouring of cycle $C_p$, first, we analyse case $p \equiv 1 \pmod 4$. Figure 3.41(a) illustrates this labelling for $C_{13}$. Consider vertices $v_1, v_2$ and $v_3$, observing that $\Pi_{N(v_1)} = \{1,3\}$, $\Pi_{N(v_2)} = \{3\}$ and $\Pi_{N(v_3)} = \{1,2\}$. This labelling induces colours $c_\pi(v_1) = 2, c_\pi(v_2) = 0$ and $c_\pi(v_3) = 1$, respectively. For vertices $v_i$ and $v_j$, $4 \leq i,j < p$, $i$ odd and $j$ even, $\Pi_{N(v_i)} = \{2,3\}$ and $\Pi_{N(v_j)} = \{3\}$.



(a) (b)

Figure 3.41: The labellings of unicyclic graphs with $p = 13$ and $p = 15$ in (a) and (b), respectively. Since $15 \equiv 3 \pmod 4$, vertices in the first level of $T_{14}$ in (b) receive label 1 so as to induce $c_\pi(v_{14}) = 2$. The "?" symbol means that these vertices have not yet been labelled.

This induces colour 1 in vertices with odd index $v_i$ and colour 0 in $v_j$. We conclude that, with the exception of $c_\pi(v_0) = 1$ and $c_\pi(v_1) = 2$, every vertex $v_i$, $2 \le i < n$, has $c_\pi(v_i) = i \bmod 2$. Therefore, $c_\pi$ is a proper colouring of cycle $C_p$.

Next, we consider the case $p \equiv 3 \pmod 4$. By inspecting Figure 3.40, we conclude that $c_\pi$ is a proper colouring of cycle $C_3$, which was defined uniquely. For $p \ge 5$, illustrated in Figure 3.41(b) for $C_{15}$, vertices $v_1, \ldots, v_{p-2}$ have their colours induced in the same way as in the case $p \equiv 1 \pmod 4$ from the previous paragraph. Thus, it remains only to consider vertices $v_0$ and $v_{p-1}$. Observe that $\Pi_{N(v_0)} = \{3\}$, inducing $c_\pi(v_0) = 0$, and $\Pi_{N(v_{p-1})} = \{1, 3\}$, which induces $c_\pi(v_{p-1}) = 2$; label 1 comes from the vertices in $L^1_{p-1}$ which, in this case, is a nonempty set.

Similar to the case of $p$ even, this labelling allows five combinations $(\pi(v_i), c_\pi(v_i))$ for vertices $v_i$, $0 \le i < p$:

(i) $(\pi(v_i), c_\pi(v_i)) = (1, 0)$

(ii) $(\pi(v_i), c_\pi(v_i)) = (2, 0)$

(iii) $(\pi(v_i), c_\pi(v_i)) = (3, 0)$

(iv) $(\pi(v_i), c_\pi(v_i)) = (3, 1)$

(v) $(\pi(v_i), c_\pi(v_i)) = (3, 2)$

We remark that pairs $(\pi(v_i), c_\pi(v_i))$ in items (i) and (ii) are exactly the same as in the case $p$ even. Contrary to the previous case, here, when $p$ is odd, $c_\pi(v_i) = 0$ is induced by $\Pi_{N(v_i)} = \{3\}$. Then, for each $u \in V(T_i)$ in items (i) and (ii), we assign

$$\pi(u) = \begin{cases} \pi(v_i), & \text{if } u \in L^j_i, j \equiv 0 \pmod 4\text{; and} \\ 3, & \text{otherwise.} \end{cases}$$

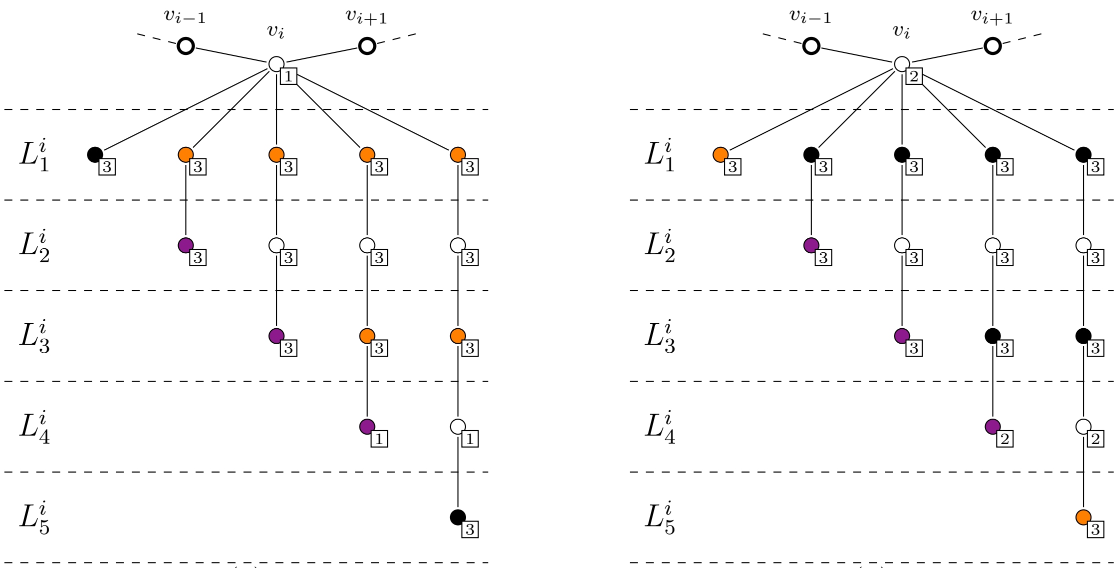
Figure 3.42 illustrates labelling $\pi$ and its induced colouring in these cases. Note that internal vertices $u \in L^j_i$ of $T_i$ have $\Pi_{N(u)} = \{3\}$ when $j$ is even, and $\Pi_{N(u)} = \{3, \pi(v_i)\}$, otherwise. This induces $c_\pi(u) = 0$ for internal vertices $u$ in even levels of $T_i$ and $c_\pi(u) = 3 - \pi(v_i) \ne 0$, in odd levels. For leaves $w \in L^j_i$, with $N(w) = \{v\}$, if $j \equiv 0, 2, 3 \pmod 4$, then $\pi(v) = 3$. Otherwise, if $j \equiv 1 \pmod 4$, then $\pi(v) = \pi(v_i) \ne 0$. In both cases, we have $c_\pi(w) \ne c_\pi(v)$ and, hence, $c_\pi$ is a proper colouring of $T_i$.

It remains to consider items (iii), (iv) and (v), in which the label assigned to $v_i$ is 3.

*Case 1. $v_i$ corresponds to item (iii).*
For every $u \in L^j_i$, assign:

$$\pi(u) = \begin{cases} 2, & \text{if } u \in L^j_i, j \equiv 2 \pmod 4\text{; and} \\ 3, & \text{otherwise.} \end{cases}$$

This case is illustrated in Figure 3.43(a). Similar to the proof of the previous cases, note that internal vertices $u \in L^j_i$ have $\Pi_{N(u)} = \{3\}$ when $j$ is even, and $\Pi_{N(u)} = \{2, 3\}$,




(a) (b)

Figure 3.42: The labellings of $T_i$ for items (i) and (ii), in (a) and (b), respectively. White vertices have induced colour 0, black vertices, colour 1, orange vertices, colour 2, and violet vertices, colour 3.

otherwise. Therefore, $c_\pi(u) = j \mod 2$. For leaves $w \in L_i^j$ of $T_i$, adjacent to $N(w) = \{v\}$, note that $c_\pi(w) = \pi(v) \in \{2, 3\}$ and $c_\pi(v) \in \{0, 1\}$. Therefore, $c_\pi(w) \neq c_\pi(v)$.

*Case 2. $v_i$ corresponds to item (iv).*
In this case, assign:

$$\pi(u) = \begin{cases} 2, & \text{if } u \in L_i^j, j \equiv 1 \pmod 4; \text{ and} \\ 3, & \text{otherwise.} \end{cases}$$

Case 2 is illustrated in Figure 3.43(b). In this case, leaves $w$ of $T_i$, adjacent to vertices $v \in N(w)$, have $c_\pi(w) = \pi(v) \in \{2, 3\}$, while internal vertices $u \in L_i^j$ are labelled such that $\Pi_{N(u)} = \{3\}$ when $j$ is odd, and $\Pi_{N(u)} = \{2, 3\}$, otherwise. This induces $c_\pi(u) = 1 + (j \bmod 2)$ and, therefore $c_\pi(u) \in \{0, 1\}$.

*Case 3. $v_i$ corresponds to item (v).*
Lastly, assign:

$$\pi(u) = \begin{cases} 1, & \text{if } u \in L_i^j, j \equiv 1 \pmod 4; \text{ and} \\ 3, & \text{otherwise.} \end{cases}$$

This final case is illustrated in Figure 3.43(c). Here, internal vertices $u \in L_i^j$ have $\Pi_{N(u)} = \{3\}$ when $j$ is odd, and $\Pi_{N(u)} = \{1, 3\}$, otherwise. Therefore, $c_\pi(u) = 0$ when $j$ is odd, and $c_\pi(u) = 2$, when $j$ is even. For the leaves $w \in L_i^j$ of $T_i$, adjacent to $v$, when $j \equiv 0, 1, 3 \pmod 4$, $\pi(w) = 3 \neq c_\pi(v)$. Otherwise, that is, $j \equiv 2 \pmod 4$, note that $v$ is in level $j \equiv 1 \pmod 4$ and, therefore, has $\pi(v) = 1$. Moreover, $c_\pi(v) = 0$. This implies that $c_\pi(w) = \pi(v) \neq c_\pi(v)$.

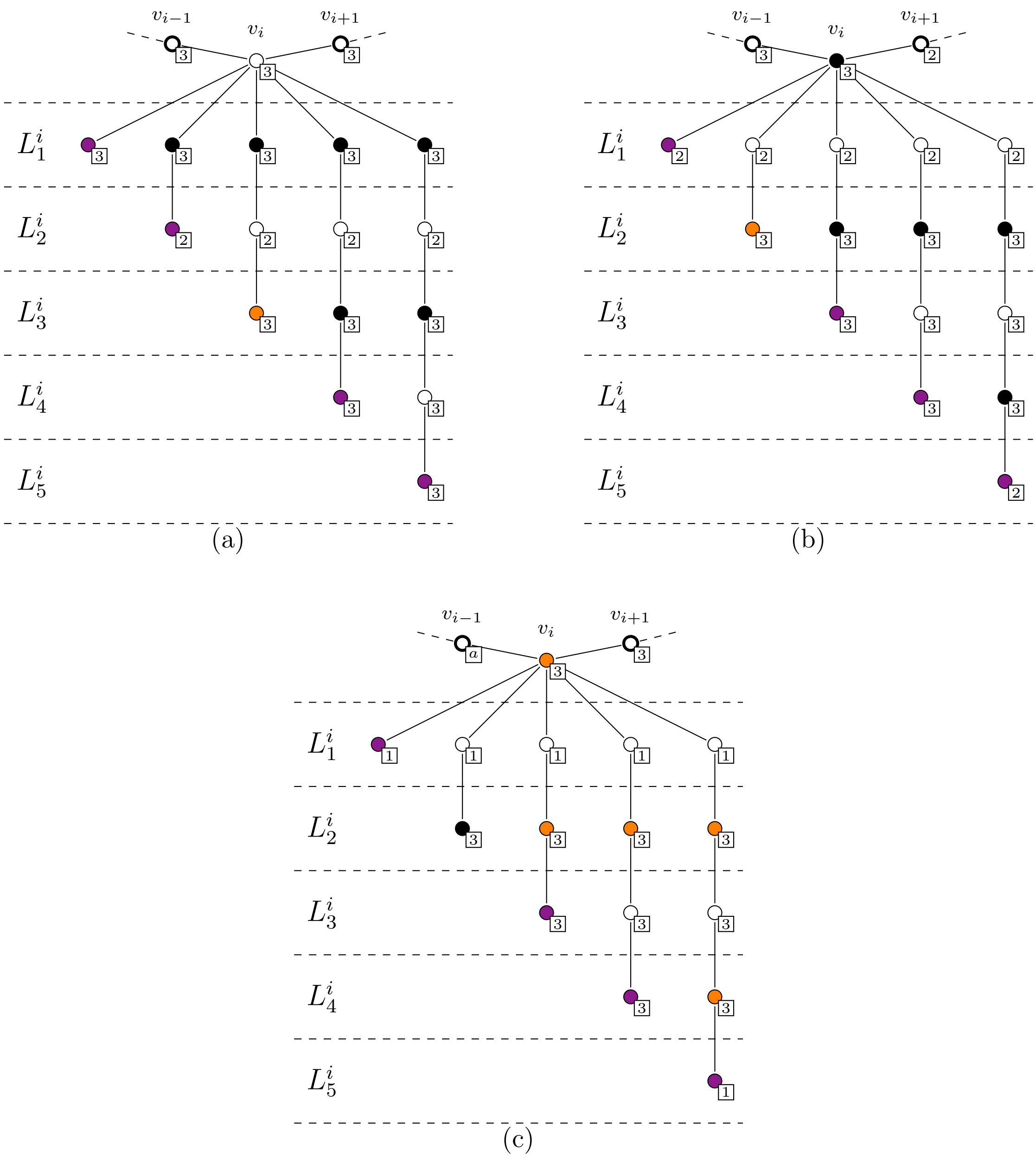



Figure 3.43: The labellings of vertices in $T_i$ for cases (iii), (iv) and (v) in (a), (b) and (c), respectively. Here, we use the same colouring scheme used in Figure 3.42.

With every possible case considered, we conclude that $c_\pi$ is a proper vertex-colouring of the graph. This completes the proof. □

We remark that the labelling of vertices $v_i$ in unicyclic graphs is similar to the one presented for cycles $C_n$. Looking back at the proof of Theorem 3.10, as well as the alternative proof for the cases of $n \equiv 1,2,3 \pmod 4$, the labelling presented here for unicyclic graphs is, to some extent, the complementary labelling of $\pi$ for cycles. Although this concept is not formally defined for gap-$[k]$-vertex-labellings, a *complementary labelling* $\bar{\pi}$ of $G$ is (usually) derived from a labelling $\pi : V(G) \to \{1,\dots,k\}$ and defined as $\bar{\pi}(v) = k - \pi(v)$, for every $v \in V(G)$. In many proper labellings, the complementary labelling gives some insight to structural properties of certain labellings and, here, this concept helped our findings of gap-$[\chi(G)]$-vertex-labelling of unicyclic graphs.

As a continuation of this research, we questioned whether the family of Cactus graphs admits a labelling based on our proposed labellings of unicyclic graphs. Due to time constraints, we could not fully investigate this problem. However, in a preliminary analysis of this class, we strongly believe that, with some modifications, it is possible to extend our labellings of unicyclic graphs to cactus graphs, and leave this question as a problem for future research.

**Problem 3.16.** *Is it possible to extend our gap-$[\chi(G)]$-vertex-labellings of unicyclic graphs $G$ to the family of Cactus graphs?*

### 3.3.5 Cubic bipartite graphs

Our study of the gap-$[k]$-vertex-labellings of graphs was motivated by questions proposed by Dehghan et al. [8] in 2013, where they ask if it is possible to determine the computational complexity of deciding whether a cubic bipartite graph $G$ admits a gap-[2]-vertex-labelling. This problem is proposed in the context of gap-[2]-vertex-labellings of $r$-regular bipartite graphs. The authors proved that every $r$-regular bipartite graph admits a gap-[2]-vertex-labelling when $r \geq 4$. In this work, we prove that this is also the case when $r = 2$, which are even cycles. However, the authors claim that not all 3-regular bipartite graphs admit gap-[2]-vertex-labellings and cite the Fano Plane as an example. Figure 3.44 illustrates the Fano Plane and the 3-regular bipartite graph obtained from it. This graph does not admit[5] a gap-[2]-vertex-labelling.

As previously mentioned, Dehghan et al. considered the computational complexity of deciding whether a cubic bipartite graph admits a gap-[2]-vertex-labelling. Despite several attempts, we could not find a *no* instance of this problem apart from the Heawood Graph. This led us to conjecture that every (other) cubic bipartite graph admits a gap-[2]-vertex-labelling. In the pursuit of verifying this conjecture, we started considering the subclass of cubic bipartite hamiltonian graphs. A *hamiltonian graph* is a graph that contains a cycle $C$ such that $V(C) = V(G)$. Herein, all graphs considered in this section are cubic bipartite hamiltonian graphs, and we refer to them as CBH-graphs.

Let $G = (V,E)$ be a CBH-graph of order $n$ and let $v_0, v_1, \dots, v_{n-1}$ denote the vertices of $G$ in the order of a hamiltonian cycle $C$ of $G$. This notation is used in all proofs

[5]The proof of this result is presented in Lemma 3.19.

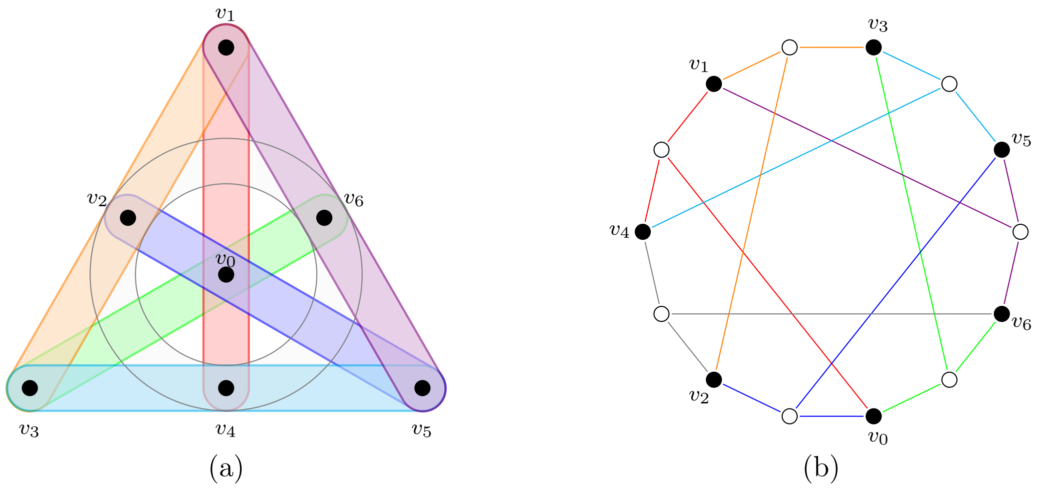



Figure 3.44: In (a), the Fano Plane $H$, with each hyperedge represented in a different colour; and in (b) the Heawood Graph. This graph is obtained by representing each hyperedge in $H$ as a (white) vertex in $G$, and each vertex of $H$, as a (black) vertex in $G$.

regarding CBH-graphs. An edge $e \in E$ that connects two nonadjacent vertices in $C$ is a *chord*. On the other hand, every edge $v_iv_{i+1}$ is a *cycle-edge*. The *reach* $r(e)$ of a chord $e$ is the size of the smallest path $v_i, \ldots, v_l$ using only cycle-edges. For example, every chord in the graph from Figure 3.44(b) has reach 5. If every chord in a CBH-graph has the same reach, we say the graph has *homogeneous chords*. For these graphs, in some cases, we denote the reach of the chords simply by $r$.

We remark that every CBH-graph has two properties: the order $n$ of the graph is always even; and every chord of the graph has odd reach. The smallest CBH-graph is the complete bipartite graph $K_{3,3}$ and it has an important role later in this section.

Our first approach for CBH-graphs considered the results for cycles established in Theorem 3.10. In fact, the gap-[2]-vertex-labelling of cycles $C_n$, $n \equiv 0 \pmod 4$, can be used to properly label CBH-graphs of order $n \equiv 0 \pmod 4$ as well. Theorem 3.17 presents this result.

**Theorem 3.17.** *Let $G$ be a CBH-graph of order $n \equiv 0 \pmod 4$. Then, $\chi^g_V(G) = 2$.*

*Proof.* Let $G$ be the graph stated in the hypothesis. By Corollary 3.8, in order to prove the result, it is sufficient to provide a gap-[2]-vertex-labelling of $G$. Define a labelling $\pi : V(G) \to \{1, 2\}$ as follows. For every vertex $v_i \in V$, let

$$\pi(v_i) = \begin{cases} 2, & \text{if } i \equiv 3 \pmod 4\text{; and} \\ 1, & \text{otherwise.} \end{cases}$$

Define colouring $c_\pi$ as usual. Note that this labelling is the same for cycles $C_n$, $n \equiv 0 \pmod 4$. Every vertex with even index has received the same label 1, while labels 1 and 2 alternate along the odd-index vertices in the hamiltonian cycle. Therefore, for vertices $v_i$, with $i$ odd, both their neighbours $v_{i-1}$ and $v_{i+1}$ received the same label. Furthermore, $v_i$ is adjacent to some $v_j$ by chord $e = v_iv_j$ and, since $r(e) \equiv 1 \pmod 2$, it follows that $j \equiv 0 \pmod 2$. Hence, $\pi(v_j) = 1$. Therefore, $\Pi_{N(v_i)} = \{1\}$ for all $v_i$, $i$ odd, which induces $c_\pi(v_i) = 0$.

Next, consider vertices $v_j$, $j$ even. Since $\pi$ uses only two labels and there are no degree-one vertices in $G$, the set of induced colours is $\{0, 1\}$. Therefore, in order to induce colour 1 in any vertex $v$ of $G$, it suffices to have two of its vertices labelled with 1 and 2. Since labelling $\pi$ alternates labels 1 and 2 in vertices with odd index and $n \equiv 0 \pmod 4$, every vertex $v_j$, $j$ even, has $\Pi_{N(v_j)} = \{1, 2\}$. Thus, we conclude that for every $v_i \in V(G)$, $c_\pi(v_i) = (i + 1) \bmod 2$, which completes the proof. □

With Theorem 3.17 established, it remains to consider CBH-graphs of order $n \equiv 2 \pmod 4$. It is important to remark that not all CBH-graphs have homogeneous chords. This is exemplified in Figure 3.45, where the illustrated graph of order 14 has chords of reach 3, 5 and 7. In fact, we prove in Theorem 3.18 that the existence of a chord $e \in E(G)$ in a CBH-graph $G$ such that $r(e) \equiv 3 \pmod 4$ is a sufficient condition for $G$, of order $n \equiv 2 \pmod 4$, to admit a gap-[2]-vertex-labelling.

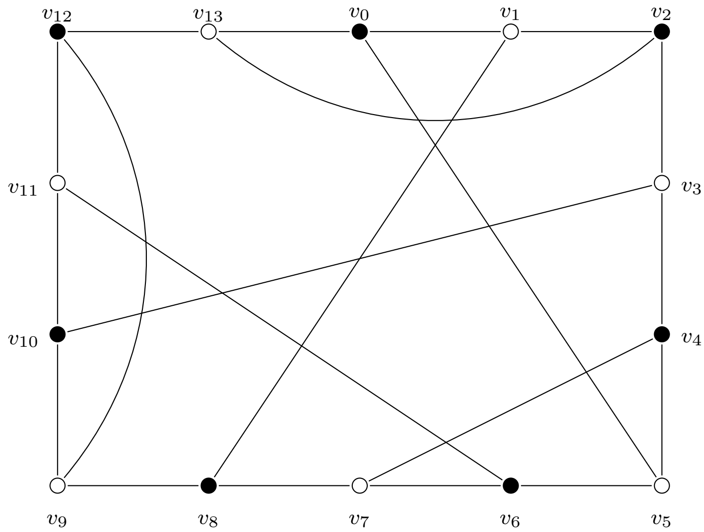



Figure 3.45: A CBH-graph of order $n = 14$. Observe, for instance, that $r(v_0v_{11}) = r(v_2v_5) = 3$, while $r(v_4v_9) = 5$ and $r(v_1v_8) = 7$.

**Theorem 3.18.** *Let $G$ be a CBH-graph of order $n \equiv 2 \pmod 4$. If there exists a chord $e \in E(G)$ such that $r(e) \equiv 3 \pmod 4$, then $\chi_V^g(G) = 2$.*

*Proof.* Let $G$ be a graph as stated in the hypothesis and $e$, a chord of $G$ with reach $r(e) \equiv 3 \pmod 4$. Adjust notation so that $e = v_0v_l$, for $l \equiv 3 \pmod 4$. By Corollary 3.8, it suffices to show a gap-[2]-vertex-labelling of $G$.

Define a labelling $\pi$ of $G$ as follows. For every vertex $v_j$, $j$ even, let $\pi(v_j) = 1$. Next, assign labels 1, 2, alternately, to vertices $v_1, v_3, \ldots, v_{n-3}, v_{n-1}$, starting with $\pi(v_1) = 1$. Define colouring $c_\pi$ as usual. This labelling is exemplified in Figure 3.46.

In order to prove that $(\pi, c_\pi)$ is a gap-[2]-vertex-labelling, we show that $c_\pi$ is a proper colouring of $G$. First, since every even-index $v_j$ receives label 1 and $G$ is connected, every $v_i$, $i$ odd, has $\Pi_{N(v_i)} = \{1\}$, inducing $c_\pi(v_i) = 0$. Next, observe that, for every $v_j$, $2 \leq j \leq n - 2$ and even, we have $\{\pi(v_{j-1}), \pi(v_{j+1})\} = \{1, 2\}$, which implies $c_\pi(v_j) = 1$ for these vertices. The last vertex to be considered is $v_0$. Since $l \equiv 3 \pmod 4$, we know

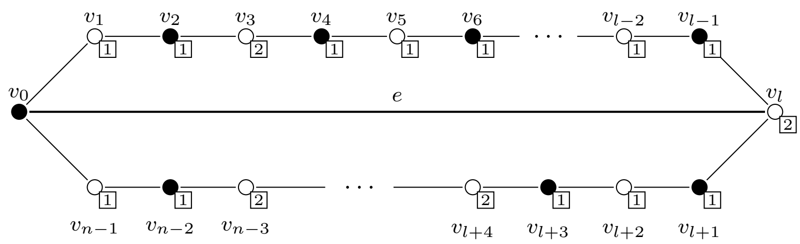



Figure 3.46: The gap-[2]-vertex-labelling $(\pi, c_\pi)$ of a graph $G$ of order $n \equiv 2 \pmod 4$. Chord $e$ has reach $r(e) \equiv 3 \pmod 4$.

that $v_l$ has received label 2. Also, $\pi(v_1) = \pi(v_{n-1}) = 1$. Therefore, $\Pi_{N(v_0)} = \{1, 2\}$, and we conclude that $c_\pi(v_0) = 1$. Thus, $c_\pi(v_l) = (l + 1) \bmod 2$ for every $v_l \in V(G)$. and the result follows. $\square$

Theorems 3.17 and 3.18 already provide a large coverage of the family of CBH-graphs. The only graphs in this class that remain to be considered are CBH-graphs $G$ with order $n \equiv 2 \pmod 4$ such that every chord $e \in G$ has $r(e) \equiv 1 \pmod 4$. In fact, Dehghan et al.'s [8] counterexample, the Heawood Graph, is such a graph. Now, we prove that this graph does not admit a gap-[2]-vertex-labelling.

**Lemma 3.19.** *Let $G$ be the Heawood Graph. Then, $\chi^g_V(G) = 3$.*

*Proof.* Let $G$ be the Heawood Graph. Every chord $e \in E(G)$ has reach $r(e) = 5$, as can be observed in Figure 3.44(b). In order to prove the result, we first show a gap-[3]-vertex-labelling $(\pi, c_\pi)$ of $G$ in Figure 3.47. By inspection, one can see that $c_\pi$ is a proper vertex-colouring of $G$.

The proof that $G$ does not admit a gap-[2]-vertex-labelling is by contradiction and essentially the same as the proof of Property 3.3. It is included here for completeness.

Suppose $(\pi, c_\pi)$ is a gap-[2]-vertex-labelling of $G$. Since $G$ is bipartite, we know that colours 0 and 1 alternate in the vertices of $G$. Adjust notation so that $c_\pi(v_i) = (i{+}1) \bmod 2$

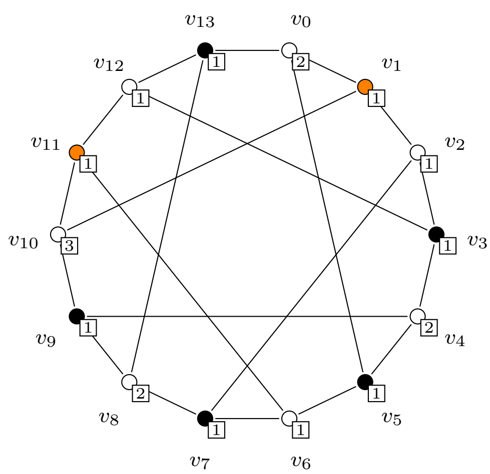
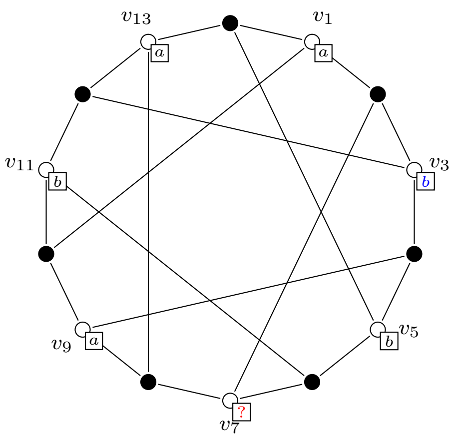



Figure 3.47: A gap-[3]-vertex-labelling $(\pi, c_\pi)$ of the Heawood Graph. White, black and orange vertices have induced colours 0, 1 and 2, respectively.

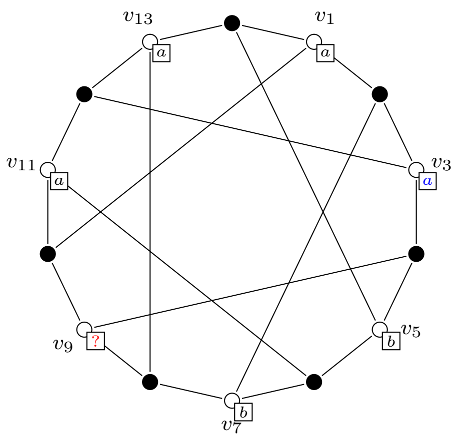
and every chord $e \in E(G)$ connects vertices $v_i$ and $v_{i+5}$. Operations on the indices are taken modulo $n$.

Since the colour of every vertex $v_i$, $i$ odd, is zero and $G$ is connected, it follows that every vertex $v_j$, $j$ even, receives the same label $\pi(v_j) = c \in \{1, 2\}$. Thus, it remains to determine the labels of odd-index vertices $v_i$. Consider $N(v_0) = \{v_1, v_{13}, v_5\}$. Since $c_\pi(v_0) = 1$, two different vertices of $N(v_0)$ receive labels $a, b \in \{1, 2\}$, $a \neq b$. First, suppose $\pi(v_1) = \pi(v_{13}) = a$ and $\pi(v_5) = b$. Since $v_1 \in N(v_2)$ and $c_\pi(v_2) = 1$, we consider the labels of $v_3$ and $v_{10}$, the other vertices in $N(v_2)$.

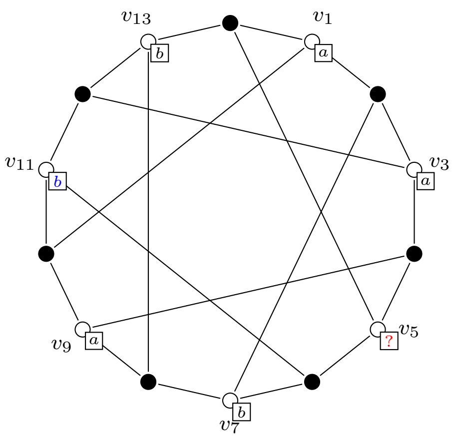
Suppose $\pi(v_3) = b$, as illustrated in Figure 3.48(a). Then, since $c_\pi(v_4) = 1$, $\Pi_{N(v_4)} = \{a, b\}$. It follows that $\pi(v_9) = a$. This, in turn, implies that $\pi(v_{11}) = b$ since $N(v_{10}) = \{v_9, v_{11}, v_1\}$ and $\pi(v_1) = \pi(v_9) = a$. However, note that there is no possible label for vertex $v_7$: if $\pi(v_7) = a$, then $\Pi_{N(v_8)} = \{a\}$; and if $\pi(v_7) = b$, then $\Pi_{N(v_6)} = \{b\}$. In both cases, we reach a contradiction. Therefore, $\pi(v_3) \neq b$, that is, $\pi(v_3) = a$. Figure 3.48(b) illustrates this case. However, a similar contradiction is reached: since $\pi(v_3) = \pi(v_1) = a$, it follows that $\pi(v_7) = b$ so as to induce $c_\pi(v_2) = 1$. This, in turn, implies that $\pi(v_{11}) = a$. Then, we have $\Pi_{N(v_{12})} = \{a\}$, inducing $c_\pi(v_{12}) = 0$, which is a contradiction.



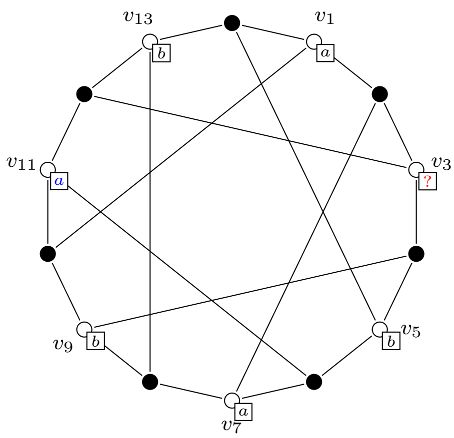

(a) Supposing $\pi(v_1) = \pi(v_{13}) = a$, and $\pi(v_3) = b$ (in blue).




(b) Supposing $\pi(v_1) = \pi(v_{13}) = \pi(v_3) = a$ (in blue).




(c)




(d)

Figure 3.48: The supposed gap-[2]-vertex-labelling of the Heawood Graph. Labels of vertices $v_j$, $j$ odd, have been omitted.

Therefore, our initial hypothesis, $\pi(v_1) = \pi(v_{13})$, is incorrect, and we conclude, without loss of generality, that $\pi(v_1) = a$ and $\pi(v_{13}) = b$. First, suppose $\pi(v_{11}) = b$, as depicted in Figure 3.48(c). Since $v_{11}, v_{13} \in N(v_{12})$ and $\pi(v_{11}) = \pi(v_{13})$, it follows that $\pi(v_3) = a$ so as to induce $c_\pi(v_{12}) = 1$. This, in turn, implies that $\pi(v_7) = b$ since $N(v_2) = \{v_1, v_3, v_7\}$ and $\pi(v_1) = \pi(v_3)$. Then, we have $\pi(v_{13}) = \pi(v_7) = b$ and, since $c_\pi(v_8) = 1$ and $v_7, v_{13} \in N(v_8)$, we conclude that $\pi(v_9) = a$. However, notice that vertex $v_5$ cannot be properly labelled: if $\pi(v_5) = a$, then $\Pi_{N(v_4)} = \{a\}$; and if $\pi(v_5) = b$, then $\Pi_{N(v_6)} = \{b\}$. Both cases induce colour 0 in a vertex with even index, which is a contradiction. Therefore, $\pi(v_{11}) = a$, as illustrated in Figure 3.48(d). By following the same line of reasoning, we conclude (sequentially) that $\pi(v_9) = b$, $\pi(v_7) = a$ and $\pi(v_5) = b$. Then, there is no label for vertex $v_3$ that induces a proper colouring of $G$, which is also a contradiction. Thus, the Heawood Graph does not admit a gap-[2]-vertex-labelling. $\square$

For the remainder of this section, only CBH-graphs with homogeneous chords are considered, that is, every graph $G$ is a CBH-graph of order $n \equiv 2 \pmod 4$ such that every chord $e \in E(G)$ has the same reach $r(e) \equiv 1 \pmod 4$. These graphs are denoted by $C_{n,\text{reach}=r}$. For example, the Heawood Graph in Figure 3.44(b) is $C_{14,\text{reach}=5}$ since it has order $n = 14$ and reach $r = 5$ for every chord.

For these graphs, we propose eight different techniques that create gap-[2]-vertex-labellings, depending on the values of $n$ and $r$: Techniques $T_1$, $T_2$ and $T_3$ create proper labellings for CBH-graphs $G$ in which the order of $G$ can be written as a multiple of the reach $r$; Techniques $T_4$ through $T_7$ use the fact that there are known labellings for other CBH-graphs which can be used to create gap-[2]-vertex-labellings of graphs of even greater order; lastly, Technique $T_8$ uses the concept of automorphism within the family of CBH-graphs.

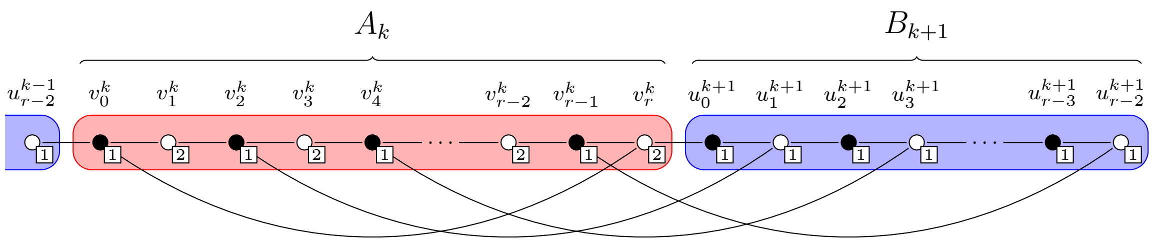
Before presenting the techniques, recall that in a gap-[2]-vertex-labelling of $G$ without degree-one vertices, the only possible induced colours are 0 and 1. Since only labels 1 and 2 are assigned to the vertices of $G$, it suffices to have two (of its three neighbours) with different labels.

**Technique $T_1$: $n = 2\alpha r$**

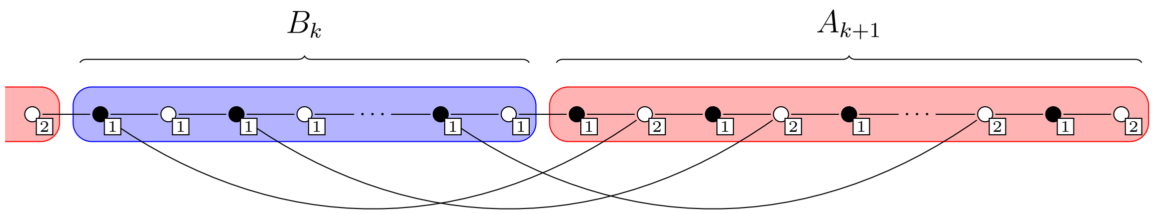
Let $G \cong C_{n,\text{reach}=r}$. Since $n = 2\alpha r$, $\alpha \in \mathbb{Z}_{>0}$, following the order of the indices of the vertices of $G$, partition $V(G)$ into $2\alpha$ blocks $A_1, B_2$, $A_3$, $B_4$, ..., $A_{2\alpha-1}$, $B_{2\alpha}$, such that $A_i = \{v^i_0, v^i_1, \ldots, v^i_r\}$, $B_i = \{u^i_0, u^i_1, \ldots, u^i_{r-2}\}$ and $v_0 = v^1_0$. Note that $|A_i| = r + 1$ and $|B_i| = r - 1$. Observe that, since $r \equiv 1 \pmod 4$, we have $|A| = 2 \pmod 4$ and $|B| \equiv 0 \pmod 4$. Also, since $n \equiv 2 \pmod 4$, we know that $\alpha$ is an odd number.

Assign label 1 to every vertex $v_j$, $j$ even. For blocks $A$, assign label 2 to the remaining odd-index vertices, and for blocks $B$, assign label 1. This labelling is displayed in Figure 3.49. Define colouring $c_\pi$ as usual. We prove that Technique $T_1$ properly labels the graph.

Since every vertex $v_j$, $j$ even, receives the same label, it follows that $c_\pi(v_i) = 0$ for every vertex $v_i \in V(G)$, $i$ odd. Therefore, we need only consider the induced colours of vertices $v_j$.




(a) Block $A_k$, adjacent to $B_{k+1}$. Note that every $v_j$, $j$ even, in $A_k$ is adjacent to a $u_i$, $i$ odd, in $B_{k+1}$, which is labelled with 1. This induces $c_\pi(v_i) = 1$ in every $v_i$, $i$ even.




(b) Block $B_k$, adjacent to $A_{k+1}$. Analogous to (a), every $u_j$, $j$ even, in $B_k$ is adjacent to a $v_i$, $i$ odd, in $A_{k+1}$, which is labelled with 2. This induces $c_\pi(v_i) = 1$ in every $v_i$, $i$ even.

Figure 3.49: The gap-[2]-vertex-labelling $(\pi, c_\pi)$ of $C_{n,\text{reach}=r}$ as described in the text

Consider an arbitrary block $A_k$, as depicted in Figure 3.49(a). Given that the size of $A_k$ is $r+1$, vertex $v_0^k$ is connected to the last vertex in $A_k$ by chord $v_0^k v_r^k$. Also, recall that $v_0^k$ is adjacent to $v_1^k$, which is assigned label 2, and vertex $u_{r-2}^{k-1}$, which receives label 1. Therefore, $\Pi_{N(v_0^k)} = \{1, 2\}$, inducing $c_\pi(v_0^k) = 1$.

Next, we consider vertices $v_j^k$, $2 \leq j \leq r$ and even. Observe that, for every $v_j^k$, two of its neighbours are in block $A_k$, namely $v_{j-1}^k$ and $v_{j+1}^k$. Thus, these vertices receive label 2. Since every chord in $G$ has reach $r \equiv 1 \pmod 4$ and the size of block $A_k$ is $r+1$, every $v_j^k$ is adjacent to $u_{j-1}^{k+1} \in V(B_{k+1})$, which has been assigned label 1. Therefore, $\Pi_{N(v_j^k)} = \{1, 2\}$ for every $v_j^k$, also inducing $c_\pi(v_j^k) = 1$. Figure 3.49(a) exemplifies this case.

Now, consider an arbitrary block $B_k$ and its adjacent blocks $A_{k-1}$ and $A_{k+1}$. Recall that $|B_k| = r - 1$. Therefore, every even-index vertex $u_j^k$ in $B_k$ is adjacent to vertex $v_{j+1}^{k+1}$ in block $A_{k+1}$, which is labelled with 2, and to $u_{j+1}^k$, which receives label 1. Therefore $\Pi_{N(u_j)} = \{1, 2\}$, and $c_\pi(u_j^k) = 1$. Therefore, $\pi$ induces a proper colouring of $B_k$, which can be observed in Figure 3.49(b).

Since every vertex $v_l \in V(G)$ has $c_\pi(v_l) = (l+1) \bmod 2$, we conclude that the proper labelling $(\pi, c_\pi)$ created by Technique $T_1$ is, in fact, a gap-[2]-vertex-labelling of $G$. Considering that $\alpha$ is an odd number and that $r \equiv 1 \pmod 4$ is bound by $\frac{n}{2}$, we present in Table 3.1 some values for $n$ and $r$ for which CBH-graphs $C_{n,\text{reach}=r}$ admit a gap-[2]-vertex-labelling created by Technique $T_1$.

An example of this labelling and the block partition is presented in Figure 3.50 for CBH-graph $C_{54,\text{reach}=9}$. In this graph, blocks $A_k$ are highlighted in red, and blocks $B_k$, in blue. The labelling scheme used for blocks $A$ and $B$ is also used in other techniques in this section. Thus, we define a *red labelling* as an assignment of alternating labels 1, 2 to the vertices of a given block, starting with 1. For instance, blocks $A_k$ in Technique $T_1$ are

| | $r = 5$ | $r = 9$ | $r = 13$ | $r = 17$ | $r = 21$ | $r = 25$ |
|---|---|---|---|---|---|---|
| $\alpha = 1$ | $C_{10,\text{reach}=5}$ | $C_{18,\text{reach}=9}$ | $C_{26,\text{reach}=13}$ | $C_{34,\text{reach}=17}$ | $C_{42,\text{reach}=21}$ | $C_{50,\text{reach}=25}$ |
| $\alpha = 3$ | $C_{30,\text{reach}=5}$ | $C_{54,\text{reach}=9}$ | $C_{78,\text{reach}=13}$ | $C_{102,\text{reach}=17}$ | $C_{126,\text{reach}=21}$ | $C_{150,\text{reach}=25}$ |
| $\alpha = 5$ | $C_{50,\text{reach}=5}$ | $C_{90,\text{reach}=9}$ | $C_{130,\text{reach}=13}$ | $C_{170,\text{reach}=17}$ | $C_{210,\text{reach}=21}$ | $C_{250,\text{reach}=25}$ |
| $\alpha = 7$ | $C_{70,\text{reach}=5}$ | $C_{126,\text{reach}=9}$ | $C_{182,\text{reach}=13}$ | $C_{238,\text{reach}=17}$ | $C_{294,\text{reach}=21}$ | $C_{350,\text{reach}=25}$ |
| $\alpha = 9$ | $C_{90,\text{reach}=5}$ | $C_{162,\text{reach}=9}$ | $C_{234,\text{reach}=13}$ | $C_{306,\text{reach}=17}$ | $C_{378,\text{reach}=21}$ | $C_{450,\text{reach}=25}$ |

Table 3.1: Examples of CBH-graphs covered by Technique $T_1$.

assigned a red labelling. Similarly, we define a *blue labelling* as the assignment of label 1 to every vertex in a block. This is the case for blocks $B_k$.

Technique $T_1$ shows that if a *red block* $A_i$, of cardinality $r+1$, is adjacent to a *blue block* $B_{i+1}$, of cardinality $r-1$, then the induced colours in vertices of $A_i$ alternate between 1 and 0, with $c_\pi(v_0^i) = 0$. Now, consider a red block $A$ of cardinality $r-1$ adjacent to a blue block $B$, also of cardinality $r-1$. Note that all even-index vertices $v_i$ in $A$ have (at least) one neighbour in $A$ and are adjacent to vertex $u_{i+r}$ in $B$. These vertices are labelled with 2 and 1, respectively. A similar reasoning applies to even index vertices of blue blocks. Thus, although the cardinality of the red block is different, the induced colouring remains a proper vertex-colouring of the CBH-graph. By using this slight modification, the next technique properly labels some graphs which are not covered by Technique $T_1$.

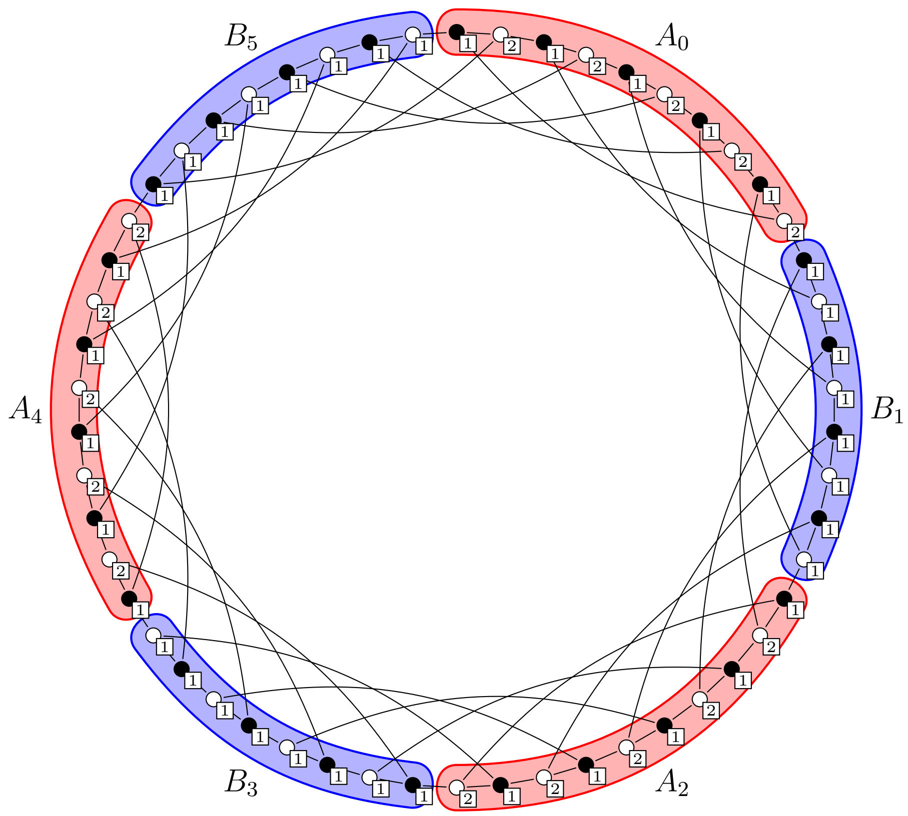



Figure 3.50: The gap-[2]-vertex-labelling of $C_{54,\text{reach}=9}$ obtained by Technique $T_1$. In this case, $\alpha = 3$.

**Technique $T_2$: $n = (r+1) + \alpha(r-1)$, $\alpha$ odd**

Let $G \cong C_{n,\text{reach}=r}$. In this case, we create a gap-[2]-vertex-labelling of $G$ similarly to Technique $T_1$. First, partition the $n$ vertices of $G$ into a single red block $A_0$, of size $r+1$, and an odd number $\alpha$ of alternating blue and red blocks, each of size $r-1$. Define colouring $c_\pi$ as usual. We remark that the proof that $(\pi, c_\pi)$ is a gap-[2]-vertex-labelling of $G$ is similar to that of Technique $T_1$. An example of this labelling technique is presented in Figure 3.51 for $C_{42,\text{reach}=5}$, a graph which was not covered by Technique $T_1$. In Table 3.2, we present some values of $n$ and $r$ that are covered by Technique 2.

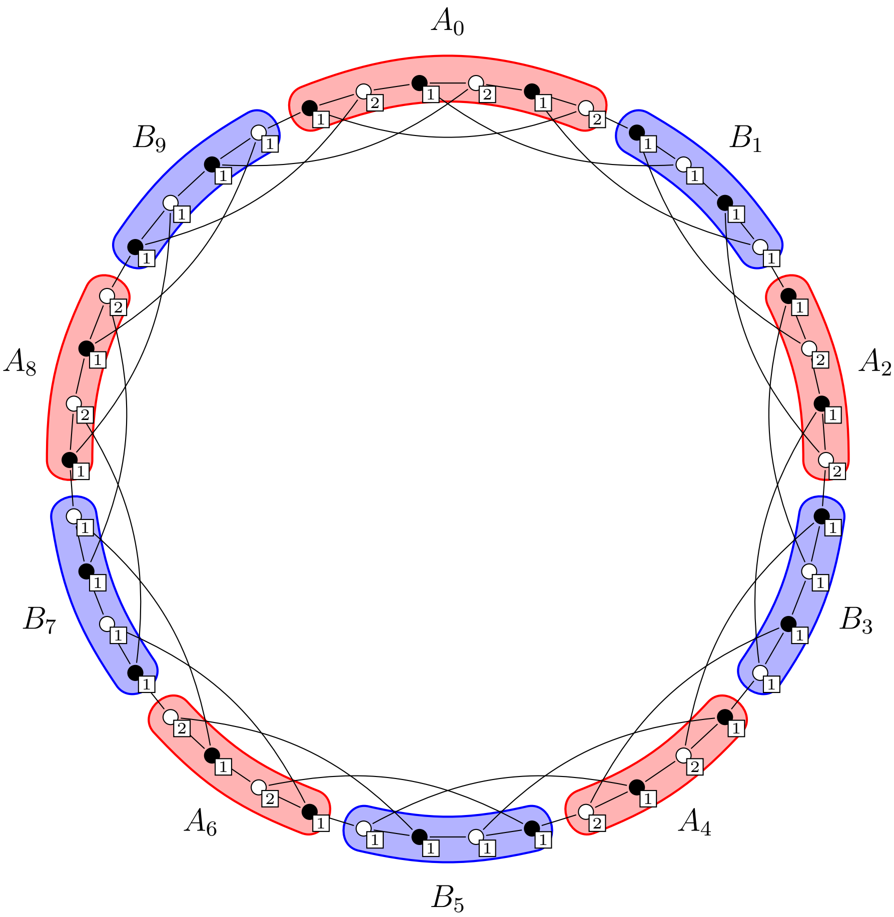



Figure 3.51: The gap-[2]-vertex-labelling of graph $C_{42,\text{reach}=5}$ by Technique $T_2$. For this graph, we have $\alpha = 9$.

| | $r=5$ | $r=9$ | $r=13$ | $r=17$ | $r=21$ | $r=25$ |
|---|---|---|---|---|---|---|
| $\alpha=1$ | $C_{10,\text{reach}=5}$ | $C_{18,\text{reach}=9}$ | $C_{26,\text{reach}=13}$ | $C_{34,\text{reach}=17}$ | $C_{42,\text{reach}=21}$ | $C_{50,\text{reach}=25}$ |
| $\alpha=3$ | $C_{18,\text{reach}=5}$ | $C_{34,\text{reach}=9}$ | $C_{50,\text{reach}=13}$ | $C_{66,\text{reach}=17}$ | $C_{82,\text{reach}=21}$ | $C_{98,\text{reach}=25}$ |
| $\alpha=5$ | $C_{26,\text{reach}=5}$ | $C_{50,\text{reach}=9}$ | $C_{74,\text{reach}=13}$ | $C_{98,\text{reach}=17}$ | $C_{122,\text{reach}=21}$ | $C_{146,\text{reach}=25}$ |
| $\alpha=7$ | $C_{34,\text{reach}=5}$ | $C_{66,\text{reach}=9}$ | $C_{98,\text{reach}=13}$ | $C_{130,\text{reach}=17}$ | $C_{162,\text{reach}=21}$ | $C_{194,\text{reach}=25}$ |
| $\alpha=9$ | $C_{42,\text{reach}=5}$ | $C_{82,\text{reach}=9}$ | $C_{122,\text{reach}=13}$ | $C_{162,\text{reach}=17}$ | $C_{202,\text{reach}=21}$ | $C_{242,\text{reach}=25}$ |

Table 3.2: Examples of CBH-graphs covered by Technique $T_2$.

Another modification can be made to this labelling technique by increasing the number of blocks of size $r+1$. This modification allows us to properly label new CBH-graphs, and we present this result in Technique $T_3$.

**Technique $T_3$: $n = \beta(r+1) + \alpha(r-1)$, $\alpha, \beta$ odd**

Let $G \cong C_{n,\text{reach}=r}$. If $n = \beta(r+1) + \alpha(r-1)$, we create a gap-[2]-vertex-labelling of $G$ as follows. Partition the $n$ vertices of $G$ into $\beta + \alpha$ blocks as follows: let $A_0, B_1, \ldots, A_{\beta-1}$ be the $\beta$ first blocks of size $r+1$, which alternate between red and blue labellings; the remaining vertices are partitioned into $\alpha$ alternating blue and red blocks $B_\beta, A_{\beta+1}, \ldots, B_{\beta+\alpha-1}$, each of size $r-1$. Once again, the partition carries the gap-[2]-vertex-labelling since it is done considering the red and blue blocks.

We prove that Technique $T_3$ properly labels $G$ by induction on $\beta$. The basis case is $\beta = 1$, that is, when $G$ is partitioned into a single block of size $(r+1)$ and an odd number $\alpha$ of blocks of size $(r-1)$. Note that this is the partition of $G$ in Technique $T_2$, which has been proven to properly label CBH-graphs.

Now, suppose that $(\pi, c_\pi)$ is a gap-[2]-vertex-labelling obtained by Technique $T_3$ of a CBH-graph $G = C_{n,\text{reach}=r}$, with order $n = \beta(r+1) + \alpha(r-1)$, $\alpha, \beta$ odd and $\beta \geq 1$. We consider a new CBH-graph $G' = C_{n',\text{reach}=r}$ such that $n' = n + 2(r+1)$. Note that $G'$ has two more blocks of size $(r+1)$. Hence, we can write $\beta' = \beta + 2$. Also, since $r \equiv 1 \pmod 4$, note that $n' = n + 2(r+1)$ and, therefore, $n' \equiv 2 \pmod 4$.

Let $A_0, B_1, \ldots, A_{\beta-1}, B_\beta, A_{\beta+1}, \ldots, B_{\beta+\alpha-1}$ be the partition of $V(G)$ into $\alpha+\beta$ blocks. Recall that, by our hypothesis, this partition defines a gap-[2]-vertex-labelling of $G$. We create a gap-[2]-vertex-labelling $(\pi', c_{\pi'})$ of $G'$ by adding two more $(r+1)$-sized blocks between blocks $B_{\beta+\alpha-1}$ and $A_0$. Let $A_{\beta+\alpha}, B_{\beta+\alpha+1}$ be these blocks.

Now, for every $v \in V(G)$, define label $\pi'(v) = \pi(v)$. Observe that all blocks $A_i, B_i$ from $G$ have their respective red and blue labellings copied to labelling $\pi'$ of $G'$. Next, assign a red labelling to block $A_{\beta+\alpha}$ and a blue labelling to block $B_{\beta+\alpha+1}$. Then, in order to prove that $(\pi', c_{\pi'})$ is a gap-[2]-vertex-labelling of $G$, it suffices to show that $c_{\pi'}$ is a proper colouring of $G$.

First, consider blocks $B_1, \ldots, A_{\beta+\alpha-2}$ and notice that their adjacent blocks remain unchanged: red (blue) blocks in $G$ continue to be red (blue) blocks in $G'$. Therefore, since $c_{\pi'}$ is a proper colouring of $G$, restricted to these vertices, it follows that the induced colours of vertices in these blocks in $G'$ remains unchanged. Thus, no adjacent vertices in $B_1, \ldots, A_{\beta+\alpha-2}$ have the same induced colour.

Now, consider block $B_{\beta+\alpha-1}$ and recall that this block, in $G$, is adjacent to blocks $A_{\beta+\alpha-2}$ and $A_0$, whose sizes are $(r-1)$ and $(r+1)$, respectively. By our construction of $G$, block $B_{\beta+\alpha-1}$ is now adjacent to $A_{\beta+\alpha-2}$ and to a new red block $A_{\beta+\alpha}$, which also have sizes $(r-1)$ and $(r+1)$, respectively. Therefore, it follows that the colours of vertices in $B_{\beta+\alpha-1}$ also remain unchanged.

It remains to consider blocks $A_{\beta+\alpha}, B_{\beta+\alpha+1}$ and $A_0$. Note that all these blocks are of size $r+1$. Similarly to the proofs of Techniques $T_1$ and $T_2$, observe that when a red (blue) block $X_i$, of size $(r+1)$, is adjacent to two blue (red) blocks of the same size, $Y_{i-1}$ and $Y_{i+1}$, every vertex $v_i \in V(X_i)$, $i$ even, has $\Pi'_{N(v_i)} = \{1, 2\}$. Therefore, we conclude that $c_{\pi'}$ is a proper colouring of $G$. Therefore, Technique $T_3$ creates a gap-[2]-vertex-labelling. Figure 3.52 illustrates the gap-[2]-vertex-labelling obtained by Technique $T_3$ for CBH-graph $C_{70,\text{reach}=9}$, which has $\beta = 3$ and $\alpha = 5$.

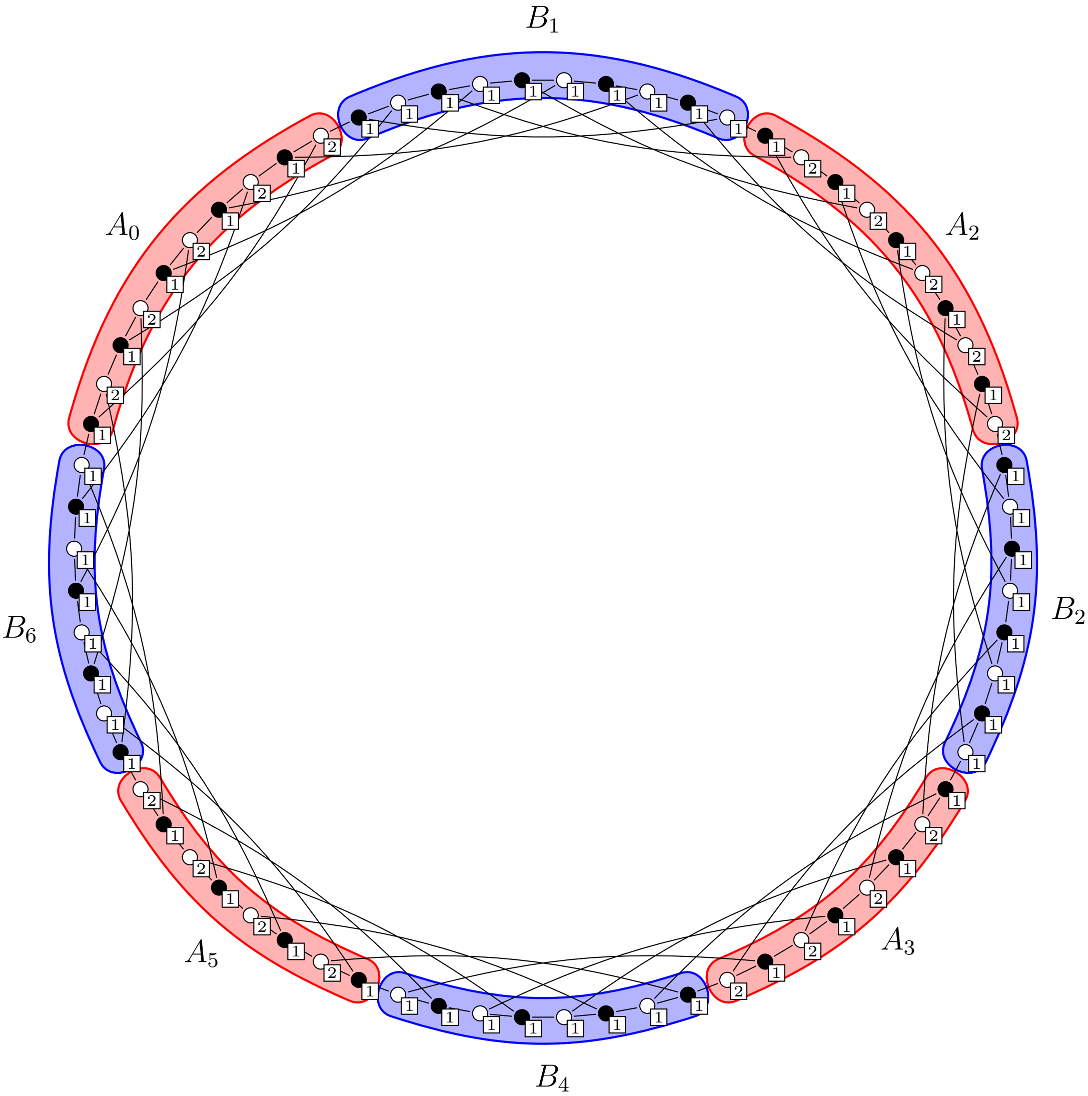



Figure 3.52: The gap-[2]-vertex-labelling of graph $C_{70,\text{reach}=9}$ by Technique $T_3$. For this graph, $\beta = 3$ and $\alpha = 5$.

In Table 3.3, we present some values for $n$ and $r$ which are covered by Technique $T_3$, depending on the values of $\alpha$ and $\beta$.

The three techniques presented thus far have one common factor: the division of blocks is done as a function of $r$. This implies that, for large values of $r$, there will be larger and larger "gaps" between the values of $n$ covered by them. For example, consider $r = 25$ and $\beta = 3$ in Table 3.3. Between two consecutive odd values of $\alpha$, for example $C_{102,\text{reach}=25}$ and $C_{150,\text{reach}=25}$, there are twelve CBH-graphs which Technique $T_3$ does not cover: $C_{106,\text{reach}=25}, C_{110,\text{reach}=25}, \ldots, C_{146,\text{reach}=25}$. With this in mind, we decided to take a different approach. In the following techniques, we partition the vertices of $G$ into fixed-size blocks. Before we present these techniques, let us define a 6-block.

| $\beta = 3$ | | | | | | |
|---|---|---|---|---|---|---|
| | $r = 5$ | $r = 9$ | $r = 13$ | $r = 17$ | $r = 21$ | $r = 25$ |
| $\alpha = 1$ | $C_{22,\text{reach}=5}$ | $C_{38,\text{reach}=9}$ | $C_{54,\text{reach}=13}$ | $C_{70,\text{reach}=17}$ | $C_{86,\text{reach}=21}$ | $C_{102,\text{reach}=25}$ |
| $\alpha = 3$ | $C_{30,\text{reach}=5}$ | $C_{54,\text{reach}=9}$ | $C_{78,\text{reach}=13}$ | $C_{102,\text{reach}=17}$ | $C_{126,\text{reach}=21}$ | $C_{150,\text{reach}=25}$ |
| $\alpha = 5$ | $C_{38,\text{reach}=5}$ | $C_{70,\text{reach}=9}$ | $C_{102,\text{reach}=13}$ | $C_{134,\text{reach}=17}$ | $C_{166,\text{reach}=21}$ | $C_{198,\text{reach}=25}$ |
| $\alpha = 7$ | $C_{46,\text{reach}=5}$ | $C_{86,\text{reach}=9}$ | $C_{126,\text{reach}=13}$ | $C_{166,\text{reach}=17}$ | $C_{206,\text{reach}=21}$ | $C_{246,\text{reach}=25}$ |
| $\alpha = 9$ | $C_{54,\text{reach}=5}$ | $C_{102,\text{reach}=9}$ | $C_{150,\text{reach}=13}$ | $C_{198,\text{reach}=17}$ | $C_{246,\text{reach}=21}$ | $C_{294,\text{reach}=25}$ |
| $\beta = 5$ | | | | | | |
| | $r = 5$ | $r = 9$ | $r = 13$ | $r = 17$ | $r = 21$ | $r = 25$ |
| $\alpha = 1$ | $C_{34,\text{reach}=5}$ | $C_{58,\text{reach}=9}$ | $C_{82,\text{reach}=13}$ | $C_{106,\text{reach}=17}$ | $C_{130,\text{reach}=21}$ | $C_{154,\text{reach}=25}$ |
| $\alpha = 3$ | $C_{42,\text{reach}=5}$ | $C_{74,\text{reach}=9}$ | $C_{106,\text{reach}=13}$ | $C_{138,\text{reach}=17}$ | $C_{170,\text{reach}=21}$ | $C_{202,\text{reach}=25}$ |
| $\alpha = 5$ | $C_{50,\text{reach}=5}$ | $C_{90,\text{reach}=9}$ | $C_{130,\text{reach}=13}$ | $C_{170,\text{reach}=17}$ | $C_{210,\text{reach}=21}$ | $C_{250,\text{reach}=25}$ |
| $\alpha = 7$ | $C_{58,\text{reach}=5}$ | $C_{106,\text{reach}=9}$ | $C_{154,\text{reach}=13}$ | $C_{202,\text{reach}=17}$ | $C_{250,\text{reach}=21}$ | $C_{298,\text{reach}=25}$ |
| $\alpha = 9$ | $C_{66,\text{reach}=5}$ | $C_{122,\text{reach}=9}$ | $C_{178,\text{reach}=13}$ | $C_{234,\text{reach}=17}$ | $C_{290,\text{reach}=21}$ | $C_{346,\text{reach}=25}$ |

Table 3.3: Examples of CBH-graphs covered by Technique $T_3$.

**Definition 3.20.** *Let $G \cong C_{6,reach=3}$ and let $(\pi, c_\pi)$ be the labelling of $G$ presented in Figure 3.53(a). A* 6-block $\Gamma$ *is the group of labelled vertices obtained by removing edge $v_0v_5$ and all chords of $G$. An illustration of $\Gamma$ is presented in Figure 3.53(b).*

For a 6-block $\Gamma^i$, let $V(\Gamma^i) = \{v_0^i, v_1^i, \ldots, v_5^i\}$ denote its vertex set; superscript $i$ is added to the vertex names so as to indicate which 6-block they belong to. Consider, first, vertices $v_2^i$ and $v_4^i$. Observe that their respective neighbours $v_1^i, v_3^i$ and $v_3^i, v_5^i$ are labelled such that $\Pi_{N(v_2^i)} = \Pi_{N(v_4^i)} = \{1, 2\}$. Therefore, these two vertices have induced colour 1.

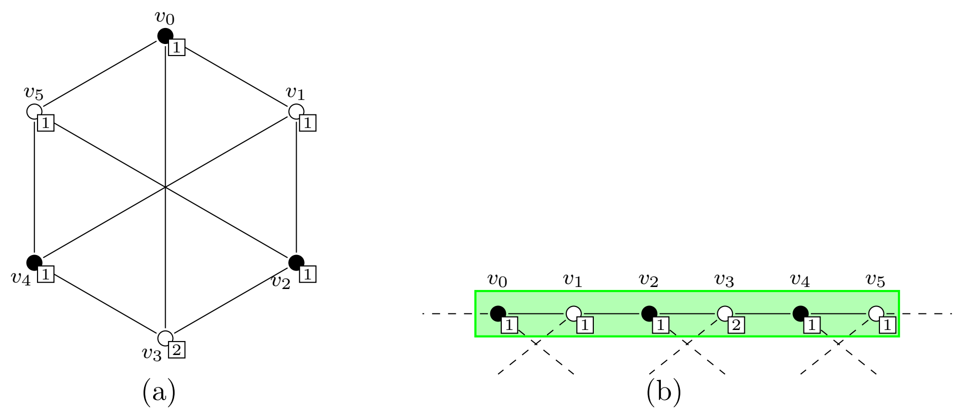



Figure 3.53: In (a), graph $C_{6,\text{reach}=3}$; and in (b), the 6-block.

Next, consider vertex $v_0^i$ and chord $v_0^iv_3^i$, removed from $C_{6,\text{reach}=3}$. This chord link $v_3$ and $v_0$ in the original graph. This, in turn, induces $c_\pi(v_0) = 1$ since $\pi(v_1) = \pi(v_5) = 1$ and $\pi(v_3) = 2$. Therefore, in order to preserve the proper vertex-colouring of $\Gamma^i$, it is sufficient for $v_0^i$ to be adjacent to a vertex $v_3^j$ in some (other) 6-block $\Gamma^j$. Then, $\Pi_{N(v_0^i)} = \{1, 2\}$ induces colour 1 in $v_0^i$ as desired. Moreover, in particular for chord $e = v_0^iv_3^j$, its reach can be determined by $r(e) = 3 + 6[(j - i) \bmod \alpha]$. By considering chords with reach $r = 6\gamma + 3$, $\gamma$ can be interpreted as the distance, or *skip*, between 6-blocks $\Gamma^i$ and $\Gamma^j$ containing vertices $v_0^i$ and $v_3^j$.

The first use of 6-blocks to create gap-[2]-vertex-labellings of CBH-graphs is presented in Technique $T_4$.

**Technique $T_4$: 6-blocks when $n \equiv 0 \pmod 6$ and $r = 6\gamma + 3$.**

Let $G$ be a CBH-graph of order $n = 6\alpha$. Partition $V(G)$ into $\alpha$ 6-blocks $\Gamma^1, \Gamma^2, \ldots, \Gamma^\alpha$, with $v_0^1 = v_0$. Since every $v_l$, $l$ even, receives label 1 and $G$ is connected, we conclude that every vertex with odd index has induced colour 0. Thus, it remains to consider the induced colours of even-index vertices.

First, consider the case where $\gamma = 1$, that is, every chord skips only one block. Consequently, every $v_0^i$ is adjacent to $v_3^{i+1}$, as illustrated in Figure 3.54(a). Note that the adjacencies from the original blocks are preserved and, thus, the proper colouring from $C_{6,\text{reach}=3}$ is maintained in every $\Gamma^i$. Therefore, this partition properly labels $G$. This also happens for $\gamma = 2$, illustrated in Figure 3.54(b). In this case, $v_0^i$ is adjacent to $v_3^{i+2}$ and, consequently, chord $v_0^i v_3^{i+2}$ has reach $15 \equiv 3 \pmod 4$. However, CBH-graphs with chords with reach $r(e) \equiv 3 \pmod 4$ are already covered by Theorem 3.18.

Consider, now, the case $\gamma = 3$, which is illustrated in Figure 3.54(c). Once again, colouring $c_\pi$ is preserved in the graph. Moreover, every chord has reach $21 \equiv 1 \pmod 4$. In fact, $\gamma$ odd implies $r \equiv 1 \pmod 4$ and Figure 3.54(d) illustrates this general case. Note that in all cases, chord $v_0^i v_3^{i+\gamma}$ guarantees that $c_\pi(v_0^i) = 1$. Thus, every CBH-graph of order $n = 6\alpha$ with homogeneous chords with reach $r = 6\gamma + 3$, with $\gamma$ odd, can be properly labelled by Technique $T_4$.

In Figure 3.55, we illustrate Technique $T_4$ for two graphs with different values of $\gamma$. Figure 3.55(a) illustrates the case where $r = 9$, that is, the connection is made between adjacent 6-blocks. On the other hand, Figure 3.55(b) exemplifies a case where $r = 21$ and, thus, the skip is 3.

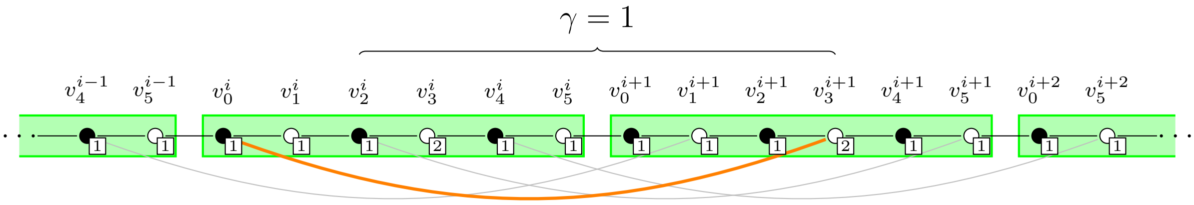
Technique $T_4$ covers all CBH-graphs of order $n \equiv 0 \pmod 6$ with chords $r \equiv 3 \pmod 6$. In Techniques $T_5$ and $T_6$, we continue to address graphs with chords of reach $6\gamma + 3$, covering cases of $n \equiv 2 \pmod 6$ and $n \equiv 4 \pmod 6$, respectively.

**Technique $T_5$: 6-blocks when $n \equiv 2 \pmod 6$ and $r = 6\gamma + 3$.**

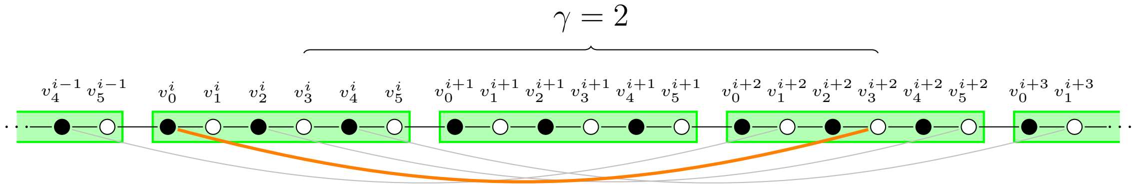
Let $G$ be a CBH-graph with $n = 6\alpha + 2$. Technique $T_5$ creates a gap-[2]-vertex-labelling $(\pi, c_\pi)$ of $G$ as follows. Partition the vertex set of $G$ into $\alpha$ 6-blocks and a *residual block* $\Gamma'$ containing the remaining two vertices. We refer to this residual block $\Gamma' = \{v_{n-2}, v_{n-1}\}$ as the *tail* of $G$, and assign a blue labelling to it – that is, $\pi(v_{n-2}) = \pi(v_{n-1}) = 1$. Finally, we alter the labels of the last vertex in each of the $\gamma + 1$ last 6-blocks of $G$. Thus, for $\alpha - \gamma \leq i \leq \alpha$, assign $\pi(v_5^i) = 2$. Figure 3.56 illustrates this labelling for CBH-graph $C_{26,\text{reach}=9}$. In this example, chords have reach $r = 9$ and, therefore, $\gamma = 1$.

In order to prove that $(\pi, c_\pi)$ is a gap-[2]-vertex-labelling of $G$, it suffices to show that $c_\pi$ is a proper vertex-colouring of the graph. Consider, initially, vertices $v_l$, $l$ odd. Since label 1 is assigned to every even-index vertex and $G$ is connected, $c_\pi(v_l) = 0$. It remains to consider vertices with even index.

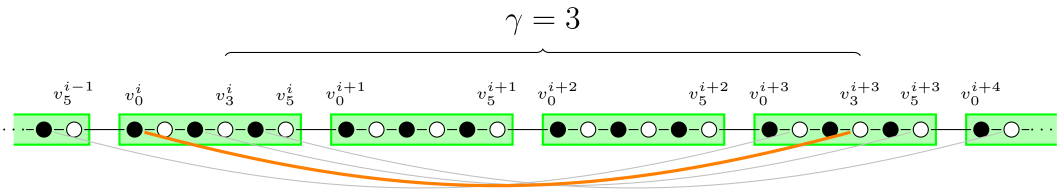
The label of vertices in block $\Gamma^j$, $1 \leq j < \alpha - \gamma$, is the same of the original 6-block. Therefore, $\Pi_{N(v_2^j)} = \Pi_{N(v_4^j)} = \{1, 2\}$ and, thus, $c_\pi(v_2^j) = c_\pi(v_4^j) = 1$. Vertex $v_0^j$ is adjacent



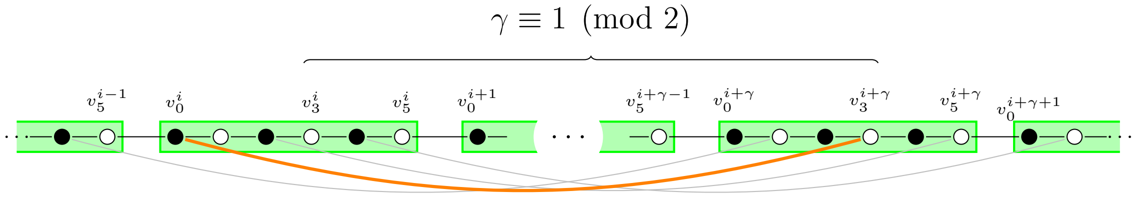

(a) Two adjacent 6-blocks, with the highlighted chord connecting vertices $v_0^i$ and $v_3^{i+1}$. In this case, $r = 9$ and $\gamma = 1$, where $\gamma$ is the skip of the chord.




(b) A representation of three adjacent 6-blocks. In this case, the highlighted chord has reach $r \equiv 3 \pmod 4$ since $\gamma$ is even.




(c) A representation of four adjacent 6-blocks. Here, the skip of a chord is $\gamma = 3$ and its reach, $r = 21 \equiv 1 \pmod 4$.




(d) A representation of adjacent 6-blocks. When $\gamma$ is odd, reach $r \equiv 1 \pmod 4$.

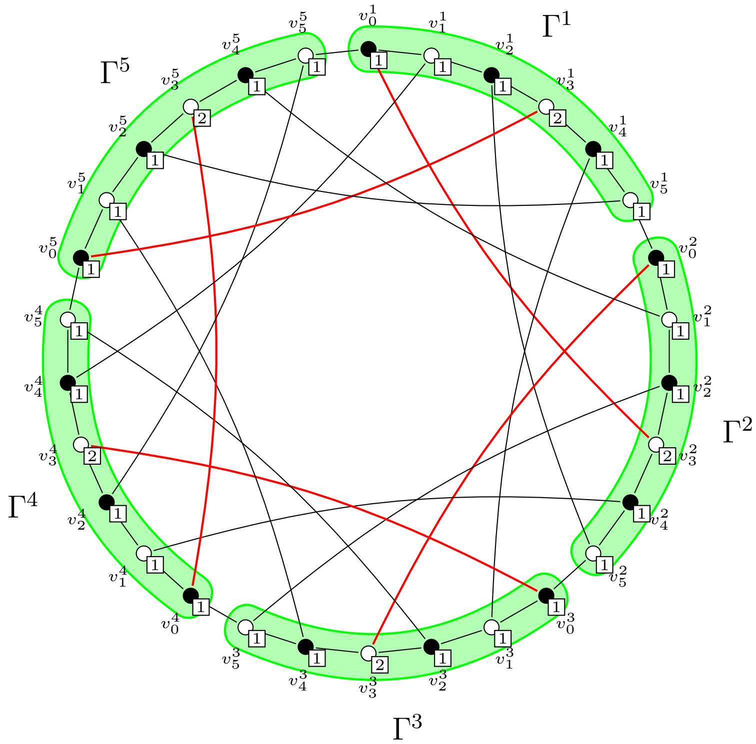
Figure 3.54: The usage of 6-blocks in CBH-graphs.

to $v_1^j$, which receives label 1, and to $v_3^{j+\gamma}$, whose label is 2. Therefore, $c_\pi(v_0^j) = 1$.

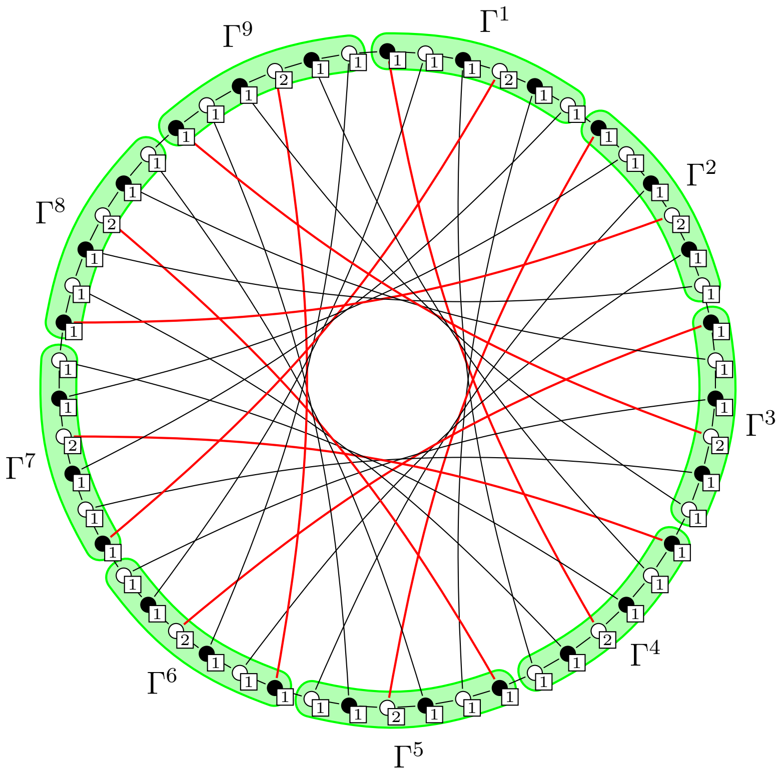
Next, consider block $\Gamma^{\alpha-\gamma}$. Note that $c_\pi(v_0^{\alpha-\gamma}) = 1$ and $c_\pi(v_2^{\alpha-\gamma}) = 1$ since their adjacencies preserve the properties of 6-blocks. Consider $v_4^{\alpha-\gamma}$. Note that chord $v_{n-1}v_4^{\alpha-\gamma}$ exists in $G$ and, since $\pi(v_{n-1}) = 1$ and $\pi(v_3^{\alpha-\gamma}) = 2$, we conclude that $c_\pi(v_4^{\alpha-\gamma}) = 1$.

Now, consider 6-block $\Gamma^j$, $\alpha - \gamma + 1 \leq j \leq \alpha$. Since $\{v_5^{j-1}, v_1^j\} \subseteq N(v_0^j)$, $\pi(v_5^{j-1}) = 2$ and $\pi(v_1^j) = 1$, we have that $c_\pi(v_0^j) = 1$. For vertex $v_2^j$, its neighbours in the cycle are labelled with 1 and 2. Thus, $c_\pi(v_2^j) = 1$. Finally, for $v_4^j$, both its neighbours in the cycle are labelled with 2. However, note that the chord which has $v_4^j$ as an end has $v_5^l$, $l = (j+\gamma) \bmod \alpha$ as the other. Since all vertices in $\Gamma^l$ keep their original labels, $\pi(v_5^l) = 1$ and we conclude that $c_\pi(v_4^j) = 1$.

It remains to consider $v_{n-2}$. Observe that this vertex is adjacent to $v_{n-1}$, which receives label 1, and $v_5^\alpha$, whose label is 2. Thus, $c_\pi(v_{n-2}) = 1$ and we conclude that $G$ is




(a)




(b)

Figure 3.55: The gap-[2]-vertex-labelling obtained by Technique $T_4$ of graphs (a) $C_{30,\text{reach}=9}$; and (b) $C_{54,\text{reach}=21}$. The highlighted chords connect vertices $v_0^i$ and $v_3^{i+\gamma}$, thus inducing $c_\pi(v_0^i) = 1$.

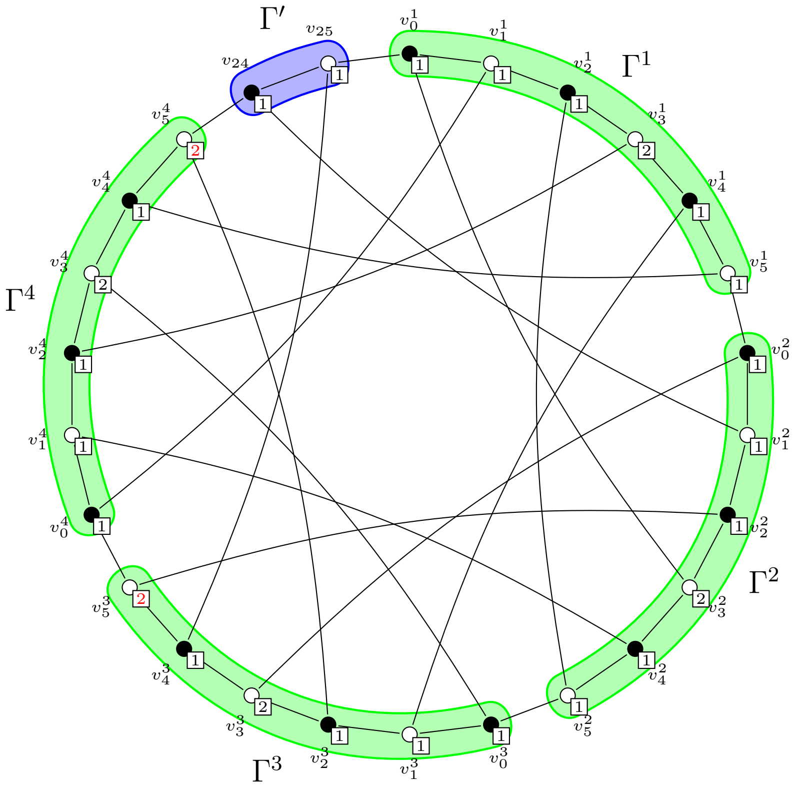



Figure 3.56: The gap-[2]-vertex-labelling of $C_{26,\text{reach}=9}$ created by Technique $T_5$. The modified labels in the $\gamma + 1$ last 6-blocks of $G$ are highlighted in red.

properly labelled by Technique $T_5$.

Thus, it remains to consider the case of CBH-graphs with chords $r \equiv 3 \pmod 6$ of order $n \equiv 4 \pmod 6$. In this last case, the tail of $G$ has 4 vertices. Technique $T_6$ presents a gap-[2]-vertex-labelling for these graphs, also based on the labellings of 6-blocks.

**Technique $T_6$: 6-blocks when $n \equiv 4 \pmod 6$ and $r = 6\gamma + 3$.**

Let $G$ be a CBH-graph with $n = 6\alpha + 4$. Partition $V(G)$ into $\alpha$ 6-blocks and a residual block $\Gamma' = \{v_{n-4}, v_{n-3}, v_{n-2}, v_{n-1}\}$, to which we assign a red labelling. Thus, $\pi(v_{n-4}) = \pi(v_{n-2}) = 1$ and $\pi(v_{n-3}) = \pi(v_{n-1}) = 2$. Finally, we alter the label of the second vertex, $v_1^i$, in the $\gamma - 1$ last 6-blocks of $G$, assigning $\pi(v_1^i) = 2$ in these blocks. Figure 3.57 exemplifies this labelling for CBH-graph $C_{58,\text{reach}=21}$, a case where chords have reach $r = 21$.

In order to prove that Technique $T_6$ properly labels $G$, it suffices to show that $c_\pi$ is a proper vertex-colouring. We begin by considering vertices $v_l$, $l$ odd. Since $\pi(v_i) = 1$ for every $v_i \in V(G)$ with $i$ even, we conclude that $c_\pi(v_l) = 0$, for every odd-index vertex $v_l \in V(G)$.

Consider $\Gamma^j$, $1 \leq j \leq \alpha - \gamma$. Since the 6-block labelling is preserved, we have $\Pi_{N(v_2^j)} = \Pi_{N(v_4^j)} = \{1, 2\}$, which induces colour 1 in these vertices. Vertex $v_0^j$ is adjacent to $v_1^j$ and to $v_3^{j+\gamma}$ which receive labels 1 and 2, respectively. Therefore, $c_\pi(v_0^j) = 1$.

Next, consider the singular case of $\Gamma^j$, $j = \alpha - \gamma + 1$. The labelling for this block remains unchanged. Therefore, $c_\pi(v_2^j) = c_\pi(v_4^j) = 1$. Vertex $v_0^j$, in this block, is adjacent

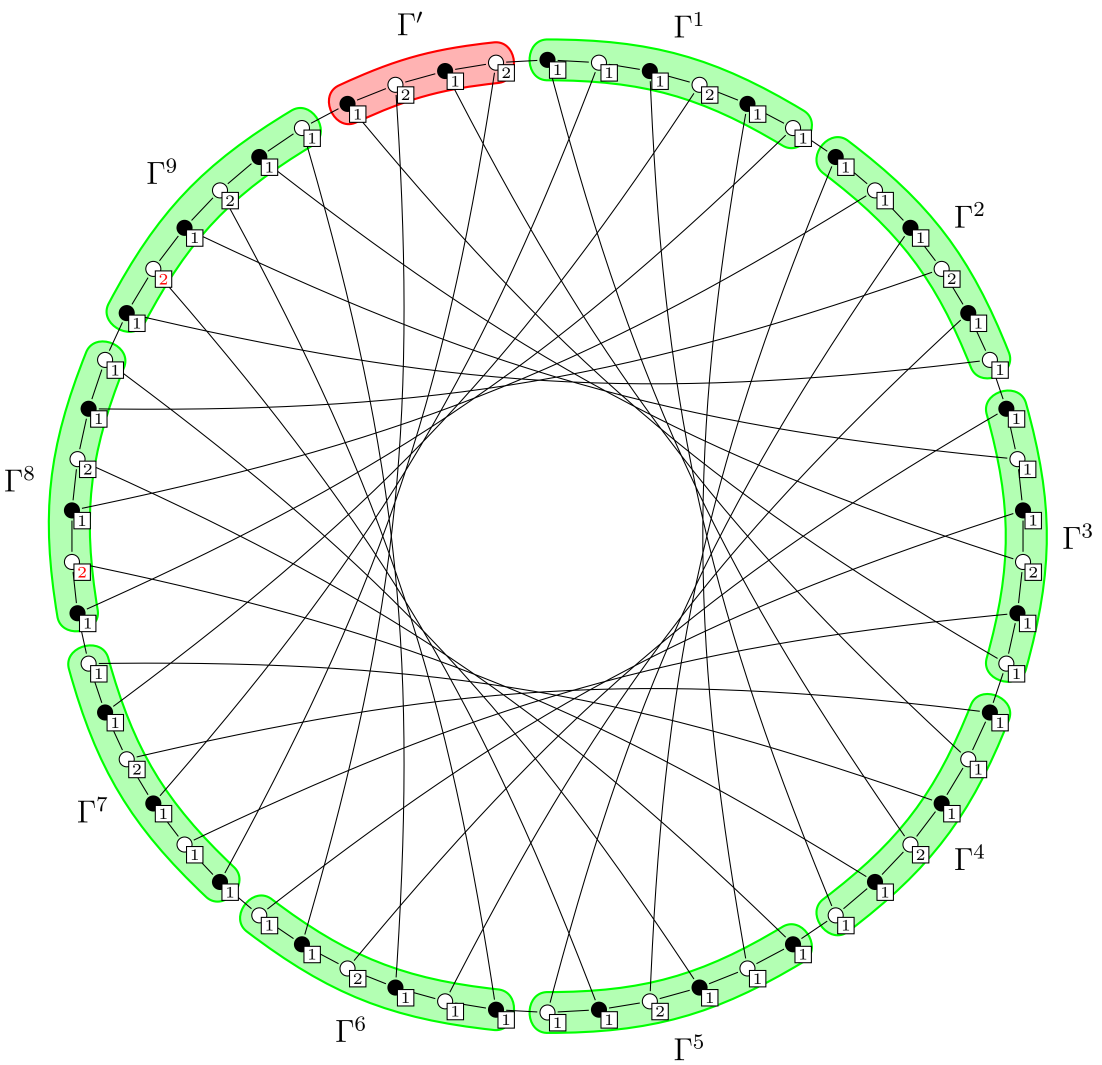



Figure 3.57: The gap-[2]-vertex-labelling of CBH-graph $C_{58,\text{reach}=21}$, as created by Technique $T_6$. Vertices highlighted in red had their labels modified from the original 6-block labelling.

to $v_1^j$ and $v_{n-1}$, whose labels are 1 and 2, respectively. Therefore, $c_\pi(v_0^j) = 1$ also.

Now, consider the $\gamma - 1$ last 6-blocks, namely $\Gamma^j$, $\alpha - \gamma + 2 \leq j \leq \alpha$. Note that vertices $v_0^j$ are adjacent to $v_5^{j-1}$, which receive label 1, and $v_1^j$, whose label is 2. Thus, $c_\pi(v_0^j) = 1$. Vertices $v_4^j$ have their neighbours in the cycle labelled with 2 and 1 which also induces colour 1. Now, both neighbours of $v_2^j$ are labelled with 2. However, the chord which has $v_2^j$ as an end links this vertex to $v_1^{(j+\gamma) \bmod \alpha}$, whose label is 1. Thus, $c_\pi(v_2^j) = 1$.

It remains to consider vertices $v_{n-2}$ and $v_{n-4}$ in the residual block of $G$. The latter is adjacent to $v_{n-3}$, labelled with 2, and to $v_{n-5}$, which belongs to 6-block $\Gamma^\alpha$ and, hence, receives label 1. Thus, $c_\pi(v_{n-4}) = 1$. Finally, since $\Pi_{N(v_{n-2})} = \{1, 2\}$, $c_\pi(v_{n-2}) = 1$, and we conclude that $c_\pi$ is a proper vertex-colouring of $G$.

**Technique $T_7$: Self-sufficient blocks.**

The previous techniques are based on the use of 6-blocks – subgraphs which were obtained from CBH-graph $C_{6,\text{reach}=3}$. In particular, in Technique $T_4$, a gap-[2]-vertex-labelling

$(\pi, c_\pi)$ of $C_{6,\text{reach}=3}$ is used to create the proper labelling of an infinite number of CBH-graphs $C_{n',\text{reach}=r'}$, such that $n' = 6\alpha$ and $r = 6\gamma + 3$. We questioned whether this approach could be used with different values of $n$ and $r$ from other graphs that admit gap-[2]-vertex-labellings.

Consider, for example, CBH-graph $C_{10,\text{reach}=3}$. In this case, $r \equiv 3 \pmod 4$ and Theorem 3.18 establishes that this graph admits a gap-[2]-vertex-labelling. This labelling is presented in Figure 3.58(a). For this graph in particular, chord $v_0v_3$ is responsible for adding a vertex labelled with 2 in $N(v_0)$ so as to induce $c_\pi(v_0) = 1$. All other even-index vertices have their neighbours in the cycle labelled with 1, 2, which also induces colour 1.

Now, consider $C_{30,\text{reach}=13}$. So far, none of our techniques can be applied to create a proper labelling for this graph. However, its vertex set can be partitioned into blocks of size 10 – a *10-block*. Furthermore, every chord has its endpoints in two adjacent blocks. Consider, for example, chord $v_0v_{13}$; note that $r = 3 \pmod{10}$. Then, we use the (known) gap-[2]-vertex-labelling of $C_{10,\text{reach}=3}$ to properly label $C_{30,\text{reach}=13}$, as demonstrated by Figures 3.58(b) and 3.58(c).

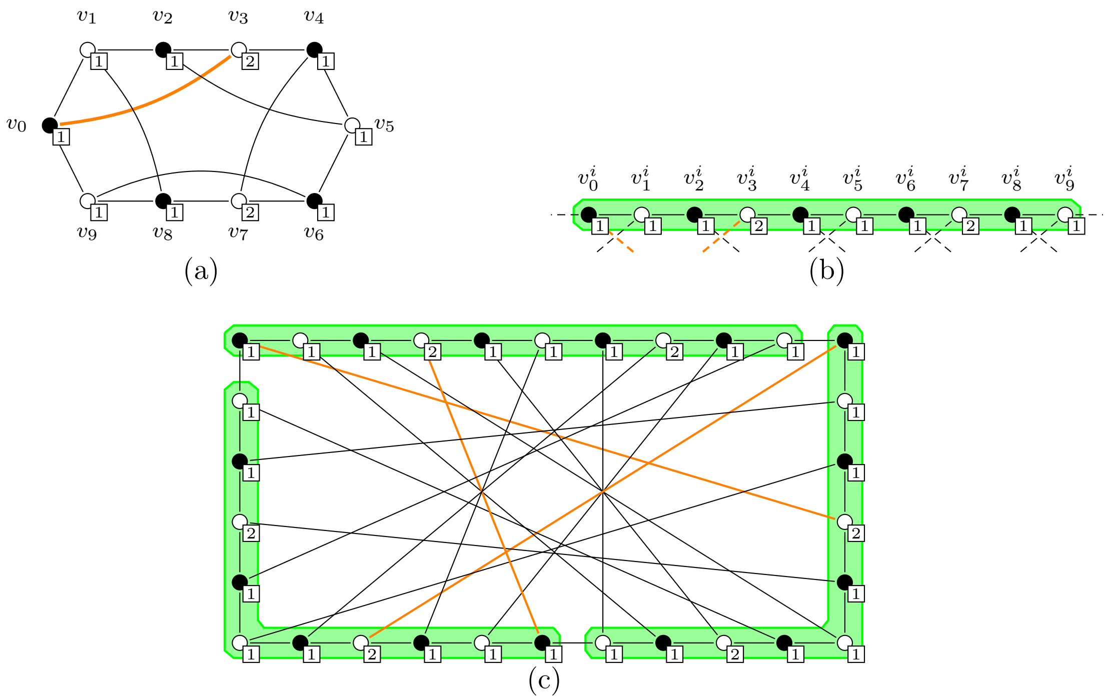



Figure 3.58: CBH-graph $C_{10,\text{reach}=3}$ in (a); the 10-block in (b); and its use to properly label graph $C_{30,\text{reach}=13}$ in (c).

Here, we have applied the same idea of the 6-block: we use a known gap-[2]-vertex-labelling for $C_{10,\text{reach}=3}$ and create a 10-block, thus covering every CBH-graph with $n' \equiv 0 \pmod{10}$ and $r' \equiv 3 \pmod{10}$. Thus, each known gap-[2]-vertex-labelling of CBH-graph $C_{n,\text{reach}=r}$ can be used as a *self-sufficient $n$-block* to properly label a new CBH-graph $C_{n',\text{reach}=r'}$, such that $n' = \alpha n$ and $r' \equiv r \pmod n$.

As another example, recall that Technique $T_1$ provides a gap-[2]-vertex-labelling of $C_{10,\text{reach}=5}$. By the same approach, we can use this labelling and, thus, properly label CBH-graphs $C_{n,\text{reach}=r}$ with $n \equiv 0 \pmod{10}$ and $r \equiv 5 \pmod{10}$. An example is presented in

Figure 3.59: $C_{70,\text{reach}=25}$ has its vertex set partitioned into self-sufficient 10-blocks, labelled according to Figure 3.59(b); each chord now has ends in two blocks at distance $\gamma = 2$ from each other; and the labels at the ends of the chord match the labels from the original graph $C_{10,\text{reach}=5}$, presented in Figure 3.59(a). The chords in Figure 3.59(c) are coloured to indicate the corresponding pairs of vertices, from Figure 3.59(a), in different 10-blocks of $C_{70,\text{reach}=25}$.

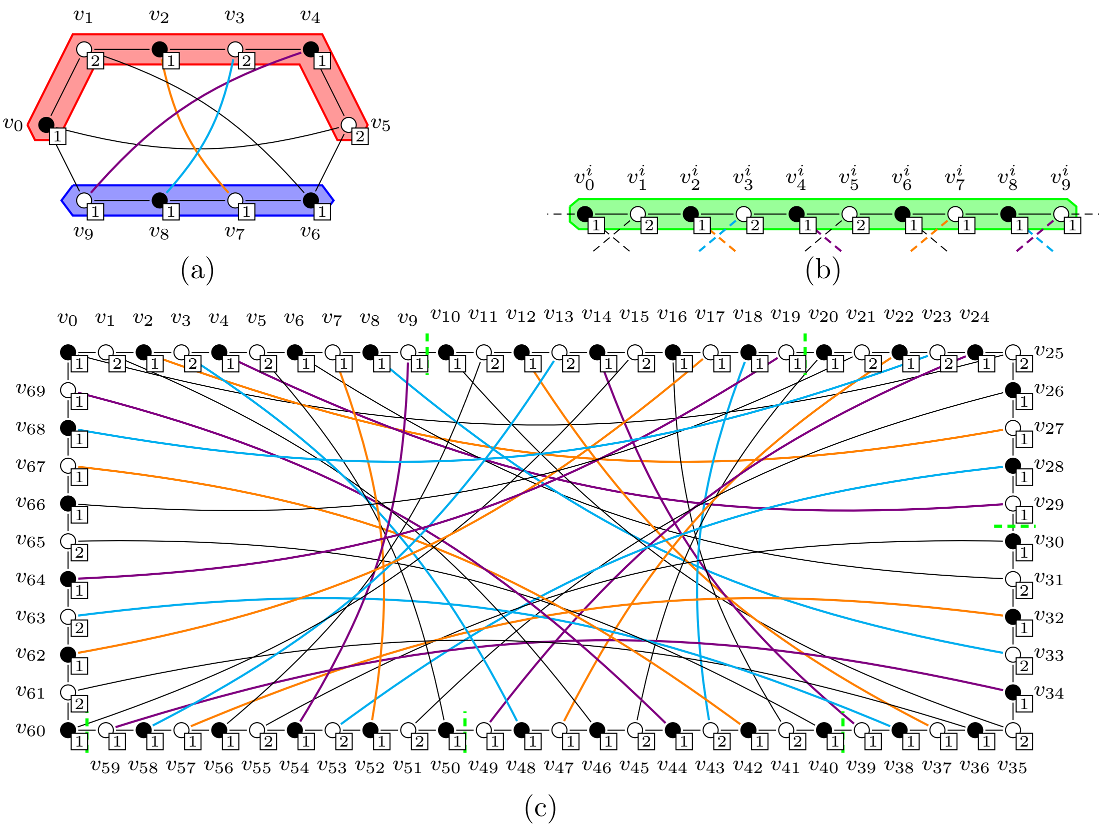



Figure 3.59: In (a), CBH-graph $C_{10,\text{reach}=5}$; in (b), the newly-created self-sufficient 10-block; and in (c); a labelling of $C_{70,\text{reach}=25}$ by Technique $T_7$.

Some examples of CBH-graphs by this technique are presented in Table 3.4.

| $n$-block | $r = 3$ | $r = 5$ | $r = 7$ |
|---|---|---|---|
| $n = 10$ | $C_{10\alpha,\text{reach}=13}$, $C_{10\alpha,\text{reach}=33}$, ... | $C_{10\alpha,\text{reach}=25}$, $C_{10\alpha,\text{reach}=45}$, ... | $C_{10\alpha,\text{reach}=17}$, $C_{10\alpha,\text{reach}=37}$, ... |
| $n = 18$ | $C_{18\alpha,\text{reach}=21}$, $C_{18\alpha,\text{reach}=57}$, ... | $C_{18\alpha,\text{reach}=41}$, $C_{18\alpha,\text{reach}=77}$, ... | $C_{18\alpha,\text{reach}=25}$, $C_{10\alpha,\text{reach}=61}$, ... |
| $n = 22$ | $C_{22\alpha,\text{reach}=25}$, $C_{22\alpha,\text{reach}=69}$, ... | $C_{22\alpha,\text{reach}=49}$, $C_{22\alpha,\text{reach}=93}$, ... | $C_{22\alpha,\text{reach}=29}$, $C_{22\alpha,\text{reach}=73}$, ... |
| $n = 26$ | $C_{26\alpha,\text{reach}=29}$, $C_{26\alpha,\text{reach}=81}$, ... | $C_{26\alpha,\text{reach}=57}$, $C_{26\alpha,\text{reach}=109}$, ... | $C_{26\alpha,\text{reach}=33}$, $C_{26\alpha,\text{reach}=85}$, ... |
| $n = 30$ | $C_{30\alpha,\text{reach}=33}$, $C_{30\alpha,\text{reach}=93}$, ... | $C_{30\alpha,\text{reach}=65}$, $C_{30\alpha,\text{reach}=125}$, ... | $C_{30\alpha,\text{reach}=37}$, $C_{30\alpha,\text{reach}=97}$, ... |
| $n$-block | $r = 9$ | $r = 11$ | $r = 13$ |
| $n = 18$ | $C_{18\alpha,\text{reach}=45}$, $C_{18\alpha,\text{reach}=81}$, ... | $C_{18\alpha,\text{reach}=29}$, $C_{18\alpha,\text{reach}=65}$, ... | $C_{10\alpha,\text{reach}=49}$, $C_{18\alpha,\text{reach}=85}$, ... |
| $n = 22$ | $C_{22\alpha,\text{reach}=53}$, $C_{22\alpha,\text{reach}=97}$, ... | $C_{22\alpha,\text{reach}=33}$, $C_{22\alpha,\text{reach}=77}$, ... | $C_{22\alpha,\text{reach}=57}$, $C_{22\alpha,\text{reach}=101}$, ... |
| $n = 26$ | $C_{26\alpha,\text{reach}=61}$, $C_{26\alpha,\text{reach}=113}$, ... | $C_{26\alpha,\text{reach}=37}$, $C_{26\alpha,\text{reach}=89}$, ... | $C_{26\alpha,\text{reach}=65}$, $C_{26\alpha,\text{reach}=117}$, ... |
| $n = 30$ | $C_{30\alpha,\text{reach}=69}$, $C_{30\alpha,\text{reach}=129}$, ... | $C_{30\alpha,\text{reach}=41}$, $C_{30\alpha,\text{reach}=103}$, ... | $C_{30\alpha,\text{reach}=73}$, $C_{30\alpha,\text{reach}=131}$, ... |

Table 3.4: Examples of CBH-graphs covered by Technique $T_7$,

We conclude this section with one final technique, which uses the concept of isomorphism within the family of CBH-graphs.

**Technique $T_8$: Isomorphism.**

Our notation for CBH-graphs states that every vertex $v_i$ is named according to its order in a fixed hamiltonian cycle $C = (v_0, v_1, v_2, \ldots, v_{n-2}, v_{n-1}, v_0)$. Consequently, every chord $e$ of the graph can be written as $e = v_i v_{(i+r) \bmod n}$. However, some CBH-graphs have more than one hamiltonian cycle as a subgraph. For instance, consider $G = C_{10,\text{reach}=3}$, which is illustrated in Figure 3.60(a), and recall that $G$ has $r(e) = 3$ for every chord $e \in E(G)$. Let us consider a different hamiltonian cycle of $G$ defined by $C^+ = (v_0, v_3, v_4, v_7, v_8, v_1, v_2, v_5, v_6, v_9, v_0)$. This cycle is illustrated in Figure 3.60(b). In the image, the orange oriented edges represent the order in which vertices appear in $C^+$. We say, in this case, that $C^+$ *covers* these chords and edges in $G$.

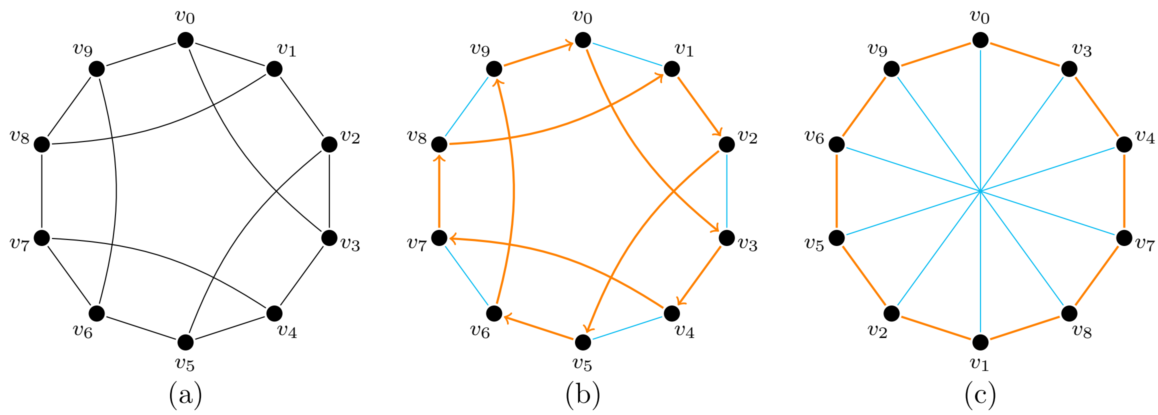



Figure 3.60: In (a), graph $G \cong C_{10,\text{reach}=3}$; in (b), cycle $C^+$ highlighted in orange and chords $e'$, in blue; lastly, in (c), graph $G' \cong C_{10,\text{reach}=5}$ obtained from $C^+$.

The "+" symbol in the superscript of $C^+$ is used to indicate that we are covering cycle-edges of $G$ in a "forward" manner. In order to clarify this statement, observe the first chord in $C^+$, i.e. $v_0v_3$. The next edge of $G$ covered by $C^+$ is $v_3v_4$. Thus, we move "forward" in the indices of the vertices. Next, $C^+$ covers chord $v_4v_7$ and, once again, moves forward by covering edge $v_7v_8$. By continuously covering (cyclically) these *forward cycle-edges* in $C_{10,\text{reach}=3}$, we obtain a new CBH-graph $G'$, which is illustrated in Figure 3.60(c). In this new graph, edges $v_0v_1, v_2v_3, v_4v_5, v_6v_7$ and $v_8v_9$ are the chords, each of which has reach five. Therefore, $G' \cong G$ is also isomorphic to $C_{10,\text{reach}=5}$.

Alternately, it is also possible to obtain a different hamiltonian cycle by following cycle-edges in a "backwards" manner. For example, consider cycle $C^- = (v_0, v_3, v_2, v_5, v_4, v_7, v_6, v_9, v_8, v_1, v_0)$. In this case, the chords of this new graph are $v_1v_2, v_3v_4, v_5v_6, v_7v_8$ and $v_9v_0$. By inspecting graph $G''$ obtained from $C^-$, it is possible to conclude that $G'' \cong C_{10,\text{reach}=3} = G$.

Formally, for a CBH-graph $G$, we define $C^+ \subset G$ as the cycle obtained by following chords and forward cycle-edges in sequential, alternating order, starting at $v_0$. The *step* of $C^+$ is defined as $s_+ = r + 1$. If $V(C^+) = V(G)$, we say that $C^+$ is a *spanning subgraph* of $G$. Figure 3.61 sketches the construction of cycle $C^+$ and illustrates step $s_+$.

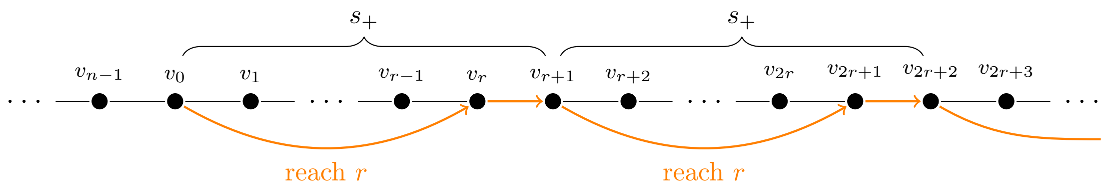



Figure 3.61: A sketch of cycle $C^+$ of a CBH-graph $G$. The orange oriented edges represent the chords and edges of $G$ covered by $C^+$.

Analogously, we define $C^- \subset G$ as the cycle obtained by following chords and backwards cycle-edges in the same, sequential manner; also, we define step $s_- = r - 1$. The construction of cycle $C^-$ and step $s_-$ are illustrated in Figure 3.62. Herein, we continue our discussion considering only cycle $C^+$. We remark, however, that analogous definitions, reasonings and results may also be obtained for $C^-$.

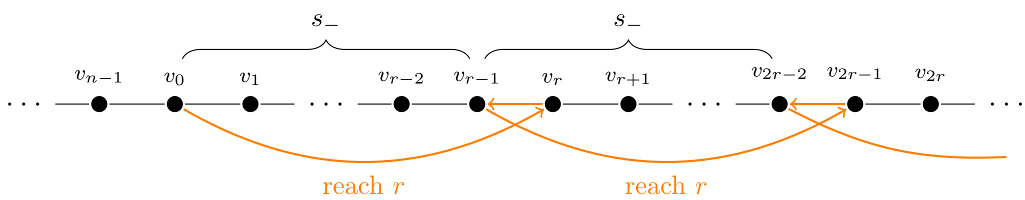



Figure 3.62: A sketch of the construction of cycle $C^-$.

In a CBH-graph $G$, each pair of vertices $v_{j-1}, v_j$, with $j$ even, is referred to as a *block* of $G$. If edge $v_{j-1}v_j$ is covered by $C^+$, we say that the block containing vertices $v_{j-1}, v_j$ is also covered. Now, if $C^+$ is a spanning subgraph of $G$, then exactly $\frac{n}{2}$ blocks are covered by the cycle. Since each step taken in $C^+$ covers exactly one block in $G$, it follows that we require the same number $\frac{n}{2}$ of steps to be taken. In particular, note that after the $\lceil \frac{n}{s_+} \rceil$-th step, cycle $C^+$ "passes over" the starting vertex $v_0$ for the first time. This only occurs when $n$ is not a multiple of $s_+$. In fact, if $n = l \cdot s_+$, for some $l \in \mathbb{N}$, we immediately conclude that cycle $C^+$ is not a spanning subgraph of $G$. This observation is made quite clear when inspecting Figure 3.63, which depicts cycle $C^+$ in CBH-graph $C_{18,\text{reach}=5}$. However, there are cases where $n \neq l \cdot s_+$ and $C^+$ is not a spanning subgraph of $C_{n,\text{reach}=r}$, as exemplified by $C_{42,\text{reach}=17}$.

Consider cycle $C^+$ in a CBH-graph of order $n \neq l \cdot s_+$. After the first step, edges $v_0v_r$ and $v_rv_{r+1}$ are covered. Note that there are $\frac{r-1}{2}$ uncovered blocks between $v_0$ and $v_r$, as illustrated in Figure 3.64(a). In turn, after the $\lceil \frac{n}{s_+} \rceil$-th step, a block containing two vertices $v_{j-1}, v_j$, $2 \leq j \leq r-1$ and even, is now covered by $C^+$. We remark that $v_j$ is the first even-index vertex covered after cycle $C^+$ "passes over" the initial vertex $v_0$. This case is illustrated in Figure 3.64(b).

Thus, we define the *pass* of $C^+$ as $p_+ = (\lceil \frac{n}{s_+} \rceil \cdot s_+) \bmod n$. Observe that $p_+$ is the index of vertex $v_j$ mentioned in the previous paragraph. If $\frac{p_+}{2}$ and $\frac{s_+}{2}$ are relatively prime[6], we are able to conclude that cycle $C^+$ is a spanning subgraph of $G$. Moreover, we

[6]We divide pass $p_+$ and step $s_+$ by two so as to consider only the blocks, rather than the vertices themselves.

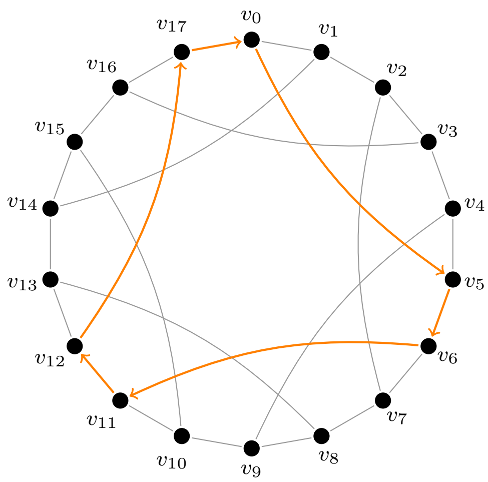
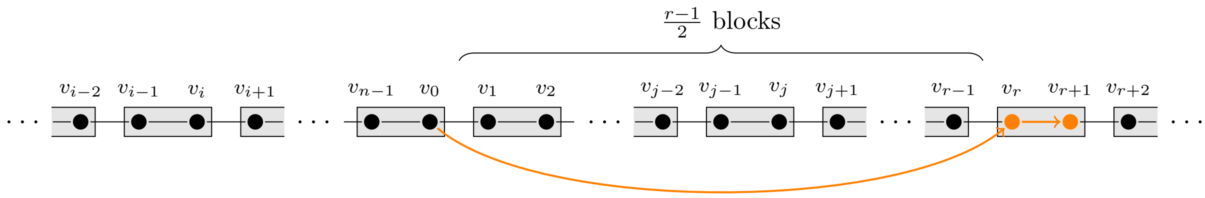
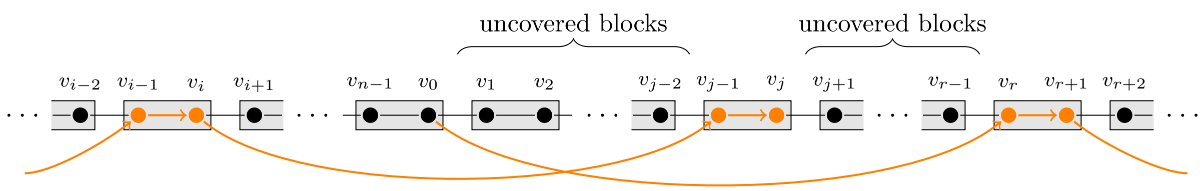



Figure 3.63: In orange, cycle $C^+$ in graph $G \cong C_{18,\text{reach}=5}$, with $V(C^+) = \{v_0, v_5, v_6, v_{11}, v_{12}, v_{17}\}$.




(a) 1st step in $C^+$.




(b) $\lceil \frac{n}{s_+} \rceil$-th step.

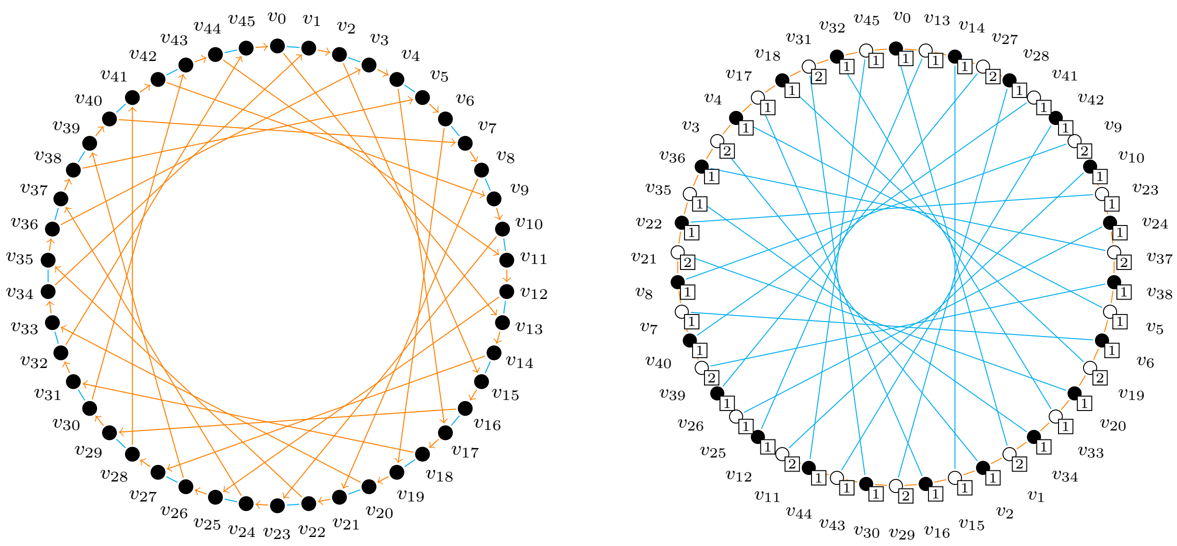
Figure 3.64: The blocks (in gray) between $v_0$ and $v_{s_+}$, and vertices $v_{j-1}$ and $v_j$ covered after the $\lceil \frac{n}{s_+} \rceil$-th step. The orange vertices indicate when a block is covered by $C^+$.

observed that when this is not the case, cycle $C^+$ does not cover every block of $G$ and, consequently, $C^+$ is not a spanning subgraph of $G$.

We investigated this (apparent) equivalence with the aid of a computer program that analyses cycle $C^+$ in CBH-graphs and checks: (i) whether the cycle is a spanning subgraph of $G$; and (ii) if $\frac{p_+}{2}$ and $\frac{s_+}{2}$ are relatively prime. For every CBH-graph of order $n \leq 1002$, our algorithm indicated that $C^+$ is a spanning subgraph of $G$ if and only if $\gcd(s_+, p_+) = 2$. With these preliminary observations, we state the following conjecture.

**Conjecture 3.21.** *Let $G$ be a CBH-graph and $C^+ \subset G$. Cycle $C^+$ is a spanning subgraph of $G$ if and only if $\frac{p_+}{2}$ and $\frac{s_+}{2}$ are relatively prime.*

We are now ready to present Technique $T_8$, which we exemplify by analysing CBH-graph $G \cong C_{46,\text{reach}=13}$. So far, this graph is not covered by any of the previous techniques. Now, consider $C^+$ in $G$, as illustrated in Figure 3.65(a). In this case, step $s_+ = 14$ and pass $p_+ = 10$. Note that, in this case, $C^+$ is a spanning subgraph and $\gcd(14, 10) = 2$, verifying Conjecture 3.21. Furthermore, $C^+$ spans the CBH-graph $C_{46,\text{reach}=19}$, in which chords have reach $r = 19 \equiv 3 \pmod 4$. By Theorem 3.18, this graph admits a gap-[2]-vertex-labelling. The new graph and its proper labelling are illustrated in Figure 3.65(b). Then, since $G' \cong G$, we conclude that $G$ also admits a gap-[2]-vertex-labelling.




(a) (b)

Figure 3.65: CBH-graphs: (a) $C_{46,\text{reach}=13}$; and (b) $C_{46,\text{reach}=19}$.

For CBH-graphs $G$, Technique $T_8$ consists of using the hamiltonian cycle $C^+ \subset G$, when it is a spanning subgraph, to create a new CBH-graph $G'$. Then, we verify if $G'$ can be properly labelled by any of the other labelling techniques. This allows us to cover a large variety of CBH-graphs which had not yet been addressed in our work. Some examples of graphs for which techniques $T_1$ through $T_7$ are not applicable, but admit gap-[2]-vertex-labellings by Technique $T_8$, are presented in Table 3.5.

| $G$ | $C^+$ | $C^-$ | Covered by |
|---|---|---|---|
| $C_{34,\text{reach}=13}$ | - | $C_{34,\text{reach}=7}$ | Theorem 3.18 |
| $C_{38,\text{reach}=13}$ | $C_{38,\text{reach}=17}$ | $C_{38,\text{reach}=5}$ | Technique $T_2$ |
| $C_{46,\text{reach}=13}$ | $C_{46,\text{reach}=19}$ | - | Theorem 3.18 |
| $C_{46,\text{reach}=17}$ | $C_{46,\text{reach}=11}$ | $C_{46,\text{reach}=7}$ | Theorem 3.18 |
| $C_{54,\text{reach}=17}$ | - | $C_{54,\text{reach}=19}$ | Theorem 3.18 |
| $C_{58,\text{reach}=13}$ | - | $C_{58,\text{reach}=11}$ | Theorem 3.18 |
| $C_{58,\text{reach}=17}$ | $C_{58,\text{reach}=25}$ | $C_{58,\text{reach}=23}$ | Theorem 3.18 |
| $C_{62,\text{reach}=13}$ | $C_{62,\text{reach}=17}$ | $C_{62,\text{reach}=9}$ | Technique $T_5$ |
| $C_{62,\text{reach}=25}$ | $C_{62,\text{reach}=23}$ | $C_{62,\text{reach}=27}$ | Theorem 3.18 |
| $C_{62,\text{reach}=29}$ | $C_{62,\text{reach}=5}$ | $C_{62,\text{reach}=21}$ | Technique $T_2$ |
| $C_{66,\text{reach}=13}$ | $C_{66,\text{reach}=29}$ | - | Technique $T_7$ |

Table 3.5: Some CBH-graphs covered by Technique $T_8$, considering cycles $C^+$ and $C^-$.

To conclude this section, we present in Table 3.6 values for $n$ and $r$ which are covered by one of Techniques $T_1$ to $T_8$. In the table, the orange cell refers to the Heawood Graph, for which Lemma 3.19 states that there is no gap-[2]-vertex-labelling. On the other hand, the blue cells are CBH-graphs we know (empirically) that admit gap-[2]-vertex-labelling. However, they are not covered by any of the techniques. Lastly, gray cells represent invalid combinations for $n$ and $r$.

Although much works still needs to be done to prove that every CBH-graph, up to the Heawood Graph, admits a gap-[2]-vertex-labelling, our research lead us to pose the following conjecture

**Conjecture 3.22.** *Let $G$ be a CBH-graph not isomorphic to $C_{14,reach=5}$. Then, $\chi_V^g(G) = 2$.*

To further strengthen our conjecture, we devised an Integer Linear Programming formulation to find a gap-[2]-vertex-labelling of every CBH-graph of order $n \equiv 2 \pmod 4$ with homogeneous chords.

**Integer Linear Programming**

For each vertex $v$ of a CBH-graph, we create two variables, $l_v$ and $c_v$, that correspond to a label and colour to be assigned to $v$, respectively. Our ILP formulation is presented below.

$$\text{minimize} \sum_{v \in V(G)} 0 \cdot c_v \tag{3.1}$$

subject to:

$$c_u + c_v = 1, \qquad \forall uv \in E(G) \tag{3.2}$$

$$c_u \leq \sum_{v \in N(u)} (l_v - 1), \qquad \forall u \in V(G) \tag{3.3}$$

$$c_u \leq d(u) - \sum_{v \in N(u)} (l_v - 1), \qquad \forall u \in V(G) \tag{3.4}$$

$$c_u \geq (l_v - 1) - (l_w - 1), \qquad \forall u \in V(G), \forall v, w \in N(u), v \neq w \tag{3.5}$$

$$c_u \geq (l_w - 1) - (l_v - 1), \qquad \forall u \in V(G), \forall v, w \in N(u), v \neq w \tag{3.6}$$

$$l_v \in \{1, 2\} \tag{3.7}$$

$$c_v \in \{0, 1\} \tag{3.8}$$

Since we are interested in a labelling with $k = 2$ and there are no vertices of degree one in $G$, restrictions (3.7) and (3.8) follow naturally. Restriction (3.2) establishes that no two variables for adjacent vertices $u, v \in V(G)$ can be assigned colour 1. The upper bound provided in restrictions (3.3) and (3.4) imply that the colour of a vertex is bound both by the labels assigned to its neighbours and to its degree. Finally, the lower bounds in restrictions (3.5) and (3.6) are used to determine the induced colour of every vertex $v \in V(G)$.

We executed this program on all CBH-graphs with homogeneous chords for $n \leq 1002$. With the exception of $C_{14,\text{reach}=5}$, in all cases, the program found a gap-[2]-vertex-labelling. We remark that many of the labelling techniques presented in the previous section were obtained from the results provided by this program.

| r \ n | 10 | 14 | 18 | 22 | 26 | 30 | 34 | 38 | 42 | 46 | 50 | 54 | 58 | 62 | 66 | 70 | 74 | 78 | 82 | 86 | 90 | 94 | 98 | 102 | 106 | 110 | 114 | 118 | 122 | 126 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | $T_1$ | | $T_3$ | $T_2$ | $T_3$ | $T_1$ | $T_3$ | $T_2$ | $T_3$ | $T_2$ | $T_1$ | $T_2$ | $T_3$ | $T_2$ | $T_3$ | $T_1$ | $T_3$ | $T_2$ | $T_3$ | $T_2$ | $T_1$ | $T_2$ | $T_3$ | $T_2$ | $T_3$ | $T_1$ | $T_3$ | $T_2$ | $T_3$ | $T_2$ |
| 9 | | | $T_1$ | $T_6$ | $T_5$ | $T_4$ | $T_2$ | $T_3$ | $T_4$ | $T_6$ | $T_2$ | $T_1$ | $T_6$ | $T_5$ | $T_2$ | $T_3$ | $T_5$ | $T_4$ | $T_2$ | $T_5$ | $T_1$ | $T_6$ | $T_2$ | $T_3$ | $T_6$ | $T_5$ | $T_2$ | $T_3$ | $T_5$ | $T_1$ |
| 13 | | | | | $T_1$ | $T_7$ | $T_8$ | $T_8$ | | $T_8$ | $T_2$ | $T_3$ | $T_8$ | $T_8$ | $T_8$ | $T_7$ | $T_2$ | $T_1$ | $T_8$ | | $T_7$ | | $T_2$ | $T_3$ | $T_8$ | $T_7$ | | $T_8$ | $T_2$ | $T_3$ |
| 17 | | | | | | | $T_1$ | $T_8$ | | $T_8$ | $T_7$ | $T_8$ | $T_8$ | $T_8$ | $T_2$ | $T_3$ | $T_8$ | $T_8$ | $T_8$ | $T_8$ | $T_7$ | | $T_2$ | $T_1$ | $T_8$ | $T_7$ | | $T_8$ | $T_8$ | |
| 21 | | | | | | | | | $T_1$ | $T_6$ | $T_5$ | $T_4$ | $T_6$ | $T_5$ | $T_4$ | $T_6$ | $T_5$ | $T_4$ | $T_2$ | $T_3$ | $T_4$ | $T_6$ | $T_5$ | $T_4$ | $T_6$ | $T_5$ | $T_4$ | $T_6$ | $T_2$ | $T_3$ |
| 25 | | | | | | | | | | | $T_1$ | $T_7$ | $T_8$ | $T_8$ | $T_7$ | $T_7$ | $T_8$ | | $T_8$ | $T_8$ | $T_7$ | $T_8$ | $T_2$ | $T_3$ | $T_8$ | $T_7$ | $T_8$ | $T_8$ | $T_8$ | $T_7$ |
| 29 | | | | | | | | | | | | | $T_1$ | $T_8$ | $T_7$ | | $T_8$ | $T_7$ | $T_8$ | $T_8$ | $T_7$ | $T_8$ | $T_8$ | $T_8$ | $T_7$ | $T_8$ | $T_2$ | $T_3$ | $T_8$ | $T_7$ |
| 33 | | | | | | | | | | | | | | | $T_1$ | $T_6$ | $T_5$ | $T_4$ | $T_6$ | $T_5$ | $T_4$ | $T_6$ | $T_5$ | $T_4$ | $T_6$ | $T_5$ | $T_4$ | $T_6$ | $T_5$ | $T_4$ |
| 37 | | | | | | | | | | | | | | | | | $T_1$ | $T_7$ | $T_8$ | $T_8$ | $T_7$ | $T_8$ | | $T_8$ | $T_8$ | $T_7$ | | $T_8$ | $T_8$ | $T_8$ |
| 41 | | | | | | | | | | | | | | | | | | | $T_1$ | $T_8$ | $T_7$ | | $T_8$ | $T_8$ | $T_8$ | | | $T_8$ | $T_8$ | $T_7$ |
| 45 | | | | | | | | | | | | | | | | | | | | | $T_1$ | $T_6$ | $T_5$ | $T_4$ | $T_6$ | $T_5$ | $T_4$ | $T_6$ | $T_5$ | $T_4$ |
| 49 | | | | | | | | | | | | | | | | | | | | | | | $T_1$ | $T_8$ | $T_8$ | $T_7$ | $T_8$ | $T_8$ | $T_8$ | $T_7$ |
| 53 | | | | | | | | | | | | | | | | | | | | | | | | | $T_1$ | $T_7$ | $T_8$ | $T_8$ | $T_8$ | $T_8$ |
| 57 | | | | | | | | | | | | | | | | | | | | | | | | | | | $T_1$ | $T_6$ | $T_5$ | $T_4$ |
| 61 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | $T_1$ | $T_7$ |

Table 3.6: Values of $n$ and $r$ for which CBH-graphs $C_{n,\text{reach}=r}$ admit gap-[2]-vertex-labellings.

## *List of Techniques*

- *Technique $T_1$: $n = 2\alpha r$;*
- *Techniques $T_2$ and $T_3$: $n = \beta(r+1) + \alpha(r-1)$, $\alpha, \beta$ odd;*
- *Technique $T_4$: 6-block on $n \equiv 0 \pmod 6$;*
- *Technique $T_5$: 6-block on $n \equiv 2 \pmod 6$;*
- *Technique $T_6$: 6-block on $n \equiv 4 \pmod 6$;*
- *Technique $T_7$: $n'$-block.*
- *Technique $T_8$: Isomorphism.*

**Theorem 3.23.** *With the exception of $C_{14,reach=5}$, every CBH-graph of order $n \leq 1002$ admits a gap-$[2]$-vertex-labelling.* □

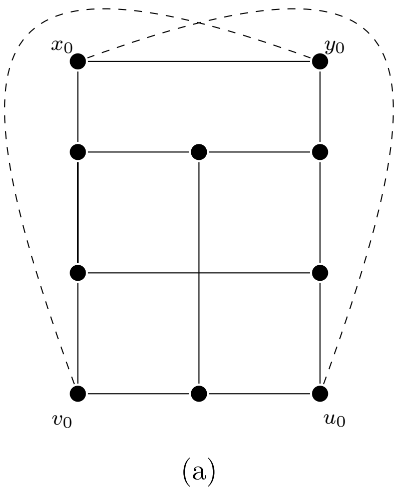
### 3.3.6 Snarks

In the previous section, we approached cubic bipartite graphs motivated by the question raised by Dehghan et al. [8]. Still motivated by the role of cubic graphs, in this section, we investigate the vertex-gap number of snarks (which play an important role in Graph Theory, especially in the field of Graph Colourings).

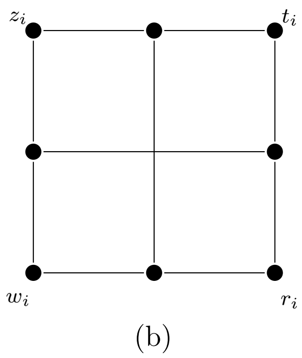
Recall the definition of snarks presented in Chapter 2. A *snark* is a bridgeless, cubic graph with chromatic index four, without cycles of length two or three. We establish the vertex-gap number for the families of Blanuša, Flower, Goldberg and Twisted Goldberg snarks in the following subsections. Although the labelling presented for each of these families is distinct, we use a similar technique to the one used for the edge-version, presented in Section 2.2.5. This technique consists of assigning labels to the "building blocks" of each snark that, together, induce a proper colouring of the graph.

#### Blanuša snarks

The family of Generalised First Blanuša Snarks is defined in Section 2.2.5. Recall that Generalised First Blanuša Snark $B_i^1$ uses a copy of graph $B_0^1$, depicted in Figure 3.66(a), and $i$ copies of block $B$, illustrated in Figure 3.66(b). A sketch of $B_i^1$ is shown in Figure 3.67. The vertex-gap number for this first family is stated in Theorem 3.24.




(a)




(b)

Figure 3.66: In (a), the first block $B_0^1$ used in the construction of Blanuša snark $B_i^1$; and in (b), the iterating blocks $B_i$.

**Theorem 3.24.** *Let $G$ be a Generalised First Blanuša Snark. Then, $\chi_V^g(G) = 3$.*

*Proof.* Let $G$ be the Generalised First Blanuša Snark $B_i^1$, $i \geq 3$ and odd. Recall that $\chi(G) = 3$. Then, in order to prove the result, by Corollary 3.8, it suffices to exhibit a gap-[3]-vertex-labelling of $G$. First, we define a labelling $\pi$ of block $B_0^1$ and of blocks $B_j$, which are presented in Figures 3.68(a) and 3.68(b), respectively. Define colouring $c_\pi$ as usual.

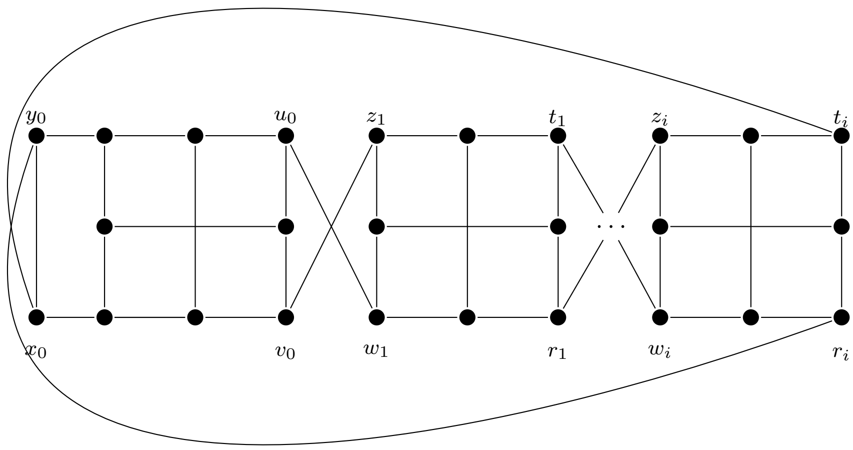


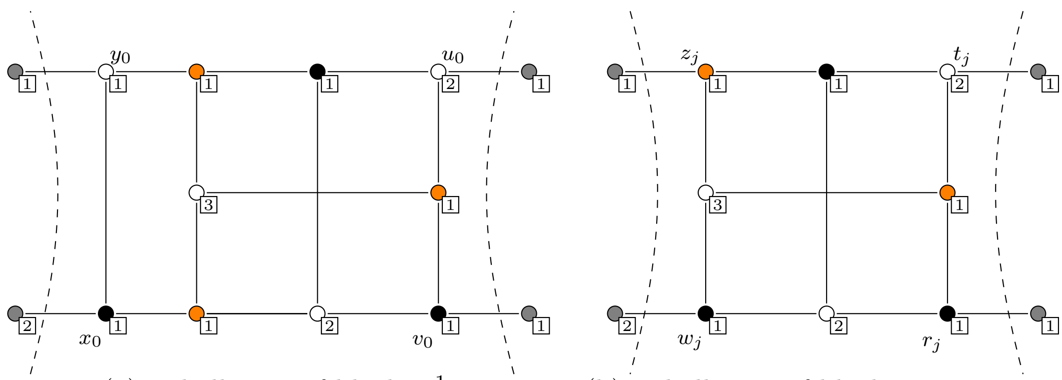

Figure 3.67: The construction of Generalised First Blanuša Snark $B^1_i$, $i \geq 1$.

Consider block $B^1_0$. By inspecting the unnamed vertices in Figure 3.68(a), we conclude that their entire neighbourhood is contained in $V(B^1_0)$, that is, they are not adjacent to any vertex in blocks $B_j$. Furthermore, observe that no two adjacent unnamed vertices have the same induced colour. Now, we analyse the remaining vertices, namely $x_0$, $y_0$, $v_0$ and $u_0$.

By construction of $B^1_i$, vertices $u_0$, $v_0$, $x_0$ and $y_0$ are adjacent to $w_1$, $z_1$, $t_i$ and $r_i$, respectively. By inspecting Figure 3.68(b), note that these last vertices receive the same labels as the gray vertices adjacent to $u_0$, $v_0$, $x_0$ and $y_0$ in Figure 3.68(a), respectively. Furthermore, by inspecting 3.68(a) considering the labels of gray vertices, we conclude that the named vertices have induced colours different from their neighbours. Therefore, $c_\pi$ is a proper colouring of block $B^1_0$. Next, we consider blocks $B_j$, $1 \leq j \leq i$.

An analogous reasoning can be applied for the unnamed vertices in Figure 3.68(b), and we conclude that no two unnamed adjacent vertices in $B_j$ have the same induced colour. It remains to consider vertices $z_j$, $t_j$, $w_j$ and $r_j$.

By construction, $z_j$ is adjacent to $v_0$, if $j = 1$, and $r_{j-1}$, otherwise. In both cases,




(a) Labelling $\pi$ of block $B^1_0$. (b) Labelling $\pi$ of blocks $B_j$, $1 \leq j \leq i$.

Figure 3.68: In (a), the first block $B^1_0$ used in the construction of Blanuša snark $B^1_i$; and in (b), the iterating blocks $B_i$. White, black and orange vertices have induced colours 0, 1 and 2, respectively. Gray vertices belong to adjacent blocks, which are delimited by the dashed lines.

these vertices are assigned label 1, represented by the gray vertex adjacent to $z_j$ in Figure 3.68(b). A similar reasoning can be applied to vertex $w_j$: it is adjacent to $u_0$, if $j = 0$ and $t_{j-1}$, otherwise; both vertices receive label 2. Analogously, considering vertices $t_j$ and $r_j$, we conclude that these vertices are always adjacent to a vertex which receives label 1. By inspection, we conclude that $c_\pi$ is a proper colouring of each block $B_j$. Furthermore, since the bottommost connecting vertices of each block, which have induced colour 1, always connect with the topmost vertices of its adjacent blocks, with colours 0 or 2, there are no adjacent vertices in neighbouring blocks with the same induced colour. Figure 3.69 illustrates this labelling and colouring for $B_3^1$. We conclude that $(\pi, c_\pi)$ is a gap-[3]-vertex-labelling of $G$, and the result follows. □

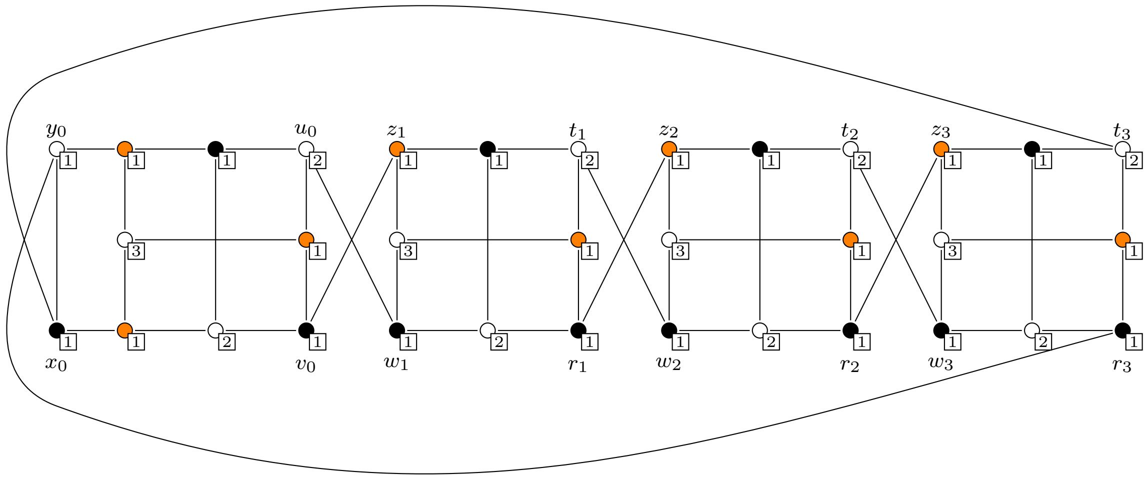



Figure 3.69: The gap-[3]-vertex-labelling $(\pi, c_\pi)$ of graph $B_3^1$.

The family of Generalised Second Blanuša Snarks is also defined in Section 2.2.5. Due to time constraints, we did not extend the results for the Generalised Second Blanuša Snarks. We believe, however, that minor adjustments to the labelling can be done in order to establish the vertex-gap number for this family. Therefore, it is presented here as a problem for future work.

**Problem 3.25.** *Determine the vertex-gap number for the family of Generalised Second Blanuša snarks.*

### Flower snarks

In Section 2.2.5, we describe the construction of Flower Snark $J_l$, $l \geq 3$ and odd. Figure 3.70 illustrates snark $J_l$ with its notation. The vertex-gap number for this family of graphs is presented in Theorem 3.26.

**Theorem 3.26.** *Let $G$ be a Flower Snark. Then,* $\chi_V^g(G) = \chi(G) = 3$.

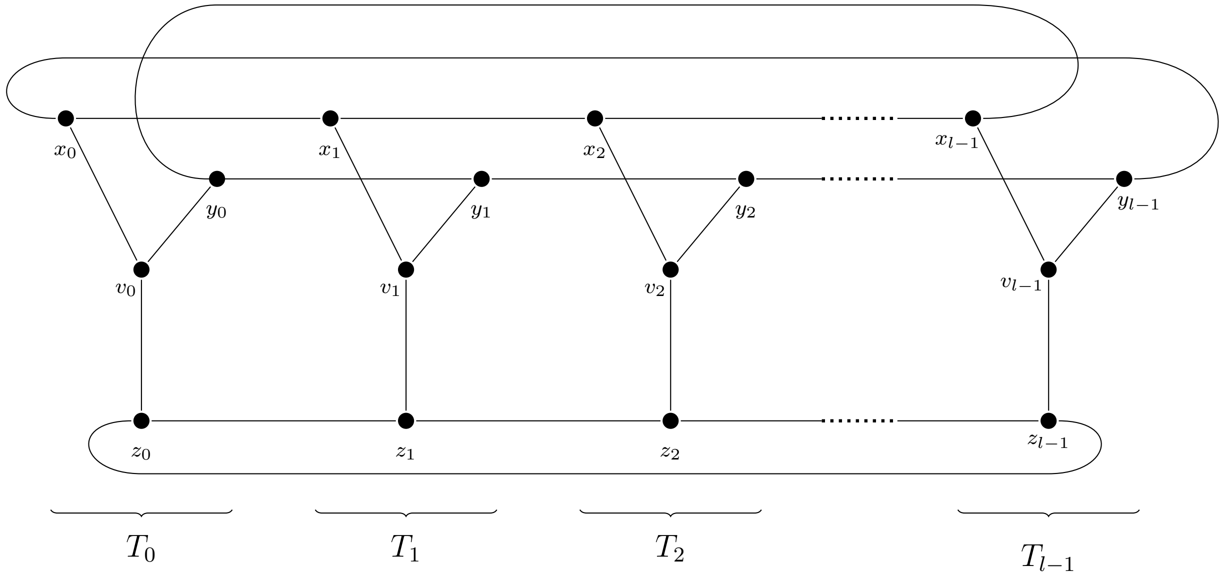



Figure 3.70: Flower Snark $J_l$.

*Proof.* Let $G$ be a Flower Snark constructed from $l$ copies of star $T_i \cong S_3$. Recall that $V(T_i) = \{x_i, y_i, z_i, v_i\}$, where $v_i$ is the central vertex, as defined in Section 2.2.5. Also, in the construction of $J_l$, every $T_i$ is connected to $T_{i-1}$ and $T_{i+1}$ through vertices $x_i$, $y_i$ and $z_i$ as indicated in Figure 3.70. Since $\chi(G) = 3$, in order to prove the result, we show a gap-[3]-vertex-labelling $(\pi, c_\pi)$ of $G$.

Let us define three labellings $\pi$ of $T_i$, for $0 \leq i < l$, for the following cases: (i) $T_i$, $i \leq l-3$ and even; (ii) $T_i$, $i \leq l-2$ and odd; and, finally, for (iii) $T_{l-1}$. For cases (i) and (ii), assign label 1 to every vertex $x_i$, $y_i$ and $z_i$. For vertices $v_i$, let

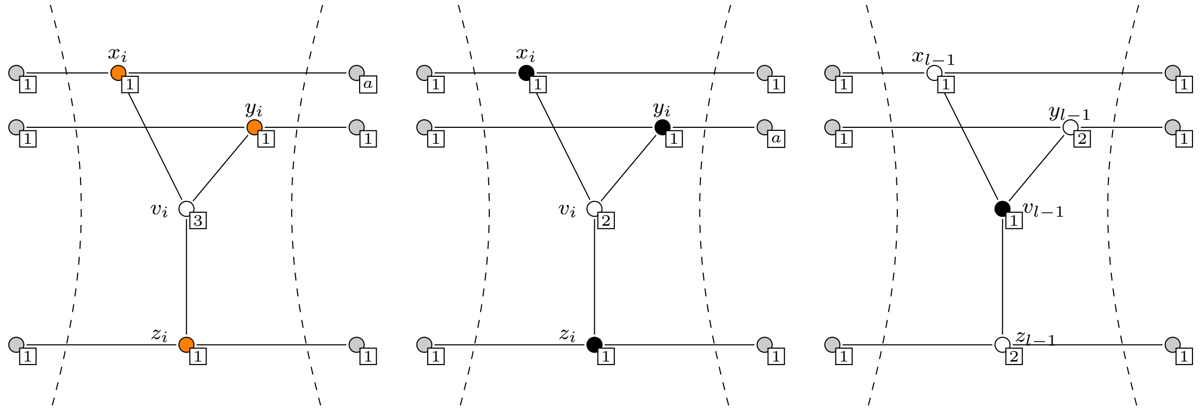
$$\pi(v_i) = \begin{cases} 3, & \text{if } i \text{ even; and} \\ 2, & \text{otherwise.} \end{cases}$$

For $T_{l-1}$, assign $\pi(x_{l-1}) = \pi(v_{l-1}) = 1$ and $\pi(y_{l-1}) = \pi(z_{l-1}) = 2$. The labellings in these three cases are exhibited in Figure 3.71. Colouring $c_\pi$ is defined as usual.

In the construction of a Flower snark $J_l$, every $T_i$ is connected with $T_{i-1}$ and $T_{i+1}$ through vertices $x_i$, $y_i$ and $z_i$. These connections are represented in Figure 3.71 by the adjacent vertices, in gray. Note that the labelling of $G$ is such that the labellings in Figures 3.71(a) and 3.71(b) alternate following the order of $T_i$, $i < l-1$. Also, observe that, with the exception of $y_{l-1}$ and $z_{l-1}$, every vertex $x_i$, $y_i$ and $z_i$ has been assigned label 1. As an example, Figure 3.72 illustrates $(\pi, c_\pi)$ for Flower snark $J_7$.

In order to obtain that $c_\pi$ is a proper vertex-colouring of $G$, first, observe that $\Pi_{N(v_{l-1})} = \{1, 2\}$ and, therefore, $c_\pi(v_{l-1}) = 1$. Furthermore, since $T_{l-1}$ is connected to $T_{l-2}$ and $T_0$, for every $w \in \{x_{l-1}, y_{l-1}, z_{l-1}\}$, we know that $\Pi_{N(w)} = \{1\}$, inducing colour 0 in these three vertices. In Figure 3.72, the colouring for this case is displayed in $T_6$.

Next, we analyse the induced colours of vertices in every $T_i$, $0 \leq i \leq l-2$. Note that




(a) $i$ even. (b) $i$ odd. (c) $i = l-1$.

Figure 3.71: The gap-[3]-vertex-labelling of graphs $T_i$ as described in the text. The vertices connected to $x_i, y_i$ and $z_i$ in each image represent their adjacencies in $J_l$.

every $x_i$, $y_i$ and $z_i$ has been labelled with 1, which implies $c_\pi(v_i) = 0$ for every $v_i$. Now, let $w_i$ be any of vertices $x_i$, $y_i$ and $z_i$ of even index. Observe that $v_i \in N(w_i)$ in this case is labelled with 3. Furthermore, all these vertices have a vertex $u$ in their neighbourhood such that $\pi(u) = 1$: vertices $w_i$ with $1 \leq i \leq l-3$ have both neighbours $w_{i-1}$ and $w_{i+1}$ with label 1; vertices $w_0$ is adjacent to $w_1$, which has received label 1; and vertices $w_{l-2}$ are adjacent to $w_{l-3}$, also labelled with 1. Therefore, for all $i < l-1$, we have $\{1, 3\} \subseteq \Pi_{N(w_i)}$ and we conclude that $c_\pi(w_i) = 2$ for all three vertices $x_i$, $y_i$ and $z_i$. This case is represented in Figure 3.72 by $T_0$, $T_2$ and $T_4$.

A similar line of reasoning allows us to determine the colour for every $w_i$ with odd index. First, observe that every $w_i$ is adjacent to $v_i$ which has received label 2 in $\pi$. Similarly to the case of $i$ even, every $w_i$ is adjacent to a vertex $u$ with $\pi(u) = 2$. Therefore, set $\{1, 2\}$ is necessarily a subset of $\Pi_{N(w_i)}$. Moreover, label 3 is only assigned to vertices $v_i$ with even index, which are not adjacent to any $w_i$ in this case. Therefore, we have $\Pi_{N(w_i)} = \{1, 2\}$ for every $w_i$, $i$ odd, and we conclude that $c_\pi(w_i) = 1$ for these vertices. This case is exhibited by $T_1$, $T_3$ and $T_5$ in Figure 3.72.

Since we have exhausted every labelling of $T_i$ and concluded that there are no adjacent vertices with the same induced colour, $(\pi, c_\pi)$ is, in fact, a gap-[3]-vertex-labelling of $G$.

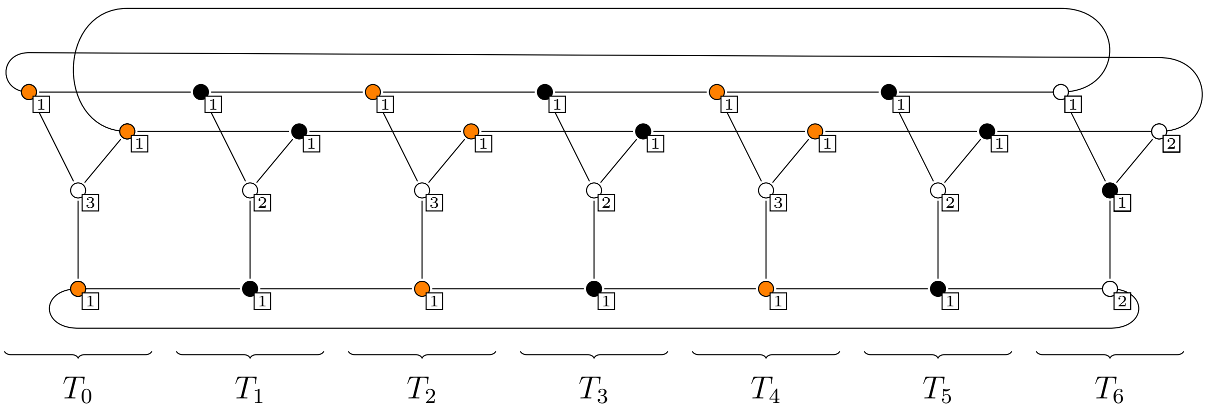



Figure 3.72: The gap-[3]-vertex-labelling $(\pi, c_\pi)$ of $J_7$, as described in the text.

This completes the proof. □

The last classes considered are the family of Goldberg and Twisted Goldberg Snarks.

### Goldberg snarks

The family of Goldberg Snarks $G_l$, $l \geq 3$ and odd, was described in Section 2.2.5. To recall, we present in Figure 3.73(a) block $B$ used in the construction of $G_l$ and, in Figure 3.73(b), a sketch of $G_l$.

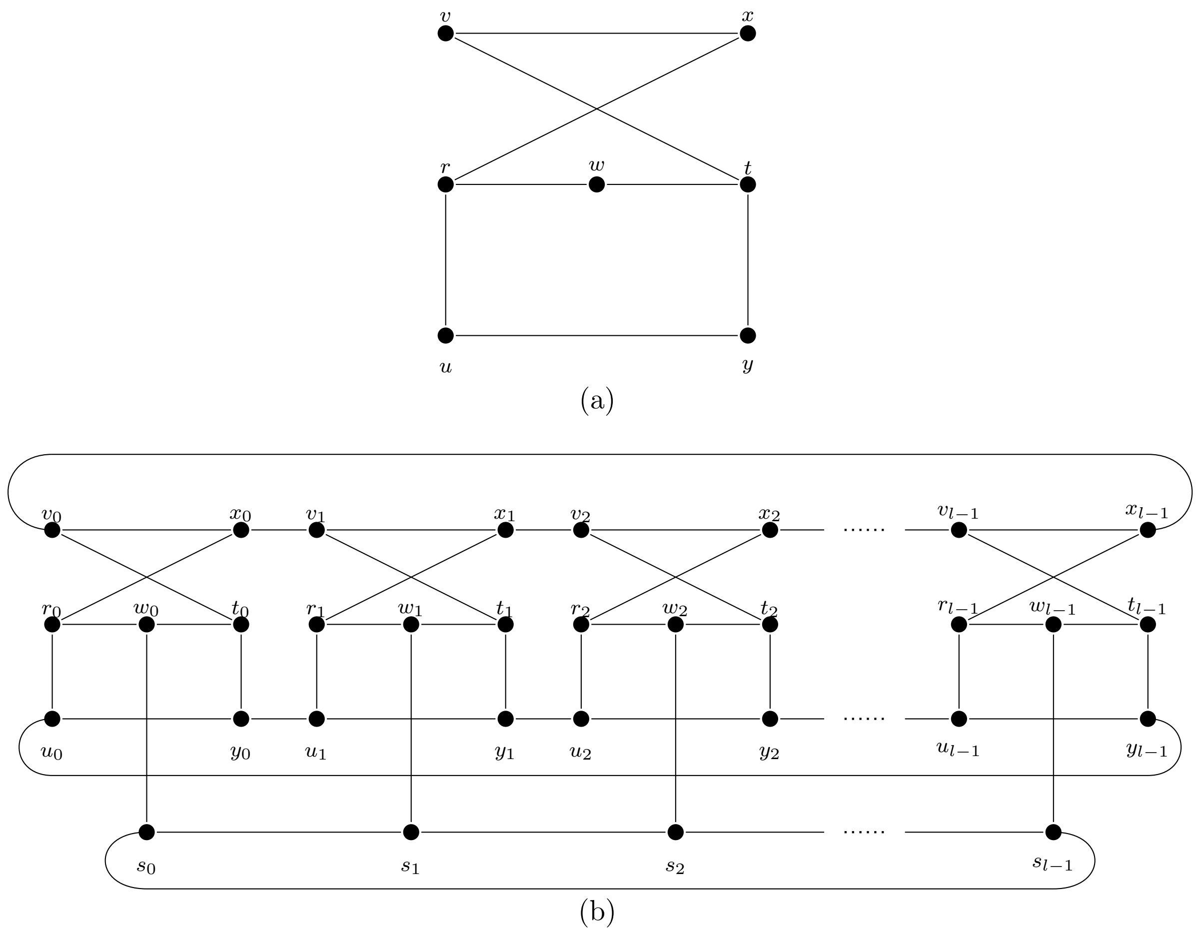



Figure 3.73: In (a), block $B$ used in the construction of Goldberg Snark $G_l$; and in (b), the resulting graph obtained by using $l$ copies of block $B$.

For this family, we determined the vertex-gap number, which is presented in Theorem 3.27.

**Theorem 3.27.** *Let $G \cong G_l$, $l \geq 3$. Then, $\chi_V^g(G) = 3$.*

*Proof.* Let $G$ be Goldberg Snark $G_l$, $l \geq 3$. It is known that $\chi(G) = 3$ for every $l$. Therefore, by Corollary 3.8, it is sufficient to show a gap-[3]-vertex-labelling of $G$ to prove the result. First, we consider the case $l = 3$, which is a unique construction in the family. The gap-[3]-vertex-labelling $(\pi, c_\pi)$ of $G_3$ is presented in Figure 3.74. By inspection, one can see that $c_\pi$ is a proper colouring of $G$, and the result follows. It remains to consider the case $l \geq 5$.

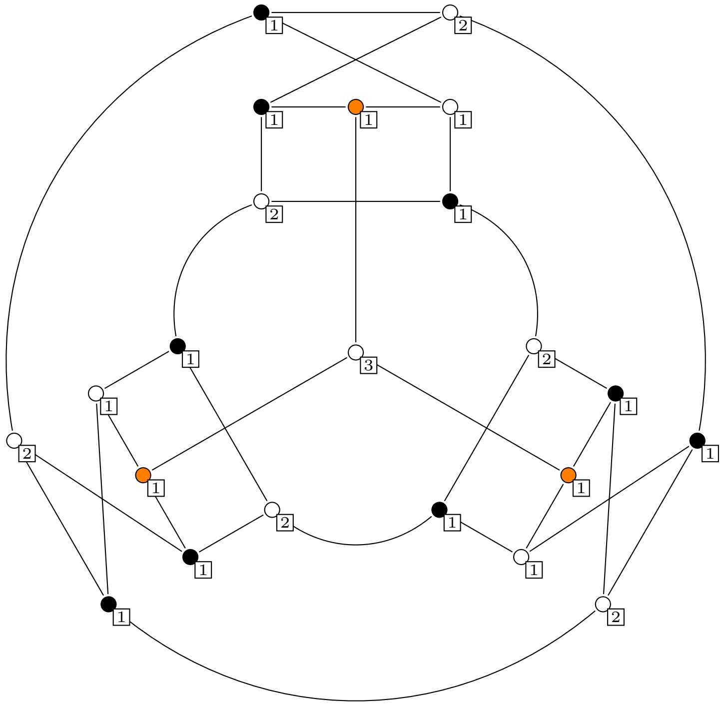
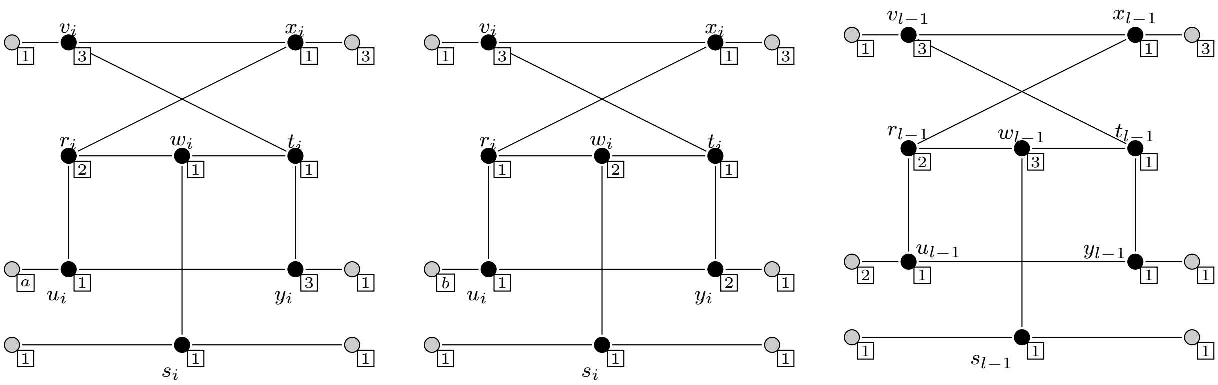


Figure 3.74: The gap-[3]-vertex-labelling of Goldberg Snark $G_3$. White vertices have induced colour 0, black vertices, colour 1 and orange vertices, colour 2.

We construct a gap-[3]-vertex-labelling $(\pi, c_\pi)$ of $G_l$, $l \geq 5$, similarly to the construction done for Flower snarks in the proof of Theorem 3.26. We define three labellings for blocks $B_i$, $0 \leq i < l$, for the following cases: $i \leq l-2$ and even; (ii) $i \leq l-2$ and odd; and (iii) $i = l-1$. These three labellings are presented in Figure 3.75; items (i), (ii) and (iii) correspond to subfigures (a), (b) and (c), respectively. An example of labelling $\pi$ can be seen in Figure 3.76 for Goldberg snark $G_5$.

First, consider vertices $s_i$, for all $i < l$. Since $\pi(s_i) = 1$ in all cases, we have $\Pi_{N(s_i)} =$



(a) Labelling of blocks $B_i$, $i \leq l-2$ and even.

(b) Labelling of blocks $B_i$, $i \leq l-2$ and odd.

(c) Labelling used only for block $B_{l-1}$.

Figure 3.75: The labellings of blocks $B_i$ for cases (i), (ii) and (iii) in (a), (b) and (c), respectively. Vertices $y_{i-1}$ have their labels $a \in \{1, 2\}$ and $b \in \{1, 3\}$, considering the possible blocks $B_{i-1}$ connected to $B_i$.

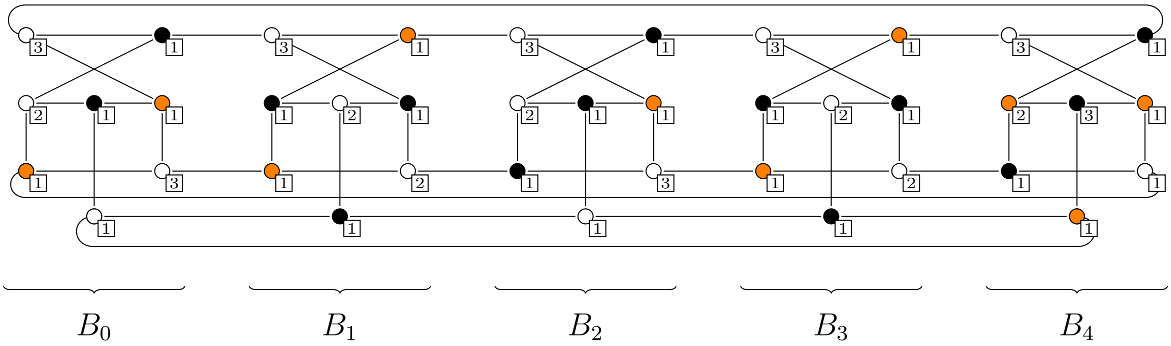



Figure 3.76: The gap-[3]-vertex-labelling of Goldberg snark $G_5$.

$\{1, \pi(w_i)\}$. Notice that labelling $\pi$ alternates labels 1,2 in $w_i$ for blocks $B_i$, $0 \leq i \leq l-2$, and block $B_{l-1}$ has $\pi(w_{l-1}) = 3$. We conclude that the cycle induced by vertices $s_i$ is properly coloured, as shown in Figure 3.77.

Next, we consider vertices $r_i$, $w_i$ and $t_i$. For $i$ odd, we have $\pi(r_i) = \pi(t_i) = \pi(s_i) = 1$. Since $\{r_i, t_i, s_i\} = N(w_i)$, we conclude that $c_\pi(w_i) = 0$ for every $w_i$ with odd index. If $i$ is even, observe that $\Pi_{N(w_i)} = \{1, 2\}$ since $\pi(r_i) = 2$ and $\pi(t_i) = \pi(s_i) = 1$. This implies $c_\pi(w_i) = 1$. Since $c_\pi(s_i) = 1$ when $i$ is odd and $c_\pi(s_i) \in \{0, 2\}$ otherwise, there is no conflict between the induced colours of vertices $s_i$ and $w_i$ in any block $B_i$.

Now, consider the cycle induced by vertices $u_i$ and $y_i$. Observe that every vertex in $N(y_i) = \{u_i, u_{i+1}, t_i\}$ received label 1. Therefore, $c_\pi(y_i) = 0$ in all blocks $B_i$. As for the induced colours of vertices $u_i$, we have the following cases. If $i$ is odd, block $B_i$ is connected to block $B_{i-1}$ which has an even index. Therefore, for all odd $i$, $\Pi_{N(u_i)} = \{1, 2, 3\}$ since $N(u_i) = \{y_{i-1}, y_i, t_i\}$. This implies that $c_\pi(u_i) = 2$ for all blocks with odd index. For blocks $B_i$ with $i$ even, we must consider two separate cases.

Since block $B_0$ is adjacent to $B_{k-1}$, we analyse vertex $u_0$ separately. Observe in Figure 3.76 that $\Pi_{N(u_0)} = \{1, 2, 3\}$. This implies that $c_\pi(u_0) = 2$. For every other even index $i$, we have $\Pi_{N(u_i)} = \{a, a+1\}$, where $a = 2$ for $i \neq l-1$, and $a = 1$, otherwise. This last case is exemplified by block $B_4$ in Figure 3.76.

It remains to consider the colours of vertices $v_i$ and $x_i$. Similarly to vertices $y_i$, observe that $N(v_i) = \{x_i, x_{i-1}, t_i\}$, all of which were assigned label 1. Therefore, $c_\pi(v_i) = 0$ for all blocks $B_i$. For vertices $x_i$, we have $\Pi_{N(x_i)} = \{\pi(r_i), 3\}$, which implies that $c_\pi(x_i) = 3 - \pi(r_i)$. Since no $r_i$ was assigned label 3, $c_\pi(x_i) \neq 0$; furthermore, by inspecting blocks $B_0$ and $B_1$, the reader can observe that in both cases of $i$ odd and even, $c_\pi(r_i) \neq c_\pi(x_i)$. Therefore, $c_\pi$ is a proper colouring of Goldberg Snark $G_k$, which completes the proof. □

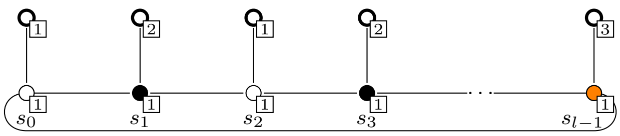



Figure 3.77: The (partial) labelling $\pi$ and colouring $c_\pi$ of the induced cycle $G[\{s_0, \dots, s_{l-1}\}]$. Every $s_i$ is adjacent to a vertex $w_i$, each of which has its colour omitted. Vertices $s_i$ filled in white have induced colour 0, vertices in black, colour 1, and the single orange vertex has induced colour 2.

**Twisted Goldberg snarks**

The family of Twisted Goldberg Snarks is formally defined in Chapter 2. To recall, the *Twisted Goldberg Snark* $TG_l$, $l \geq 3$, is obtained by *twisting* an odd number of edges connecting adjacent blocks in Goldberg Snark $G_l$. Figure 3.78 illustrates the twist operation.

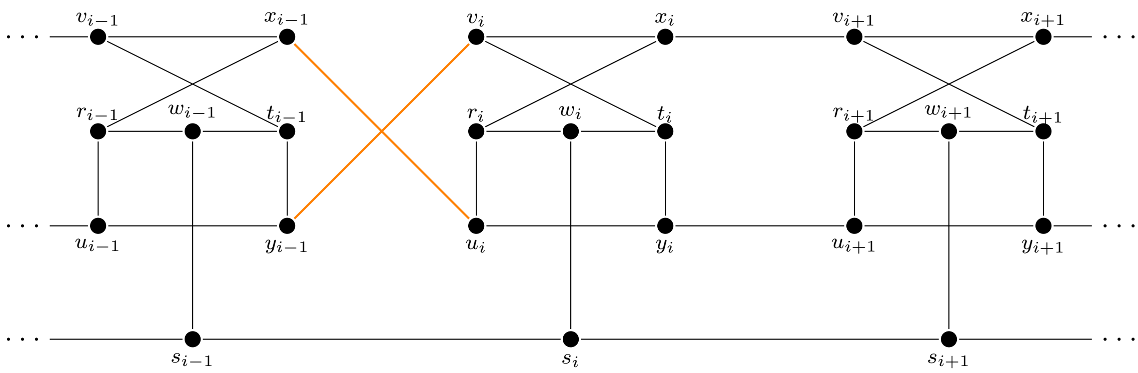



Figure 3.78: A twisted edge in Goldberg Snark $G_l$.

By using the same approach as the previous classes of snarks, we established the vertex-gap number for the family of Twisted Goldberg Snarks. This result is presented in Theorem 3.28.

**Theorem 3.28.** *Let* $G \cong TG_l$, $l \geq 3$. *Then,* $\chi^g_V(G) = 3$.

*Proof.* Let $G$ be the Twisted Goldberg Snark for $l \geq 3$. Once more, the result follows from establishing a gap-[3]-vertex-labelling of $G$. For $l = 3$, we consider graph $TG_3$, which has a unique construction. The gap-[3]-vertex-labelling of this graph is presented in Figure 3.79.

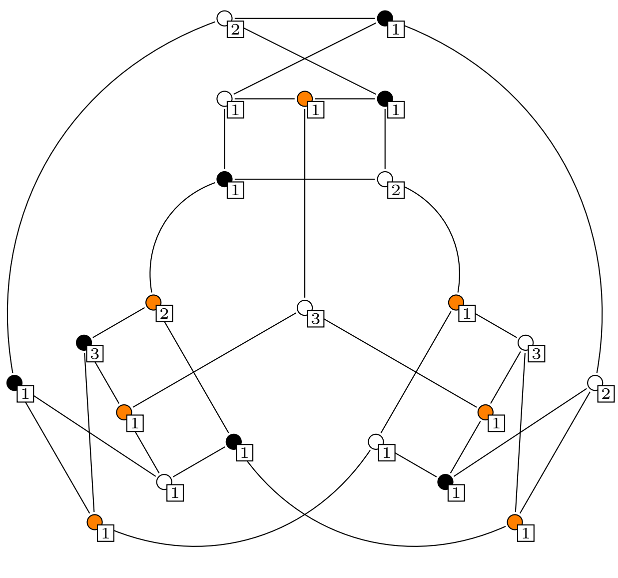
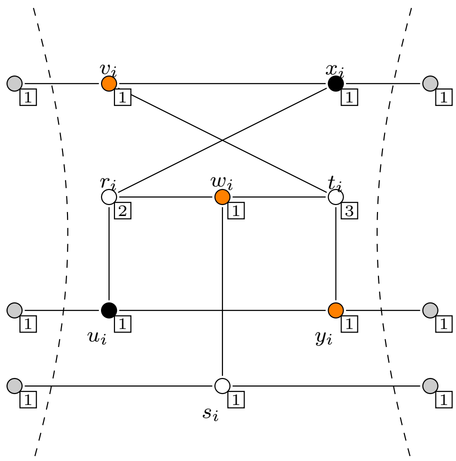
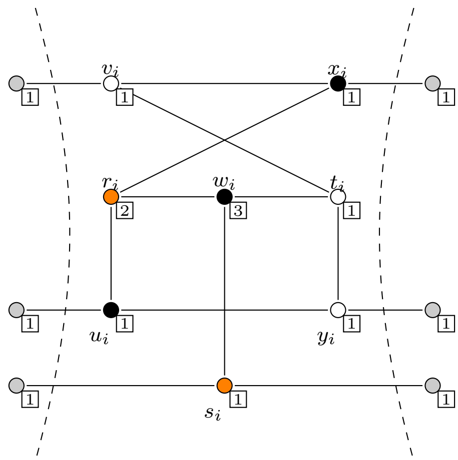


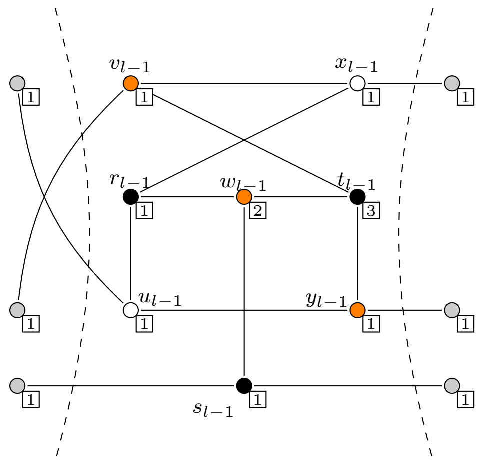
Figure 3.79: The gap-[3]-vertex-labelling of Twisted Goldberg Snark $TG_3$. Vertices filled in white, black and orange have induced colours 0, 1 and 2, respectively.




(a) Labelling of blocks $B_i$, $i \leq l-2$ and even.




(b) Labelling of blocks $B_i$, $i \leq l-2$ and odd.




(c) Labelling used only for block $B_{l-1}$. Observe the twisted edges attached to $v_{l-1}$ and $u_{l-1}$.

Figure 3.80: The labellings of blocks $B_i$ for cases (i), (ii) and (iii) in (a), (b) and (c), respectively. Observe that for Twisted Goldberg snarks, every $u_i, y_i, v_i$ and $x_i$ was assigned label 1. Vertices filled in white have induced colour 0, black, colour 1, and orange, colour 2.

For $l \geq 5$, the proof of this result is similar to the proofs of theorems 3.26 and 3.27. We define the labellings for blocks $B_i$ for three cases: (i) $i \leq l-2$ and even; (ii) $i \leq l-2$ and odd; and (iii) $i = l-1$. The labelling for each case is depicted in Figure 3.80, subfigures (a), (b) and (c), respectively. As in the proof of Goldberg Snarks, the gray vertices adjacent to $v_i$, $x_i$, $u_i$ and $y_i$ represent their adjacent vertices in neighbouring blocks. Moreover, note that all connecting vertices received label 1. By inspecting blocks $B_i$, we conclude that $c_\pi$ is a proper colouring of each block. $\square$

By observing all classes presented in this chapter, we strongly believe that there is a correlation between the vertex-gap number of a graph $G$ and its chromatic number, other than the lower bound established in Lemma 3.7. However, despite the evidence provided in this section, there is still no proof that the vertex-gap number of a graph can be inferred from its chromatic number. In fact, we leave this as an open problem.

**Problem 3.29.** *Let $G$ be an arbitrary graph and $f$, a function. Is it possible to establish $f$*

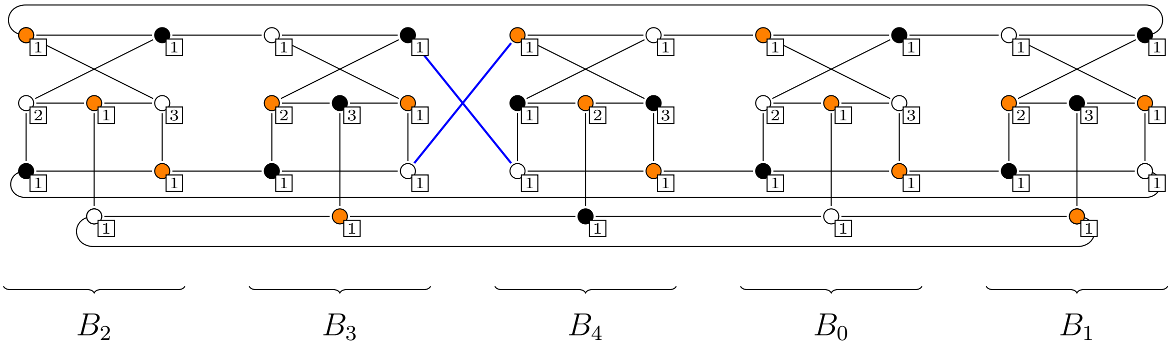



Figure 3.81: The gap-[3]-vertex-labelling of Twisted Goldberg Snark $TG_5$. The twisted edges are highlighted in blue. Vertices filled in white have induced colour 0, in black, colour 1, and in orange, colour 2.

*such that* $\chi_V^g(G) \leq f(\chi(G))$*?*

# Chapter 4

# Further discussions on gap-vertex-labellings

Our research on gap-$[k]$-vertex-labellings, presented in the previous chapter, enabled us to determine the vertex-gap number of some traditional classes of graphs. We pose a conjecture that almost every cubic bipartite hamiltonian graph admits a gap-$[2]$-vertex-labelling, and provide evidence to support this conjecture. Also, in the beginning of Section 3.3, we establish a tight lower bound for the vertex-gap number of arbitrary graphs.

All results presented thus far regarding gap-$[k]$-vertex-labelling have a common factor: they all rely on the input parameter $k$. To clarify: we determined the least number $k$ for which certain graphs admit a gap-$[k]$-vertex-labelling, and established a lower bound for the least $k$ for which a graph admits a gap-$[k]$-vertex-labelling. However, we have yet to address a fundamental question regarding gap-$[k]$-vertex-labellings: are there graphs which do not admit this proper labelling, regardless of $k$? The answer is *yes*.

In the article that introduced gap-$[k]$-vertex-labellings, Dehghan et al. [8] stated that "a graph may lack any vertex-labelling by gap[1]". However, the authors did not characterize these graphs. Here, we make note of how strong Dehghan et al.'s statement is: for certain graphs, there is no natural $k$ for which the graph admits any gap-$[k]$-vertex-labelling. In light of this, Dehghan et al. [8] proposed the following question:

**Problem 4.1** (Dehghan et al.). *Does there exist a polynomial-time algorithm to determine whether a given graph admits a gap-$[k]$-vertex-labelling?*

In this chapter, we present our discussions related to Dehghan et al.'s problem. We present two families of graphs which do not admit gap-$[k]$-vertex-labellings in Section 4.1, and comment on their importance for this particular problem. In Section 4.2, a new parameter, called the gap-strength of graphs, is introduced and we present some preliminary results for it. Finally, in Section 4.3, we prove certain structural properties regarding gap-$[k]$-vertex-labellings of graphs, when they exist. These properties are used to design a brute force algorithm that decides whether a given graph admits a gap-$[k]$-vertex-labelling, for some $k \in \mathbb{N}$. This is the first known algorithm to solve Dehghan et al.'s decision problem and it executes in $\mathcal{O}(n!)$ time. As a corollary of these properties, we also obtain a tight upper bound for the vertex-gap number of arbitrary graphs.

[1]This is the original name of gap-$[k]$-vertex-labellings as given by Dehghan et al. [8] in 2013.

In the context of deciding if and when a graph admits a gap-$[k]$-vertex-labelling, the value of $k$ is, to some extent, irrelevant. Here, we are not interested in the least $k$, or how large or small its value is. We merely inquire if there exists (any) $k \in \mathbb{N}$. Hence, for the remainder of this chapter, we refer to gap-$[k]$-vertex-labellings of graphs simply as *gap-vertex-labellings*, omitting $k$.

## 4.1 Graphs that do not admit gap-vertex-labellings

When considering Problem 4.1, we remark that there is a significant difference when comparing the gap-vertex-labelling version to its edge counterpart. In the edge version, presented in Chapter 2, Tahraoui et al. [27] showed that a graph $G$ admits a gap-$[k]$-edge-labelling, for some $k \leq 2^{|E|-1}$, if and only if $G$ does not have any connected component isomorphic to $K_2$. Therefore, we know how to decide whether a graph admits a gap-$[k]$-edge-labelling, for some $k \in \mathbb{N}$, in polynomial time.

The algorithm that solves this decision problem needs only check every connected component $H \subseteq G$ and verify if they are isomorphic to $K_2$. If this is not the case, then there exists a $k$ for which $G$ admits a gap-$[k]$-edge-labelling. Conversely, if there exists a connected component $H \cong K_2$, then the label assigned to the (singular) edge $e = uv \in E(H)$ in any edge-labelling of $G$ will induce the same colour in its endpoints. Therefore, $H$ cannot be properly labelled and, consequently, $G$ does not admit a gap-$[k]$-edge-labelling. We remark that, once again, we are not interested in establishing a value for $k$. We are only interested in determining if such $k$ exists, however large or small it may be.

Now, let us return to the decision problem of determining whether a graph $G$ does or does not admit a gap-vertex-labelling, and formalize it.

Gap-vertex-labelling (**Gvl**)
**Instance:** A graph $G = (V, E)$.
**Question:** Does $G$ admit a gap-vertex-labelling?

It is now possible to restate Dehghan et al.'s problem as simply: is **Gvl** in P?

Many decision problems regarding labellings and colourings of graphs have been proven to be NP-complete. For proper gap-labellings in particular, the beginning of chapters 2 and 3 list existing NP-completeness results in the literature for both the edge and the vertex variants, respectively. Our initial assessment of **Gvl** led us to believe that this problem is also NP-complete, contrary to its edge version.

Motivated by our results and discussions regarding the gap-[2]-vertex-labelling of subcubic bipartite graphs, and by Dehghan et al.'s [8] statement – that there are graphs which lack a gap-vertex-labelling– we began our research investigating which are these graphs for which there is no gap-vertex-labelling. The first family of graphs that do not admit any gap-vertex-labelling is the family of complete graphs $K_n$ when $n \geq 4$. Theorem 4.2 establishes this result.

**Theorem 4.2.** *Let $G \cong K_n$. Then, $G$ admits a gap-vertex-labelling if and only if $n \leq 3$.*

*Proof.* Let $G = K_n$, $n \geq 2$. Complete graph $K_1$ is a trivial graph, for which the result naturally holds.

($\Rightarrow$) For complete graphs $K_2$ and $K_3$, we present gap-vertex-labellings in Figure 4.1. By inspecting the image, we conclude that the induced colouring is a proper vertex-colouring of the graph. Therefore, complete graphs $K_n$, $n \leq 3$, admit gap-vertex-labellings.

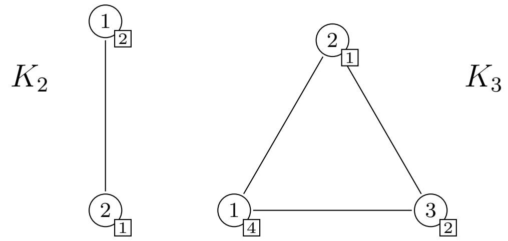



Figure 4.1: Complete graphs $K_2$ and $K_3$ with their gap-vertex-labellings.

($\Leftarrow$) Conversely, consider $n \geq 4$, and let $V = \{v_0, \dots, v_{n-1}\}$ be the vertices of $G$. Suppose that $G$ admits a gap-vertex-labelling $(\pi, c_\pi)$. Adjust notation so that $v_0$ is the vertex which is assigned the largest of all labels in $V$ and $v_1$, the smallest. Consider vertices $v_2$ and $v_3$. These vertices exist in $G$ since $n \geq 4$. Observe that $v_0, v_1 \in N(v_2)$ and $v_0, v_1 \in N(v_3)$. This implies that, regardless of the labels assigned to $v_2, v_3, \dots, v_{n-1}$, both $v_2$ and $v_3$ have their colours induced by the same gap, that is, $c_\pi(v_2) = c_\pi(v_3) = \pi(v_0) - \pi(v_1)$. This is a contradiction since $c_\pi(v_2) \neq c_\pi(v_3)$ in any proper vertex-colouring of $G$. Therefore, $c_\pi$ is not a proper colouring of $G$ and the result follows. $\square$

By a similar line of reasoning, we were able to prove that another – albeit very restricted – family of graphs also does not admit gap-vertex-labellings: a subclass of split graphs. As defined by S. Földes and P. Hammer [10], a *split graph* is a graph $G$ whose vertex set $V(G)$ can be partitioned into the disjoint union of a nonempty independent set and a complete graph, i.e. a clique. We denote a partition – a *split* – of $G$ into a clique $W$ and an independent set $U$ by $(G, W, U)$. If the split of $G$ is such that $W$ is maximal, we say that $(G, W, U)$ is a *maximal split* of $G$. In Figure 4.2(a), we exemplify a split graph with $W \cong K_4$ and $U$ containing 3 vertices. Notice that $(G, W, U)$ is not a maximal split of $G$ since the bottommost vertex of $U$ has degree four and, thus, $W$ is not maximal. In fact, the maximal split $(G, W', U')$ of this graph is presented in Figure 4.2(b), in which $W' \cong K_5$ and $|U'| = 2$.

In a preliminary study of split graphs, we considered only those in which every vertex $u \in U$ has degree one. For these graphs, we determined when they admit a gap-vertex-labelling. This result is presented in Theorem 4.3. Notice in the statement of the theorem that we are considering only split graphs in which clique $W$ has at least four vertices since a split graph with $|W| = 3$ is a unicyclic graph, for which we create a gap-[3]-vertex-labelling in Chapter 3.

**Theorem 4.3.** *Let $(G, W, U)$ be a maximal split graph with $|W| \geq 4$ and $d(u) = 1$ for every $u \in U$. Then, $G$ admits a gap-vertex-labelling if and only if there are $l \geq |W| - 3$ vertices in $W$ with at least one neighbour in $U$.*

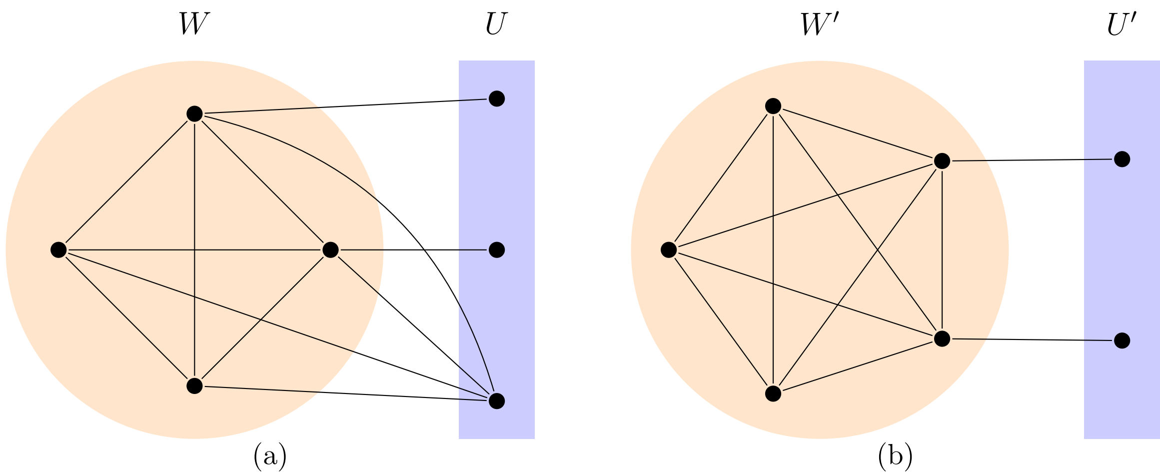



Figure 4.2: In (a), a split graph $(G, W, U)$; and in (b), a maximal split of the same graph $(G, W', U')$. The cliques and independent sets are highlighted in orange and blue, respectively.

*Proof.* Let $(G, W, U)$ as stated in the hypothesis and let $r$ and $s$ denote the sizes of parts $W$ and $U$, respectively. Adjust notation of $V(G)$ as follows: let $w_0, w_1, \ldots, w_{r-1}$ be the vertices of the clique; and for every $w_i \in W$, let $u_j^i \in U$ denote the $j$-th degree-one vertex adjacent to $w_i$. Figure 4.3 presents a sketch of the defined notation. As examples, note that $w_0$ is adjacent to two vertices in $U$, while $w_1$ and $w_{r-2}$ are adjacent to one vertex and $w_{r-1}$, to none.

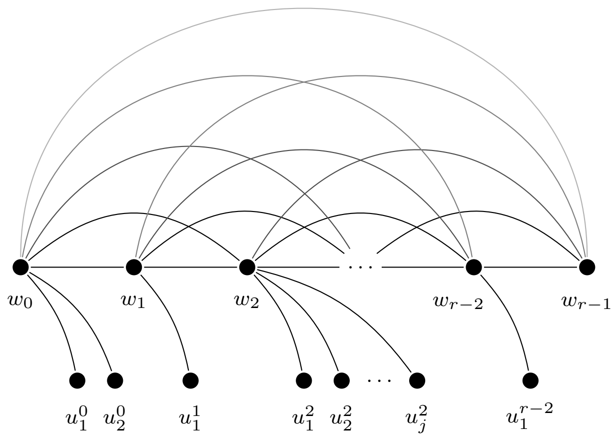



Figure 4.3: The defined notation for a maximal split graph $(G, W, U)$.

($\Rightarrow$) Suppose there are $l \geq |W| - 3$ vertices in $W$ with at least one neighbour in $U$. Adjust notation so that $w_0, \ldots, w_{l-1}$ are these vertices. Then, by our hypothesis, we know that there are (at most) three vertices, namely $w_{r-1}$, $w_{r-2}$ and $w_{r-3}$, with $d(w_i) = |W| - 1$. In order to prove the result, we define a labelling $\pi$ of $V(G)$ which induces a proper colouring $c_\pi$ of the graph. For every vertex $w_i \in W$, assign:

$$\pi(w_i) = \begin{cases} 2^2, & \text{if } i = r-1; \\ 2^0, & \text{if } i = r-2; \\ 2^1, & \text{otherwise.} \end{cases}$$

Finally, for every $u^i_j \in U$, let $\pi(u^i_j) = 2^{i+3}$. Define colouring $c_\pi$ as usual. We exemplify this labelling[2] for a split graph $(G, W, U)$ in Figure 4.4. In this graph, $W \cong K_6$ and there are $l = r-3$ degree-one vertices in $U$ adjacent to vertices $w_i$.

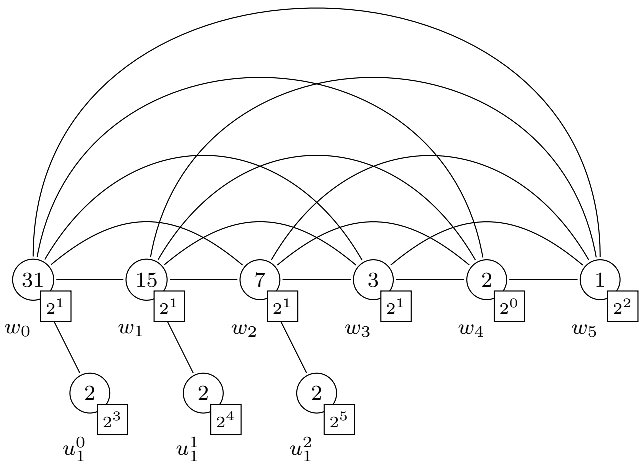



Figure 4.4: The gap-vertex-labelling of a maximal split graph $(G, W, U)$.

We show that induced colouring $c_\pi$ is a proper vertex-colouring of $G$. We start by considering vertices $u^i_j$. Recall that every vertex $u^i_j \in U$ has $d(u^i_j) = 1$ and, therefore, $c_\pi(u^i_j) = \pi(w_i) = 2$ for every $0 \le i \le r-4$. For vertices $u^{r-3}_j$, $u^{r-2}_j$ and $u^{r-1}_j$, when they exist, their induced colours are 2, 1 and 4, respectively.

Next, we consider the vertices in $W$. With the exception of $w_{r-2}$, every vertex $w_i$ in $W$ is adjacent to $w_{r-2}$, which received label 1. Moreover, note that $\pi(w_{r-2}) = 1$ is the smallest of all labels in $\Pi_{V(G)}$. This implies that every $w_i$, $i \neq r-2$, has its colour defined as $c_\pi(w_i) = \pi(v) - \pi(w_{r-2})$, for some $v \in V(G) - w_{r-2}$. Now, consider the first $0 \le i < l$ vertices $w_i$ in $W$ which are adjacent to vertices in $U$ and recall that $\pi(w_{r-1})$ is the largest of all labels in $\Pi_W$. Since every vertex $u^i_j$ received label $\pi(u^i_j) > \pi(w_{r-1}) = 4$, we conclude that every such $w_i$ has induced colour $c_\pi(w_i) = 2^{i+3} - 1$.

It remains to determine the induced colour of the last $r-l$ vertices. If they were not considered in the previous paragraph, these vertices are $w_{r-1}$, $w_{r-2}$ and $w_{r-3}$. Then, we have $\Pi_{N(w_{r-3})} = \{1, 4\}$, $\Pi_{N(w_{r-2})} = \{2, 4\}$, and $\Pi_{N(w_{r-1})} = \{1, 2\}$, inducing $c_\pi(w_{r-i}) = i$, $i \in \{1, 2, 3\}$.

Finally, we show that there are no conflicting vertices in $G$. Observe that the first $l$ vertices $w_i$ always have an odd induced colour, whereas each of their respective neighbours in $U$ have $c_\pi(u^i_j) = 2$. For the last $r-l$ vertices, $c_\pi(w_{r-i}) \neq \pi(w_{r-i})$, which implies

[2] The reader might have noticed that this labelling assigns only powers of two to the vertices of the graph. In Section 4.3, we elaborate on this particular decision of labels.

$c_\pi(w_{r-i}) \neq c_\pi(u_j^{r-i})$ when $u_j^{r-i}$ exists. Therefore, there are no conflicting vertices in $G$ and, consequently, $c_\pi$ is a proper vertex-colouring of the graph.

($\Leftarrow$) Conversely, suppose $l < r - 3$. Then, there are (at least) four vertices in $W$ whose neighbourhoods are strictly contained in $W$. Adjust notation so that $w_1, \ldots, w_{r-l}$ are these vertices.

Suppose $G$ admits a gap-vertex-labelling $(\pi, c_\pi)$ and let $w_{\max}$ and $w_{\min}$ be the vertices in $W$ with the largest and smallest labels in $\Pi_W$, respectively. Consider the case where $\{w_{\max}, w_{\min}\} \not\subset \{w_1, \ldots, w_{r-l}\}$. Then, it follows that $w_1, \ldots, w_{r-l}$ all receive the same colour in $c_\pi$, induced by $\pi(w_{\max}) - \pi(w_{\min})$. This is a contradiction since, by hypothesis, $c_\pi$ is a proper vertex-colouring of $G$. Thus, one (or both) of $w_{\max}$ and $w_{\min}$ are in set $\{w_1, \ldots, w_{r-l}\}$. In this first moment, we consider $w_{\max} \in \{w_1, \ldots, w_{r-l}\}$.

Without loss of generality, let $w_1 = w_{\max}$. Then, if $w_{\min} \notin \{w_2, \ldots, w_{r-l}\}$, by a similar reasoning, we conclude that vertices $w_2, \ldots, w_{r-l}$ have the same induced colour in $c_\pi$. Therefore, $w_{\min} \in \{w_2, \ldots, w_{r-l}\}$. Once again, without loss of generality, we consider $w_{\min} = w_2$. This implies that $c_\pi(w_3) = \ldots = c_\pi(w_{r-l}) = \pi(w_1) - \pi(w_2)$. Moreover, since $r - l \geq 4$, we know that there are at least two vertices with the same induced colour, which is a contradiction. We remark that the same conclusion is reached when first considering $w_{\min} \in \{w_1, \ldots, w_{r-l}\}$. Therefore, there is no gap-vertex-labelling $(\pi, c_\pi)$ of $G$ and the result follows. $\square$

The family of split graphs has several interesting properties and is widely studied in Graph Theory. Theorem 4.3 covers only a small subclass of this family. In fact, during our research, we encountered several split graphs which admit gap-vertex-labellings. For example, consider Figure 4.5. In this graph, each vertex in $U$ has degree two. As demonstrated in the image, this split graph admits a gap-vertex-labelling. However, if we remove the rightmost vertex in $U$, the graph resulting from this operation does not admit a gap-vertex-labelling.

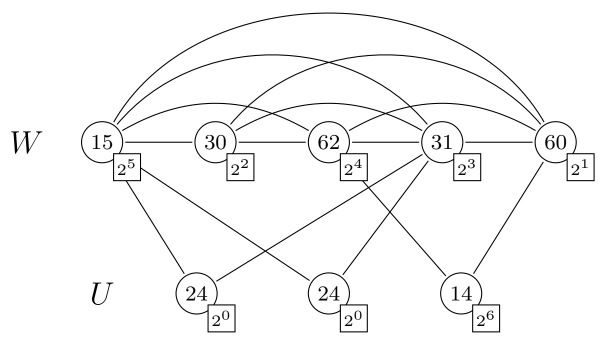



Figure 4.5: A gap-vertex-labelling of a split graph not covered by Theorem 4.3.

There is still much work to be done regarding gap-vertex-labellings of split graphs. In this context, it is interesting to establish another infinite family of graphs which does not admit such a labelling. Due to time constraints, we did not characterize split graphs that admit gap-vertex-labellings. Also, for graphs in this family which admit gap-vertex-labellings, it would be interesting to determine their vertex-gap number. We leave these problems open for future research.

**Problem 4.4.** *Characterize split graphs that do not admit gap-vertex-labellings.*

**Problem 4.5.** *Determine the vertex-gap number of split graphs.*

We have successfully established that certain graphs do not admit gap-vertex-labellings. More importantly, note that both classes considered have complete graphs of $n \geq 4$ vertices as subgraphs in their structure. In particular, the split graph in Figure 4.5 shows that removing a vertex from the graph, in this case, hinders the existence of a gap-vertex-labelling. This led us to question if, by performing operations to the structure of graphs that do not admit gap-vertex-labellings, it would be possible to create new graphs which do admit gap-vertex-labellings. With this in mind, Section 4.2 presents a discussion on the gap-strength of graphs – a new parameter associated with gap-vertex-labellings.

## 4.2 The gap-strength of graphs

As mentioned in the previous section, another interesting problem arose from our discussions on the gap-vertex-labellings of complete and split graphs, which we introduce here. Consider, for example, complete graph $K_4$ and recall the proof of Theorem 4.2. In our demonstration, while supposing that a gap-vertex-labelling exists for this graph, we consider two vertices that are labelled with the maximum and minimum labels in $\Pi_V$. Let us refer to these vertices as $v_{\max}$ and $v_{\min}$, respectively. Particularly in the case of $K_4$, the two remaining vertices, say $u$ and $v$, are adjacent to both $v_{\max}$ and $v_{\min}$ and, more importantly, to each other. Therefore, in the case of $K_4$, regardless of the labels assigned to $u$ and $v$, their induced colours will (always) be $c_\pi(u) = \pi(v_{\max}) - \pi(v_{\min}) = c_\pi(v)$.

However, what if we were to remove the edge between these two vertices? Then, although their induced colours would be the same, there would be no conflicting vertices in the graph. Thus, by removing a single edge from $K_4$, the resulting graph becomes *gap-vertex-labelable*. We illustrate this analysis in Figure 4.6.

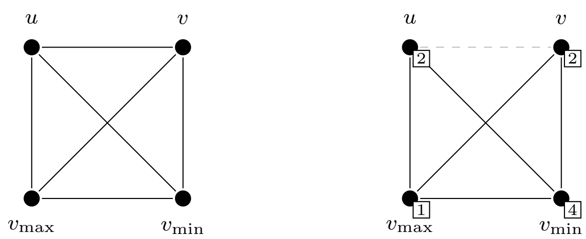



Figure 4.6: Graph $K_4$ and the graph obtained by removing edge $uv$ from $K_4$; to the right, a gap-[4]-vertex-labelling of the latter.

As another example, consider the graph $G$ obtained by removing an arbitrary edge from complete graph $K_5$, which is depicted in Figure 4.7. Suppose $G$ admits a gap-vertex-labelling and let $v_{\max}$ be an arbitrary vertex. Now, if $v_{\min}$ is adjacent to $v_{\max}$, as illustrated in Figure 4.7(a), observe that the endpoints of the highlighted edges have both $v_{\max}$ and $v_{\min}$ in their respective neighbourhoods. This implies that, regardless of the labels assigned to these vertices, they all have the same induced colour $\pi(v_{\max}) - \pi(v_{\min})$. Therefore, $v_{\max}$ and $v_{\min}$ are not adjacent. This second case is illustrated in Figure 4.7(b) and, once again, the highlighted edges indicate three vertices which have the same induced colour. Therefore, this graph does not admit a gap-vertex-labelling, which is also the case for complete graph $K_5$ (see Theorem 4.2).

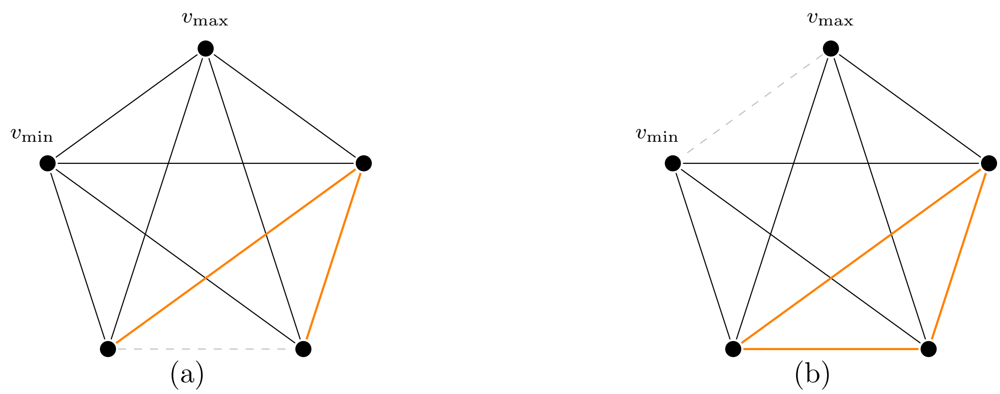



Figure 4.7: Graph $K_5$ without an edge. In (a), $v_{\max}$ and $v_{\min}$ are adjacent, while this is not the case in (b).

We conclude that removing a single edge from $K_5$ is not sufficient to create a graph which admits a gap-vertex-labelling. Alternately, let us remove two edges from $K_5$. There are two distinct graphs which can be obtained by this operation: the first is obtained by removing a maximum matching of $K_5$; and the other, by removing two adjacent edges. These cases are illustrated in figures 4.8(a) and 4.8(b), respectively. Moreover, both graphs admit gap-vertex-labellings, which are also shown in Figure 4.8.

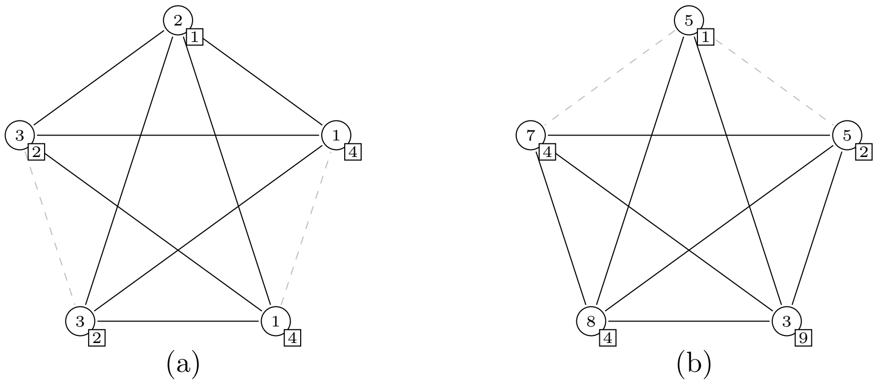



Figure 4.8: The graphs obtained by: (a) removing a maximal matching of $K_5$; and (b) removing two adjacent edges. Both graphs admit gap-vertex-labellings.

The removal of one edge from $K_4$ was sufficient for the resulting graph to admit a gap-vertex-labelling, as was the removal of two edges for complete graph $K_5$. Thus, the following question arises: what is the least number $l$ of edges that needs to be removed from complete graph $K_6$ for the resulting graph to admit a gap-vertex-labelling?

By inspecting the graphs obtained by removing one and two edges from $K_6$, we observed that none of these graphs admit a gap-vertex-labelling. (Similarly to $K_5$, this conclusion is reached upon analysing the possible combinations of $v_{\max}$ and $v_{\min}$ within the possible resulting graphs.) However, by removing a perfect matching from $K_6$, we obtain the graph depicted in Figure 4.9, which does, in fact, admit a gap-[4]-vertex-labelling.

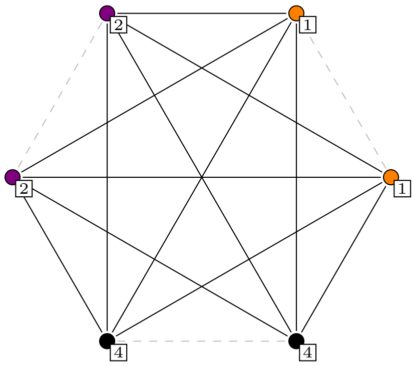



Figure 4.9: Graph $K_6$ with a perfect matching removed (dashed edges), and a gap-[4]-vertex-labelling of the resulting graph. Vertices in black, orange and violet have induced colours 1, 2 and 3, respectively.

Before we proceed, let us formally define $G^{-l}$ as the family of graphs which are obtained by removing $l$ edges, in any order, from $G$. As examples: the rightmost graph in Figure 4.6 exemplifies the (only) graph in $K_4^{-1}$; while both graphs in Figure 4.8 belong to $K_5^{-2}$. Then, we know that no graph in $K_6^{-1}$ or in $K_6^{-2}$ admits a gap-vertex-labelling, whereas there exists a graph in $K_6^{-3}$ (see Figure 4.9) which does. It is important to remark that this is not the case for every graph in $K_6^{-3}$; for example, the graph obtained by removing three adjacent edges from $K_6$ is not gap-vertex-labelable.

We were prompted, thus, with the following question: what is the least number $l$ of edges that must be removed from an arbitrary graph $G$ such that there exists a gap-vertex-labelable graph in $G^{-l}$? With this problem in mind, we introduce a new parameter associated to the gap-vertex-labelling problem. The *gap-strength* of a graph $G$ is defined as the least number $l$ for which there exists a graph $G' \in G^{-l}$ such that $G'$ admits a gap-vertex-labelling. We denote the gap-strength of $G$ by $\mathrm{str}_{\mathrm{gap}}(G)$.

We named this parameter using "strength" as the keyword in order to symbolize the main structure which we believe to be in the heart of every non-gap-vertex-labelable graph. In this sense, graph $K_6$, for example, is sufficiently *strong* that the removal of two edges is not enough to create a gap-vertex-labelable graph. Therefore, graph $K_6$ is relatively "stronger" than $K_4$, for example, since we require the removal of more edges from the former in order to create a gap-vertex-labelable graph. Similarly, by comparing $K_5$ and $K_6$, we conclude that $K_5$ is relatively "weaker".

In the following sections, we present our findings of the gap-strength of complete graphs, a family for which we know no graph of order $n \geq 4$ admits a gap-vertex-labelling. We begin by considering a rather restricted substructure in complete graphs; this particular case is be used to establish bounds for $\mathrm{str}_{\mathrm{gap}}(K_n)$ in Section 4.2.1. We conclude our discussion by presenting a dynamic programming algorithm which can be used to obtain a lower bound for the parameter in Section 4.2.2.

### 4.2.1 A restricted analysis on complete graphs

Before we present the results obtained during our investigation on the gap-strength of complete graphs, we present a similar problem, also on complete graphs, but with certain restrictions. This first analysis will be important for our discussions later in this section.

Let $K_n$ be a complete graph of order $n \geq 4$. Suppose we wish to remove a number $l'$ of edges from $K_n$ such that the resulting graph, $G$, admits a gap-vertex-labelling. We do this, however, with one restriction: in any gap-vertex-labelling of $G$, every vertex must be adjacent to a vertex, $v_{\max}$, which receives the largest label in $\Pi_{V(G)}$. Note that this implies that $v_{\max}$ remains a universal vertex in $G$, that is, no edge incident with $v_{\max}$ is removed from $E(G)$. Now, suppose $G$ admits a gap-vertex-labelling $(\pi, c_\pi)$ and let $v_{\min} \in V(G)$ denote a vertex which received the smallest of all labels in $\Pi_{V(G)}$. Then, by considering these restrictions, observe that for every $v \in V(G) \backslash \{v_{\max}, v_{\min}\}$, either:

(i) $v_{\min} \in N(v)$; or

(ii) $v_{\min} \notin N(v)$.

Let $\mathcal{I}$ and $\mathcal{X}$ be the subsets of $V(G)$ which comprise vertices that satisfy cases (i) and (ii), respectively. Thus, we have created a *decomposition* of $K_n$, which we denote by $G(\mathcal{X}, \mathcal{I})$. Given that we are analysing a restricted case of $K_n$, we will refer to $G(\mathcal{X}, \mathcal{I})$ as a *restricted decomposition of* $K_n$; the general decomposition is studied in Section 4.2.2. We illustrate a sketch of this restricted decomposition in Figure 4.10 for complete graph $K_{15}$.

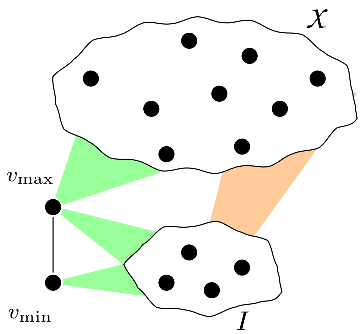



Figure 4.10: A restricted decomposition $G(\mathcal{X}, \mathcal{I})$ of $K_{15}$. The green areas symbolize the edges connecting $v_{\max}$ and $v_{\min}$ to vertices in sets $\mathcal{X}$ and/or $\mathcal{I}$. The orange areas indicate that there may be edges connecting vertices in $\mathcal{X}$ and $\mathcal{I}$. Note that there are no edges connecting $v_{\min}$ to vertices in $\mathcal{X}$; also we have omitted the edges in sets $\mathcal{X}$ and $\mathcal{I}$.

Let $x = |\mathcal{X}|$ and $i = |\mathcal{I}|$ denote the size of each set. Thus, we are able to rewrite the order of complete graph $K_n$ as $n = x + i + 2$. Also, we remark that there are $n-1$ different restricted decompositions of $K_n$ – one for each distinct combination of $x$ and $i$ whose sum equals $n-2$. In order to help the reader better understand this concept, consider, as an example, complete graph $K_5$. The $n-1 = 4$ possible restricted decompositions of $K_5$ are:

1. $x = 3$ and $i = 0$;
2. $x = 2$ and $i = 1$;

3. $x = 1$ and $i = 2$; and

4. $x = 0$ and $i = 3$.

Suppose we decompose $K_5$ according to item 2, that is, sets $\mathcal{X}$ and $\mathcal{I}$ have $x = 2$ and $i = 1$ vertices, respectively. This case is illustrated in Figure 4.11. In this image, we have assigned labels to the vertices of $G$ such that, by the definitions of $v_{\max}$ and $v_{\min}$, $b \leq c, d, e \leq a$.

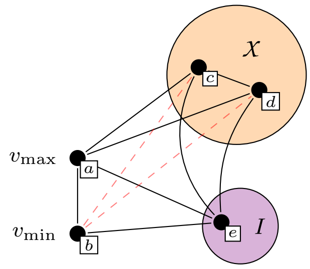



Figure 4.11: A decomposition $G(\mathcal{X}, \mathcal{I})$ of complete graph $K_5$, with $x = 2$ and $i = 1$.

In order to create this graph from $K_5$, we require the removal of $l' = 2$ edges, which connect $v_{\min}$ to the vertices in $\mathcal{X}$. The removed edges are illustrated in Figure 4.11 by the dashed, red lines. This restricted decomposition $G(\mathcal{X}, \mathcal{I})$ does, in fact, admit a gap-vertex-labelling: assign[3] $a = 2^4$, $b = 2^0$, $c = 2^1$ and $d = 2^2$ and $e = 2^3$. The reader can inspect this labelling to see that the colouring induced by these labels is a proper vertex-colouring of the graph.

Without taking into account vertices $v_{\max}$ and $v_{\min}$, the assignment of labels $c, d$ and $e$ is rather unique. In the labelling from the previous paragraph, we have $c < d < e$. If, however, label $e$ assigned to the vertex in $\mathcal{I}$ were to be strictly smaller than both $c$ and $d$, we would have both vertices in $\mathcal{X}$ with the same induced colour $a - e$. This implies that, in order for this graph, with this new labelling, to admit a gap-vertex-labelling, another edge would have to be removed – either the edge connecting the conflicting vertices in $\mathcal{X}$ or one edge connecting a vertex in $\mathcal{X}$ to the vertex in $\mathcal{I}$. If this were the case, we would have the removal of $l' = 3$.

In the context of determining the least number $l'$ of edges to be removed from a complete graph, this particular labelling does not interest us. However, it proves that we cannot simply disconsider that there may be edges removed within sets $\mathcal{X}$ and $\mathcal{I}$, or even between them. Then, for a decomposition $G(\mathcal{X}, \mathcal{I})$ of complete graph $K_n$, $n \geq 4$, let us define: $\mathcal{R}^{\mathcal{X}}$ as the number of edges removed within set $\mathcal{X}$; $\mathcal{R}^{\mathcal{I}}$, defined analogously for set $\mathcal{I}$; and, finally, let $\mathcal{R}'$ be the number of edges removed between the two sets. Then, the total number of edges $l'$ that are removed from complete graph $K_n$ such that the resulting graph $G$ admits a gap-vertex-labelling can be written as

$$l' = x + \mathcal{R}^{\mathcal{I}} + \mathcal{R}^{\mathcal{X}} + \mathcal{R}'. \tag{4.1}$$

[3]Here, we are using only powers of two for the labels of $G$. This is done in accordance with Lemma 4.12, which is presented in Section 4.3.

As another example, consider the two decompositions of complete graph $K_7$ in Figure 4.12. In subfigure (a), the sets in the decomposition have sizes $x = 2$ and $i = 3$. First, consider set $\mathcal{I}$ and recall that, by definition, every vertex $v_i \in \mathcal{I}$ has $v_{\max}, v_{\min} \in N(v_i)$. This implies that every $v_i$ has the same induced colour $c_\pi(v) = \pi(v_{\max}) - \pi(v_{\min}) = a - b$. If $G$ is to admit a gap-vertex-labelling in this decomposition, then every edge connecting vertices in $\mathcal{I}$ must also be removed. This directly implies that $\mathcal{I}$ is an independent set in any decomposition of $K_n$. Consequently, the number of edges removed in $\mathcal{I}$ is $\mathcal{R}^{\mathcal{I}} = \binom{i}{2}$.

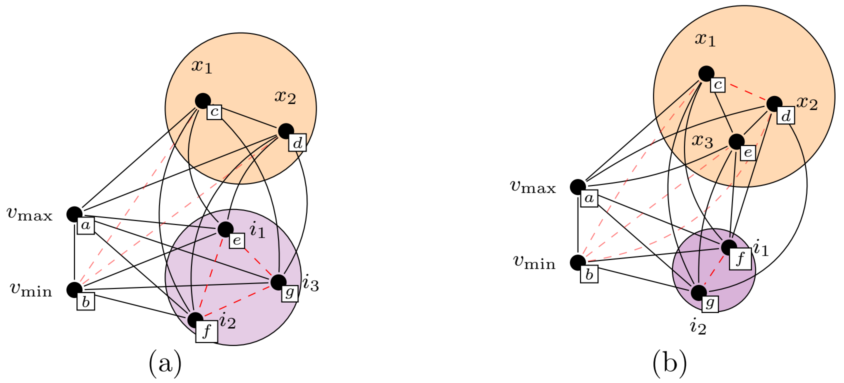



Figure 4.12: Decomposition of $K_7$ into sets $\mathcal{X}$ and $\mathcal{I}$, of sizes: (a) 2 and 3; and (b) 3 and 2.

Next, consider Figure 4.12(b), which depicts a decomposition $G(\mathcal{X}, \mathcal{I})$ of $K_7$ in which $x = 3$ and $i = 2$. We turn the readers attention to set $\mathcal{X}$, remarking that this set contains three vertices that are not adjacent to $v_{\min}$. This implies that they have their colours induced by $\pi(v_{\max})$ and $\pi(w)$ for some other $w \in V(G) - v_{\min}$. Notice that in order for this decomposition to admit a gap-vertex-labelling, we require the removal of (at least) one other edge inside set $\mathcal{X}$. We clarify this statement in the following paragraph.

Let $v'_{\min}$ be the vertex with the smallest label in $V(G) - v_{\min}$. (We remark that the subgraph induced by these vertices is also a complete graph since all edges removed so far were incident with $v_{\min}$.) Suppose $v'_{\min} \in \mathcal{I}$. This implies that every vertex in $\mathcal{X}$ has induced colour $\pi(v_{\max}) - \pi(v'_{\min})$. Then, since these vertices have conflicting colours, a number $l'' \geq 1$ of edges must be removed from $G$, either within set $\mathcal{X}$ or between the two sets. Conversely, suppose $v'_{\min} \in \mathcal{X}$. In this case, observe that, by assuming $c < e, d$ in Figure 4.12(b), we have $v'_{\min} = x_1$. Then, removing edge $x_1x_2$ is necessary for the graph to be gap-vertex-labelable. In fact, by defining $a = 2^3$, $b = 2^0$, $c = 2^1$, $d = e = f = g = 2^2$, we induce a proper vertex-colouring of the graph.

We conclude that, for this particular graph, the least number of edges removed within set $\mathcal{X}$ is $\mathcal{R}^{\mathcal{X}} = 1$ and, thus, the least number of edges removed from $K_7$ for this decomposition to admit a gap-vertex-labelling is $l' = 5$. More importantly, we remark that by removing edge $x_1x_2$ and considering the above labelling, every vertex $v_x \in \mathcal{X}$ has its colour induced either by:

1. $c_\pi(v_{\max}) - c_\pi(v'_{\min})$, which is the case of $x_3$; or
2. $c_\pi(v_{\max}) - c_\pi(w')$, for some other $w' \in \mathcal{X} + v_{\max} - v'_{\min}$.

Note that the vertices satisfying item 1 form an independent set $\mathcal{I}' \subset \mathcal{X}$, while vertices satisfying item 2 create a new set $\mathcal{X}'$ in which every vertex is adjacent to $v_{\max}$. Thus, if $x \geq 3$, we have the exact same premise upon which a restricted decomposition of the original graph $K_n$ was built: a (new) complete graph of order $n' \geq 4$, which is the subgraph induced by $\mathcal{X} + v_{\max}$, for which we require the removal of $l''$ edges such that the resulting graph is gap-vertex-labelable.

To summarize, the previous paragraphs state that in order for a decomposition $G(\mathcal{X}, \mathcal{I})$ of a complete graph $K_n$, $n \geq 4$, to admit a gap-vertex-labelling, if $x \geq 3$, then there exists a complete subgraph $G'$, induced by $\mathcal{X} + v_{\max}$, from which we also need to remove edges. Furthermore, the removal of these edges can be done by decomposing $G'$ into new subsets $\mathcal{X}'$ and $\mathcal{I}'$.

This implies that the number of edges removed from $K_n$ to create a gap-vertex-labelable graph $G$ can be computed recursively. For every combination $(x, i)$ whose sum is $x + i = n - 2$, when $x \geq 3$, we compute the number of edges removed in subproblem $\mathcal{X} + v_{\max}$ and take this value into account. Now, it is possible to establish a formula which determines the least number $l'(n)$ of edges that are required to be removed from $K_n$ in order to create a decomposition which admits a gap-vertex-labelling. Note that when $n \leq 3$, by Theorem 4.2, $K_n$ admits a gap-vertex-labelling. This is the base for our recursion.

$$l'(n) = \begin{cases} 0, & \text{if } n \leq 3; \\ \min\limits_{x+i=n-2} \{x + \frac{i(i-1)}{2} + \mathcal{R}^{\mathcal{X}} + \mathcal{R}'\}, & \text{otherwise.} \end{cases} \tag{4.2}$$

In our research, we focused on establishing bounds for the gap-strength of complete graphs; the same holds for this restricted case. In the pursuit of a lower bound for $l'(n)$, we consider the least number of edges whose removal is mandatory in order to create a decomposition $G(\mathcal{X}, \mathcal{I})$. Thus, we assume that there are no edges removed between sets $\mathcal{X}$ and $\mathcal{I}$, that is, $\mathcal{R}' = 0$. Then, we have

$$l'(n) \geq \min_{x+i=n-2} \{x + \frac{i(i-1)}{2} + l'(x+1)\}. \tag{4.3}$$

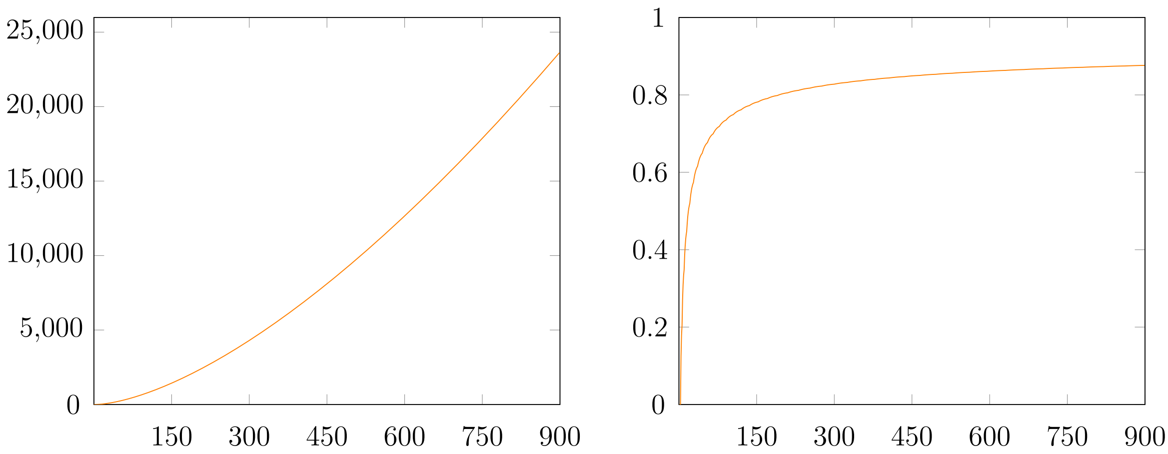
Notice that we have replaced $\mathcal{R}^{\mathcal{X}}$, from equation (4.2), with $l'(x+1)$ to account for the recursive decomposition of $G' = G[\mathcal{X} + v_{\max}]$. With equations (4.2) and (4.3) in mind, we designed a dynamic programming algorithm that computes a lower bound for the least number $l'(n)$ of edges that need to be removed from a complete graph $K_n$ such that the resulting graph admits a gap-vertex-labelling. The pseudocode is presented in Algorithm 1.

Line 1 of Algorithm 1 establishes the base case $n \leq 3$. For each value $j \leq n$, we consider the $j-1$ possible combinations of values $x$ and $i$ whose sum is $j-2$; these values represent the sizes of sets $\mathcal{X}$ and $\mathcal{I}$ in different restricted decompositions of each $K_j$, respectively. Then, in each of these restricted decompositions, it calculates the least number $l'(j)$ of edges required to be removed from $K_j$. This value is stored in the two-dimensional array `Current_Sol`. When $x \geq 3$, the recursive subproblem arises in set $\mathcal{X}$, as discussed in the previous paragraphs. Therefore, in Line 8, we consider the best computed solution for the complete graph of order $x + 1$. Line 11 stores the desired lower bound for each $K_j$ in array `Restr_Solution`.

**Algorithm 1** Given the order of a complete graph $K_n$, computes a lower bound for $l'(n)$, the least number of edges that must be removed to create a restricted decomposition $G(\mathcal{X}, \mathcal{I})$.

```
Restr_Solution[1], Restr_Solution[2], Restr_Solution[3] ← 0

for j ← 4 to n do
    Restr_Solution[j] ← ∞
    for i ← 0 to n − 2 do
        x ← j − 2 − i
        Current_Sol[j][i] ← x + i(i−1)/2
        if x ≥ 3 then
            Current_Sol[j][i] ← Current_Sol[j][i] + Restr_Solution[x + 1]
        end if

        if Current_Sol[j][i] < Restr_Solution[j] then
            Restr_Solution[j] ← Current_Sol[j][i]
        end if
    end for
end for

return Restr_Solution[n]
```

In Figure 4.13, we show the calculated results for $l'(n)$ for each value of $n < 900$ in subfigure (a), and subfigure (b) shows the value of $\frac{l'(n)}{n\sqrt{n}}$ for each $n$.




(a) (b)

Figure 4.13: Results from the execution of Algorithm 1 for $K_n$, $n < 900$.

We draw the readers attention to the graph in Figure 4.13(b), remarking that it is possible to observe that for increasing values of $n$, the bound for $l'(n)$ grows at a smaller rate when divided by $n\sqrt{n}$. We use this observation to state the following conjecture.

**Conjecture 4.6.** *Let $K_n$ be a complete graph. Then,*

$$\mathrm{str}_{gap}(K_n) \in \Omega(n\sqrt{n}).$$

By performing small modifications to Algorithm 1, we stored the sizes of sets $\mathcal{X}$ and $\mathcal{I}$ for each value of $n$. This was done to observe if any patterns emerged when considering the calculated values of $x$ and $i$ that lead to the desired lower bound. In the next section, we use these observations to create restricted decompositions of complete graphs, thus establishing an upper bound for $l'(n)$.

### An upper bound for $l'(n)$

In a modification of Algorithm 1, we stored the values of $x$ and $i$ of an optimal decomposition of $K_n$ in an array. In Table 4.1, we present some values obtained for the sizes of these sets. By observing these results, we noticed a certain pattern in the decompositions of $K_n$ for growing values of $n$.

| $n$ | $x$ | $i$ | $n$ | $x$ | $i$ |
|---|---|---|---|---|---|
| 4 | 1 | 1 | 13 | 8 | 3 |
| 5 | 2 | 1 | 14 | 9 | 3 |
| 6 | 2 | 2 | 15 | 9 | 4 |
| 7 | 3 | 2 | 16 | 10 | 4 |
| 8 | 4 | 2 | 17 | 11 | 4 |
| 9 | 5 | 2 | 18 | 12 | 4 |
| 10 | 5 | 3 | 19 | 13 | 4 |
| 11 | 6 | 3 | 20 | 14 | 4 |
| 12 | 7 | 3 | 21 | 14 | 5 |

Table 4.1: Sizes of sets $\mathcal{X}$ and $\mathcal{I}$ in optimal restricted decompositions of $K_n$.

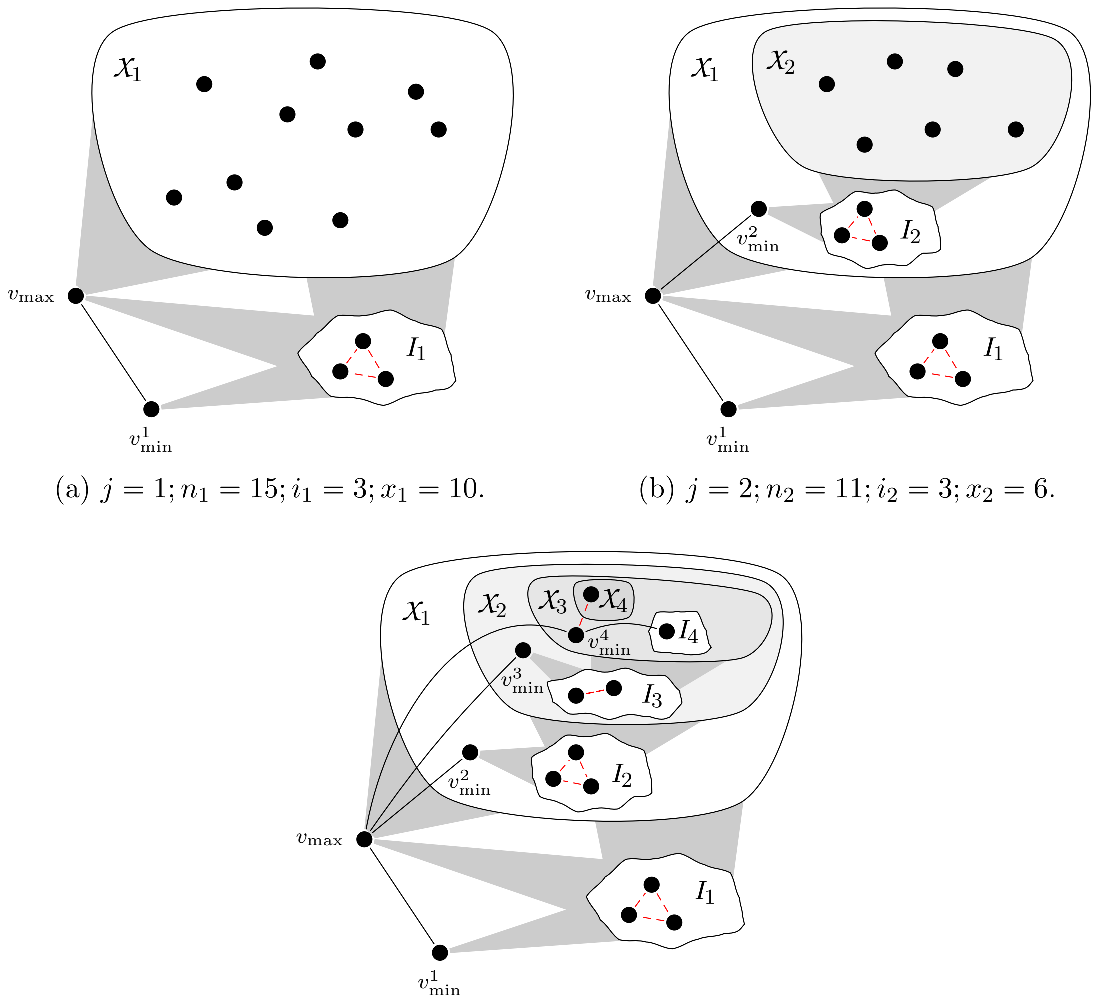
Motivated by these observations, we designed a decomposition of $K_n$ into sets $\mathcal{X}$ and $\mathcal{I}$ and, more importantly, created a gap-vertex-labelling of the resulting graph. Since we are determining the sizes of each part in the decomposition, we are able to establish an upper bound for $l'(n)$. This result is presented in Theorem 4.7.

**Theorem 4.7.** *Let $K_n$ be a complete graph. Then,*

$$\mathrm{str}_{gap}(K_n) \in \mathcal{O}(n\sqrt{n}).$$

*Proof.* Let $K_n$ be a complete graph of order $n \geq 4$. We create a restricted decomposition $G(\mathcal{X}, \mathcal{I})$ of $K_n$ by a recursive process. Each iteration $j$ in our construction will partition the (current) vertex set of a complete graph, $V_j$, into sets $\mathcal{X}_j$ and $\mathcal{I}_j$. Let $v_{\max}$ be an arbitrary vertex in $K_n$. In the first iteration $j = 1$, we have $n_1 = n$ and $V_1 = V(K_n) - v_{\max}$. For the $j$-th iteration of the construction, partition $V_j$ into:

- an arbitrary vertex $v^j_{\min}$;
- set $\mathcal{I}_j$ of size $i_j = \lfloor\sqrt{n_j}\rfloor$; and
- set $\mathcal{X}$ of size $x_j = n_j - i_j - 2$.




(a) $j = 1; n_1 = 15; i_1 = 3; x_1 = 10$.

(b) $j = 2; n_2 = 11; i_2 = 3; x_2 = 6$.

(c) $j = 4; n_4 = 4; i_4 = 1; x_4 = 1$.

Figure 4.14: Decomposition process for $K_{15}$. Gray areas symbolize all edges connecting vertices in different sets. Observe that no $v^j_{\min}$ is adjacent to vertices in $V_{j+1}$.

If $x_j \geq 3$, define $n_{j+1} = x_j + 1$, $V_{j+1} = \mathcal{X}_j$ and continue on iteration $j + 1$. Otherwise, we end our construction. In Figure 4.14 we exemplify the first, second and last iterations of our recursive process for complete graph $K_{15}$.

Next, we assign labels to each vertex of $V(G)$ as follows. Assign: $\pi(v_{\max}) = 2^{n-1}$; $\pi(v^j_{\min}) = 2^{j-1}$ for every $j \geq 1$; $\pi(v) = 2^{n-2}$ for every $v \in I_j$, $j \geq 1$. It remains to assign labels to the vertices in $\mathcal{X}_{j'}$ of the last iteration $j'$. We remark that this set has either one or two vertices, by construction. Now, if $x_{j'} = 1$, assign label $2^{j'}$ to that vertex. Otherwise, there are exactly two vertices in $\mathcal{X}_{j'}$, and we assign labels $2^{j'}$ and $2^{j'+1}$ to these vertices, in any order. Colouring $c_\pi$ is defined as usual. In Figure 4.15, we exhibit a different representation of our restricted decomposition obtained from $K_{15}$. We also show our gap-vertex-labelling $(\pi, c_\pi)$. For this graph, our construction executes $j' = 4$ iterations. In the image, vertices $v_j$ belong to set $I_j$ and vertex $v_x$ is the singular vertex in $\mathcal{X}_4$. The removed edges are displayed as red, dashed lines between vertices.

Let $f'(n)$ denote the number of edges removed in our construction. In order to complete the proof, we have to show: that colouring $c_\pi$ is a proper vertex-colouring of $G$;

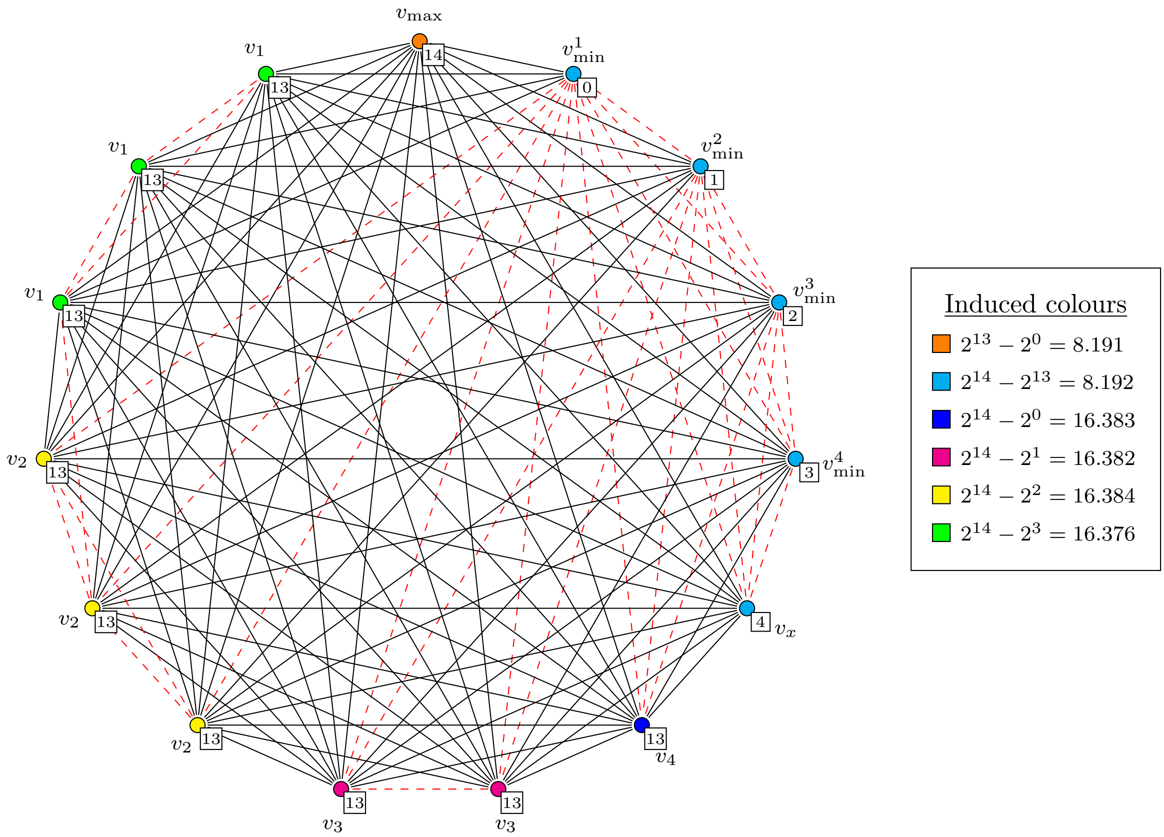



Figure 4.15: Graph $G$ obtained by our decomposition of $K_{15}$, accompanied with the gap-vertex-labelling described in the text. The value $i$ in each box next to the vertices corresponds to the label $2^i$ assigned to that vertex. The induced colours are discriminated in the table to the right of the graph.

and we removed $f'(n) \in \mathcal{O}(n\sqrt{n})$ edges from $K_n$. We start by showing the former. First, we draw the readers attention to the labels assigned to the vertices of $G$. The label set used in $\pi$ is[4] $\{2^0, 2^1, \dots, 2^{j'}, 2^{j'+1}, 2^{n-2}, 2^{n-1}\}$, where $j'$ denotes the last iteration of the recursive construction. Moreover, with the exception of $2^{n-2}$, i.e. the label assigned to vertices $v \in I_j$, $j \geq 1$, every label in the set is assigned to exactly one vertex. Now, consider $v_{\max}$ and observe that, since $v_{\max}$ is a universal vertex, the largest and smallest labels in $\Pi_{N(v_{\max})}$ are the largest and smallest label in $\Pi_{V(G)-v_{\max}}$, namely $2^{n-2}$ and $2^0$. We conclude that $c_\pi(v_{\max}) = 2^{n-2} - 1$.

Next, we consider the vertices in each set $I_j$, referring to these vertices as $v_j$. Recall that, by construction, each $I_j$ is an independent set. Also, every $v_j$ is adjacent to $v_{\max}$, which received label $2^{n-1}$. When $j = 1$, we have $v_{\max}, v^1_{\min} \in N(v_1)$, which induces $c_\pi(v_1) = 2^{n-1} - 2^0$. Hence, $c_\pi(v_1) \neq c_\pi(v_{\max})$. For every $j \geq 2$, recall that vertices in $I_j$ are not adjacent to any $v^l_{\min}$, $l < j$, since $I_j \in \mathcal{X}_{j-1}$. Moreover, $\pi(v^j_{\min}) < \pi(v^{j+l}_{\min})$ for all $j + l \leq j'$. Therefore, the smallest label in $\Pi_{N(v_j)}$ is the label assigned to $v^j_{\min}$, and we conclude that $c_\pi(v_j) = 2^{n-1} - 2^{j-1}$ for every $v_j \in I_j$. With the exception of $j = 1$, which we mention in the beginning of the paragraph, we conclude that $c_\pi(v_j)$ is always an even

[4] We remark that label $2^{j'}$ only belongs in this set if $x_{j'} = 2$ in part $\mathcal{X}_{j'}$ of the last iteration $j'$. Otherwise, the label set is $\{2^0, 2^1, \dots, 2^{j'}, 2^{n-2}, 2^{n-1}\}$.

number. Therefore, $c_\pi(v_j) \neq c_\pi(v_{\max})$ since $c_\pi(v_{\max})$ is always odd.

Now, consider the vertices in $\mathcal{X}_{j'}$. As previously stated, this set has either one or two vertices. First, suppose $|\mathcal{X}_{j'}| = 1$, and let $v_x$ be the vertex in this set. By construction, $v_x$ is adjacent to: $v_{\max}$, which received label $2^{n-1}$; to every $v_j \in I_j$, $1 \leq j \leq j'$, all of which received label $2^{n-2}$; and no other vertex. This implies that $c_\pi(v_x) = 2^{n-1} - 2^{n-2}$. Thus, $c_\pi(v_x) \neq c_\pi(w)$ for every $w \in N(v_x)$. Conversely, suppose $|\mathcal{X}_{j'}| = 2$, and let $v_x$ and $v'_x$ be the two vertices in $\mathcal{X}_{j'}$. Also, recall that $v_x$ and $v'_x$ received labels $2^j$ and $2^{j'+1}$, in any order. Without loss of generality, let $\pi(v_x) = 2^{j'}$. Now, since $\pi(v_x) < \pi(v'_x) < \pi(w)$ for every other $w \in N(v_x)$ and $w \in N(v'_x)$, it follows that $c_\pi(v_x) = 2^{n-1} - 2^{j'+1}$ and $c_\pi(v'_x) = 2^{n-1} - 2^{j'}$. This, in turn, implies that $c_\pi(v_x) \neq c_\pi(v'_x)$ and, moreover, that these induced colours do not conflict with that of the vertices in their respective neighbourhoods.

Lastly, we consider the induced colours of vertices $v^j_{\min}$. For every $1 \leq j < j'$, we remark that $N(v^j_{\min})$ consists only of $v_{\max}$ and vertices $v_j \in I_j$; these vertices received labels $2^{n-1}$ and $2^{n-2}$, respectively. Then, we conclude that every $v^j_{\min}$ has colour $c_\pi(v^j_{\min}) = 2^{n-1} - 2^{n-2} = 2^{n-2}$. It follows that $c_\pi(v^j_{\min}) \neq c_\pi(v_{\max})$. It is important to remark that the number of iterations $j' < n-1$ and, therefore, $c_\pi(v_{\min}) \neq c_\pi(v_j)$ for all $v_j \in I_j$. We conclude that there are no conflicting vertices in $G$ and, consequently, that $c_\pi$ is a proper vertex-colouring of the graph.

Thus, it remains to prove that our construction removes $f'(n) \in \mathcal{O}(n\sqrt{n})$ from $K_n$. Equivalently, we show that $f'(n) \leq 3n\sqrt{n}$, for $n \in \mathbb{N}$. We prove this result by (strong) induction on $n$. When $n \leq 3$, the inequality naturally holds since $f'(n) = 0$. Now, suppose $f'(n-1) \leq 3(n-1)\sqrt{n-1}$ for every $n \geq 1$. Let us consider the number $f'(n)$ of edges removed from $K_n$. By construction, we have

$$\begin{aligned} f'(n) &= x + \binom{i}{2} + f'(x+1) \\ &= n - \lfloor\sqrt{n}\rfloor - 2 + \frac{\lfloor\sqrt{n}\rfloor(\lfloor\sqrt{n}\rfloor - 1)}{2} + f'(n - \lfloor\sqrt{n}\rfloor - 1). \end{aligned} \tag{4.4}$$

We want to prove that $f'(n) \leq 3n\sqrt{n}$. First note that $n - \lfloor\sqrt{n}\rfloor - 1 < n$. Then, by our hypothesis, we know that $f'(n - \lfloor\sqrt{n}\rfloor - 1) \leq 3(n - \lfloor\sqrt{n}\rfloor - 1)\sqrt{n - \lfloor\sqrt{n}\rfloor - 1}$, and we can rewrite equation (4.4) as

$$f'(n) \leq n - \lfloor\sqrt{n}\rfloor - 2 + \frac{\lfloor\sqrt{n}\rfloor(\lfloor\sqrt{n}\rfloor - 1)}{2} + 3(n - \lfloor\sqrt{n}\rfloor - 1)\sqrt{n - \lfloor\sqrt{n}\rfloor - 1}.$$

Now, since $\sqrt{n} - 1 \leq \lfloor\sqrt{n}\rfloor \leq \sqrt{n}$, we have

$$\begin{aligned} f'(n) &\leq n - (\lfloor\sqrt{n}\rfloor - 1) - 1 + \frac{\lfloor\sqrt{n}\rfloor(\lfloor\sqrt{n}\rfloor - 1)}{2} + 3(n - (\lfloor\sqrt{n}\rfloor - 1))\sqrt{n - (\lfloor\sqrt{n}\rfloor - 1)} \\ &\leq \frac{3}{2}(n - \sqrt{n}) - 1 + 3(n - \sqrt{n})\sqrt{n - \sqrt{n}} \\ &\leq \frac{3}{2}(n - \sqrt{n}) - 3\sqrt{n}\sqrt{n - \sqrt{n}} + 3n\sqrt{n - \sqrt{n}}. \end{aligned} \tag{4.5}$$

We draw the readers attention to the rightmost part of equation (4.5), remarking that $3n\sqrt{n - \sqrt{n}} \leq 3n\sqrt{n}$ for $n \geq 1$. Therefore, if $\frac{3}{2}(n - \sqrt{n}) - 3\sqrt{n}\sqrt{n - \sqrt{n}} \leq 0$, then the desired result holds. For the sake of contradiction, suppose the contrary:

$$\begin{aligned}\frac{3}{2}(n-\sqrt{n})-3\sqrt{n}\sqrt{n-\sqrt{n}}>0 &\iff \frac{3}{2}(n-\sqrt{n})>3\sqrt{n}\sqrt{n-\sqrt{n}}\\ &\iff n-\sqrt{n}>2\sqrt{n}\sqrt{n-\sqrt{n}}\\ &\iff n^2-2n\sqrt{n}+n>4n(n-\sqrt{n})\\ &\iff 3n^2-2n\sqrt{n}-n<0. \end{aligned} \tag{4.6}$$

Since $n \geq 1$, we can divide equation (4.6) by $n$, obtaining $3n - 2\sqrt{n} - 1 \leq 0$. This inequality is only satisfied when $0 \leq n < 1$. However, since we are considering only $n \geq 1$, we conclude that

$$\begin{aligned}\frac{3}{2}(n-\sqrt{n})-3\sqrt{n}\sqrt{n-\sqrt{n}}&\leq 0\\ \frac{3}{2}(n-\sqrt{n})-1-3\sqrt{n}\sqrt{n-\sqrt{n}}+3n\sqrt{n-\sqrt{n}}&\leq 3n\sqrt{n}\\ f'(n)&\leq 3n\sqrt{n}.\end{aligned}$$

This completes the proof. □

### 4.2.2 Bounds for $\mathrm{str}_{\mathbf{gap}}(K_n)$

We now return to analysing the gap-strength of complete graphs $K_n$. In the previous section, we use restricted decompositions to remove certain edges from $K_n$ such that the resulting graph is gap-vertex-labelable. Here, we create a decomposition without the restrictions in the previous section.

Let $K_n$ be a complete graph of order $n$, and let $G \in K_n^{-l}$ be a graph obtained by removing $l$ edges from $K_n$ such that $G$ is gap-vertex-labelable. Let $(\pi, c_\pi)$ be a gap-vertex-labelling of $G$ and let $v_{\max}$ and $v_{\min}$ be two distinct arbitrary vertices in $G$ which received the highest and lowest labels in $\Pi_{V(G)}$, respectively. Then, for every vertex $v \in V(G)\backslash\{v_{\max}, v_{\min}\}$, either:

(i) $v_{\max}, v_{\min} \in N(v)$;

(ii) $v_{\max} \in N(v)$ and $v_{\min} \notin N(v)$; or, alternately

(iii) $v_{\max} \notin N(v)$ and $v_{\min} \in N(v)$; and, finally,

(iv) $v_{\max}, v_{\min} \notin N(v)$.

We define sets $\mathcal{I}, \mathcal{X}, \mathcal{Y}$, and $\mathcal{Z}$ as the subsets of $V(K_n)$ comprising, respectively, the vertices that satisfy cases (i), (ii), (iii) and (iv). Note that sets $\mathcal{X}$ and $\mathcal{I}$ are defined exactly as in the previous section. Furthermore, set $\mathcal{Y}$ can be seen as symmetric to set $\mathcal{X}$ in the sense that every vertex $v_x \in \mathcal{X}$ is adjacent to $v_{\max}$ (and not $v_{\min}$), whereas every $v_y \in \mathcal{Y}$, to $v_{\min}$ (and not $v_{\max}$). This (new) decomposition of $G$ is denoted by $G(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{I})$. In Figure 4.16, we present a sketch of this decomposition.

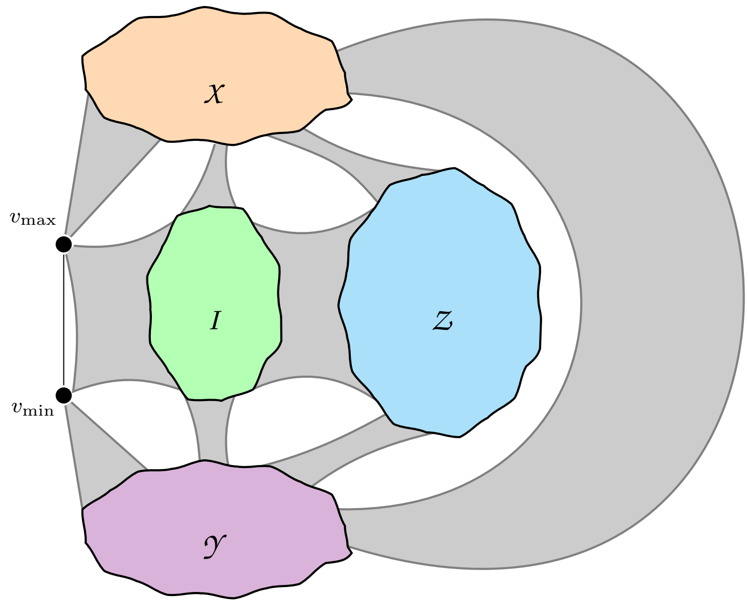



Figure 4.16: The decomposition $G(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, I)$ of graph $G$, with each subset of $V$. The gray areas indicate all edges connecting vertices in distinct sets.

Let $i = |I|$, $x = |\mathcal{X}|$, $y = |\mathcal{Y}|$ and $z = |\mathcal{Z}|$ denote the size of each set in a decomposition of $G$. Then, in $G(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, I)$, we can write the order of $G$ as $n = x + y + z + i + 2$. Also, observe that to obtain such a decomposition of $G$, it is required to remove at least: $x$ edges connecting $v_{\min}$ to vertices in $\mathcal{X}$; $y$ edges connecting $v_{\max}$ to vertices in $\mathcal{Y}$; and $2z$ edges connecting vertices in $\mathcal{Z}$ to $v_{\max}$ and $v_{\min}$. Then, we can write $l(n)$ as a function of $x, y, z$ and $i$ as follows:

$$l(n) = x + y + 2z + \mathcal{R}' + \mathcal{R}^{\mathcal{X}} + \mathcal{R}^{\mathcal{Y}} + \mathcal{R}^{\mathcal{Z}} + \mathcal{R}^{I}. \tag{4.7}$$

Similar to the previous section, in equation (4.7), $\mathcal{R}'$ denotes the number of edges removed between two distinct sets and $\mathcal{R}^S$ denotes the number of edges removed inside set $S \in \{\mathcal{X}, \ldots, I\}$ of a decomposition of $G$. Thus, the gap-strength of a complete graph $K_n$ can be determined by the following modification to the equation:

$$\mathrm{str}_{\mathrm{gap}}(K_n) = \min_{x,y,z,i \in \mathbb{Z}_{\geq 0}} \{x + y + 2z + \mathcal{R}' + \mathcal{R}^{\mathcal{X}} + \mathcal{R}^{\mathcal{Y}} + \mathcal{R}^{\mathcal{Z}} + \mathcal{R}^{I}\}. \tag{4.8}$$

It is important to remark that a decomposition of $K_n$ in which $\mathcal{Y}$ and $\mathcal{Z}$ are empty is equivalent to a restricted decomposition of $G$, for which we established an upper bound in the previous section. Also, notice that $\mathcal{R}' \geq 0$. In the context of establishing lower bounds for the gap-strength of complete graphs, we omit this value in equations herein.

Now, consider set $I$. Analogously to the restricted case, every vertex in $v_i \in I$ has vertices $v_{\max}$ and $v_{\min}$ in $N(v_i)$. This implies that every such $v_i$ has the same induced colour $c_\pi(v_i) = \pi(v_{\max}) - \pi(v_{\min})$. This, in turn, implies that no two vertices in $I$ are adjacent and, once again, $I$ is an independent set. Thus, the number of removed edges from within set $I$ is:

$$\mathcal{R}^{I} = \binom{i}{2} = \frac{i(i-1)}{2}. \tag{4.9}$$

Next, consider set $\mathcal{X}$. Every $v_x \in \mathcal{X}$ is, by definition, adjacent to $v_{\max}$, i.e. the vertex which received the largest of all labels in $\Pi_{V(G)}$. Observe that a similar reasoning can be applied here as to that of Section 4.2.1. In the restricted decomposition of $K_n$, when $x \geq 3$, we required the removal of edges within set $\mathcal{X}$ in order for the graph to be gap-vertex-labelable. This removal was done considering the same problem recursively within set $\mathcal{X}$. Here, a similar reasoning applies: we can recursively decompose the complete subgraph induced by set $\mathcal{X}$ and $v_{\max}$ into sets $\mathcal{X}', \ldots, I'$. Moreover, since no edge can be removed connecting vertices in $\mathcal{X}$ to $v_{\max}$, then sets $\mathcal{Y}'$ and $\mathcal{Z}'$ are empty. Then, in order to determine $\mathcal{R}^{\mathcal{X}}$, we consider the restricted decomposition of set $\mathcal{X}$ and, by modifying equation (4.7), we obtain

$$l(n) \geq x + y + 2z + \frac{i(i-1)}{2} + l'(x+1) + \mathcal{R}^{\mathcal{Y}} + \mathcal{R}^{\mathcal{Z}}. \tag{4.10}$$

Now, regarding set $\mathcal{Y}$, observe that in order to determine $\mathcal{R}^{\mathcal{Y}}$, we can apply a symmetric reasoning to that of set $\mathcal{X}$. Every vertex $v_y \in \mathcal{Y}$ is adjacent to $v_{\min}$, but not $v_{\max}$. Let $v'_{\max} \in \mathcal{Y}$ be a vertex such that $\pi(v'_{\max})$ is the largest of all labels in $\Pi_{\mathcal{Y}}$. Supposing a gap-vertex-labelling exists, the edges removed within $\mathcal{Y}$ are such that for every $v_y \in \mathcal{Y} + v_{\min}$, either $v'_{\max} \in N(v_y)$ or $v'_{\max} \notin N(v_y)$. By considering these cases, we create two subsets $I'$ and $\mathcal{Y}'$. Moreover, if $y \geq 3$, then the subgraph induced by $\mathcal{Y} + v_{\min}$ is also a complete graph of order $n' \geq 4$, which we can recursively decompose in a restricted manner. Hence, we conclude that

$$\mathcal{R}^{\mathcal{Y}} \geq l'(y+1). \tag{4.11}$$

It remains to consider set $\mathcal{Z}$, a set in which every vertex is not adjacent to $v_{\max}$ nor $v_{\min}$. Since we are considering that no edges have been removed connecting distinct sets in the decomposition, every vertex $v_z$ has $N(v_z) = V(G)\backslash\{v_{\max}, v_{\min}\}$. If $z \geq 4$, we have a complete subgraph induced by the vertices in $\mathcal{Z}$ alone. This subgraph also requires the removal of edges in order for $G$ to be properly labelled. Moreover, this can be done by decomposing the complete subgraph $G[\mathcal{Z}]$ into new sets $\mathcal{X}', \ldots, I'$. Then, in the context of establishing a lower bound for $\mathrm{str}_{\mathrm{gap}}(K_n)$, when $z \geq 4$, we know that

$$\mathcal{R}^{\mathcal{Z}} \geq l(z). \tag{4.12}$$

Applying these bounds to equation (4.10), we obtain the following recurrence:

$$l(n) \geq \begin{cases} 0, & \text{if } n \leq 3; \\ \min\limits_{x,y,z,i \in Z_{\geq 0}} \{x + y + \frac{i(i-1)}{2} + l'(x+1) + l'(y+1) + l(z)\}, & \text{otherwise.} \end{cases} \tag{4.13}$$

As was done in the restricted case, we designed a dynamic programming algorithm which calculates a lower bound for $l(n)$; this algorithm is presented in Algorithm 2.

**Algorithm 2** Given the order of a complete graph $K_n$ and `Restr_Solution`, computes a lower bound for $l(n)$ – the least number of edges that must be removed to create a decomposition $G(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, I)$.

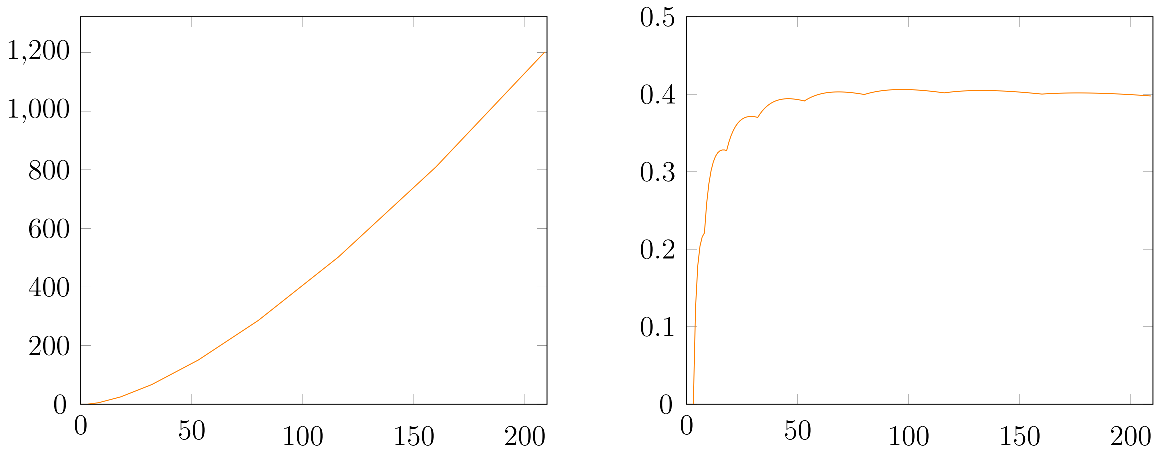
```
Gen_Solution[1], Gen_Solution[2], Gen_Solution[3] ← 0

for j ← 4 to n do
    Gen_Solution[j] ← ∞
    for i ← 0 to n − 2 do
        for z ← 0 to n − 2 − i do
            for y ← 0 to n − 2 − i − z do
                x → n − 2 − i − z − y
                Cur_Sol[j][i][z][y] ← x + y + 2z + i(i−1)/2
                if x ≥ 3 then
                    Cur_Sol[j][i][z][y] ← Cur_Sol[j][i][z][y] + Restr_Solution[x + 1]
                end if
                if y ≥ 3 then
                    Cur_Sol[j][i][z][y] ← Cur_Sol[j][i][z][y] + Restr_Solution[y + 1]
                end if
                if z ≥ 4 then
                    Cur_Sol[j][i][z][y] ← Cur_Sol[j][i][z][y] + Gen_Solution[z]
                end if
                if Cur_Sol[j][i][z][y] < Gen_Solution[j] then
                    Gen_Solution[j] ← Cur_Sol[j][i][z][y]
                end if
            end for
        end for
    end for
end for

return Gen_Solution[n]
```

Note that Algorithm 2 uses the lower bounds for $l'(n)$ calculated by Algorithm 1 from the previous section. We consider that these values are stored in the array `Restr_Solution`. We coded this algorithm in C++ and calculated the lower bound for $l(n)$ for values of $n \leq 210$. The results from the execution are presented in Figure 4.17.

We draw the readers attention to subfigure (b), in which the computed value for the lower bound of $l(n)$ is divided by $n\sqrt{n}$, as was done in the restricted decomposition. Observe that for growing values of $n$, the graph remains (relatively) constant. This graph provides evidence that $l(n)$ has asymptotic growth equal to that of $n\sqrt{n}$. With this in mind, we conjecture that $n\sqrt{n}$ is a lower bound to $l(n)$ and, consequently, the gap-strength of complete graphs.

**Conjecture 4.8.** *Let $K_n$ be a complete graph of order $n \geq 4$. Then,*

$$\mathrm{str}_{gap}(K_n) \in \Omega(n\sqrt{n}).$$




(a)

(b)

Figure 4.17: Results for: (a) $l(n)$; and (b) $\frac{l(n)}{n\sqrt{n}}$.

Thus, we conclude our preliminary study of the gap-strength of complete graphs. For future work, it would be interesting to determine the exact formula which calculates the least numbers of edges $l'(n)$ and $l(n)$ that need to be removed from complete graph $K_n$ in order to obtain gap-vertex-labelable restricted and general decompositions. Also: to investigate the gap-strength of other families of graphs which do not admit gap-vertex-labellings, such as the subclass of split graphs presented in Section 4.1. We leave these problems open for future research.

**Problem 4.9.** *Determine the gap-strength of $K_n$ as a function of $n$.*

## 4.3 An algorithm for Gvl

Until this point in our work, our studies on gap-vertex-labellings revolved around discovering families for which we know the least number $k$ of labels required to create proper gap-labellings (Chapter 3), investigating graphs which do not admit any gap-vertex-labelling (Section 4.1), or that require a modification in their structure in order for them to become gap-vertex-labelable (Section 4.2). In this final section of our work, we address a fundamental question regarding gap-vertex-labellings, proposed by Dehghan et al. [8].

**Problem 4.10** (Dehghan et al.)**.** *Does there exist a polynomial-time algorithm to determine whether a given graph admits a gap-vertex-labelling?*

In Section 4.1, we named the decision problem of determining whether a given graph $G$ admits a gap-vertex-labelling as **Gvl**. Here, we determine structural properties of gap-vertex-labellings which enable us to create an algorithm that executes in $\mathcal{O}(n!)$ to solve **Gvl**. This is the first algorithm that solves Dehghan et al.'s decision problem. Before we establish this result, we require the proof of the following lemma.

**Lemma 4.11.** *Let $G$ be a connected simple graph. Then, $G$ admits a gap-vertex-labelling $(\pi, c_\pi)$ if and only if $G$ admits a gap-vertex-labelling $(\pi', c_{\pi'})$ such that for every pair of distinct vertices $u, v \in V(G)$, $\pi'(u) \neq \pi'(v)$.*

*Proof.* Let $G = (V, E)$ be a connected simple graph of order $n$ and suppose $G$ admits a gap-vertex-labelling $(\pi, c_\pi)$. Note that if every vertex has a distinct label in $\pi$, then the sufficient condition naturally holds. Therefore, in order to prove the result, it suffices to show the necessary condition.

($\Rightarrow$) Adjust notation of $V$ as $\{v_0, \ldots, v_{n-1}\}$, such that $\pi(v_0) \leq \pi(v_1) \leq \ldots \leq \pi(v_{n-1})$. Define labelling $\pi'$ of $G$ as follows: for every vertex $v_i \in V(G)$, let $\pi'(v_i) = \pi(v_i) \cdot 2n + i$. Colouring $c_{\pi'}$ is defined as usual.

First, we prove that $\pi'$ is a labelling of $G$ such that each vertex received a distinct label. Suppose, for the sake of contradiction, that $\pi'(v_i) = \pi'(v_j)$ for two distinct vertices $v_i, v_j \in V$. Without loss of generality, we assume $i < j$.

$$\pi'(v_i) = \pi'(v_j) \Rightarrow [\pi(v_i) - \pi(v_j)] \cdot 2n = j - i$$

Since $i < j$, the right side of the equation is always larger than 0. However, we know that $\pi(v_i) \leq \pi(v_j)$ by the defined notation, which implies that the left side of the equation is always a nonnegative number. Therefore, there are no values for $i$ and $j$ which satisfy the equation, and we conclude that $\pi'$ is a labelling of $G$ in which every vertex is assigned a distinct label. Furthermore, it is important to remark that $\pi'$ is defined as an *order preserving* function of $\pi$. This means that if $\pi'(v_i) < \pi'(v_j)$ for two vertices $v_i, v_j \in V(G)$, then $\pi(v_i) \leq \pi(v_j)$ in the first gap-vertex-labelling $(\pi, c_\pi)$.

Next, we prove that colouring $c_{\pi'}$ is a proper vertex-colouring of $G$ by contradiction. Suppose there are two adjacent vertices $v_i, v_j \in V$ such that $c_{\pi'}(v_i) = c_{\pi'}(v_j)$. Since the colour of a vertex is induced differently for vertices $v$ with $d(v) = 1$ and $d(v) \geq 2$, we must address two cases: (i) if $d(v_i) \geq 2$ and $d(v_j) \geq 2$; and (ii) if $d(v_i) \geq 2$ and $d(v_j) = 1$. The case $d(v_i) = d(v_j) = 1$ implies that $G \cong K_2$, which can be inspected.

*Case (i). $d(v_i) \geq 2$ and $d(v_j) \geq 2$.*
Let $v_a$ and $v_b$ be the neighbours of $v_i$ such that $c_{\pi'}(v_i) = \pi'(v_a) - \pi'(v_b)$, and $v_x$ and $v_y$, the neighbours of $v_j$ such that $c_{\pi'}(v_j) = \pi'(v_x) - \pi'(v_y)$. Note that not necessarily $a \neq x$, $a \neq y$ or $b \neq y$. We express the equality as

$$\begin{aligned} c_{\pi'}(v_i) = c_{\pi'}(v_j) &\Rightarrow \pi'(v_a) - \pi'(v_b) = \pi'(v_x) - \pi'(v_y) \\ &\Rightarrow (\pi(v_a) - \pi(v_b) - \pi(v_x) + \pi(v_y)) \cdot 2n = x - y - a + b. \end{aligned} \tag{4.14}$$

Since $1 \leq a, b, x, y \leq n$, we have $|x - y - a + b| < 2n$. From the left side of equation (4.14), we consider two subcases: if $|\pi(v_a) - \pi(v_b) - \pi(v_x) + \pi(v_y)| \geq 1$; and if $\pi(v_a) - \pi(v_b) - \pi(v_x) + \pi(v_y) = 0$. In the former, we have

$$|(\pi(v_a) - \pi(v_b) - \pi(v_x) + \pi(v_y))| \geq 1 \Rightarrow |(\pi(v_a) - \pi(v_b) - \pi(v_x) + \pi(v_y)) \cdot 2n| \geq 2n.$$

Since there are no values for $a, b, x, y$ for which $|x - y - a + b| \geq 2n$, this case cannot be satisfied. Therefore, the equality can only hold in the latter. However, this implies

$$\pi(v_a) - \pi(v_b) - \pi(v_x) + \pi(v_y) = 0 \Rightarrow \pi(v_a) - \pi(v_b) = \pi(v_x) - \pi(v_y). \tag{4.15}$$

Since $\pi'$ is order preserving, if $v_a$ and $v_b$ are the vertices that define colour $c_{\pi'}(v_i)$, then $c_\pi(v_i)$ is computed by $\pi(v_a) - \pi(v_b)$. An analogous reasoning holds for $v_j$. Then, we have $\pi(v_a) - \pi(v_b) = c_\pi(v_i)$ and $\pi(v_x) - \pi(v_y) = c_\pi(v_j)$, implying that $c_\pi(v_i) = c_\pi(v_j)$ by equation (4.15). This contradicts the fact that $(\pi, c_\pi)$ is a gap-vertex-labelling of $G$, and we conclude that there are no such vertices $v_i$ and $v_j$ with the same induced colour.

*Case (ii). $d(v_i) \geq 2$ and $d(v_j) = 1$.*
Once again, let: $v_a$ and $v_b$ be the neighbours of $v_i$ such that $c_{\pi'}(v_i) = \pi'(v_a) - \pi'(v_b)$; and, since $d(v_j) = 1$ and $v_j$ is adjacent to $v_i$, $v_j$ has its colour induced by $c_{\pi'}(v_j) = \pi'(v_i)$.

$$\begin{aligned} c_{\pi'}(v_i) = c_{\pi'}(v_j) &\Rightarrow \pi'(v_a) - \pi'(v_b) = \pi'(v_i) \\ &\Rightarrow (\pi(v_a) - \pi(v_b) - \pi(v_i)) \cdot 2n = i - a + b. \end{aligned} \tag{4.16}$$

Following the same line of reasoning as Case (i), notice that the right side of equation (4.16) is strictly smaller than $2n$. Now, if $|\pi(v_a) - \pi(v_b) - \pi(v_i)| > 1$, then the left side is strictly larger than $2n$ and, thus, the equation cannot be satisfied. This implies that the equation only holds when $\pi(v_a) - \pi(v_b) - \pi(v_i) = 0$. Once again, given that $\pi'$ is order preserving, we have

$$\pi(v_a) - \pi(v_b) = \pi(v_i) \Rightarrow c_\pi(v_i) = c_\pi(v_j).$$

This is a contradiction since $c_\pi$ is a proper vertex-colouring of $G$. Since all cases have been considered, we conclude that there are no two adjacent vertices $v_i$ and $v_j$ with $c_{\pi'}(v_i) = c_{\pi'}(v_j)$. Consequently, $(\pi', c_{\pi'})$ is a gap-vertex-labelling of $G$ in which every vertex receives a distinct label. This completes the proof. □

Observe that with Lemma 4.11 established, we can safely assume that if a graph admits a gap-vertex-labelling, then there are exactly two vertices which received the maximum and minimum labels. Moreover, it allows us to also assume that all labels are distinct. Thus, we are able to prove another (stronger) result about graphs which admit gap-vertex-labellings.

**Lemma 4.12.** *Let $G$ be a connected simple graph. Then, $G$ admits a gap-vertex-labelling $(\pi, c_\pi)$ if and only if $G$ admits a gap-vertex-labelling $(\pi', c_{\pi'})$ in which all vertex labels are distinct powers of two.*

*Proof.* Let $G$ be a connected simple graph of order $n$, and suppose $G$ admits a gap-vertex-labelling $(\pi, c_\pi)$. By Lemma 4.11, we can safely assume that $\pi(u) \neq \pi(v)$ for every pair of distinct vertices $u, v \in V(G)$. As was the case in the previous proof, the sufficient condition naturally holds: if $G$ admits a gap-vertex-labelling in which all vertex-labels are distinct powers of two, then $G$ admits a gap-vertex-labelling. Therefore, in order to establish the result, it remains to prove the necessary condition.

($\Rightarrow$) First, adjust notation of $V(G)$ as $v_0, \ldots, v_{n-1}$ such that $\pi(v_0) < \ldots < \pi(v_{n-1})$. Define a new labelling $\pi'$ of $G$ as follows. For every $v \in V(G)$, let $\pi'(v) = 2^{\pi(v)}$. Define colouring $c_{\pi'}$ as usual. Once again, note that $\pi'$ is order preserving. In order to prove that $(\pi', c_{\pi'})$ is a gap-vertex-labelling of $G$, we show that induced colouring $c_{\pi'}$ is a proper colouring of $G$.

Suppose there are two adjacent vertices $v_i, v_j \in V(G)$ such that $c_{\pi'}(v_i) = c_{\pi'}(v_j)$. Similar to the proof of Lemma 4.11, we consider two cases depending on the degrees of $v_i$ and $v_j$: (i) if $d(v_i) \geq 2$ and $d(v_j) \geq 2$; and (ii) if $d(v_i) \geq 2$ and $d(v_j) = 1$. Once again, case $d(v_i) = d(v_j) = 1$ implies that $G \cong K_2$, which can be inspected.

*Case (i). $d(v_i) \geq 2$ and $d(v_j) \geq 2$.*
Let $v_a, v_b \in N(v_i)$ such that $c_{\pi'}(v_i) = \pi'(v_a) - \pi'(v_b)$ and $v_x, v_y \in N(v_j)$ such that $c_{\pi'}(v_j) = \pi'(v_x) - \pi'(v_y)$. Then, we have

$$\begin{aligned} c_{\pi'}(v_i) = c_{\pi'}(v_j) &\Rightarrow \pi'(v_a) - \pi'(v_b) = \pi'(v_x) - \pi'(v_y) \\ &\Rightarrow 2^{\pi(v_a)} - 2^{\pi(v_b)} = 2^{\pi(v_x)} - 2^{\pi(v_y)}. \end{aligned}$$

Without loss of generality, consider $\pi(v_b) \leq \pi(v_y)$ (this assumption can be made since considering the opposite is equivalent to exchange the left and right sides of the equation, and the same result follows). Dividing the equation by $2^{\pi(v_b)}$, which is, by hypothesis, the smallest among all labels, we obtain

$$2^{\pi(v_a)-\pi(v_b)} - 1 = 2^{\pi(v_x)-\pi(v_b)} - 2^{\pi(v_y)-\pi(v_b)}. \tag{4.17}$$

Since $\pi(v_a) > \pi(v_b)$ and $\pi(v_b) \leq \pi(v_y) < \pi(v_x)$, it follows that: $2^{\pi(v_a)-\pi(v_b)} > 1$; $2^{\pi(v_x)-\pi(v_b)} > 1$; and $2^{\pi(v_y)-\pi(v_b)} \geq 1$. This implies that the left side of equation (4.17) is always an odd number and, therefore, the equation can only be satisfied if $\pi(v_y) = \pi(v_b)$. This, in turn, implies that $v_y = v_b$ since all labels are distinct, and the equation is reduced to

$$2^{\pi(v_a)-\pi(v_b)} - 1 = 2^{\pi(v_x)-\pi(v_b)} - 1,$$

which can only be satisfied if $\pi(v_a) = \pi(v_x)$ and, consequently, $v_a = v_x$. But if this is the case, then $v_i$ and $v_j$ have their respective colours induced by the labels of the same two vertices, as illustrated in Figure 4.18.

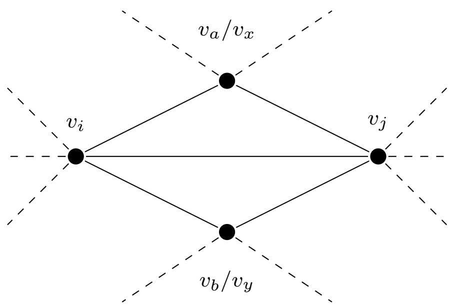



Figure 4.18: Adjacent vertices $v_i$ and $v_j$ of graph $G$, and vertices $v_a = v_x$ and $v_b = v_y$.

Since $\pi'$ is order preserving, if $\pi'(v_a)$ and $\pi'(v_b)$ define colours $c_{\pi'}(v_i)$ and $c_{\pi'}(v_j)$, it follows that $c_\pi(v_i)$ and $c_\pi(v_j)$ are also computed as $\pi(v_a) - \pi(v_b)$. This, however, implies that $c_\pi(v_i) = c_\pi(v_j)$, which is a contradiction since $(\pi, c_\pi)$ is a gap-vertex-labelling of $G$. We conclude that there are no such vertices $v_i, v_j$ in $G$.

*Case (ii). $d(v_i) \geq 2$ and $d(v_j) = 1$.*
Let $v_a$ and $v_b$ be the neighbours of $v_i$ such that $c_{\pi'}(v_i) = \pi'(v_a) - \pi'(v_b)$. Since $d(v_j) = 1$, then $c_{\pi'}(v_j) = \pi'(v_i)$. Then, we have

$$\begin{aligned} c_{\pi'}(v_i) = c_{\pi'}(v_j) &\Rightarrow \pi'(v_a) - \pi'(v_b) = \pi'(v_i) \\ &\Rightarrow 2^{\pi(v_a)} - 2^{\pi(v_b)} = 2^{\pi(v_i)}. \end{aligned}$$

Similar to Case (i), we consider $\pi(v_b) \leq \pi(v_i)$ without loss of generality. Dividing the equation by $2^{\pi(v_b)}$, we obtain

$$2^{\pi(v_a)-\pi(v_b)} - 1 = 2^{\pi(v_i)-\pi(v_b)}. \tag{4.18}$$

Observe that equation (4.18) cannot be satisfied when $\pi(v_i) > \pi(v_b)$ since the left side is always an odd number. Thus, $\pi(v_i) = \pi(v_b)$ and, consequently, $v_i = v_b$. Note, however, that this implies that $c_{\pi'}(v_i) = \pi(v_i)$. Since all labels are distinct and $v_i \notin N(v_i)$, this case is also impossible. We conclude that $(\pi', c_{\pi'})$ is a gap-vertex-labelling of $G$ in which every label is a distinct power of two. □

Note that the gap-vertex-labelling created in the proof of Lemma 4.12 has no bound for the size of the label set. In the next theorem, we establish the first bound for the number of labels required to properly label the graph.

**Theorem 4.13.** *If $G$ is a gap-vertex-labelable graph, then $\chi^g_V(G) \leq 2^{n-1}$.*

*Proof.* Let $G = (V, E)$ be a simple graph and suppose $G$ admits a gap-vertex-labelling $(\pi, c_\pi)$. By Lemma 4.12, we can safely assume that $\pi(v)$ is a distinct power of two for every $v \in V$. Adjust notation of the vertices of $G$ such that for $v_0, v_1, \ldots, v_{n-1}$ we have $\pi(v_0) < \pi(v_1) < \ldots < \pi(v_{n-1})$.

Define a labelling $\pi'$ of $G$ as follows. For every $v_i \in V(G)$, let $\pi'(v_i) = 2^i$. Define colouring $c_{\pi'}$ as usual. We remark that $\pi'$ is defined from the ordering of vertices obtained by $\pi$. Evidently, the largest label value is $2^{n-1}$, which was assigned to vertex $v_{n-1}$. Therefore, in order to prove the result, it suffices to prove that $(\pi', c_{\pi'})$ is a gap-vertex-labelling of $G$.

We start by considering vertices $v_i \in V(G)$ with $d(v_i) = 1$, which are adjacent to vertices $v_j$ with $d(v_j) \geq 2$. Suppose, for the sake of contradiction, that $c_{\pi'}(v_i) = c_{\pi'}(v_j)$, and let $v_a$ and $v_b$ be the vertices in $N(v_j)$ with the largest and smallest labels in $\Pi_{N(v_j)}$, that is, $c_{\pi'}(v_j) = \pi'(v_a) - \pi'(v_b)$. Then, we have

$$\begin{aligned} c_{\pi'}(v_i) = c_{\pi'}(v_j) &\Rightarrow \pi'(v_j) = \pi'(v_a) - \pi'(v_b) \\ &\Rightarrow 2^j = 2^a - 2^b. \end{aligned}$$

Similar to the reasoning in Lemma 4.12, this case can only be satisfied if $j = b$ and $a = b + 1$. Consequently, $c_{\pi'}(v_j)$ is induced by its own label $\pi'(v_j)$. Since all labels are distinct, this case is impossible, and we conclude that there are no two such vertices.

It remains to consider the case of vertices $v_i$ and $v_j$ both having degrees $d(v_i) \geq 2$ and $d(v_j) \geq 2$. Then, similar to the proof of Lemma 4.12, we are able to conclude that if $v_i$ and $v_j$ have conflicting induced colours in $c_{\pi'}$, then their induced colours in the $c_\pi$ are also the same. Since $(\pi, c_\pi)$ is a gap-vertex-labelling of $G$, this case is impossible, and we conclude that no such vertices $v_i$ and $v_j$ exists.

Having exhausted all cases, we conclude that if a graph $G$ admits a gap-vertex-labelling, it is sufficient to use label set $\{1, 2, 4, \ldots, 2^{n-1}\}$. Therefore, $\chi^{\mathrm{g}}_{V}(G) \leq 2^{n-1}$. □

We remark that the bound in Theorem 4.13 is tight since $\chi^{\mathrm{g}}_{V}(K_3) = 4 = 2^{3-1}$. Although the labellings created in the proofs of Lemma 4.11 and 4.12 had no bound for the size of the label set, with Theorem 4.13 established, we can now design a factorial-time algorithm to decide whether a graph $G$ admits a gap-vertex-labelling. This algorithm consists of assigning every possible combination of powers of two, from $2^0$ to $2^{n-1}$, to the vertices of $G$. For each assignment, we calculate the induced colours of the vertices and verify if there are any conflicting vertices. Given that determining the induced colour of a vertex and verifying its adjacencies (for conflicting colours) can be done in polynomial time, the following corollary holds.

**Corollary 4.14.** **Gvl** *can be solved in* $\mathcal{O}(n!)$ *time.*

# Chapter 5

# Conclusions

Graph Colourings and, in particular, Proper Graph Labellings are important and quite challenging fields of study in Theoretical Computer Science. In fact, several decision problems in this area have been proved to be NP-complete. In our work, we study the edge and vertex variants of proper gap-labellings. This type of labelling concerns the assignment of labels to some elements of a graph so as to induce a proper vertex-colouring, by using the largest gap among labels from a set of labelled elements.

Initially, we study the edge-gap and the vertex-gap numbers of some families of graphs. These are the least $k \in \mathbb{N}$ for which a graph admits a gap-$[k]$-edge-labelling and a gap-$[k]$-vertex-labelling, respectively. Our results for these parameters are compiled in Table 5.1. Regarding the edge-gap and vertex-gap numbers, we leave the following problems open for future research.

First, consider unicyclic graphs with even cycles. For these graphs, we know that $\chi^{\mathrm{g}}_{E}(G) \in \{2, 3\}$ as established by Brandt et al. [4]. In Chapter 2, we show graphs that admit a gap-[2]-edge-labelling and, on the other hand, we also know of unicyclic graphs for which no gap-[2]-vertex-labelling exists. These examples are presented in Figure 2.21. In this context, it would be interesting to characterize which graphs $G$ in this class have $\chi^{\mathrm{g}}_{E}(G) = 2$.

**Problem 2.6.** *Determine the edge-gap number for unicyclic graphs with even cycles.*

In Chapter 3, we establish the vertex-gap number for all unicyclic graphs, regardless of parity. Unicyclic graphs are a subfamily of Cacti graphs. *Cacti* are connected simple graphs for which any two cycles have at most one vertex in common. Therefore, it seems that the labellings of unicyclic graphs could be used to establish the parameter for Cacti.

**Problem 3.16.** *Is it possible to extend our gap-$[\chi(G)]$-vertex-labelling of unicyclic graphs $G$ to the family of Cactus graphs?*

Still for the vertex version of proper gap-labellings, we recall that we did not determine $\chi^{\mathrm{g}}_{V}$ for Generalised Second Blanuša Snarks due to time constraints. However, we believe that the vertex-gap number for this family can be determined adjusting the labelling created for the Generalised First Blanuša family.

**Problem 3.25.** *Determine the vertex-gap number for the family of Generalised Second Blanuša Snarks.*

| Class | Edge-gap number | Vertex-gap number | Theorems |
|---|---|---|---|
| Cycles | $\chi^{g}_{E}(C_n) = \chi^{g}_{V}(C_n) \begin{cases} 4, & \text{if } n = 3; \\ 2, & \text{if } n \equiv 0 \pmod 4; \\ 3, & \text{otherwise.} \end{cases}$ | | 2.2 and 3.10 |
| Crowns | $\chi^{g}_{E}(R_n) = \chi^{g}_{V}(R_n) = \chi(R_n)$. | | 2.3 and 3.12 |
| Wheels | $\chi^{g}_{E}(W_n) = \begin{cases} 4, & \text{if } n = 4; \\ \chi(W_n), & \text{otherwise.} \end{cases}$ | $\chi^{g}_{V}(W_n) = \begin{cases} 3, & \text{if } n \geq 8 \text{ and even;} \\ 4, & \text{if } n \geq 5 \text{ and odd.} \end{cases}$ | 2.4 and 3.13 |
| Unicyclic graphs | $\chi^{g}_{E}(G) = 3$, if $p$ is odd.<br>$\chi^{g}_{E}(G) \in \{2, 3\}$, otherwise [4]. | $\chi^{g}_{V}(G) = \begin{cases} 2, & \text{if } p \text{ is even and } G \ncong C_n, n \equiv 2 \pmod 4; \\ 3, & \text{otherwise.} \end{cases}$ | 2.5 and 3.15 |
| First Blanuša Snarks | $\chi^{g}_{E}(B^1_0) = \chi^{g}_{V}(B^1_0) = 3$ | | 2.7 and 3.24 |
| Second Blanuša Snarks | *open* | $\chi^{g}_{V}(B^2_i) = 3$ | 2.8 |
| Flower Snarks | $\chi^{g}_{E}(J_l) = \chi^{g}_{V}(J_l) = 3.$ | | 2.9 and 3.26 |
| Goldberg Snarks | $\chi^{g}_{E}(G_l) = \chi^{g}_{V}(G_l) = 3.$ | | 2.10 and 3.27 |
| Twisted Goldberg Snarks | $\chi^{g}_{E}(TG_l) = \chi^{g}_{V}(TG_l) = 3$ | | 2.10 and 3.28 |

Table 5.1: Results for the edge-gap and vertex-gap numbers for classes of graphs.

In particular, for gap-$[k]$-vertex-labellings, we also addressed one of the problems posed by A. Dehghan et al. [8] in 2013, namely that of finding the algorithmic complexity of deciding whether a cubic bipartite graph admits a gap-[2]-vertex-labelling. By considering subcubic bipartite graphs $G$, we proved that it is $\mathsf{NP}$-complete to decide whether $G$ admits a gap-[2]-vertex-labelling. This result is presented in Theorem 3.1. For cubic bipartite hamiltonian graphs, which we refer to as CBH-graphs, we devised several techniques that properly label subfamilies of these graphs using only two labels (see Section 3.3.5). The following theorem states our results for this family of graphs.

**Theorem 5.1.** *Let $G$ be a CBH-graph. Then, $\chi^{g}_{V}(G) = 2$ if:*

*(i)* $n \equiv 0 \pmod 4$*;*

*(ii)* $n \equiv 2 \pmod 4$ *and there exists a chord* $e \in E(G)$ *such that* $r(e) \equiv 3 \pmod 4$*;*

*(iii)* $n \equiv 2 \pmod 4$ *and* $n = \beta(r+1) + \alpha(r-1)$*, for* $\alpha, \beta$ *odd;*

*(iv)* $n \equiv 2 \pmod 4$ *and* $r(e) \equiv 3 \pmod 6$ *for every chord* $e \in E(G)$*.* □

Additionally, our computational experiments on CBH-graphs with homogeneous chords, with up to 1002 vertices, have shown that the only graph in this class without a gap-[2]-vertex-labelling is the Heawood Graph. Therefore, we pose the following conjecture.

**Conjecture 3.22.** *Let $G$ be a CBH-graph not isomorphic to $C_{14,reach=5}$. Then, $\chi_V^g(G) = 2$.*

One last question regarding the vertex-gap number of graphs is raised. In 2016, Brandt et al. [4] established that the edge-gap number of arbitrary graphs is bound by $\chi(G)$ and $\chi(G)+1$ for every graph that is not isomorphic to star $S_n$, $n \geq 2$. In the vertex version, all classes of graphs we addressed also seem to have their vertex-gap number closely related to the chromatic number of the graph. In fact, the difference between $\chi_V^g(G)$ and $\chi(G)$ for every family we approached differs in at most one. This provides evidence that it may be possible to obtain a similar result to that of the edge-gap number, that is, the vertex-gap number of arbitrary graphs $G$ may be closely bounded by a function of $\chi(G)$.

**Problem 3.29.** *Let $G$ be an arbitrary graph and $f$, a function. Is it possible to establish $f$ such that $\chi_V^g(G) \leq f(\chi(G))$?*

The second part of our work addresses the algorithmic complexity of decision problems associated with gap-$[k]$-vertex-labellings of arbitrary graphs. In the edge version, we know that deciding whether a graph $G$ admits a gap-$[k]$-edge-labelling, for some $k \in \mathbb{N}$, can be done in polynomial time since one needs only check every connected component of $G$ and verify that none of them are isomorphic to $K_2$. The complexity of the vertex-version, however, remains unknown. In Chapter 4, we show two infinite families of graphs for which no gap-$[k]$-vertex-labelling exists: complete graphs $K_n$, $n \geq 4$, and a subfamily of split graphs. Based on these results, we conjecture that cliques of size $n \geq 4$ are at the heart of every graph that does not admit a gap-$[k]$-vertex-labelling. Additionally, in the same chapter, we prove structural properties which allow us to create an $\mathcal{O}(n!)$-time algorithm that decides whether an arbitrary graph admits a gap-vertex-labelling. As a result of these properties, we establish a tight upper bound for the vertex-gap number of arbitrary graphs. In addition, our work in Chapter 3 also establishes a tight lower bound for this parameter. Combining these results, which are presented separately in Theorems 3.7 and 4.13, we have the following corollary.

**Corollary 5.2.** *Let $G$ be a gap-vertex-labelable graph. Then, $\chi(G) \leq \chi_V^g(G) \leq 2^{n-1}$ unless $G \cong S_n$, $n \geq 2$, in which case $\chi_V^g(G) = 1 = \chi(G) - 1$.* □

We also define a new parameter associated with gap-vertex-labellings: the gap-strength of graphs. Denoted by $\mathrm{str}_{\mathrm{gap}}(G)$, the gap-strength of a graph $G$ is the least number $l$ of edges that need to be removed from $G$ so as to obtain a new graph that is gap-vertex-labelable. For complete graphs, we prove that $\mathrm{str}_{\mathrm{gap}}(K_n) \in \mathcal{O}(n\sqrt{n})$ and provide evidence that $\mathrm{str}_{\mathrm{gap}}(K_n) \in \Omega(n\sqrt{n})$. The research on the gap-strength of graphs is still in its early stages. An interesting continuity would be to formally establish the lower bound for $\mathrm{str}_{\mathrm{gap}}(K_n)$ and, thus, answer the following conjecture.

**Conjecture 5.3.** *Let $K_n$ be the complete graph of order $n$. Then, $\mathrm{str}_{gap}(K_n) \in \Theta(n\sqrt{n})$.*

# Bibliography

[1] B. D. Acharya, S. Arumugam, and A. Rosa. *Labelings of Discrete Structures and its Applications*. Narosa Publishing House, 2008.

[2] K. Appel and W. Haken. Every planar map is four colorable. *Bull. Am. Math. Soc.*, 82:711–712, 1976.

[3] D. Blanuša. Problem cetiriju boja (Le problème des quatre couleurs). *Glasnik Mat.-Fiz. Astronom. Ser.*, II:31–42, 1946.

[4] A. Brandt, B. Moran, K. Nepal, F. Pfender, and D. Sigler. Local gap colorings from edge labelings. *Australasian Journal of Combinatorics*, 65(3):200 – 211, 2016.

[5] R. L. Brooks. On colouring the nodes of a network. *Mathematical Proceedings of the Cambridge Philosophical Society*, 37:194–197, 1941.

[6] G. Chartrand, M. Jacobson, J. Lehel, O. Oellerman, S. Ruiz, and F. Saba. Irregular networks. *Proceedings of the 250th Anniversary Conference on Graph Theory, Frt Wayne, Indiana*, 1986.

[7] A. Dehghan. On strongly planar not-all-equal 3SAT. *Journal of Combinatorial Optimization*, 32(3):721 – 724, 10 2016.

[8] A. Dehghan, M. Sadeghi, and A. Ahadi. Algorithmic complexity of proper labeling problems. *Theoretical Computer Science*, 495:25 – 36, 2013.

[9] P. Feofillof. Complexidade computacional e problemas NP-completos. `http://www.ime.usp.br/pf/analise_de_algoritmos/aulas/NPcompleto2.html`. 25-08-2016.

[10] S. Földes and P. L. Hammer. Split Graphs. *Proceedings of the Eighth Southeastern Conference on Combinatorics, Graph Theory and Computing*, pages 311–315, 1977.

[11] J. A. Gallian. A dynamic survey of graph labeling. *The Electronic Journal of Combinatorics*, (#DS6), 2016.

[12] M. Gardner. Mathematical Games: Snarks, Boojums and Other Conjectures Related to the Four-Color-Map Theorem. *Scientific American*, 234:126 – 130, 1976.

[13] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[14] M. Ghebleh. The circular chromatic index of Goldberg Snarks. *Discrete Mathematics*, pages 3220–3225, 2007.

[15] M. K. Goldberg. Construction of class 2 graphs with maximum vertex degree 3. *Journal of Combinatorial Theory, Series B*, 31:282–291, 1981.

[16] S. W. Golomb. How to number a graph. *Graph Theory and Computing Academic Press, New York*, pages 23 – 27, 1972.

[17] R. Isaacs. Infinite families of non-trivial trivalent graphs which are not Tait colorable. *Amer. Math. Monthly*, 82:221–239, 1975.

[18] M. Karoński, T. Łuczak, and A. Thomason. Edge weights and vertex colours. *Journal of Combinatorial Theory, Series B*, 91(1):151 – 157, 2004.

[19] D. E. Knuth. *The Art of Computer Programming, Volume 3: (2Nd Ed.) Sorting and Searching*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1998.

[20] S. C. López and F. A. Muntaner-Batle. *Graceful, Harmonious and Magic Type Labelings - Relations and Techniques*. Springer Briefs in Mathematics. Springer International Publishing, 2017.

[21] A. M. Marr and W. D. Wallis. *Magic graphs*. Birkhäuser Basel, 2013.

[22] J. Petersen. Sur le théoreme de Tait. *L'Intermédiaire des Mathématiciens*, pages 225–227, 1898.

[23] N. Robertson, D. Sanders, P. Seymour, and R. Thomas. A new proof of the four-colour theorem. *Electron. Res. Announc. Amer. Math. Soc.*, 2:17–25, 1996.

[24] A. Rosa. On certain valuations of the vertices of a graph. *Theory of Graphs (International Symposium, Rome*, pages 349 – 355, 1967.

[25] R. Scheidweiler and E. Triesch. New estimates for the gap chromatic number. *SIAM Journal of Discrete Mathematics*, 328:42 – 43, 2014.

[26] R. Scheidweiler and E. Triesch. Gap-neighbour-distinguishing colouring. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 94:205 – 214, 2015.

[27] M. A. Tahraoui, E. Duchêne, and H. Kheddouci. Gap vertex-distinguishing edge colorings of graphs. *Discrete Mathematics*, 312(20):3011 – 3025, 2012.

[28] P. G. Tait. On the Colouring of Maps. *Proc. Royal Soc. Edinburgh*, 10:501–503, 1880.

[29] P. G. Tait. Remarks on the previous Communication. *Proc. Royal Soc. Edinburgh*, 10:729, 1880.

[30] J. J. Watkins. On the construction of snarks. *Ars Combinatoria*, 16:111–124, 1983.

[31] J. J. Watkins. Snarks. *Annals of the New York Academy of Sciences*, 576:606–622, 1989.

[32] P. Zhang. *Color-Induced Graph Colorings*. Springer Briefs in Mathematics. Springer International Publishing, 2015.