FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

ASSIGNMENT OF MASTER'S THESIS

Title:	Sum Graphs
Student:	Bc. Šimon Schierreich
Supervisor:	RNDr. Dušan Knop, Ph.D.
Study Programme:	Informatics
Study Branch:	Computer Science
Department:	Department of Theoretical Computer Science
Validity:	Until the end of winter semester 2021/22

Instructions

A graph G is a sum graph if there exists a labeling f of its vertices, i.e., a function assigning a positive integer to every vertex, such that there is an edge {u,v} if and only if there exists a vertex w with f(u)+f(v)=f(w). Clearly, no connected graph is sum graph. Thus, for connected graph G we study the least number of isolated vertices one has to add to G so that the new graph is a sum graph. Furthermore, the same applies to graph classes, that is, we ask what is the least number of vertices one as to add to a graph coming from a particular graph class so that it is a sum graph; this is know as the sum number. This question has been answered for very simple graph classes such as paths, cycles, wheels, or trees.

The aim of the thesis is to survey the known results on this topic and to obtain new results by e.g. showing a new (preferably simpler) labeling scheme for graph classes with known sum number or to find sum number of other graph classes.

References

Will be provided by the supervisor.

doc. Ing. Jan Janoušek, Ph.D.
Head of Department

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
Dean

Prague April 24, 2020

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master's thesis

Sum Graphs

Bc. Šimon Schierreich

Department of Theoretical Computer Science
Supervisor: RNDr. Dušan Knop, Ph.D.

July 30, 2020

Acknowledgements

I would like to use the following lines to express my sincerest thanks and appreciation to my supervisor Dr. Dušan Knop. Not only for his enthusiasm, advice, invaluable support, guidance, criticism, and encouragement but also for hours spent with a cup of tea and many delicate problems to solve. You are a scientist in the truest sense of the word, and I will always be inspired by you.

Last but not least, I would like to thank my family, which was my support not only when I was writing this thesis but also during my entire studies.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46 (6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the "Work"), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity. However, all persons that makes use of the above license shall be obliged to grant a license at least in the same scope as defined above with respect to each and every work that is created (wholly or in part) based on the Work, by modifying the Work, by combining the Work with another work, by including the Work in a collection of works or by adapting the Work (including translation), and at the same time make available the source code of such work at least in a way and scope that are comparable to the way and scope in which the source code of the Work is made available.

In Prague on July 30, 2020 . . . .

Citation of this thesis

Schierreich, Šimon. *Sum Graphs.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2020.

Abstrakt

Neorientovaný graf $G = (V, E)$ nazveme součtovým grafem, pokud existuje injektivní funkce $\sigma\colon V \to \mathbb{N}$ přiřazující vrcholům kladná celá čísla tak, že pro každé dva vrcholy $u, v \in V$ platí, že jsou v grafu G spojeny hranou právě tehdy, když existuje třetí vrchol $w \in V$ takový, že $\sigma(w) = \sigma(u) + \sigma(v)$.

Není těžké nahlédnout, že žádný souvislý graf nemůže být součtovým grafem, jelikož vrchol s největším přiřazeným číslem nemůže mít žádného souseda. Proto se snažíme najít součtové číslo grafu G, což je minimální počet izolovaných vrcholů, které je třeba ke grafu G připojit, abychom z něj součtový graf udělali.

Zmíněný problém studujeme v celé jeho šíři. Nejdříve se věnujeme formálnímu zavedení všech pojmů a studiu vlastností součtových grafů. Pro vybrané třídy grafů pak uvádíme deterministické algoritmy, které z grafu G vytvoří součtový graf za použití minimálního možného počtu přidaných vrcholů. V poslední části práce prezentujeme náš exaktní exponenciální algoritmus, který dokáže najít součtové číslo libovolného grafu.

Klíčová slova teorie grafů, značkování grafů, součtové grafy, součtové číslo

Abstract

A simple undirected graph $G = (V, E)$ is said to be a sum graph if there is an injective function $\sigma\colon V \to \mathbb{N}$ such that for every $u, v \in V$ there is an edge $\{u, v\}$ in E if and only if there exists a vertex $w \in V$ such that $\sigma(w) = \sigma(u) + \sigma(v)$.

It is easy to see that none connected graph is a sum graph since the vertex v with the largest label is not adjacent to any other vertex. Therefore, we investigate the sum number of a graph G, which is the minimum number of isolated vertices we must add to G to obtain a sum graph.

We study this problem in its entirety. We provide complete definitions of sum graphs and their properties. For some graph families, we investigate their sum numbers and provide exact labeling algorithms to find an optimal labeling. In the last part of this thesis, we present exact exponential algorithms that find a sum number for an arbitrary graph.

Keywords graph theory, graph labeling, sum graphs, sum number

Contents

List of Figures

List of Tables

Introduction

Graphs and graph theory have become in the last century one of the main fields of study for not only mathematicians and computer scientists but play a significant role even in chemistry, physics, biology, social sciences, economics, and other traditional fields. A graph, as a natural model is closely related to many real-life situations, such as finding a shortest path, designing computer chips, telecommunications, scheduling, and many others.

In this thesis, we deal with the idea of sum graphs. The sum graph is a concept linking together subsets of positive integers and graphs associated with this subset. Frank Harary introduced the idea in his talk at the Nineteenth Southeastern Conference on Combinatorics at Baton Rouge in 1988 [1].

Let S be a finite subset of the set $\{1, 2, \ldots\}$ of positive integers. Harary [2] defined the sum graph of the set S as $G^+(S) = (V, E)$, where $V = S$ and $E = \{\{x, y\} \mid x + y \in S\}$. Such graphs have very beneficial property because they do not have to be stored in computer's memory as both sets of vertices and edges, but it is sufficient to store only the set of vertices, since the edges are implicitly encoded in vertex labels and presence of an edge $\{x, y\}$ in $E(G^+(S))$ can be easily calculated. In 2006 Slamet, Sugeng, and Miller [3] presented yet another application of sum graphs in the distribution of secret information among a group of participants.

The attempt at various alternative representations of graphs has been part of graph theory since its inception. Let us recall at this point at least a few of them. One of the oldest ways of graph representation are *interval graphs* [4, 5]. "A graph is called an interval graph if each of its vertices can be associated with an interval on a real line such that two vertices are adjacent if and only if the associated intervals have non-empty intersection" [6, Definition 6.3.4]. Another example are *string graphs*. A graph G is a string graph if there exists a set of curves (called strings) drawn in the plane such that no three strings intersect at a single point, each vertex is associated with a different string, and there is an edge for each intersection of strings [7].

From Harary's introduction to the problem, researches do not make too much progress. In these days, we know sum labeling algorithms for not so many graph families. Moreover, Harary's original question about the complete characterization of sum graphs is, as well as a general algorithm that extends arbitrary graph G to become a sum graph using the minimum number of isolates, still unanswered and challenges the researchers for more than 30 years.

One of the main goals for a theoretical computer scientist is to determine whether a given problem is “easy” to solve, or there is no “feasible” decision procedure. Even such characterization in terms of computational complexity classes, which Harary [2] introduced as an open problem in his original article, remains unsolved.

Goals

The main goal of this thesis is to summarize all known results about sum graphs. Based on this survey we would like to (a) generalize and/or simplify known algorithms which produce sum labeling for some graph families, (b) come up with formal proofs of statements and algorithms related to sum graphs, which are in many cases hidden in the personal correspondence of the authors, (c) introduce sum labeling algorithms for graphs families for which the optimal sum labeling schema is still not known and (d) try to present the sum labeling algorithm for an arbitrary graph and investigate the problem from a computational complexity perspective.

Our contribution

The first part of this thesis brings the reader formal definitions of all the concepts related to sum graphs. Besides that, we introduce some well-known sum graphs properties for which there were only fragments of proofs or no proofs at all, in literature.

In the second third of our work, we introduce labeling algorithms for the sum labeling of both already known graph families and entirely new ones. Many already known algorithms were generalized, and possibly new proofs of them are introduced.

In the last chapter, we introduce the first generally applicable algorithm for sum labeling arbitrary graph while using an optimal number of extra isolates. Despite that the running time of our algorithm is far from optimal, we believe that this algorithm can be significantly useful in further research on the presented problem.

Thesis organization

We have organized the remainder of this thesis in the following way. In Chapter 1, we introduce general notation necessary for understanding the rest of our work.

Chapter 2 contains formal definitions of all concepts related to sum graphs, and we introduce here some basic properties of sum graphs, which are mostly used in Chapter 3. In Section 2.1, we give a brief overview of related problems, and especially in Section 2.2, we describe the natural motivation for studying this area of graph theory.

In Chapter 3, we present algorithms for sum labeling of selected graph families, and the last chapter contains several ideas about the general-purpose algorithm, which finds the sum labeling of an arbitrary graph G using an optimal number of isolated vertices.

Chapter **1**

Definitions and notation

In this chapter, we introduce the notation used in the thesis. We mainly follow the basic notation of number and graph theory by Diestel [8]. The notation of Fibonacci numbers and sequence is by Vorobiev [9], and asymptotic analysis is from the book of Graham, Knuth, and Patashnik [10]. In complexity theory notation, we follow the monograph of Arora and Barak [11]. We introduce other necessary definitions in the relevant chapters.

1.1 Number theory

By $\mathbb{N}$ we denote the set of natural numbers, excluding zero. $\mathbb{N}^0$ denotes the set of all non-negative integers including zero. For any integer $n \geq 1$ we abbreviate the set $\{1, 2, \ldots, n-1, n\}$ with $[n]$.

Let S be a set. By $[S]^k$ we denote[1] the set of all k-element subsets of set S. Note that since $[S]^k$ is a set of k-element subsets, there are no two elements $X, Y \in [S]^k$ such that $X = Y$ up to permutation. Moreover, all elements of $X \in [S]^k$ are distinct elements of set S.

We write logarithms as $\log_b x$, where b is its base. If the base is omitted, logarithms are taken at base 2. We denote natural logarithm by ln.

1.1.1 Asymptotics

In many situations, such as when examining our algorithms' running time, it is useful to have a tool that allows us to compare the growth of given functions and decide which algorithm scales well with large data.

Let n denote the size of the input of our algorithm. There is not much sense in comparison for small inputs, since the small instances are processed very fast even with a slow algorithm; thus, we are interested in behavior in terms of running-time for considerably large inputs.

[1] In related publications (e.g., [12]), it is relatively common to denote this set as $\binom{S}{k}$.

We are usually also not very interested in multiplicative and additive constants, as they usually depend on implementation details and concrete hardware architecture. All the preceding leads to the following definitions of the *asymptotic* behavior of functions.

For the rest of this section, we denote by f and g two given function $f\colon \mathbb{N}^0 \to \mathbb{N}^0$ and $g\colon \mathbb{N}^0 \to \mathbb{N}^0$.

If there exists a constant $c \in \mathbb{R}^+$ and $n_0 \in \mathbb{N}$ such that for every $n \in \mathbb{N}$ with $n \geq n_0$ it holds that

$$f(n) \leq c \cdot g(n),$$

we say that $f(n)$ is at most of order $g(n)$ and we write $f(n) \in \mathcal{O}(g(n))$. In other words, by $\mathcal{O}$-notation, we express an upper bound for a function growth up to multiplicative and additive constants.

In opposite, sometimes we would like to express a lower bound for function growth. It is defined similarly to an upper bound but with usage of the Ω-notation. If there is a constant $c \in \mathbb{R}^+$ and $n_0 \in \mathbb{N}$ such that for every $n \in \mathbb{N}$ with $n \geq n_0$ it holds that

$$f(n) \geq c \cdot g(n),$$

we say that $f(n)$ is at least of order $g(n)$ and we write $f(n) \in \Omega(g(n))$.

These two concepts given together directly make a definition of an asymptotically tight bound for function growth. We say that f is of the same order as g if there exist constants $c_1, c_2 \in \mathbb{R}^+$, and $n_0 \in \mathbb{N}$ such that for every $n \in \mathbb{N}$, $n \geq n_0$, it holds that

$$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n).$$

We denote this tight bound by $f(n) \in \Theta(g(n))$.

1.1.2 Fibonacci numbers

Many of our labeling schemas use heavily the Fibonacci numbers, which can be formally defined as follows:

Definition 1 (Fibonacci numbers)**.** For a non-negative integer $n \geq 0$, the value of n-th Fibonacci number, denoted F_n, is defined as

$$F_n = \begin{cases} 0 & n = 0, \\ 1 & n = 1, \\ F_{n-1} + F_{n-2} & n \geq 2. \end{cases}$$

Example. *First* 15 *members of Fibonacci sequence are:*

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, \ldots$$

For the proofs of our hardness results following well-known lemma will be appropriate. We include the proof for self-containment of the thesis.

Lemma 1. *There is a constant $c < 1$ such that $F_n \leq 2^{cn}$ for all $n \geq 0$.*

Proof. The lemma obviously holds for F_0 and F_1 for every $c > 0$. Let us now suppose that the lemma holds for every value until F_{n-1}.

By definition the n-th Fibonacci number is

$$F_n = F_{n-1} + F_{n-2}.$$

We can now rewrite the equation using induction hypothesis as

$$2^{cn} \geq 2^{c(n-1)} + 2^{c(n-2)},$$

which gives us

$$\begin{aligned}
2^{cn} &\geq 2^{cn} \cdot 2^{-c} + 2^{cn} \cdot 2^{-2c} \\
2^{cn} &\geq 2^{cn} \cdot (2^{-c} + 2^{-2c}) \\
1 &\geq 2^{-c} + 2^{-2c} \\
2^{2c} &\geq 2^{c} + 1 \\
c &\geq \frac{\ln(1+\sqrt{5}) - \ln(2)}{\ln(2)} \\
c &\gtrapprox 0.694242.
\end{aligned}$$

Together with the condition from Lemma 1, we showed that any c from interval $(0.6943, 1)$ works and the proof is complete. $\square$

It follows that Fibonacci numbers grow at worst exponentially. This fact is significant because exponentially large numbers can be encoded using $\mathcal{O}(\log 2^n) = \mathcal{O}(n)$ bits. Thus, the certificate of existence of an appropriate sum labeling has polynomial-size, and algorithms that use Fibonacci numbers belong to **P** or **NP** complexity class (for formal definition, please see Section 1.3).

1.2 Graph theory

A *graph* is a pair (V, E), where V is a nonempty set of vertices, and $E \subseteq [V]^2$ is a set of edges. For any graph G, we denote by $V(G)$ the set of vertices of a graph G and similarly by $E(G)$ the set of all edges of G. We say that a graph G is *finite* if and only if the size of the set of vertices $V(G)$ is finite. Unless otherwise stated, all our graphs are finite.

For a vertex $v \in V(G)$, we say that v is *incident* to an edge $e \in E(G)$, and v is an *endpoint* of e if $v \in e$. Two vertices $u, v \in V(G)$ are *neighbors*, or *adjacent*, if $\{u, v\} \in E(G)$. By $N_G(v)$, we denote the set of all neighbors of a vertex $v \in V(G)$ in a graph G.

Our definition of graphs directly forbid *loops*, which are the edges with both endpoints equal to the same vertex $v \in V(G)$. By our definition of the graph, there is also at most one edge between two distinct vertices. Hence, we assume only simple graphs.

For a vertex $v \in V(G)$ we define its *degree* $\deg(v)$ as the number of neighbors of v in G. Vertices with degree 0 are called *isolates*. By $\delta(G) := \min\{\deg(v) | v \in V(G)\}$ we denote the *minimum degree* of G and by $\Delta(G) := \max\{\deg(v) | v \in V(G)\}$ we denote the *maximum degree* of a graph G. If for every vertex $v \in V(G)$ it holds that $\deg(v) = k$, we say that G is *k-regular* graph.

A graph $G = (V, E)$ is *connected* if for any two vertices $u, v \in V(G)$ there is a path connecting u and v in G. When G is not connected, we say that G is *disconnected*.

For a graph G its *complement* graph $\overline{G}$ has $V(\overline{G}) = V(G)$ and $E(\overline{G}) = [V]^2 \setminus E(G)$.

Let G and H be two graphs. We say that $G \cup H := (V(G) \cup V(H), E(G) \cup E(H))$ is a *(disjoint) union* of graphs G and H. If there is a bijection $f\colon V(G) \to V(H)$ such that for every pair $u, v \in V(G)$ it holds that $\{u, v\} \in E(G)$ if and only if $\{f(u), f(v)\} \in E(H)$, we say that graphs G and H are *isomorphic*, we write $G \simeq H$, and f is their *isomorphism*.

1.3 Computational complexity

In addition to examining the asymptotic complexity of proposed algorithms, we have also, closely related, theory for classification of concrete computational problems[2] (such as sorting, Maximum flow, Hamiltonian cycle, 3-coloring, and many others) into categories called *complexity classes*.

Computational problems are investigated from the point of view of different computational models, which simplify their analysis. There are many of them, such as the RAM model [13]. We construct our complexity theory based on Turing machines [14].

Definition 2 (Turing machine [11])**.** A deterministic Turing machine $\mathcal{M}$ is a tuple (Σ, Q, δ), where

- Σ is a finite set of symbols $\mathcal{M}$ works with. We assume that Σ contains unique "blank" symbol B, "start" symbol $\triangleright$, and the numbers 0 and 1. We call Σ the *alphabet* of $\mathcal{M}$.

- Q is a finite set of states of the machine $\mathcal{M}$. We assume that Q contains a start state, denoted q_{start}, and a halting state denoted q_{stop}.

- $\delta\colon Q \times \Sigma^k \to Q \times \Sigma^k \times \{\leftarrow, \bullet, \rightarrow\}^k$ is a *transition function* describing the rule $\mathcal{M}$ uses in performing each step.

One can imagine a Turing machine as a machine containing $k \in \mathbb{N}$ infinite one-directional *tapes* divided into cells, each of which holds a symbol from Γ, and k *tape heads* that read or write symbols to the tape. At each step of computation, each tape head reads the symbol on the current head position, and according to the transition function, replace the current symbol, change the current state and change the head

[2] We would like to highlight the difference between the asymptotics and computational complexity. As the first investigate the complexity of a concrete algorithm for a given problem, the second examines the complexity of the given computational problem regardless of a particular algorithm.

position one cell left or right. For an exhaustive overview of Turing machines and their properties, we refer the reader to [11].

We can finally define our first complexity class. It is called **NP** and contains the vast majority of well-known computational problems.

Definition 3. (**NP** complexity class [11]) A language $L \subseteq \{0,1\}^*$ is in complexity class **NP** if there exists a polynomial $p\colon \mathbb{N}^0 \to \mathbb{N}^0$ and a polynomial-time Turing machine $\mathcal{M}$ such that for every $x \in \{0,1\}^*$, $x \in L$ if and only if there exists $u \in \{0,1\}^{p(|x|)}$ such that $\mathcal{M}(x,u) = 1$. If $x \in L$ and $M(x,u) = 1$, then we call u a certificate for x.

In other words, class **NP** contains all the problems, for which we can easily verify the solution. Nevertheless, it follows that this class contains both classes of problems, that are easy to solve, such as sorting, and the notoriously hard problems, such as Clique. Therefore, for a more exceptional division of problems, we define the following complexity class.

Definition 4 (**P** complexity class [11])**.** Let $T\colon \mathbb{N}^0 \to \mathbb{N}^0$ be some computable function. We let $\mathbf{DTIME}(T(n))$ be the set of all Boolean functions that are computable in $c \cdot T(n)$-time for some constant $c > 0$. Complexity class **P** is equal to $\bigcup_{c \geq 1} \mathbf{DTIME}(n^c)$.

"The class **P** is felt to capture the notion of decision problems with 'feasible' decision procedures" [11]. It is easy to see that $\mathbf{P} \subset \mathbf{NP}$, but it remains one of the most critical problems of computer science, whether $\mathbf{P} = \mathbf{NP}$. To capture the difference between classes **P** and **NP**, we classify problems using the property called **NP**-*hardness*.

Definition 5 (Karp reduction, **NP**-hardness, and **NP**-completeness [11])**.** We say that a language $A \subseteq \{0,1\}^*$ is *polynomial-time Karp reducible* to a language $B \subseteq \{0,1\}^*$ denoted by $A \leq_p B$ if there is a polynomial-time computable function $f\colon \{0,1\}^* \to \{0,1\}^*$ such that for every $x \in \{0,1\}^*$, $x \in A$ if and only if $f(x) \in B$. We say that B is **NP**-*hard* if $A \leq_p B$ for every $A \in \mathbf{NP}$. We say that B is **NP**-*complete* if B is **NP**-*hard* and $B \in \mathbf{NP}$.

The first **NP**-*complete* problem is 3-SAT due to the work of Cook [15]. Later Karp [16] showed **NP**-*completeness* of many other computational problems. Showing that some problem is **NP**-*hard* is taken as evidence that the problem is not polynomial-time solvable, since most researchers believe that $\mathbf{P} \neq \mathbf{NP}$.

Chapter 2

Sum graphs and their properties

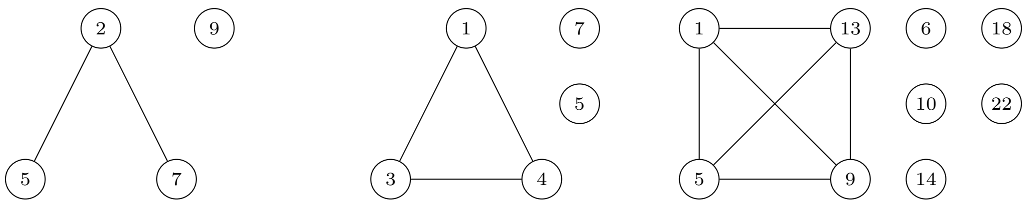
Harary, in his talk [1] and the follow-up pioneer paper [2], defined sum graphs as follows.

Definition 6 (Sum graph [2]). Let $S \subset \mathbb{N}$ be a finite set of positive integers. We say that the graph $G^+(S) = (V, E)$ is a *sum graph compatible with the set S* if $V = S$ and $E = \{\{x, y\} \mid x + y \in S\}$. By extension, a general graph G is called *sum graph* if there exists $S \subset \mathbb{N}$ such that $G^+(S) \simeq G$.

Example. *For an example, the sum graph compatible with the set* $S = \{1, 2, 3\}$ *is the graph* $G^+(S) = (V, E)$, *where* $V = S = \{1, 2, 3\}$ *and* $E = \{\{1, 2\}\}$, *since from all combinations of pairs of elements from* S *only the sum of* $x = 1$ *and* $y = 2$ *belongs to* S. *Based on the form of sets of vertices and edges of* $G^+(S)$ *we can say that the graph* $P^1 \cup K^1$ *is a sum graph.*

Similarly, for the set $S = \{5, 7, 8, 10, 12, 15, 18\}$ *the compatible graph* $G^+(S)$ *has vertices* $V(G^+(S)) = S$ *and edges* $E(G^+(S)) = \{\{5, 7\}, \{5, 10\}, \{7, 8\}, \{8, 10\}\}$. *The resulting graph is isomorphic with the graph* $C_4 \cup \overline{K_3}$, *i.e.,* $C^4 \cup \overline{K^3}$ *is, as well as* $P^1 \cup K^1$, *a sum graph.*

In Figure 2.1, we offer the reader a few more examples of different sum graphs and their compatible sets.



(a) A sum graph asociated with set $S = \{2, 5, 7, 9\}$

(b) A sum graph associated with set $S = \{1, 3, 4, 5, 7\}$

(c) A sum graph associated with set $S = \{1, 5, 6, 9, 10, 13, 14, 18, 22\}$

Figure 2.1: Examples of three sum graphs

It is not hard to see that not every graph $G = (V, E)$ is a sum graph. As an example of such a graph, we can take the cycle graph C^3. This graph has three vertices and three edges. Suppose that there is a set $S = \{a, b, c\}$ of positive integers such that $G^+(S) \simeq C^3$. Without loss of generality, we can assume that $a < b < c$. But we can see that from the numbers a, b, and c, we are able to construct at most one edge, which is strictly less than three edges of C^3.

The following lemma shows why Definition 6 discards so many common graph classes from the subject of interest.

Lemma 2. *Every sum graph $G = (V, E)$ contains an isolated vertex.*

Proof. Suppose that v is the vertex with the largest label in G, i.e.,

$$\forall u \in V(G) \setminus \{v\} \colon \sigma(u) < \sigma(v).$$

Furthermore, suppose for a contradiction that there is a vertex $w \in V(G)$ such that $\{v, w\} \in E(G)$. Since G is a sum graph, it holds that for every edge $\{x, y\} \in E(G)$, there is at least one vertex z such that $\sigma(z) = \sigma(x) + \sigma(y)$. Nevertheless, we selected v as the vertex with the largest label, so there cannot be a vertex w connected with v. It follows that v is an isolated vertex in G. $\square$

By Lemma 2, it holds that every sum graph with at least two vertices is disconnected. Harary stated [2] that "Trivially, almost no graphs are sum graphs, since it is known [17] that almost all graphs are connected".

That is also the reason why the following alternative, but equivalent, definition has been adopted by the community.

Definition 7 (Sum labeling)**.** Let $G = (V, E)$ be a graph and let $\sigma \colon V \to \mathbb{N}^+$ be an injective function. We say that σ is a *sum labeling* of the graph G if for any two distinct vertices $u, v \in V(G)$, $\{u, v\}$ is an edge of G if and only if $\sigma(u) + \sigma(v) = \sigma(w)$ for some other vertex $w \in V(G)$.

If the above definition holds not only for two distinct vertices u and v but even for possibly equals vertices $u = v$, then the sum labeling is called *strong sum labeling*. With Definition 7, we can finally give the promised second definition of sum graphs.

Definition 8 (Sum graph)**.** A graph $G = (V, E)$ is called a *sum graph* if and only if there exists a sum labeling σ of G.

As stated in Lemma 2, there is no connected graph G with $|V(G)| \geq 2$ such that G is a sum graph. So it becomes an interesting question of how many isolated vertices we must add to a general graph G to obtain a sum graph.

Definition 9 (Sum number)**.** The *sum number* $\sigma(G)$ of a graph $G = (V, E)$ is the least integer, such that $G' = (V(G) \cup \{v_0, \ldots, v_{\sigma(G)-1}\}, E(G))$, where $\forall i \in \{0, \sigma(G) - 1\} \colon v_i \notin V(G)$, is a sum graph.

It is clear that the sum number of sum graphs is 0, since there is no need to tackle any isolates to the graph because it is already a sum graph. Another particular class of graphs is a class of graphs with a sum number equal to 1.

Definition 10 (Unit graph [18])**.** A graph G is called *unit graph* if $\sigma(G) = 1$.

Smyth [18] proposed constructive method to prove that there is no unit graph such that its number of edges is greater than $\lfloor |V(G)|^2/4 \rfloor$, and for each p, $|V(G)| - 1 \le p \le \lfloor |V(G)|^2/4 \rfloor$, there is at least one unit graph with p edges.

For graphs that are not sum graphs by itself, we have the following folklore trivial lower bound mentioned by many authors without a formal proof.

Lemma 3 (Trivial lower bound)**.** *For every graph G it holds that $\sigma(G) \ge \delta(G)$.*

Proof. Let $G = (V, E)$ be a graph, $G^+ = G \cup \overline{K_{\sigma(G)}}$ corresponding sum graph, and let $\sigma\colon V(G^+) \to \mathbb{N}$ be a sum labeling function according to Definition 7. By definition we know, that for every $u, v \in V(G^+)\colon \sigma(u) = \sigma(v) \iff u = v$. Let $w \in V(G)$ be a vertex with the largest label over all vertices in $V(G)$. Since w has the largest label in $V(G)$, it is clear that the added isolated vertices must cover all edges incident with w, i.e., there must be at least $\deg(w)$ isolates.

So the previous paragraph claims that every labeling scheme must use at least as many isolated vertices as is the degree of the vertex with the largest label in G. Therefore, there cannot be a graph with $\sigma(G) < \delta(G)$. □

This lower bound on the sum number of an arbitrary graph gives us a straightforward characterization of graphs, which can be, from a certain point of view, optimally sum labeled.

Definition 11 (k-optimum summable graph [19])**.** We say that a nontrivial connected graph G is *k-optimum summable*, where $k \ge 1$, if and only if $\sigma(G) = \delta(G) = k$.

For examples of k-optimally summable graphs, we can return to Figure 2.1. If we imagine the graph from Figure 2.1a as $G_a = (\{2, 5, 7\}, \{\{2, 5\}, \{2, 7\}\})$ we can see, that $\delta(G_a) = 1$. Moreover, there is only one added isolated vertex with label 9, i.e., $\sigma(G_a)$ is equal to the minimum degree, and G_a is 1-optimum summable. Similarly with the graph C^3 from Figure 2.1b. On the other hand, the graph K^4 in Figure 2.1c is not 3-optimum summable because there are 5 added isolates, which is, as we prove later in Section 3.5, an optimal number of added isolates.

From practical point-of-view, it is also interesting to question what is the upper bound on sum number, i.e., what is the number of isolates which we must add to an arbitrary graph to be sure that obtained graph is a sum graph. We can find claims about trivial upper bound in the work of Harary [2] and Hao [20], but the rigorous proof is missing.

Lemma 4 (Trivial upper bound)**.** *Every connected graph $G = (V, E)$ can be transformed into the sum graph G^+ by adding $|E(G)|$ isolated vertices.*

Proof. Let $G = (V, E)$ be a graph with vertices $v_1, \ldots, v_n$. We assign to every vertex $v_i \in V(G)$ label $\sigma(v_i) = 10^i$ and for each edge $\{v_i, v_j\} \in E(G)$ we introduce in the corresponding sum graph G_+ single isolate $v_{i,j}$ with label $\sigma(v_{i,j}) = 10^i + 10^j$.

It is clear from the definition of our labeling schema that for each edge $\{v_i, v_j\} \in E(G)$ there exists a vertex z such that $\sigma(z) = \sigma(v_i) + \sigma(v_j)$. It is the added isolate $v_{i,j}$. We must only verify that for every $\{u, v\} \notin E(G^+)$ there is no vertex z such that $\sigma(z) = \sigma(u) + \sigma(v)$.

We can imagine each label as an n-digit number of base 10. For every $v \in V(G)$, the label contains a single 1, and on the rest of the positions are 0. On the other hand, vertices $z \in V(G^+) \setminus V(G)$, i.e., the added isolates, have 1 on two positions.

With such imagination it is easy to see that for $v_i, v_j \in V(G)$ such that $\{v_i, v_j\} \notin E(G)$ there will be no vertex $w \in V(G^+)$ such that $\sigma(w) = \sigma(v_i) + \sigma(v_j)$ since w must contain two 1 in its label, but there is no such vertex in $V(G)$, and we did not add an isolate with such label.

Also, the case when we have $v \in V(G)$ and $z \in V(G^+) \setminus V(G)$ is clear. A witness of edge $\{v, z\}$ contains either three 1 in its label, or single 1 and single 2, but such a vertex does not exist in $V(G^+)$. The same arguments hold when both v, z are added isolates. In this case, the potential witness contains either four 1 in its label or at least a single 2. $\square$

Note that the base of the labeling schema in the proof of Lemma 4 must not be specifically number 10. Hao [20] proposed base 3, but any labeling with base strictly greater than 2 works.

We conclude this brief introduction of sum graphs properties with the lemma of Harary [2] about the multiplication of labels and our lemma about added isolate. Note that in the original paper, the proof of the Harary's lemma is omitted.

Lemma 5. *Let $G^+(S)$, where $S = \{x_1, x_2, \ldots, x_n\}$, be a sum graph and let $k \in \mathbb{N}$ be a constant. Then graph $G^+(S')$, $S' = \{k \cdot x_1, k \cdot x_2, \ldots, k \cdot x_n\}$, is isomorphic with $G^+(S)$.*

Proof. The proof directly follows from the distributive property of a multiplication over addition in natural numbers. $\square$

Lemma 6. *Let $G = (V, E)$ be a graph, $k \in \mathbb{N}^0$ be a constant and $G^+(S)$, where $S = \{x_1, x_2, \ldots, x_{|V(G)|}, \ldots, x_{|V(G)|+k}\}$ be a sum graph such that $G \cup \overline{K^k} \simeq G^+(S)$. Then there is a sum graph $G^+(S')$, $S = \{x_1, x_2, \ldots, x_{|V(G)|}, \ldots, x_{|V(G)|+k}, x_{|V(G)|+k}\}$ such that $G \cup \overline{K^{k+1}} \simeq G^+(S')$.*

Proof. Let $S = \{x_1, \ldots, x_n\}$ be an ordered set such that $G^+(S)$ is a sum graph and $G^+(S) \simeq G \cup \overline{K^k}$. If we create set $S' = S \cup \{\gamma\}$, where $\gamma \in \mathbb{N}$ and $\gamma > x_{n-1} + x_n$, then the added number γ is the new largest label in S' and it follows from Lemma 2 that γ corresponds to an isolated vertex z in $G^+(S')$.

Moreover, the graph $G^+(S')$ remains a sum graph, because the isolated vertex z cannot be a witness of an edge, since its label is bigger than the sum of the second and the third largest labels in S'. The label of z is just too high, so the lemma holds. $\square$

2.1 Related work

The concept of sum graphs was introduced by Harary [2] in 1990. Harary investigated the relationship between graphs and subsets of the set of positive integers $\{1, 2, \ldots\}$.

Since the problem of decision, whether a given graph G is a sum graph challenges researchers for decades, some less restrictive variants of sum graphs were introduced. In 1994 Harary [21] introduced the *integral sum graphs* where S is not restricted to be a subset of positive integers, but it is permitted to be a subset of all integers. Even earlier, in 1990, Boland, Laskar, Turner, and Domke [22] proposed generalization of sum graphs called *modular sum graphs*, where a graph G is called modular sum graph if there exists a positive integer $\mathcal{Z}$ such that $V(G) \subseteq \{1, \ldots, \mathcal{Z} - 1\}$ and $E(G) = \{\{x, y\} | ((x + y)(\bmod \mathcal{Z})) \in V(G)\}$. Harary, Hentzel, and Jacobs [23] also defined *real sum graphs* by allowing S to be a finite set of real numbers and proved that every real sum graph is a sum graph.

Similar concepts were investigated even for various mathematical operations. Bloom et al. [24, 25] and Harary [2] independently discovered *difference graphs* $G^-(S)$ defined analogously to sum graphs as $V(G^-) = S$ but with $\{x, y\} \in E(G^-)$ if and only if $|x - y| \in S$. Bergstrand et al. [26] proposed, in 1992, *product graphs* by excluding number 1 from the set S and defining that $\{x, y\} \in E(G^*)$ if and only if $x \cdot y \in V(G^*)$. Moreover, in the same paper, the authors proved "that every product graph is a sum graph and vice versa" [27].

For a comprehensive overview of research and papers related to sum graphs and similar concepts, we refer the reader to the survey of Gallian [27, pp. 230–238].

2.2 Motivation

The genuine motivation of the first research on sum graphs was a pure fascination of problem-solving. The practical usage of this theory was found years later in the work of T. Hao [20] and expressly in several papers of Mirka Miller [28, 29, 3].

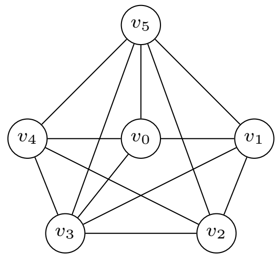
2.2.1 Graph compression

The problem of finding sum labeling has straightforward utilization in graph compression when we store them digitally, i.e., in computer. A graph is, by definition, a pair of sets of vertices and edges. It also leads to two traditional ways of their representation in computer memory, where both vertices and edges are stored in different ways, but explicitly.

Adjacency matrix The first option of graph storage is using its *adjacency matrix*. Let $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$, be a graph. The adjacency matrix $\mathbf{A}_G = (a_{i,j})_{|V(G)| \times |V(G)|}$ of graph G is a square matrix such that

$$a_{i,j} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \in E(G), \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to see that such representation has space complexity $\mathcal{O}(|V(G)|^2)$, since the overall matrix is stored. On the other hand, if we want to check whether $\{v_i, v_j\}$ is an edge of the graph G, it can be done in constant time by looking at the proper element of adjacency matrix $\mathbf{A}_G$. For an example of a graph and its adjacency matrix, please see Figure 2.2.



$$\begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Figure 2.2: Graph and the corresponding adjacency matrix

Adjacency list The second most common option for storing graphs is with the usage of *adjacency lists*. This representation uses an array of length $|V(G)|$ where on index i is the first element of a linked list containing, possibly in sorted order, all neighbors of vertex v_i. Linked list representation of a simple graph is shown in Figure 2.3.

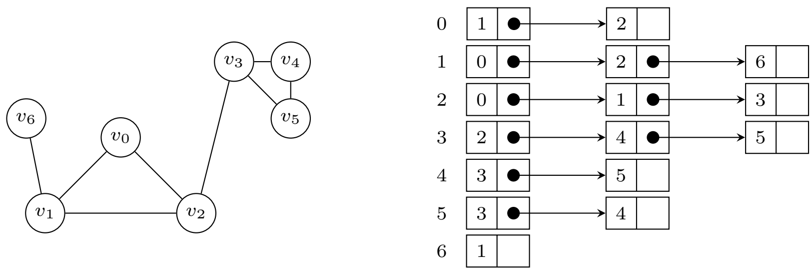


Figure 2.3: Graph and the corresponding adjacency list representation

In contrast with the adjacency matrix, linked lists representation has space complexity only $\mathcal{O}(|V(G)| + |E(G)|)$. However, if we decide to check whether there exists an edge between two vertices v_i and v_j, we must traverse the whole linked list containing neighbors of vertex v_i (or v_j), which leads to time complexity $\mathcal{O}(\Delta(G)) = \mathcal{O}(|V(G)|)$.

Based on previous paragraphs, we can say that graph representation using the adjacency matrix is optimal for dense graphs, wheres adjacency list representation is more suitable for problems over sparse graphs.

At first glance, it might seem that sum graphs can be stored only as a set of vertices, since all the edges are encoded explicitly in vertex labels. However, as stated before, very few graphs are sum graphs, so there is the need to introduce some isolated vertices to make some graph a sum graph — this number of isolates is known as the *sum number* (see Definition 9).

A natural question is about the upper bound on this number for a general graph. In Chapter 3, we introduce sum numbers for selected graphs families, for which it always applies that $\sigma(G) \in \mathcal{O}(|E(G)|)$. Nevertheless, Gould and Rödl [30] proved that there exists a class of graphs such that $\sigma(G) \in \Theta(|V(G)|^2)$. It is worth pointing out that their proof is not constructive — it uses probabilistic arguments — and it remains a challenging open problem to find such a graph class explicitly[3]. Even so, we can say that compression graphs using corresponding sum graphs can be, in some cases, contraproductive and is useful only for specific graph classes.

2.2.2 Secret sharing scheme

Slamet, Sugeng, and Miller [3] found another compelling motivation to study sum graphs in cryptography.

There are many situations where we need to, due to security reasons, make something inaccessible to a single person. For example, in the Czech Republic, we have Crown Jewels of Bohemia, which are stored in the secured and hardly-accessible chapel in St. Vitus cathedral. The door to the Crown Jewels chamber has seven locks, and each key is entrusted to a different person. All the key-holders must be convened to facilitate the opening of the door. [31]

The original motivation for the secret sharing scheme comes from Liu [32], who proposed the following problem: "Eleven scientists are working on a secret project. They wish to lock up the documents in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present. What is the smallest number of locks needed? What is the smallest number of keys to the locks each scientist must carry?"

The theory of secret sharing was invented independently by Shamir [33] and Blakley [34]. The concept is defined as follows. Let D be a piece of information (e.g., cryptographic key, password, safe combination). A *secret* (n, t)*-sharing scheme* is a method to distribute D to a set P of n participants such that

- every subset $Q \subseteq P$ of participants of size at least t is able to reconstruct original information and
- every subset $Q \subseteq P$ of size $t - 1$ or less gain no information about the original secret.

As Slamet, Sugeng, and Miller [3] pointed out

> *Many mathematical structures are used to create secret sharing schemes. For example, Brickell and Davenport [35] generated an ideal secret sharing scheme based on matroid theory.*

Later Stinson [36] proposed a secret sharing scheme based on graph access structures. Inspired by Stinson's work, Slamet, Sugang, and Miller [3] proved that sum graphs can

[3] Paper of Gould and Rödl [30] also disproved the conjecture of P. Erdös mentioned by Harary [2] whether "2p - 3 is the maximum value of $\sigma(G)$ for graph G with p vertices."

be used as access structure for a secret $(n, 2)$-sharing scheme with possible extension to arbitrary t using sum hypergraphs [37].

Chapter 3

Sum numbers of selected graph families

At the beginning of our study of sum labeling, we investigated the properties of some basic graphs families. Let us first formally define what graph family denotes.

Definition 12 (Graph family)**.** Let $\mathcal{G}$ be the class of all simple graphs and let $\upsilon\colon \mathcal{G} \to \{0,1\}$ be a function indicating the presence of some graph property Υ. A subclass $\mathcal{G}_\Upsilon \subseteq \mathcal{G}$ is called a *graph family* with respect to the property Υ if and only if

$$\forall G \in \mathcal{G}\colon G \in \mathcal{G}_\Upsilon \iff \upsilon(G) = 1.$$

Exact labeling schemas for many graph families were hidden in the authors' private communication, so there was the need to reinvent the labeling algorithms and come with proofs of them again.

3.1 Paths

The first family of graphs we were studied was *path graphs*, ordinarily referred to as P^n. These graphs consist of $n+1$ vertices $v_0,\ldots,v_n$ and n edges $\{v_i, v_i+1\}$ for every $i \in \{0,\ldots,n-1\}$.

The basic case of path graphs is P^0. This simple graph contains only single vertex and no edges at all, so there is no need for any isolates — P^0 is a sum graph. To a single vertex v_0, we can assign any positive natural number $\alpha \in \mathbb{N}$. Based on this observation, we can say that $\sigma(P^0) = 0$.

For any nontrivial case, a little bit sophisticated labeling schema must be found. Hao proposed [20] labeling paths with Fibonacci numbers, but the proof of correctness of such labeling is not given in [20]. Based on that, we come with the following more general algorithm.

Algorithm 1 (Sum labeling of paths)**.** *Let P^n, where $n \geq 1$, be a path graph. Make union of graph P^n with graph K^1 consisting out of a single vertex named v_{n+1} and assign*

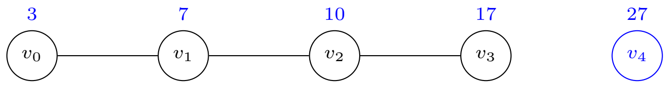


Figure 3.1: Sum labeling of path graph P^3 with vertex set $V(P^3) = \{v_0, v_1, v_2, v_3\}$ and edges $E(P^3) = \{\{v_0, v_1\}, \{v_1, v_2\}, \{v_2, v_3\}\}$. We select minimum label $\alpha = 3$ and proceed Algorithm 1. According to definition of the algorithm each vertex $v_i \in V(P^3)$ obtains label $F_{i+2} \cdot \alpha + F_i$ and the added isolate v_4 obtains label $F_6 \cdot \alpha + F_4 = 8 \cdot 3 + 3 = 27$.

to each vertex v_i of newly created graph label $\sigma(v_i) = F_{i+2} \cdot \alpha + F_i$, where $\alpha \in \mathbb{N}$ is any selected minimal label.

For an example of the resulting sum graph generated by the preceding algorithm on a path graph with four vertices, we refer the reader to Figure 3.1. Let us now introduce the following lemma to prove the correctness of Algorithm 1.

Lemma 7. *Let P^n, $n \geq 1$, be a path graph and $\alpha \in \mathbb{N}$ be a minimum required label. Then for the given graph P^n and minimum required label α Algorithm 1 produces sum graph G^+ such that $G^+ \simeq P^n \cup K^1$ and*

$$\forall u, v \in V(G^+)\colon \{u, v\} \in E(G^+) \iff \exists w \in V(G^+)\colon \sigma(u) + \sigma(v) = \sigma(w).$$

Proof.
($\Rightarrow$) To proof the correctness of Lemma 7 we must at first show that for each edge $\{v_i, v_{i+1}\}$, where $0 \leq i \leq n-1$, there exists a vertex v_k with label $\sigma(v_i) + \sigma(v_{i+1})$. As below equation shows, this condition is fulfilled by vertex v_{i+2}, which always exists, thanks to the added isolate v_n.

$$\begin{aligned}
\sigma(v_i) + \sigma(v_{i+1}) &= F(i+2) \cdot \alpha + F(i) + F(i+3) \cdot \alpha + F(i+1) \\
&= (F(i+2) + F(i+3)) \cdot \alpha + (F(i) + F(i+1)) \\
&= F(i+4) \cdot \alpha + F(i+2) = \sigma(v_{i+2}).
\end{aligned}$$

($\Leftarrow$) To complete the proof, we must also show that for any two selected vertices v_i and v_j which are not adjacent, i.e., $|i - j| \geq 2$, there is no vertex v_k such that $\sigma(v_i) + \sigma(v_j) = \sigma(v_k)$. Without loss of generality, we can assume that $1 \leq i + 1 < j < k \leq n$.

Suppose for contradiction that the Algorithm 1 produced labeling such that for two vertices v_i, v_j not connected by edge there exists third vertex v_k with label equal to $\sigma(v_i) + \sigma(v_j)$. When we write down the equation, we get the following result:

$$\begin{aligned}
\sigma(v_i) + \sigma(v_j) &= \sigma(v_k) \\
F(i+2) \cdot \alpha + F(i) + F(j+2) \cdot \alpha + F(j) &= F(k+2) \cdot \alpha + F(k) \\
(F(i+2) + F(j+2) - F(k+2)) \cdot \alpha &= F(k) - F(j) - F(i) \\
\alpha &= \frac{F(k) - F(j) - F(i)}{F(i+2) + F(j+2) - F(k+2)}
\end{aligned}$$

Because of assumed boundaries for i, j and k it follows that the used initial label α was a negative number. But that is a contradiction with the algorithm definition. Thus, the algorithm never assigns sum of labels of not connected vertices to the third vertex in a graph.

We have successfully proved that the proposed algorithm works and gives us a proper sum labeling of path graphs. □

We note that Algorithm 1 always terminates, since it visits every vertex of P^n exactly once. The correctness of the generated result follows from Lemma 7. We also note that Algorithm 1 has time complexity $\mathcal{O}(n^2)$, since it, according to Lemma 1, assigns to every vertex at most exponentially big number. By Definition 4 it follows that sum labeling of paths is in complexity class **P**.

With the completed proof of an algorithm for the sum labeling of paths, we can state the final theorem about the sum number of path graphs.

Theorem 8. *Let $n \in \mathbb{N}^0$ and P^n be a path graph. It holds that*

$$\sigma(P^n) = \begin{cases} 0 & \text{if } n = 0, \\ 1 & \text{otherwise.} \end{cases}$$

Proof. The first case when $n = 0$ is obvious. For the latter case, we recall Lemma 3, which shows that the sum number is lower bounded by a minimal degree of a graph. We introduced a deterministic algorithm that found the sum labeling of any path graph while using single isolated vertex, so the theorem holds. □

The above theorem directly implies that path graphs are 1-optimal summable, and all path graphs with at least a single edge are unit graphs. Moreover, it is easy to see that Algorithm 1 produces a strong sum labeling of path graphs.

3.2 Cycles

Let P denote an arbitrary path graph with at least three vertices $\{v_0, v_1, v_2, \ldots, v_{n-1}\}$. Then graph $C := (V(P), E(P) \cup \{\{v_{n-1}, v_0\}\})$ is called a *cycle graph* (or simply a *cycle*). We ordinarily denote cycles by C^n, where n is the number of edges (or vertices).

For every vertex v of a cycle graph it holds that $\deg(v) = 2$, consequently all cycle graphs are 2-regular.

The smallest representative of cycle graphs is C^3. From Lemma 3 it is clear that we must add at least 2 isolates to make C^3 a sum graph. Let $\{v_0, v_1, v_2\}$ be the set of vertices and $\{\{v_0, v_1\}, \{v_1, v_2\}, \{v_0, v_2\}\}$ the set of edges of C^3 with the following labels assignment

$$v_0 = 1, \quad v_1 = 3, \quad v_2 = 4.$$

It is easy to see, that the edge $\{v_0, v_1\}$ is covered by vertex v_2, since $\sigma(v_0) + \sigma(v_1) = 1 + 3 = 4 = \sigma(v_2)$. It remains to cover edges $\{v_1, v_2\}$ and $\{v_0, v_2\}$. To cover the first, we must add an isolate v_3 with label 7. For the least edge, we introduce another isolate

v_4 with corresponding label 5. There is no doubt about the correctness of such labeling, and we obtained a sum graph from graph C^3 while using 2 added isolates, which is, by Lemma 3, optimal.

A fascinating fact is that for the graph C^4 we must add at least 3 isolated vertices to obtain a sum graph.

Lemma 9. *For a cycle graph C^4, it holds that $\sigma(C^4) \geq 3$.*

Proof. By Lemma 3, we know that $\sigma(C^4)$ is at least 2, so we are supposed to prove only that $\sigma(C^4) \neq 2$.

Suppose for contradiction that $\sigma(C^4) = 2$. We denote the vertices of C^4 by $\{v_0, v_1, v_2, v_3\}$ and the added isolates by v_4 and v_5. Moreover, we can assume that $E(C^4) = \{\{v_0, v_1\}, \{v_1, v_2\}, \{v_2, v_3\}, \{$

Without loss of generality, we can assume that the vertex v_3 has the largest label d out of all vertices of C^4 in sum labeling σ of the graph $C^4 \cup \overline{K^2}$. We know that v_3 is adjacent to two vertices v_0 and v_2 with labels $\sigma(v_0) = a$ and $\sigma(v_2) = c$. Since v_3 has the largest label, it follows that $a < d$ and $c < d$. Moreover, it also follows that $\sigma(v_4) = a+d$ and $\sigma(v_5) = c + d$.

Let b denote the label corresponding to vertex v_1 in the sum labeling σ. At this time, we have covered two edges out of four. The remaining edges are those incident with the vertex v_1.

We take, for example, the edge $\{v_0, v_1\}$. From the definition, there must be a vertex with label $a + b$. It is clear, since all the labels must be greater than 0, that the witness can be neither the vertex v_0 nor the vertex v_1. The vertex v_4 is also out of the question, since its label is $a + d$, which implies $b = d$.

The remaining possible witnesses are vertices v_3 and v_5. Let us assume that the required witness is the vertex v_3. In such case we have $d = \sigma(v_3) = \sigma(v_0)+\sigma(v_1) = a+b$ and if we substitute for d in all labels, then we obtain

$$\sigma(v_0) = a,\ \sigma(v_1) = b,\ \sigma(v_2) = c,\ \sigma(v_3) = a + b,\ \sigma(v_4) = 2a + b,\ \sigma(v_5) = a + b + c.$$

Now we must find a witness for the edge $\{v_1, v_2\}$. For an obvious reason, the vertices v_1, v_2 are forbidden. Also, the vertex v_3 cannot be a witness, since it implies $a = c$, and we do not have to think of the vertex v_5 because its label is too high.

If we take as the witness the vertex v_0, then after the substitution we get $\sigma(v_4) = 3b + 2c = b + 2b + 2c = \sigma(v_1) + \sigma(v_5)$, but this edge is forbidden. It follows that v_0 cannot be a witness.

Let us take as the witness the vertex v_4. In this case we obtain that $2a+b = b+c \implies c = 2a$ and after substitution we get $\sigma(v_5) = 3a + b = a + (2a + b) = \sigma(v_0) + \sigma(v_4)$, but this edge is also forbidden.

All the previous paragraphs imply that the vertex v_3 cannot be the witness for the edge $\{v_0, v_1\}$. To complete the proof we must exclude the label v_5 from the list of possible witnesses.

With the label v_5 as the witness we obtain, after substitution, following labels

$$\sigma(v_0) = a,\ \sigma(v_1) = b,\ \sigma(v_2) = c,\ \sigma(v_3) = d,\ \sigma(v_4) = a + d,\ \sigma(v_5) = c + d = a + b.$$

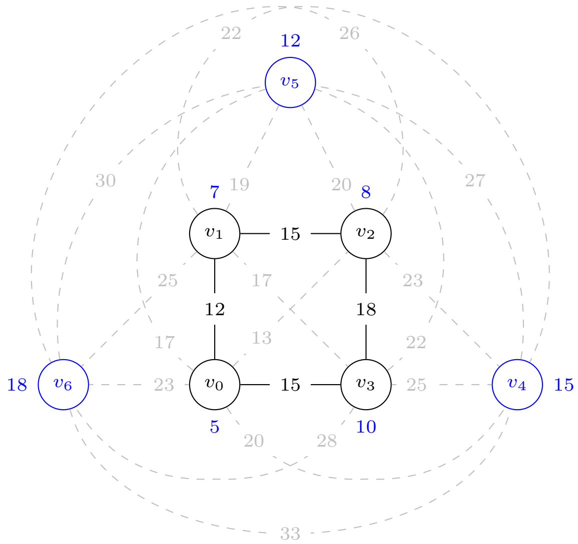


Figure 3.2: Sum labeling of cycle graph C^4. One can easily verify, that resulting graph is a sum graph.

Now, we need to find the witness for the last remaining edge $\{v_1, v_2\}$. Vertices v_1, v_2, v_5 are all forbidden due to a similar argument as with the witness v_3. We must check only for v_0 and v_4.

From the least case, we obtain after substitution that $c = a$ and the case when the witness is v_0 induces forbidden edge $\{v_1, v_5\}$.

We have examined all possible cases of labels assignments and we proved that there is no sum labeling of C^4 with 2 added isolates. It follows that at least 3 isolated vertices are required, i.e., $\sigma(C^4) \geq 3$, and the proof is complete. $\square$

To prove that $\sigma(C^4) = 3$, we refer the reader to Figure 3.2, which displays one possible proper sum labeling of a graph $C^4 \cup \overline{K^3}$.

For cycles with at least 5 vertices, we can, as we prove later in this section, again use an algorithm similar to our example of labeling C^3, i.e., we will suffice with 2 isolates.

Algorithm 2 (Sum labeling of cycles). *Let C_n, where $n \geq 3$ and $n \neq 4$, be a cycle graph with vertices $\{v_0, \ldots, v_{n-1}\}$ and edges $\{\{v_i, v_{i+1}\} \mid i \in \{0, \ldots, n-2\}\}\} \cup \{\{v_{n-1}, v_0\}\}$ and $\alpha \in \mathbb{N}$ be a minimal required label. We create sum graph G^+ with $V(G^+) = V(C^n) \cup \{v_n, v_{n+1}\}$ and $E(G^+) = E(C^n)$. Then we assign to each vertex of $V(G^+)$ a label according to the following rule*

$$\sigma(v_i) = \begin{cases} F_{i+2} \cdot \alpha + F_i & \text{if } i \in \{0, \ldots, n\}, \\ \sigma(v_0) + \sigma(v_{n-1}) & \text{if } i = n+1. \end{cases}$$

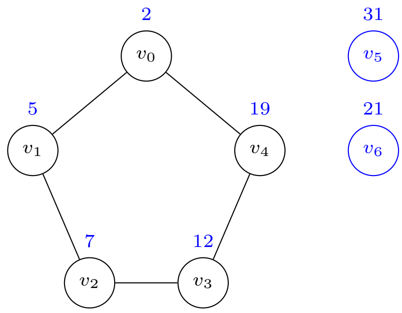


Figure 3.3: Sum labeling of cycle graph C^5 with vertex set $V(C^5) = \{v_0, v_1, v_2, v_3, v_4\}$. We select minimum label $\alpha = 2$ and simulated Algorithm 2. According to the definition of the algorithm each vertex $v_i \in V(C^5)$ obtains label $F_{i+2} \cdot \alpha + F_i$, first added isolate v_5 obtains label $F_7 \cdot \alpha + F_5 = 13 \cdot 2 + 5 = 31$ and second added isolate v_6 obtains a label $\sigma(v_0) + \sigma(v_4) = 2 + 19 = 21$

For an example of the output of Algorithm 2 executed with graph C^5 and initial label $\alpha = 2$ we refer the reader to Figure 3.3. Let us now prove an auxiliary lemma which help us to prove the correctness of the preceding algorithm.

Lemma 10. *Let C^n, $n \geq 3$, $n \neq 4$, be a cycle graph and $\alpha \in \mathbb{N}$ be a required initial label. Then for a given graph C^n and initial label α Algorithm 2 produces a sum graph G^+ such that $G^+ \simeq C^n \cup \overline{K^2}$ and*

$$\forall u, v \in V(G^+)\colon \{u, v\} \in E(G^+) \iff \exists w \in V(G^+)\colon \sigma(u) + \sigma(v) = \sigma(w).$$

Proof.
($\Rightarrow$) Let $i \in \mathbb{N}^0$, $0 \leq i < n-1$. For each edge $\{v_i, v_{i+1}\}$ there must exists vertex v_k, $k \in \{1, \ldots, n\}$ such that $\sigma(v_i) + \sigma(v_{i+1}) = \sigma(v_k)$. We can rewrite the preceding equation according to labels definition and we obtain

$$\begin{aligned} \sigma(v_i) + \sigma(v_{i+1}) &= \sigma(v_k) \\ F_{i+2} \cdot \alpha + F_i + F_{i+3} \cdot \alpha + F_{i+1} &= F_{k+2} \cdot \alpha + F_k \\ F_{i+4} \cdot \alpha + F_{i+2} &= F_{k+2} \cdot \alpha + F_k. \end{aligned} \tag{3.1}$$

Equation (3.1) has clearly a solution $k = i + 2$ which corresponds to assigned labels. Last not yet checked edge is $\{v_0, v_{n-1}\}$. For this edge we introduced special isolated node v_{n+1}, so the edge has witness. The implication therefore apply.

($\Leftarrow$) Now, we must show that for all vertices $u, v \in V(G^+)$ such that $\{u, v\} \notin E(G^+)$ there is no witness for the edge. We can be sure, thanks to Lemma 2, that vertex v_n is not adjacent to any $v \in V(G^+)$ since it has the largest label.

Let $i \in \{0, \ldots, n\}$. We know that vertex v_{n+1} cannot be adjacent to any other $v \in V(G^+)$. Let us now verify this fact. For a contradiction we assume that there is vertex v_i, $i \in \{0, \ldots, n-2\}$ and vertex v_k, $i < k \leq n-1$ such that $\sigma(v_i) + \sigma(v_{n+1}) = \sigma(v_k)$.

We rewrite the equation according to defined labeling schema and we get

$$\begin{aligned}
\sigma(v_i) + \sigma(v_{n+1}) &= \sigma(v_k) \\
F_{i+2} \cdot \alpha + F_i + F_2 \cdot \alpha + F_0 + F_{n+1} \cdot \alpha + F_{n-1} &= F_{k+2} \cdot \alpha + F_k \\
\alpha \cdot (F_{i+2} + F_2 + F_{n+1} - F_{k+2}) &= F_k - F_i - F_0 - F_{n-1} \\
\alpha &= \frac{F_k - F_i - F_0 - F_{n-1}}{F_{i+2} + F_2 + F_{n+1} - F_{k+2}} \\
\alpha &= \frac{F_k - F_i - F_{n-1}}{1 + F_{i+2} + F_{n+1} - F_{k+2}}. \qquad (3.2)
\end{aligned}$$

We recall that $0 \leq i < k \leq n-1$. According to this boundaries we can say that dividend on the right side of Equation (3.2) will be always negative or 0, since F_k is at most F_{n-1}. On the other hand, the divisor is always, by the same argument, positive. It implies that initial label α is negative number or zero, which is contradiction, i.e., there is no edge incident with v_{n+1}.

Edges between $\{v_i, v_j\}$, $i+1 \neq j$ are also forbidden. We now assume that $1 \leq i+1 < j < k < n$. As in the previous case, we rewrite the labels according to Algorithm 2

$$\begin{aligned}
\sigma(v_i) + \sigma(v_j) &= \sigma(v - k) \\
F_{i+2} \cdot \alpha + F_i + F_{j+2} \cdot \alpha + F_j &= F_{k+2} \cdot \alpha + F_k \\
\alpha \cdot (F_{i+2} + F_{j+2} - F_{k+2}) &= F_k - F_i - F_j \\
\alpha &= \frac{F_k - F_i - F_j}{F_{i+2} + F_{j+2} - F_{k+2}}. \qquad (3.3)
\end{aligned}$$

If we think about values, which can α acquire, we get that that dividend on the right side of Equation (3.3) is always positive, since at the worst case $F_j = F_{k-1}$, but the largest possible value for F_i is F_{k-3}. When we subtract this extremal values we always get a positive reminder. For the divisor, exactly the opposite is true. It implies that the divisor is always lower than zero, i.e., the initial α must be negative, which is contradiction.

Last not yet discussed case is for edges $\{v_i, v_k\}$, $1 \leq i+1 < j < n-1$, with exception of edge $\{v_0, v_{n-1}\}$, and possible witness in a vertex v_{n+1}. If we rewrite the equation $\sigma(v_i) + \sigma(v_j) = \sigma(v_{n+1})$ under the preceding condition, then we again obtain a contradiction on value of α.

We discuss all possible cases and it follows, that our labeling is valid. So the lemma holds. □

To complete the proof of the correctness of Algorithm 2 we note that the algorithm always terminates, since it perform constant amount of operations for finite number of vertices. Our algorithm, with the same argumentation as in case of Algorithm 1, has time complexity $\mathcal{O}(n^2)$, since it uses exponentially big numbers. It follows that sum labeling of cycles belongs to complexity class **P**.

Theorem 11. *Let $n \in \mathbb{N}$, $n \geq 3$, and C^n be a cycle graph. It holds that*

$$\sigma(C^n) = \begin{cases} 3 & \text{if } n = 4, \\ 2 & \text{otherwise.} \end{cases}$$

Proof. The first case follows from Lemma 9 and Figure 3.2. For the latter case we introduce deterministic that find the sum labeling of any given cycle graph with two added isolates. According to Lemma 3 there cannot be a sum labeling with less added isolates, i.e., the theorem holds. □

The preceding theorem implies that cycle graphs except the case of C^4 are all 2-optimum summable.

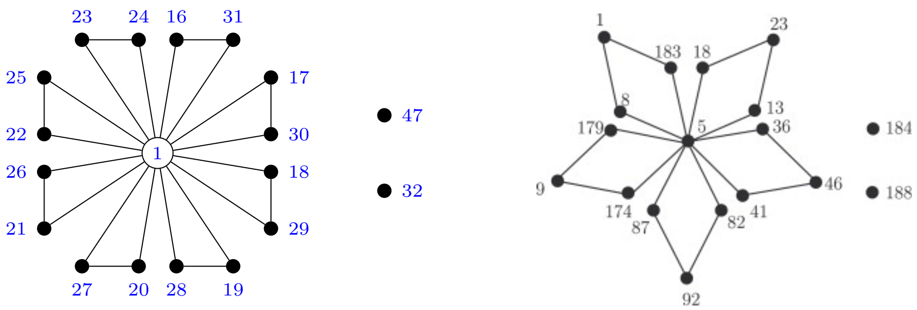
3.3 Flowers

The case of 4-cycle from Section 3.2 is quite curious. It is intellectually provocative to accept that it is impossible to label C^4 using 2 isolates. This leads to an article of Fernau, Ryan, and Sugeng [38], who investigated a sum number of *flower graphs* or *generalized friendship graphs*.

A flower $f^{q,p}$ is a collection of p, $p \geq 2$, q-cycles, $q \geq 3$, with a common vertex called the *center*. The q-cycles are called *petals*. We follow the notation of Fernau, Ryan, and Sugeng [38], and denote the central vertex by c, and for i-th, $1 \leq i \leq p$, q-cycle we denote its vertices as v^i_j, where $1 \leq j \leq q$ and $v^i_q = c$. For examples of flowers, please see Figure 3.4.

As stated before, the main motivation for study of flower graphs was the strange property of graph C^4 with respect to sum numbers. Based on that, Fernau, Ryan, and Sugeng [38, Lemma 2] proved that flower graphs $f^{4,p}$, i.e., p 4-cycles with single common vertex, are, in contrast with C^4, 2-optimal summable. They also proposed exact sum labeling schema for such graphs. An example of a graph $f^{4,p}$ with its sum labeling is in the Figure 3.4b.

In addition, Fernau, Ryan, and Sugeng [38] showed a complete characterization of all flower graphs in terms of sum numbers. We present an algorithm for labeling flower graphs with petals of size 3. Other cases can be found in the original article [38].



(a) The flower $f^{3,8}$ and its labeling   (b) The flower $f^{4,5}$ and its labeling [38]

Figure 3.4: Flowers and its labeling

Algorithm 3. *Let $f^{3,q}$, $q \geq 3$ be a flower graph and $\alpha \in \mathbb{N}$, $\alpha \geq 2p$ sufficiently large constant. We assign to vertices labels according to the following rule*

$$\sigma(v_j^i) = \begin{cases} \alpha + (i-1) & \text{if } j = 1, \\ \alpha + (2p - i) & \text{if } j = 2 \\ 1 & v_j^i = v_3^i = c. \end{cases}$$

With such assignment we need to add a single isolate z_1 with label $\alpha + (2p-1)$ to cover all the edges of the form $\{v_1^i, v_2^i\}$, and an isolate z_2 with label $\alpha + 2p$ to cover the edge incident with center and vertex with the largest label.

It follows that the central vertex c obtains label 1. Moreover, the labels of v_1^i vertices increase with i from α to $\alpha + (p-1)$ and labels of vertices v_2^i, on the other hand, with increasing i decrease from $\alpha + (2p-1)$ down to $\alpha + p$. It is also not hard to see that the proposed labeling does not induce any further edges. The sum of the lowest and the second lowest labels on petals is grater than the label of the isolate z_2. Since all the labels are distinct positive integers, the isolate z_1 cannot induce edges other than those on petals. For example of the algorithm result, please see Figure 3.4a.

Theorem 12 (Fernau, Ryan, and Sugeng [38])**.** *The flower graph $f^{p,q}$, where $p \geq 2$ and $q \geq 3$, has sum number* 2*.*

For the complete proof, we refer the reader to the original article [38]. Theorem 12 implies that all flower graphs are 2-optimum summable.

3.4 Stars

Another graph family we study is known as *star* graphs. A star graph, denoted by S^n, where $n \in \mathbb{N}$, is a graph with a vertex set $\{c, v_1, \ldots, v_{n-1}\}$ and edges $\{\{c, v_i\} \mid i \in [n-1]\}$. A vertex c is denoted as the *center*, and remaining vertices all called *beams*. It follows that the central vertex has degree $n-1$, and all other vertices have degree 1. The star graphs find their usage in computer networks design and distributed computing.

A star S^1 is a graph with one isolated vertex. Such a graph is isomorphic with P^0, and it follows that it is a sum graph. Likewise, a star graph S^2 is isomorphic with P^1, and S^3 is isomorphic with P^2, i.e., all these graphs have sum number 1.

The first little bit complicated case is a star graph S^4. We provide the following sum labeling. To the center c, we assign a label 1. Each vertex $v_i \in v_1, v_2, v_3$ than obtain a label $2+i$. For each edge $\{c, v_i\}$, excluding the edge $\{c, v_3\}$, the sum is located on vertex v_{i+1}. The edge $\{c, v_3\}$ forces us to add isolate z with label 6. It is easy to see that no new edge is induced by the added isolate. The center c is adjacent to all other vertices in $V(S^4)$ and cannot be incident with the isolate, since z has the largest label. Moreover, the sum of the two smallest labels between all beams is $3 + 4 = 7$, which is greater than $\sigma(z)$. Thus there is no induced edge and $\sigma(S^4) = 1$. According to Lemma 3, this number of isolates is optimal.

We use a similar approach inspired by the work of Ellingham [39] even for a star graph with more than 4 vertices.

Algorithm 4. *Let S^n, $n \geq 2$, be a star graph, and $\alpha \in \mathbb{N}$ a given constant. We assign to the central vertex c label α and add a single isolate v_n. Each vertex v_i, $i \in [n]$, including the isolate v_n, then obtain a label $\sigma(v_i) = n\alpha + i\alpha$.*

An alternative approach is proposed by Hao [20]. He assigns a label 1 to vertex v_1, and the center c obtains label 2. The vertex v_3 obtain a label $\sigma(c) + \sigma(v_1)$ and all the remaining vertices, including an added isolate, v_i obtain label $\sigma(v_{i-1}) + \sigma(c)$. We leave this algorithm without a proof.

However, what we formally prove is the correctness of Algorithm 4. Let us now prove the following auxiliary lemma.

Lemma 13. *Let S^n, $n \geq 4$, be a star graph and $\alpha, \beta \in \mathbb{N}$, $\beta > 2\alpha n$, be given constants. Then for a given graph S^n Algorithm 4 produces a sum graph G^+ and corresponding sum labeling σ such that $G^+ \simeq S^n \cup K^1$ and*

$$\forall u, v \in V(G^+)\colon \{u, v\} \in E(G^+) \iff \exists w \in V(G^+)\colon \sigma(u) + \sigma(v) = \sigma(w).$$

Proof.

($\Rightarrow$) At first, we must show that for each edge $\{c, v_i\}$, $i \in [n-1]$, there exists a vertex $v_k \in V(G^+)$ such that $\sigma(c) + \sigma(v_i) = \sigma(v_k)$. If we rewrite the equation, then we obtain

$$\begin{aligned}\sigma(v_k) &= \sigma(c) + \sigma(v_i)\\ \sigma(v_k) &= \alpha + n\alpha + i\alpha\\ \sigma(v_k) &= n\alpha + (i+1)\alpha. \end{aligned} \tag{3.4}$$

Equation (3.4) has always a solution $k = i + 1$. It is easy to see that thanks to the added isolate, all the edges have the corresponding witness.

($\Leftarrow$) We now prove that for two vertices $u, v \in V(S^n)$ such that $\{u, v\} \notin E(S^n)$, there is no induced edge with given labeling.

The added isolate v_n has the largest label, and according to Lemma 2, it is adjacent to any vertex $v \in V(S^n)$. Moreover, the center is connected with all vertices in $V(S^n)$. The only case we need to check is whether there is no induced edge by the added isolate.

If we sum labels of the beams with the lowest labels, we obtain $\sigma(v_1) + \sigma(v_2) = n\alpha + \alpha + n\alpha + 2\alpha = 2n\alpha + 2\alpha$. Since the largest label is $\sigma(v_n) = n\alpha + n\alpha = 2n\alpha$, we proved that there is no edge induced by this labeling. Thus the lemma holds. $\square$

Since the validity of the produced output of Algorithm 4 directly implies from Lemma 13, we only note that the algorithm always terminates. It performs a constant number of operations for every vertex of S^n, and the same procedure is executed even for the added isolate. Hence, the algorithm has running time $\sigma(n)$, and the sum labeling of stars belongs to complexity class **P**.

Theorem 14. *Let S^n, where $n \in \mathbb{N}$, be a star graph. It holds that*

$$\sigma(S^n) = \begin{cases} 0 & \text{if } n = 1, \\ 1 & \text{otherwise.} \end{cases}$$

Proof. The first case follows from our discussion that $S^1 \simeq P^0$. For the latter case, we introduced a deterministic polynomial-time algorithm and proved its correctness. Moreover, according to Lemma 3, there cannot be labeling with less added isolate, i.e., the theorem holds. □

From Theorem 14 follows, that star graphs S^n with at least two vertices are 1-optimum summable and are unit graphs. Moreover, the produced sum labeling is obliviously strong sum labeling.

3.5 Complete graphs

A *complete graph* (or a *clique*) is a graph, where every vertex v is adjacent with all other vertices $w \in V(G) \setminus \{v\}$. More formally, graph $G = (V, E)$ is complete graph on n vertices if and only if $E(G) = [V(G)]^2$. We ordinarily denote complete graphs as K^n.

Many notoriously difficult graph problems are easy on complete graphs thanks to their symmetry. As an example, let us mention for example the GRAPH COLORING problem, CLIQUE problem, and VERTEX COVER.

High symmetry is advantageous even in the case of sum labeling. We are not forced to invent highly sophisticated schema because there is no need to worry about unwanted connections in the graph.

Sum labeling of complete graphs was first studied by Bergstrand et al. [40]. They introduced polynomial time algorithm which uses $2n - 3$ isolated vertices. Moreover, they proved that such number of isolates is optimal for complete graphs, i.e., $\sigma(K^n) = 2n - 3$ for $n \geq 4$. Let us now formulate little bit generalized algorithm inspired by their approach.

Algorithm 5. *Let K^n, $n \geq 4$, be a complete graph with vertices $\{v_1, \ldots, v_n\}$. We assign to each vertex v_i label $1 + 4 \cdot (i - 1)$ and we add $2n - 3$ isolates z_j, $1 \leq j \leq 2n - 3$, with corresponding labels equal to $2 + 4j$.*

Lemma 15. *Let K^n, $n \geq 4$, be a complete graph. Then Algorithm 5 produces a sum graph G^+ and a labeling $\sigma\colon V(G^+) \to \mathbb{N}$ such that $G^+ \simeq K^n \cup \overline{K^{2n-3}}$ and*

$$\forall u, v \in V(G^+)\colon \{u, v\} \in E(G^+) \iff \exists w \in V(G^+)\colon \sigma(u) + \sigma(v) = \sigma(w).$$

Proof.

($\Rightarrow$) Let v_i, v_j, $1 \leq i < j \leq n$, be two vertices. We claim that there exists an isolated vertex z_k, $1 \leq k \leq 2n - 3$, such that $\sigma(z_k) = \sigma(v_i) + \sigma(v_j)$. We can substitute into the equation and we obtain

$$\begin{aligned} \sigma(z_k) &= 1 + 4(i-1) + 1 + 4(j-1) \\ 2 + 4k &= 2 + 4(i+j-2) \\ k &= i + j - 2 \end{aligned} \tag{3.5}$$

The minimum value on the right-hand of Equation (3.5) is $i + j - 2 = 1 + 2 - 2 = 1$ and the maximum value is $(n-1) + n - 2 = 2n - 3$. All other possible combinations of i and j, with respect to assumed boundaries, fall into this interval and all of them are integers. This implies that the witness always exists.

($\Leftarrow$) We must check that there is no edge induced by added isolates. Suppose that there is a vertex $v \in V(G^+)$ such that $\sigma(v) = \sigma(v_i) + \sigma(z_j)$, where $v_i \in V(K^n)$ and $z_j \in V(G^+) \setminus V(K^n)$. If we rewrite the equation, we obtain $\sigma(v) = 1 + 4(i-1) + 2 + 4j = 3 + 4(i+j-1)$. But there is no vertex in $V(G^+)$ with label congruent to 3 (mod 4).

What remains, is to prove that there is no induced edge between two isolates. Let us suppose that there is a vertex $v \in V(G^+)$ such that for two distinct vertices $z_i, z_j \in V(G^+) \setminus V(K^n)$ it holds that $\sigma(v) = \sigma(z_i) + \sigma(z_j)$. It is easy to see that $2 + 4i + 2 + 4j = 4(1 + i + j)$, but we do not assign label congruent to 0 (mod 4) to any vertex in v. Thus, a vertex v does not exist. $\square$

Theorem 16. *Let $n \in \mathbb{N}$, $n \geq 1$, and K^n be a complete graph. It holds that*

$$\sigma(K^n) = \begin{cases} 0 & \text{if } n = 1, \\ 1 & \text{if } n = 2, \\ 2 & \text{if } n = 3, \\ 2n - 3 & \text{otherwise.} \end{cases}$$

Proof. A complete graph with a single vertex is isomorphic to P^0 and K^2 is isomorphic to path on two vertices. By Theorem 8, $\sigma(P^0) = 0$ and $\sigma(P^1) = 1$. Moreover, a graph K^3 is isomorphic to C^3 and we show in Theorem 11 that $\sigma(C^3) = 2$. So the theorem holds for the first three cases.

For the latter case we proved deterministic polynomial-time algorithm which find proper sum labeling using $2n-3$ isolates. It follows that $\sigma(K^n)$, $n \geq 4$, is at most $2n-3$. We must now show that at least $2n - 3$ isolates are necessary, i.e., $\sigma(K^n) \geq 2n - 3$. To complete the proof we need the following auxiliary lemma of Bergstrand et al. [26].

Lemma 17. *Using the above labeling of $G^+ = K^n \cup \overline{K^{2n-3}}$ there is no $i, j, k \leq n$ such that $\sigma(v_i) + \sigma(v_j) = \sigma(v_k)$.*

We now consider two sets $A = \{\sigma(v_1) + \sigma(v_i) \mid i \in \{2, \ldots, n\}\}$ and $B = \{\sigma(v_n) + \sigma(v_n) \mid i \in \{2, \ldots, n-2\}\}$. The size of a set A is $n - 1$ and the size of a set B is $n - 2$. It is easy to see that $A \cap B = \emptyset$. Thus, by Lemma 17 we obtain $\sigma(K^n) \geq |A| + |B| = n - 1 + n - 2 = 2n - 3$ and the proof is complete. $\square$

3.6 Other graph families

There are further graph families, for which the exact sum number is known. We do not study these families in full detail here, since they are not as significant as those examined above. Therefore, we present only summary of known results without formal proofs or algorithms.

Trees Let G be a graph. If G contains no cycle and is connected, then G is called a *tree* graph, ordinarily denoted by T^n, where $n = |V(G)|$. A single selected vertex of T^n is called *root* and all other vertices with degree 1 are called *leaves*.

The study of a sum number of trees was initiated by Harary [2], who proposed a sum labeling of a caterpillar graphs — a *caterpillar* is a tree with the property that the removal of its leaves creates a path graph [27] — and formed following conjecture.

Conjecture 1 (True Conjecture [2]). *Every nontrivial tree graph T^n has $\sigma(T^n) = 1$, i.e., $T^n \cup K^1$ is a sum graph.*

Efforts to confirm or disprove the conjecture were completed by Ellingham [39] in 1993. He proposed an exact labeling algorithm for any nontrivial tree and proved the Harary's conjecture.

Complete bipartite graphs We say that a graph G is *complete bipartite graph* if and only if the set of vertices can be divided into two disjoint sets A and B such that vertices from one set are adjacent to every vertex from the second set, and there is no edge between two vertices from the same set.

The first upper bound is due to Bergstrand [40] who showed that $\sigma(k^{m,n}) \leq m+n-1$, for $m, n \geq 2$. In 1992, Hartsfield and Smyth [41] proved that $\sigma(K^{m,n}) = \lceil 3m + n - 3 \rceil$, $m \geq n$, but Yan and Liu [42] found a flaw in the proof. In multiple consecutive articles [43, 44, 45, 46] the following theorem was established.

Theorem 18 ([27]). *For a complete bipartite graph $K^{m,n}$, where $2 \leq m \leq n$, it holds that*

$$\sigma(K^{m,n}) = \left\lceil \frac{n}{p} + \frac{(p+1)(m-1)}{2} \right\rceil,$$

where $p = \left\lceil \sqrt{\frac{2n}{m-1} + \frac{1}{4}} - \frac{1}{2} \right\rceil$.

Wheels A *wheel* graph, ordinarily denoted as W^n, is a graph with at least three vertices $\{c, v_1, \ldots, v_n\}$ and a set of edges $\{\{c, v_i\}, \{v_i, v_{i+1}\} \mid i \in [n-1]\} \cup \{\{c, v_n\}, \{v_1, v_n\}\}$.

The study of wheels and their sum numbers was initiated by Hartsfield and Smyth [47]. They proved that for wheels it holds $\sigma(W^n) \in \Theta(|E(W^n)|)$ which was significant result, since all the graph families for which the construction have been found, it was $\sigma(G) \in o(|E(G)|)$. Harstfield and Smyth also conjectured some exact sum numbers for wheels, but they were later disproved.

Later Miller et al. [48, 49] in two consecutive articles proposed an optimal labeling procedures with respect to sum number and provided complete proofs of them. We conclude this section with their theorem about sum numbers of wheel graphs.

Theorem 19 (Miller et al. [48, 49])**.** *The sum number of a wheel graph W^n is*

$$\sigma(W^n) = \begin{cases} 2 & \textit{if } n = 2 \\ 5 & \textit{if } n = 3 \\ 4 & \textit{if } n = 4 \\ n & \textit{if } n \geq 5,\ n \textit{ odd} \\ \frac{n}{2} + 2 & \textit{if } n \geq 6,\ n \textit{ even} \end{cases}$$

Fans A *fan* graph, ordinarily denoted as F^n, where $n \geq 1$, is a graph with a vertex set $\{c, v_1, \ldots, v_n\}$ and an edge set $\{\{c, v_i\} \mid i \in [n]\} \cup \{\{v_i, v_{i+1}\} \mid i \in [n-1]\}$. Fan graphs were studied by Dou and Gao [50] from all perspectives of sum labeling. They provided not only a complete characterization of fans and their sum numbers, but they also investigated the variant of integral and mod sum numbers. For us the significant result of they work is the following theorem.

Theorem 20 (Dou and Gao [50])**.** *The sum number of a fan graph F^n is*

$$\sigma(F^n) = \begin{cases} 1 & n = 1, \\ 2 & n = 2 \textit{ or } n = 4, \\ 3 & n = 3 \textit{ or } n \geq 6 \textit{ and } n \textit{ even}, \\ 4 & n \geq 5 \textit{ and } n \textit{ odd}. \end{cases}$$

3.7 Summary of known sum numbers

In the last section of this chapter, we would like to provide to the reader a clear summary of the sum numbers for studied graph families. In addition to the individual sum numbers we provide also references to original articles.

Table 3.1: Summary of known sum numbers

Graph family	$\sigma(G)$	**References**
path P^n	$\sigma(P^n) = \begin{cases} 0 & n = 0, \\ 1 & n \geq 1. \end{cases}$	[2, 20]
cycle C^n	$\sigma(C^n) = \begin{cases} 3 & n = 4, \\ 2 & n \geq 3, n \neq 4. \end{cases}$	[2]
flower $f^{p,q}$	$\sigma(f^{p,q}) = 2$, $p \geq 3$, $q \geq 2$.	[38]
star S^n	$\sigma(S^n) = \begin{cases} 0 & n = 1, \\ 1 & n \geq 2. \end{cases}$	[39, 20]
clique K^n	$\sigma(K^n) = \begin{cases} 0 & n = 1, \\ 1 & n = 2, \\ 2 & n = 3, \\ 2n-3 & n \geq 4. \end{cases}$	[40]
tree T^n	$\sigma(T^n) = \begin{cases} 0 & n = 1 \\ 1 & n \geq 2. \end{cases}$	[39]
bipartite $K^{m,n}$	$\sigma(K^{m,n}) = \left\lceil \frac{n}{p} + \frac{(p+1)(m-1)}{2} \right\rceil$, $p = \left\lceil \sqrt{\frac{2n}{m-1} + \frac{1}{4}} - \frac{1}{2} \right\rceil$.	[46]
wheel W^n	$\sigma(W^n) = \begin{cases} 2 & n = 2, \\ 5 & n = 3, \\ 4 & n = 4, \\ n & n \geq 5,\ n \text{ odd}, \\ \frac{n}{2}+2 & n \geq 6,\ n \text{ even}. \end{cases}$	[47, 48, 49]
fan F^n	$\sigma(F^n) = \begin{cases} 1 & n = 1, \\ 2 & n = 2 \text{ or } n = 4, \\ 3 & n = 3 \text{ or } n \geq 6 \text{ and } n \text{ even}, \\ 4 & n \geq 5 \text{ and } n \text{ odd}. \end{cases}$	[50]

Chapter **4**

General algorithm

One of the main goals in the field of sum labeling theory is to invent an algorithm that finds proper sum labeling schema for any given general graph while using the minimal number of isolated vertices. Let us now formally define this problem.

Sum number
Input: A graph $G = (V, E)$.
Output: The value of $\sigma(G)$.

None of the researchers with publications in this area made any significant breakthrough so far. Only algorithms for a few graph classes, introduced in the previous chapter, are known. Moreover, these algorithms have not many common properties, so while developing a general algorithm, it is not the best idea to start with them.

Based on this observation, we investigated different approaches during our effort to develop such a generally applicable algorithm.

At first, we investigate the computational complexity of the problem of finding sum number of a general graph G. One of the most important result on the field of computational complexity for sum number problem is the following theorem of Kratochvíl, Miller, and Nguyen [51] which upper bounds the maximum required label.

Theorem 21 (Kratochvíl, Miller, and Nguyen [51])**.** *Let $G = (V, E)$ be a graph. There is a labeling $\sigma\colon V(G) \to \mathbb{N}$ of graph $G^+ = G \cup \overline{K_{\sigma(G)}}$ such that the largest label has a value at most 4^n.*

Theorem 21 directly implies that the problem of the sum number a determination for general graph belongs to complexity class **NP**, since the certificate has polynomial-size — it consists out of an explicit set of at most n^2 numbers of size at most 4^n, i.e., the certificate has the length at most n^3, where n is a number of vertices of the given graph.

Before we introduce our generally applicable algorithms, let us now define the following auxiliary problem.

k-sum labeling

Input: A graph $G = (V, E)$ and a positive integer $k \in \mathbb{N}^0$.

Output: A graph $G^+ = G \cup \overline{K^k}$ and a function $\sigma \colon V(G^+) \to \mathbb{N}$ such that σ is sum labeling of G^+. If no such function exists, then an empty graph is returned.

It is clear that if we are able to solve the k-sum labeling problem, then the solution of the Sum number problem is easily obtained as the minimum k such that the algorithm for k-sum labeling returned non-empty graph. Moreover, as Gould and Rödl [30] proved, and as we show in Lemma 4, we must check only the $k \in \{0, \ldots, |E(G)|\}$. This number of different executions of the algorithm for k-sum labeling can be, thanks to Lemma 6, actually decreased to $\mathcal{O}(\log(|V(G)|^2)) = \mathcal{O}(\log(|V(G)|))$ if we use a binary search algorithm to find lowest applicable k.

Based on the previous paragraph, we research only algorithms for solving k-sum labeling, since the algorithm for sum number problem is obvious.

4.1 Brute-force algorithm

Brute-force algorithms are typically not the most effective algorithms for solving problems. On the other hand, they are always simple, check whole state space, and can be easily modified.

Since the brute-force algorithms always relatively straightforwardly follow the definition of the problem, without exception in our case, we would like to prepare some more theoretical background.

At first, the definition of the k-sum labeling problem has heavy use of the graph isomorphism problem, i.e., deciding whether two finite graphs are isomorphic. It is easy to see that this problem belongs to the **NP** complexity class. On the other hand, it is still unknown whether the problem belongs to **P** or is **NP**-complete. However, it is known [52] that the graph isomorphism problem is in the low hierarchy of class **NP**, which implies that this problem is not **NP**-complete unless the polynomial-time hierarchy collapses to some finite level. [52]

Currently, the best-known and a widely accepted algorithm is due to Luks [53] with running time $2^{\mathcal{O}(\sqrt{n \log n})}$. In 2017, Babai [54] published an algorithm for graph isomorphism with running time $2^{\mathcal{O}((\log n)^{\mathcal{O}(1)})}$, but the results are not entirely peer-reviewed at the time of writing this thesis.

With the necessary auxiliary procedures, we can finally introduce our brute-force algorithm for k-sum labeling of general graph G.

Algorithm 6. *Let G be a graph, $k \in \mathbb{N}^0$ be an integer, and by n, we denote the sum of $|V(G)|$ and k. For every n-subset S of a set $[4^n]$ we check whether obtained sum graph $G^+(S)$ is isomorphic with graph $G \cup \overline{K^k}$. If no subset S satisfies the previous condition, then return an empty graph. Otherwise, the solution is any matching subset S.*

Lemma 22. *Algorithm 6 is correct and has a running time*

$$\left(\binom{4^n}{n} \cdot 2^{\mathcal{O}\left(\sqrt{n \log n}\right)}\right),$$

where $n = |V(G)| + k$.

Proof. Our algorithm always terminates, since every subset is checked exactly once. Moreover, the result given by the algorithm is correct because we investigate all possible subsets of a set $[4^n]$. By Theorem 21, we can be sure that if the sum labeling of graph G with k added isolates exist, then it is induced by at least a single n-subset in $[4^n]$.

Let us now discuss the running time of algorithm 6. As stated before, the algorithm tries every n-subset of the set $[4^n]$ exactly once. For each n-subset S, the algorithm constructs the sum graph $G^+(S)$, which can be clearly done in polynomial time, and calls algorithm of Luks [53], which decides whether $G^+(S)$ is isomorphic to $G \cup \overline{K^k}$ in time $2^{\mathcal{O}(\sqrt{n \log n})}$. Since we check at most $\binom{4^n}{n}$ different n-subsets, the overall running time is $\mathcal{O}\left(\binom{4^n}{n} \cdot 2^{\mathcal{O}\left(\sqrt{n \log n}\right)}\right)$, and the lemma holds. □

4.2 Integer linear programming approach

The second algorithm we study is based on Integer linear programming (ILP). Integer linear programming is a method of mathematical optimization. A large number of optimization problems can be modeled using ILP, despite that Karp proved [16] that solving integer linear program is **NP**-*complete*.

Usually, the ILP program is given by a set X of variables $x_i, \ldots, x_n$, and by a list of linear inequalities over X called constraints. The goal is to maximize (or minimize) the sum of the values substituted for variables without violating constraints. For an exhaustive reference, we refer the reader to the monograph of Conforti, Cornuéjols, and Zambelli [55].

We propose the following integer linear program, which solves the k-Sum labeling problem for an arbitrary graph.

Algorithm 7. *The input of the algorithm is a graph G and a positive integer $k \in \mathbb{N}^0$. The goal is to determine whether there exists any sum labeling of graph $G^+ = G \cup \overline{K^k}$.*

We propose the following integer linear program. By n we denote the sum $|V(G)|+k$. The program contains a variable for each combination of a vertex $v \in V(G^+)$ and possible label $\ell \in [4^n]$. If the following program have no feasible solution, we output an empty *graph. Otherwise, the required label for vertex v is the ℓ such that $x_{v,\ell}$ is 1.*

$$
\begin{array}{rl}
\textbf{variables} & \forall v \in V(G^+)\colon \forall \ell \in [4^n]\colon x_{v,\ell} \\
\textbf{minimize} & \displaystyle\sum_{\ell=1}^{4^n} x_{v,\ell} \cdot \ell \\
\textbf{subject to} & \forall v \in V(G^+)\colon \displaystyle\sum_{\ell \in [4^n]} x_{v,\ell} = 1 \\
 & \forall \{u,v\} \in E(G)\colon \forall a \in [4^n]\colon \displaystyle\sum_{\{\ell,\ell'\} \in [a]^2} [x_{u,\ell} + x_{v,\ell'} = 2] = \sum_{w \in V(G^+)} x_{w,a} \\
\textbf{and} & \forall v \in V(G^+)\colon \forall \ell \in [4^n]\colon x_{v,\ell} \in \mathbb{N}
\end{array}
$$

We stated that ILP $\in$ **NP**-*complete*, which implies that our program runs at best in exponential time, i.e., there is no significant improvement against the algorithm from Section 4.1. In contrast to the above brute-force algorithm, there are a large number of high-performance solvers for ILP [56], which are able to solve many instances of integer linear programs very efficiently.

Conclusion

In this thesis, we provided an almost complete overview of known results on the field of sum graphs and sum numbers. We rigorously proved many sum graphs properties that repeatedly appear in different articles on the topic of study as claims only.

In the second part of this thesis, we investigate sum numbers of different graph families. We provided many possibly new or generalized labeling algorithms for many of them. All the algorithms are thoroughly proved. Thus we get rid of the addiction on the personal correspondence of the original authors, which probably no longer exists today.

The last part is devoted to general algorithms that find the sum number for an arbitrary graph G. We proposed two algorithms. The first is the brute-force approach, wheres the second one is based on INTEGER LINEAR PROGRAMMING, which is a very useful approach in practice with a lot of very powerful solvers. Although the running time of our algorithms is not optimal, we believe that they can be a good basis for future research on the field of sum graphs.

Future work

There remain many open problems. We showed that the SUM NUMBER problem belongs to complexity class **NP**, but the complete characterization in terms of **NP**-*completeness* is still missing. Resolving this characterization could open up new challenges in the area of parameterized complexity and multivariate analysis.

Besides that, there are many graph families for which the exact sum number is still not known. As stated before, Gould and Rödl [30] proved that there is an infinite graph family such that $\sigma(G) = \Theta(|V(G)|^2)$. Their proof, however, uses probabilistic arguments and no representative of this family is known so far.

Besides that, we already said that the running time of the general algorithm is far from optimal, so there is a wide space for improvement.

Bibliography

[1] Harary, F. Sum graphs and difference graphs, 1988, talk at the Nineteenth Southeastern Conference on Combinatorics, Graph Theory and Computing.

[2] Harary, F. Sum graphs and difference graphs. *Congressus Numerantium*, volume 72, 1990: pp. 101–108, ISSN 0384-9864.

[3] Slamet, S.; Sugeng, K.; et al. Sum graph based access structure in a secret sharing scheme. *Journal of Prime Research in Mathematics*, volume 2, Jan. 2006: pp. 113–119.

[4] Hajos, G. Über eine Art von Graphen. *Internationale Mathematische Nachrichten*, 1957, problem 65.

[5] Benser, S. On the topology of genetic fine structure. *Proceedings of the National Academy of Sciences of the USA*, volume 45, 1959: pp. 1607–1620.

[6] Mahadev, N.; Peled, U. Chapter 6 - The Threshold Dimension. In *Threshold Graphs and Related Topics*, *Annals of Discrete Mathematics*, volume 56, edited by N. Mahadev; U. Peled, Elsevier, 1995, pp. 123 – 138, doi:https://doi.org/10.1016/S0167-5060(06)80008-7. Available from: `http://www.sciencedirect.com/science/article/pii/S0167506006800087`

[7] Ehrlich, G.; Even, S.; et al. Intersection graphs of curves in the plane. *Journal of Combinatorial Theory, Series B*, volume 21, no. 1, 1976: pp. 8 – 20, ISSN 0095-8956, doi:https://doi.org/10.1016/0095-8956(76)90022-8. Available from: `http://www.sciencedirect.com/science/article/pii/0095895676900228`

[8] Diestel, R. *Graph Theory*, *Graduate texts in mathematics*, volume 173. Springer, fifth edition, 2017, ISBN 978-3-662-53621-6.

[9] Vorobiev, N. N. *Fibonacci Numbers*. Birkhäuser, 2002, ISBN 978-3-7643-6135-8, doi:10.1007/978-3-0348-8107-4.

[10] Graham, R. L.; Knuth, D. E.; et al. *Concrete Mathematics: A Foundation for Computer Science*. USA: Addison-Wesley Longman Publishing Co., Inc., second edition, 1994, ISBN 978-0-201-55802-9.

[11] Arora, S.; Barak, B. *Computational Complexity: A Modern Approach*. USA: Cambridge University Press, first edition, 2009, ISBN 978-0-521-42426-4.

[12] Matoušek, J.; Nešetřil, J. *Invitation to Discrete Mathematics*. Oxford University Press, second edition, Jan. 2009, ISBN 978-0-19-857042-4.

[13] Cook, S. A.; Reckhow, R. A. Time-Bounded Random Access Machines. In *Proceedings of the Fourth Annual ACM Symposium on Theory of Computing*, STOC '72, New York, NY, USA: Association for Computing Machinery, 1972, ISBN 9781450374576, p. 73–80, doi:10.1145/800152.804898. Available from: `https://doi.org/10.1145/800152.804898`

[14] Turing, A. M. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, volume s2-42, no. 1, 01 1937: pp. 230–265, ISSN 0024-6115, doi:10.1112/plms/s2-42.1.230, `https://academic.oup.com/plms/article-pdf/s2-42/1/230/4317544/s2-42-1-230.pdf`. Available from: `https://doi.org/10.1112/plms/s2-42.1.230`

[15] Cook, S. A. The Complexity of Theorem-Proving Procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, New York, NY, USA: Association for Computing Machinery, 1971, ISBN 978-1-4503-7464-4, pp. 151–158, doi:10.1145/800157.805047. Available from: `https://doi.org/10.1145/800157.805047`

[16] Karp, R. M. Reducibility among Combinatorial Problems. In *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations*, edited by R. E. Miller; J. W. Thatcher; J. D. Bohlinger, Boston, MA: Springer, Mar. 1972, ISBN 978-1-4684-2001-2, pp. 85–103, doi:10.1007/978-1-4684-2001-2_9. Available from: `https://doi.org/10.1007/978-1-4684-2001-2_9`

[17] Harary, F.; Palmer, E. M. *Graphical Enumeration*. Academic Press, May 1973, ISBN 978-0-12-324245-7, doi:10.1016/C2013-0-10826-4.

[18] Smyth, W. F. Sum graphs of small sum number. *Colloquia Mathematica Societatis Janos Bolyai*, volume 60, 1991: pp. 669–678.

[19] Koh, K. M.; Miller, M.; et al. On optimum summable graphs. *AKCE International Journal of Graphs and Combinatorics*, volume 3, no. 1, 2006: pp. 45–57.

[20] Hao, T. On Sum Graphs. *Mathematical Preprint*, volume 7, 1988.

[21] Harary, F. Sum graphs over all the integers. *Discrete Mathematics*, volume 124, no. 1, Jan. 1994: pp. 99–105, ISSN 0012-365X, doi:10.1016/0012-365X(92)00054-U. Available from: `http://www.sciencedirect.com/science/article/pii/0012365X9200054U`

[22] Boland, J.; Laskar, R.; et al. On mod sum graphs. *Congressus Numerantium*, volume 70, 1990: pp. 131–135.

[23] Harary, F.; Hentzel, I.; et al. Digitizing Sum Graphs Over the Reals. *Caribbean Journal of Mathematical and Computing Sciences*, volume 1, Jan. 1991: pp. 1–4.

[24] Bloom, G. S.; Hell, P.; et al. Collecting autographs: n-node graphs that have n-integer signatures. *Annals of the New York Academy of Sciences*, volume 319, 1979: pp. 93 – 102.

[25] Bloom, G. S.; Burr, S. A. On autographs which are complements of graphs of low degree. *Caribbean Journal of Mathematical and Computing Sciences*, volume 3, 1984: pp. 17 – 28.

[26] Bergstrand, D.; Hodges, K.; et al. Product Graphs Are Sum Graphs. *Mathematics Magazine*, volume 65, no. 4, 1992: pp. 262–264, ISSN 0025570X, 19300980.

[27] Gallian, J. A Dynamic Survey of Graph Labeling. *The Electronic Journal of Combinatorics*, volume 22, Dec. 2019.

[28] Nagamochi, H.; Miller, M.; et al. Bounds on the number of isolates in sum graph labeling. *Discrete Mathematics*, volume 240, no. 1, 2001: pp. 175 – 185, ISSN 0012-365X, doi:https://doi.org/10.1016/S0012-365X(00)00390-3. Available from: `http://www.sciencedirect.com/science/article/pii/S0012365X00003903`

[29] Miller, M.; Patel, D.; et al. Exclusive sum labeling of graphs. *Journal of Combinatorial Mathematics and Combinatorial Computing*, volume 55, Jan. 2005: pp. 137–148.

[30] Gould, R. J.; Rödl, V. Bounds on the number of isolated vertices in sum graphs. In *Graph Theory, Combinatorics, and Applications*, edited by Y. Alevi; G. Chartrand; O. R. Oellermann; A. J. Schwenk, New York: Wiley, 1991, pp. 553–562.

[31] Wikipedia contributors. Bohemian Crown Jewels — Wikipedia, The Free Encyclopedia. 2020, [Online; accessed 24-July-2020]. Available from: `https://en.wikipedia.org/w/index.php?title=Bohemian_Crown_Jewels&oldid=949326457`

[32] Liu, C. L. *Introduction to Combinatorial Mathematics*. Computer Science Series, New York: McGraw-Hill, first edition, June 1968, ISBN 978-0-07-038124-7.

[33] Shamir, A. How to Share a Secret. *Communications of the ACM*, volume 22, no. 11, Nov. 1979: p. 612–613, ISSN 0001-0782, doi:10.1145/359168.359176. Available from: https://doi.org/10.1145/359168.359176

[34] Blakley, G. R. Safeguarding cryptographic keys. In *Managing Requirements Knowledge, International Workshop on*, Los Alamitos, CA, USA: IEEE Computer Society, June 1979, p. 313, doi:10.1109/AFIPS.1979.98. Available from: https://doi.ieeecomputersociety.org/10.1109/AFIPS.1979.98

[35] Brickell, E. F.; Davenport, D. M. On the Classification of Ideal Secret Sharing Schemes. *Journal of Cryptology*, volume 4, no. 2, Jan. 1991: p. 123–134, ISSN 0933-2790, doi:10.1007/BF00196772. Available from: https://doi.org/10.1007/BF00196772

[36] Stinson, D. R. Decomposition constructions for secret-sharing schemes. *IEEE Transactions on Information Theory*, volume 40, no. 1, 1994: pp. 118–125.

[37] Sonntag, M.; Teichert, H.-M. Sum numbers of hypertrees. *Discrete Mathematics*, volume 214, no. 1, 2000: pp. 285–290, ISSN 0012-365X, doi:https://doi.org/10.1016/S0012-365X(99)00307-6. Available from: http://www.sciencedirect.com/science/article/pii/S0012365X99003076

[38] Fernau, H.; Ryan, J. F.; et al. A sum labelling for the generalised friendship graph. *Discrete Mathematics*, volume 308, no. 5, 2008: pp. 734–740, ISSN 0012-365X, doi:10.1016/j.disc.2007.07.059, selected Papers from 20th British Combinatorial Conference. Available from: http://www.sciencedirect.com/science/article/pii/S0012365X07005122

[39] Ellingham, M. N. Sum graphs from trees. *Ars Combinatorica*, volume 35, 1993: pp. 335–349.

[40] Bergstrand, D.; Harary, F.; et al. The Sum Number of a Complete Graph. *Bulletin of the Malaysian Mathematical Sciences Society*, volume 12, 1989: pp. 25–28.

[41] Hartsfield, N.; Smyth, W. F. The sum number of complete bipartite graphs. In *Graphs and Matrices*, edited by R. Rees, New York: Marcel Dekker, 1992, pp. 205–211.

[42] Yan, W.; Liu, B. Some results on integral sum graphs. *Discrete Mathematics*, volume 240, no. 1, 2001: pp. 219 – 229, ISSN 0012-365X, doi:https://doi.org/10.1016/S0012-365X(00)00118-7. Available from: http://www.sciencedirect.com/science/article/pii/S0012365X00001187

[43] Pyatkin, A. New formula for the sum number for the complete bipartite graphs. *Discrete Mathematics*, volume 239, no. 1, 2001: pp. 155 – 160, ISSN 0012-365X, doi:https://doi.org/10.1016/S0012-365X(01)00188-1. Available from: http://www.sciencedirect.com/science/article/pii/S0012365X01001881

[44] Liaw, S.-C.; Kuo, D.; et al. Integral sum numbers of graphs. *Ars Combinatorica*, volume 54, 2000: pp. 259–268.

[45] Wang, Y.; Liu, B. The sum number and integral sum number of complete bipartite graphs. *Discrete Mathematics*, volume 239, no. 1, 2001: pp. 69 – 82, ISSN 0012-365X, doi:https://doi.org/10.1016/S0012-365X(01)00195-9. Available from: `http://www.sciencedirect.com/science/article/pii/S0012365X01001959`

[46] He, W.; Shen, Y.; et al. The integral sum number of complete bipartite graphs Kr,s. *Discrete Mathematics*, volume 239, no. 1, 2001: pp. 137 – 146, ISSN 0012-365X, doi:https://doi.org/10.1016/S0012-365X(00)00172-2. Available from: `http://www.sciencedirect.com/science/article/pii/S0012365X00001722`

[47] Hartsfield, N.; Smyth, W. A family of sparse graphs of large sum number. *Discrete Mathematics*, volume 141, no. 1, 1995: pp. 163–171, ISSN 0012-365X, doi:https://doi.org/10.1016/0012-365X(93)E0196-B. Available from: `http://www.sciencedirect.com/science/article/pii/0012365X93E0196B`

[48] Miller, M.; Ryan, J.; et al. Labelling wheels for the minimum sum number. *Journal of Combinatorial Mathematics and Combinatorial Computing*, volume 28, 1998: pp. 289–297.

[49] Sutton, M.; Miller, M. On the sum number of wheels. *Discrete Mathematics*, volume 232, no. 1, 2001: pp. 185 – 188, ISSN 0012-365X, doi:https://doi.org/10.1016/S0012-365X(00)00347-2. Available from: `http://www.sciencedirect.com/science/article/pii/S0012365X00003472`

[50] Dou, W.; Gao, J. The (mod, integral) sum numbers of fans and Kn,n-E(nK2). *Discrete Mathematics*, volume 306, no. 21, 2006: pp. 2655–2669, ISSN 0012-365X, doi:https://doi.org/10.1016/j.disc.2006.05.019. Available from: `http://www.sciencedirect.com/science/article/pii/S0012365X06003906`

[51] Kratochvíl, J.; Miller, M.; et al. Sum graph labels – An upper bound and related problems. In *Proceeding of AWOCA 2001*, edited by E. Baskoro, Institut Teknologi Bandung, Indonesia, 2001, pp. 126–131.

[52] Schöning, U. Graph isomorphism is in the low hierarchy. *Journal of Computer and System Sciences*, volume 37, no. 3, 1988: pp. 312–323, ISSN 0022-0000, doi:10.1016/0022-0000(88)90010-4. Available from: `http://www.sciencedirect.com/science/article/pii/0022000088900104`

[53] Babai, L.; Luks, E. M. Canonical Labeling of Graphs. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, STOC '83, New York, NY, USA: Association for Computing Machinery, 1983, ISBN 978-0-89791-099-6, p. 171–183, doi:10.1145/800061.808746. Available from: `https://doi.org/10.1145/800061.808746`

[54] Babai, L. Graph Isomorphism in Quasipolynomial Time. 2017. Available from: https://arxiv.org/abs/1512.03547

[55] Conforti, M.; Cornuéjols, G.; et al. Cham: Springer International Publishing, 2014, ISBN 978-3-319-11008-0, doi:10.1007/978-3-319-11008-0. Available from: https://doi.org/10.1007/978-3-319-11008-0

[56] Luppold, A.; Oehlert, D.; et al. Evaluating the performance of solvers for integer-linear programming. 2018, doi:10.15480/882.1839. Available from: http://tubdok.tub.tuhh.de/handle/11420/1842

Appendix **A**

Contents of enclosed CD

```
│ readme.txt ................................ the file with CD contents description
├── src
│   └── thesis .................... the source codes for the thesis in the LaTeX format
└── text ........................................................... the thesis text
    ├── thesis.pdf .......................... the compiled thesis in the PDF format
    └── thesis.ps ............................. the compiled thesis in the PS format
```