CHARLES UNIVERSITY

FACULTY OF MATHEMATICS AND PHYSICS

LOCALLY INJECTIVE HOMOMORPHISMS

Doctoral Thesis

JIŘÍ FIALA

Department of Applied Mathematics
Malostranské nám 25
Prague, Czech Republic

2000

Mým rodičům

I would like to express my sincere thanks to professor Jan Kratochvíl, for his friendly and careful supervision during my schooldays at Charles University. Without his support and encouragement, I would hardly explore the fascinating and intriguing world of discrete mathematics.

I also thank all KAM-DIMATIA people, who carried the fruitful and friendly atmosphere at the department, and especially to professor Jaroslav Nešetřil, Jan Arne Telle, Ton Kloks and Rolf Niedermeier, who inspired and critically examined several arguments used in the thesis.

I am much obliged Alena Fialová, my closest friend, for her tender support during the nice years of our common life.

But the deepest thanks belong to my parents for their patience and love.

Prague, July 4, 2000

Prohlašuji, že jsem tuto práci vypracoval samostatně a že jsem použil pouze literaturu uvedenou v seznamu. Souhlasím se zapůjčováním této práce.

Contents

Introduction **3**

1 Definitions **7**

1.1 Graphs 8

1.1.1 Subgraphs, minors 9

1.1.2 Examples of graphs 10

1.1.3 Graph drawing 11

1.1.4 Connectivity 11

1.1.5 Morphism on graphs, coloring, factors 12

1.1.6 Covers, partial covers 14

1.2 Computational Complexity 15

2 Covers **20**

2.1 Structural behavior of covers and partial covers 20

2.1.1 Introduction 20

2.1.2 Degree refinement 21

2.1.3 Marked products 24

2.1.4 Common covers 26

2.1.5 Colored directed multigraphs 28

2.2 Computational complexity of the H-COVER problem 30

2.2.1 Complexity of covering sparse graphs 31

2.2.2 Results review 32

2.2.3 Covers of regular graphs 35

2.2.4 Covers of cyclic graphs 38

3 The H-partial cover problem **42**

3.1 Proof techniques 43

3.1.1 Subset of halfedges 43
3.1.2 Edge precoloring extension 45
3.2 Results 48
3.2.1 Two distinct parameters 49
3.2.2 Three parameters 55
3.3 Further research 61

4 The λ-labeling problem **62**
4.1 Motivation 62
4.2 The $\lambda_{(2,1)}$-labeling problem 65
4.2.1 Partial covers and generalized $L(2,1)$-labeling 66
4.2.2 Computational complexity of the $\lambda(2,1)$-labeling problem 67
4.2.3 The $\lambda_{(2,1)}$-labeling of sparse graphs 72
4.3 The general $\lambda(p,q)$-labeling problem 73
4.3.1 Complexity of the $\lambda(p,1)$-labeling problem 76

5 Conclusion **78**

Index **81**

Bibliography **84**

Introduction

The graph covering projection, in other words a graph homomorphism that is locally isomorphic, appeared during the past four decades in various graph-theoretic concepts as well as there were presented its application in computer science, e.g. in distributed computing. In this thesis we would like to exhibit further application of graph covering projections, namely in the graph-theoretic model of the channel assignment problem. Like the application of covers in distributed computing, results in the frequency assignment field have high interest in computer and telecommunication industry.

We shall start with a brief history of graph cover and its applications.

We traced the first occurrence of the notion of graph covers to Conway [4] who in early sixties used a special kind of a covering projection in the construction of highly symmetric graphs in the proof that there are infinitely many finite cubic 5-arc-transitive graphs. This approach was extended by Djoković [11] to a construction of a infinite class of finite fourregular 7-arc-transitive and by Gardiner [20] to the antipodal distance-regular graphs in 1974.

Nešetřil and Pultr [53] showed in 1971 that every locally injective mapping $G \to G$ of a connected graph G is an isomorphism of G.

The structure of the set of all k-fold coverings of a given graph was characterized in 1977 by Gross and Tucker [26] in terms of permutation voltage assignments in a symmetric group of k elements. A simplier characterization of all k-fold coverings was given by Bodlaender [7] in 1989, and the covering projection of directed multigraphs was introduced here, too.

Embeddings of covering projections of graphs was considred in 1980 by Clarke, Thomas and Waller [9].

In 1981, Biggs [5] used graph covers to prove that a k-regular graph on $2(k^2-k+2)$ vertices and girth 6 exists if and only if k or $k-2$ is a perfect square. In his later paper from 1983 [6] he showed that covering graphs admit groups of automorphisms related to the group of the base graph

In 1988, Negami [52] conjectured that the class of projective planar graphs is equal to the class of graphs that have a finite planar cover. The inclusion

$$\{H \text{ is projective planar}\} \subseteq \{H \text{ has a finite planar cover}\}$$

is trivial, but the opposite is difficult. The result could be possibly obtained by the use of Robetson-Seymour theorem of forbidden minors, and only one of 35 forbidden cases — namely the graph $K_{1,2,2,2}$ — resists to be shown that it allows no planar cover (see [29] for the case of $K_{4,4} - e$ minor). However the conjecture is not proven yet, Hliněný and Thomas [30] in 1999 showed that the conjecture can allow only upto 16 possible counterexamples (upto obvious constructions).

Hofmeister [31] in 1991 counted isomorphism classes of k-fold covering projections onto a fixed graph G.

Among recent results we shall mention the paper by Nedela and Škoviera [51] who in 2000 used graph covers to determine substantially small set of groups s.t. their Cayley graph could be a snark, i.e. cubic and non 3-edge colorable.

The graph covering projection became a standard construction in topologic and algebraic graph theory, see monographs [4, 27, 49].

Graph covers play a specific role in the computer sicence. We review the most intersting results here.

Angluin [2], Angluin and Gardiner [3] showed in early 80's that classes of graphs closed under taking covers can not be recognized by a distributed computing environment with a finite fixed set of processor types. To prove the complete characterization, they conjectured that two graphs have a finite common cover if and only if they have the same degree refinement matrix, which was proved by Leighton [44] in 1982.

Litovsky, Métivier and Zielonka showed in 1993 [45] that the families of series parallel graphs and planar graphs cannot be recognized by means of local computations. This result was extended by Courcelle and Métivier in 1994 [10], who showed that any minor-closed class distinct from the class of all connected graphs which contains a graph with at least two cycles cannot be closed under taking connected covers. In practice this means that this class cannot be recognized by local computations, too — in the sense of bounded relabelling schemes over a possibly infinite alphabet. Both results generalize the model of Angluin and Gardiner, since they concern only finite graphs. This negative characterization holds for example for the class of connected planar graphs, the class of connected partial k-trees with $k \geq 2$, etc.

Bodlaender in 1989 proved that every cover G of a connected graph H is a uniform emulation, that menas that a parallel algorithm designed for the processor network G can be emulated on H where each node of H corresponds to a constant number of nodes of G. The same paper provided the complete characterization of covers of the ring, the grid, the cube, the cube connected cycles, the tree and the complete graphs. Moreover it is shown there that the decision problem whether a graph G covers a graph H is at least as hard as the graph isomorphism problem, even if the ratio $|V_G|/|V_H|$ is fixed.

In the concluding remarks Bodlaender asked the computational complexity of the H-cover problem for fixed graph H. Abello, Fellows and Stillwell [1] showed in 1991 that there are both polynomially solvable and *NP*-complete cases. The series of paper by Kratochvíl, Proskurowski and Telle [39, 38, 37, 36] from late 90's exhibits several approaches to establish the most accurate boundary between the graphs for which the H-cover problem is polynomially solvable and the *NP*-complete instances for the H-cover problem.

Kratochvíl, Proskurowski and Telle showed that sufficiently connected regular graphs belong to *NP*-complete instances for the H-cover problem. Their proof requires the existence of a graph G which satisfies the following property: For all its vertices u, the graph G allows an extension of a local isomorphism on the neighborhood of u into a covering projection $G \to H$. The construction of this *multicover* G involves an algebraic method that generalizes the building of common covers used by Angluin, Gardiner and Leighton.

The multicover technique gave rise to the notion of the partial covering projection. Its importance increased when it was used as a tool in the proof of *NP*-completeness of the graph labeling problem with condition at distance two [18]. Here we would like to introduce some recent aspects of graph theoretic models of the channel assignment.

The concept of graph labelling satisfying constraints $(2, 1)$, that is frequently considered in this thesis was introduced by Griggs and Yeh [25, 58]. This concept was motivated by the channel assignment problem, although the telecommunication industry may in certain cases demand a more sophisticated model, see Leese's survey [43] for more details. On the other hand, Leese showed that in several other cases the general graph theoretic setup for the channel assignment problem could be sufficiently accurate and results in fast nontrivial algorithms [42].

The graph labeling with condition at distance two, the so called $L(2,1)$-labeling, resulted to an interesting graph structure and initiated both the graph-theoretical and the computational research. The main aim in past ten years is the specification of the maximum label $\lambda(G)$ that appears in an optimal $L(2,1)$-labeling of G, see papers by Chang and Kuo [8], Jonas [33], Liu and Yeh [46], and by Sakai [55] on chordal graphs and by Whittlesey, Georgess and Mauro [57] on cubes.

One can find several interesting approaches like the relation of the parameter $\lambda(G)$ to another graph invariant: the path covering number [23, 22] by Georges, Mauro and Whittlesey in 1994 or the question of which graphs allow only optimal labelings that use the full range of labels, see Fishburn and Roberts [19]. The computational approach includes the *NP*-completeness of the decision problem of whether a bipartite graph allows a consecutive $L(2,1)$-labeling with at least six different labels [24], given by Giaro in 1997.

Labelings satisfying three constraints were studied by van den Heuvel, Leese and Shepherd [56] and they provide several bounds for the span (i.e. the size of the biggest label) for possibly infinite square and triangular grid graphs as well as for paths.

In this thesis we will focus our attention on the relationship between the partial covering projection and the computational complexity of the corresponding decision problem on one side, and several models of channel assignment on the other side. For this purpose, we shall comprehensively inspect in detail the behavior of full and partial covering projections first.

The thesis is organized as follows: We introduce the notation and several traditional theorems in the first chapter. The second chapter is devoted to the structural properties of covers as well as to simple results on the computational complexity of the H-cover problem which asks whether a given graph G fully covers a fixed graph H. In the third chapter we explore computational complexity of the H-partial covering problem, and the fourth chapter exhibits a relationship between partial covering projections and models of the channel assignment problem.

Chapter 1

Definitions

In this chapter, we present the used notation and also provide some characterization theorems which are well known through the graph theory. See the monograph [50] for proofs of theorems and lemmas presented in this section. The topic is also well covered by the book [47].

We use $\mathbb{N}$ for the set of natural numbers, and $\mathbb{R}$, $\mathbb{C}$ resp. for the sets of real, respectively complex numbers. If $n \in \mathbb{N}$, then $[n]$ denotes the set $\{1, 2, ..., n\}$.

For p being a prime, symbol $\mathbb{Z}_p$ stands for the ring of residues modulo p.

An ordered pair of elements x, y is denoted by $[x, y]$, while for an unordered pair on elements x and y we use symbols (x, y) or (y, x).

If A, B are sets then $A \times B$ means the *Cartesian product* of A and B. The product is formally defined as $A \times B = \{[x, y] : x \in A, y \in B\}$.

A *mapping* or a *function* f from the set A to a set B is a subset of $A \times B$, such that for all $x \in A$, there exists a unique $y \in B : [x, y] \in f$. We denote the existence of a mapping by $f : A \to B$ or by $A \xrightarrow{f} B$ and instead $[x, y] \in f$ we write $y = f(x)$ and say that y is the image of x over f.

Identity on the set A is a mapping $i : A \to A$ such that $i(x) = x$ for all $x \in A$.

If $f : A \to B$ is a function, then the set A is called the *domain* of f, while the set of all elements of B that are images of some elements from A is called the *range*.

A mapping $f : A \to B$ is *injective* if, for each $y \in B$, there is at most one $x \in A$ s.t. $f(x) = y$. Similarly f is a *surjective* mapping if for each $y \in B$ there is at least one $x \in A, f(x) = y$. The *inverse function* $f^{-1} : B \to A$ of an injective and surjective mapping f is defined by the equality $f^{-1}(y) = x$ whenever $y = f(x)$.

When $A' \subseteq A$, we define $f(A') = \{y \in B : \exists x \in A', y = f(x)\}$ and, similarly, for $B' \subseteq B$, we set $f^{-1}(B') = \{x \in A : f(x) \in B'\}$.

The operation $f \circ g$ is called the *composition* of mappings $f : A \to B$ and $g : B \to C$ and is defined by the equality $(f \circ g)(x) = g(f(x))$.

A class $\mathcal{C}$ of mappings is *closed under composition* if, for each $f, g \in \mathcal{C}$, $Range(f) \subseteq Domain(g)$:, the composition $f \circ g$ belongs to $\mathcal{C}$ too.

1.1 Graphs

A *graph* or *simple graph* is a pair (V, E) where V is a set of *vertices* and $E \subseteq \binom{V}{2}$ is a set of pairs which we call *edges.* For our purposes, we always deal with *finite graphs*, which means that both V and E are finite sets. Vertices are usually denoted by letters u and v and, if not otherwise stated we number the vertex set of the cardinality n by $V = \{v_1, v_2, ..., v_n\}$. Similarly, m denotes the cardinality of the edge set and we write $E = \{e_1, ..., e_m\}$. If vertices u and v belong to an edge $e = (u, v)$, we say that u and v are *adjacent* and write $u \in e$ or $u \in (u, v)$, $e \ni u$, etc.

For graphs, we will use symbols G, G', H, etc. To distinguish between the vertex and edge sets of various graphs, we will use subscripts or brackets, e.g. $V(G), E(G'), V_H$.

The *complement* of a graph $G = (V, E)$ is defined as the graph whose edge set contains all pairs that do not form an edge in G. For the complement of a graph G, we use the symbol $\overline{G}$ and formally define $(V(\overline{G}), E(\overline{G}) = (V(G), \binom{V(G)}{2} \setminus E(G))$.

A *directed graph* or *digraph* is a pair $(V, \vec{E})$, where V is a finite set of vertices and $\vec{E}$ is a set of ordered pairs of V, i.e., oriented edges between distinct vertices. For a directed graph, we use symbols $\vec{G}, D$. If $\vec{e} = [u, v]$ is an oriented edge, we say that $\vec{e}$ starts in the vertex u and ends in v and that u is the *tail* or the ending vertex of $\vec{e}$ and v is the *head* or the beginning vertex of $\vec{e}$.

Only distinct vertices can be connected by an edge in the definition of the graph. If we allow that the same vertex appears on both positions in the pair, we call such an edge a *loop.* A (directed) graph which contains loops is called a *(directed) graph with loops.*

A *multigraph* is a generalization of a finite graph where the edge set E is a *finite* multiset formed from directed and undirected edges and loops (loops can be directed too). Each edge (and loop) has assigned a finite natural number, called the *multiplicity* of the edge $m(e)$, which describes how many times th edge e appears in E.

If u is a vertex of G, then the set of vertices that are adjacent to the vertex u is called the *neighborhood* of u and is denoted by $N(u)$. The set $N[u] = N(u) \cup \{u\}$ is called the *closed neighborhood* of u.

The *degree* of a vertex u in a simple graph G is the number of edges that are incident with u. Formally, $deg(u) = |\{(u,v) : (u,v) \in E(G)\}|$. If G is a graph with loops, then each loop is counted twice in the degree of the vertex, i.e. $deg(u) = |\{(u,v) : (u,v) \in E(G), u \neq v\}| + 2|\{(u,u) : (u,u) \in E(G)\}|$.

For the maximal degree in a graph G, we use the symbol $\Delta(G)$.

In a multigraph the degree of a vertex u is the sum of the multiplicities of the undirected edges incident with u, i.e.

$$deg(u) = \sum_{(u,v)\in E, u\neq v} m((u,v)) + 2 \sum_{(u,u)\in E} m((u,u))$$

A vertex of degree 1 is called a *leaf.*

If all vertices of G have the same degree equal to a constant d, we say that G is a *d-regular* graph. A 3-regular graph is also called a *cubic* graph.

If D is a directed (multi)graph, then the *outdegree* of a vertex u is the number of edges that start in u and the *indegree* is the number of edges that end in u, where both numbers are counted with the multiplicity:

$$outdeg(u) = \sum_{[u,v]\in \vec{E}} m([u,v]) \qquad indeg(u) = \sum_{[v,u]\in \vec{E}} m([v,u])$$

The *adjacency matrix* A_G of a (directed, multi-) graph G is a square matrix of order $n = |V(G)|$, where the entry $(A_G)_{ij} = a_{ij}$ is equal to the number of edges going from the vertex v_i to v_j. The matrix is symmetric for undirected (multi-) graphs and 0,1 valued for simple graphs. If the graph does not contain a loop, then all entries on the diagonal are equal to zero.

The symbol $L(G)$ denotes the *line graph* of a simple graph G and is defined as follows: $V(L(G)) = E(G)$, $E(L(G)) = \{(e,e') : e \neq e' e \cap e' \neq \emptyset\}$. The line graph shows whether a pair of edges of G shares a common vertex or not.

1.1.1 Subgraphs, minors

A graph G' is a *subgraph* of G if $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$. If $E(G') = E(G) \cap \binom{V(G')}{2}$, then G' is an *induced subgraph* of G. We also say that G' is the induced subgraph of G spanned on the vertex set $V(G')$.

Let G be a simple graph and $e \in E$. After the *removing* of the edge e from G, we get the graph $(G - e) = (V(G), E(G) \setminus \{e\})$. The *contraction* of the edge $e = (u, v)$ results in a graph $G{\cdot}e$ with the vertex set $V(G{\cdot}e) = V(G) \setminus \{v\}$ and the edge set $E(G \cdot e) = E(G) \setminus \{(v, w) : w \in N(v)\} \cup \{(u, w) : w \in N(v) \setminus N(u)\}$.

A graph G is a *minor* of H if G can be obtained from a subgraph of H by the contraction of some of its edges. If G is contracted from an induced subgraph of H, we call it an *induced minor* of H.

A class $\mathcal{C}$ of graphs is called *minor closed* whenever $G \in \mathcal{C}$ implies that $\mathcal{C}$ contains all minors of G, as well.

1.1.2 Examples of graphs

The *empty graph* on n vertices E_n has the vertex set $V(E_n) = \{u_1, ...u_n\}$ and no edge, i.e. $E(E_n) = \emptyset$.

Vertices of an empty induced subgraph form the *independent set.* The size of maximum independent set of G is called the *independence number* of G and is denoted by $\alpha(G)$.

The *path of length* $n - 1$, is denoted by P_n, has n vertices $V(P_n) = \{u_1, ...u_n\}$ and $n - 1$ edges: $E(P_n) = \{(u_i, u_{i+1}), 1 \leq i < n\}$. We usually say that P_n starts in u_1 and ends in u_n. A path in a graph G is an isomorphic image of some P_n, i.e. a (not necessarily induced) subgraph that is isomorphic to a P_n. A *walk* in G is a homomorphic image of a P_n. A *tour* is an image of an edge-injective homomorphism of some P_n. In other words, in a path, every edge and every vertex appears only once. In a tour, some vertices may appear more times but every edge is used only once, i.e., a tour may cross itself only in a vertex. Any edge or vertex may appear several times inside a walk.

The *cycle* C_n is formed from P_n and an edge joining vertices u_1 and u_n.

The *girth* of G is the length of the shortest cycle contained in the graph G as a subgraph.

If a graph has a cycle on all vertices, we call the cycle (and also the graph) *Hamiltonian.*

The *complete graph* on n vertices K_n has all possible edges, i.e., $E(K_n) = \{(u_i, u_j), \forall i \neq j\}$ or, equivalently, $K_n = \overline{E}_n$.

An complete subgraph is called a *clique.* The clique of the maximum size (measured in the number of vertices) is called the *maximum clique.* The size of a maximum clique of a graph G is called the *clique number* of G, and is denoted by $\omega(G)$.

The *complete bipartite graph* on partitions with a and b vertices $K_{a,b}$ is defined by $V(K_{a,b}) = \{u_1, ...u_a, v_1, ...v_b\}$, $E(K_{a,b}) = \{(u_i, v_j)\, \forall i = 1, .., a, j = 1, .., b\}$.

A *tree* is a connected graph that contains no cycle as a subgraph.

The *star graph* with n *rays* is denoted by S_n, and has $n+1$ vertices $V(E_n) = \{u_0, ...u_n\}$ and n edges: $E(S_n) = \{(u_0, u_i), 1 \leq i \leq n\}$. We also call u_0 the *center of the star*.

Suppose that, in the following definitions, $(j_1, .., j_k)$ is a k-tuple of positive integers.

The *flower (multi)graph* $F(j_1, .., j_k)$ contains one vertex of degree $2k$ that is the unique intersection of k cycles of length $j_1, .., j_k$. If $j_i = 1$ for some i, then this cycle forms a loop on the central vertex.

The *banana (multi)graph* $B(j_1, .., j_k)$ has two vertices of degree k connected by k paths of length $j_1, .., j_k$. If at least two parameters j_i, j_i' are one, then the central vertices are connected by the multiple edge.

The *weight (multi)graph* $W(i, j, k)$ consists of two cycles (loops) of length j and k which are joined by a path of length i.

Let k be greater or equal to three. The k-starfish graph has k vertices of degree four forming a cycle of length k and k vertices of degree two such that each pair of consecutive vertices of degree four share one common neighbor of degree two.

1.1.3 Graph drawing

We usually visualize a graph by a drawing of its vertices as distinct points in the Euclidean plane, and edges as curves (i.e. homeomorphic images of a closed real interval) that connect adjacent vertices. We say that G is a *planar* graph if there exists a *planar drawing* of G, i.e., a drawing where curves intersects only in their endpoints. The segments of the plane are called *faces*, the infinite segment is the *outerface*.

An *outerplanar graph* has a planar drawing with a Hamiltonian cycle as the boundary of the outerface.

A graph G is called a *projective planar graph* if there exists a cycle C as a subgraph of G which can be contracted to the cycle $(u_1, u_2, ..., u_k, v_1, ..., v_k)$ and $G - \{(u_1, v_1), (u_2, v_2), ..., (u_k, v_k)\}$ has a planar drawing with C as the outerface.

1.1.4 Connectivity

The graph G is *connected* if, for every pair of vertices u, v, there exists a path in G which starts in u and ends in v.

If G is not connected, then its maximal connected subgraphs are called *components* of connectivity.

The length of the shortest path connecting vertices u and v from the same component is called the *distance*, and is denoted by $dist(u,v)$. Note that any shortest path is always an induced path.

The greatest distance between a pair of vertices of a connected graph G is called the *diameter* of G, and is indicated by $diam(G)$.

The simple graph G is *vertex k-connected* if, for every pair of vertices u, v, there exist at least k paths connecting u and v that are pairwise disjoint on their inner vertices.

The simple graph G is *edge k-connected* if, for every pair of vertices u, v, at least k edge disjoint paths join u to v.

The maximal vertex 2-connected induced subgraphs of G are called *blocks* of G.

A set of vertices $V' \subset V(G)$ is called the *cutset* of G if the subgraph spanned on $V(G) \setminus V(G')$ has more components than G.

A set of edges $E' \subset E(G)$ is called the *edge cutset* of G if $(V(G), E')$ has more components than G.

The one-vertex cutset is called the *articulation* or the *cutvertex*, and the edge-cutset of size one is called the *bridge.*

There are two well-known theorems that characterize the connectivity of a graph in words of cutsets:

Theorem 1.1 (Ford-Fulkerson)
A connected simple graph G on $n+2$ vertices is k-vertex-connected if it has no cutset of size at most k.

Theorem 1.2 (Menger)
A connected simple graph G on $n+1$ vertices is k-edge-connected if it has no edge cutset of size at most k.

1.1.5 Morphism on graphs, coloring, factors

A mapping $f : V(G) \to V(H)$ is called the *graph homomorphism* from the graph G to the graph H if the existence of any edge (u, v) of G implies that the pair $(f(u), f(v))$ is an edge of H. The digraph homomorphism is defined in the same way, i.e., by considering the directed edges instead of undirected.

With each (simple graph or digraph) homomorphism $f : V(G) \to V(H)$, there is assigned a unique *edge homomorphism* $f_E : E(G) \to E(H)$ defined by $f_E((u, v)) = (f(u), f(v))$.

A homomorphism $G \to G$ is called an *endomorphism.* An injective and surjective homomorphism is called an *isomorphism.* An isomorphism $G \to G$ is an *automorphism.* The group of automorphisms of a graph G with compositions is denoted by $Aut(G)$.

An automorphism which maps each vertex onto itself is called the *identity.* The other automorphisms are called non-trivial. A *rigid graph* has no automorphism except the identity.

The *proper vertex coloring* or, simply, the *coloring* of a graph G using k colors, is any homomorphism $G \to K_k$, i.e., a labeling of the vertex set $V(G)$ by numbers from $[k]$, s.t. adjacent vertices get different labels. When G allows a coloring with k-colors, we say that G is k-colorable. The minimum k, s.t. G is k-colorable, is called the *chromatic number of* G, and is denoted by $\chi(G)$.

When the number k is small, we prefer to use names of real colors like black, white, red, etc., to denote the vertex label.

All vertices of a clique of a graph are colored by distinct colors under any proper vertex coloring. Hence, the chromatic number of a graph is bounded by the size of its largest clique: $\chi(G) \geq \omega(G)$. A graph G is called the *perfect graph* if, for each induced subgraph $H \subseteq G$, the chromatic number of H is equal to the size of its largest clique.

The graph G is *bipartite* if $\chi(G) \leq 2$. In particular, every tree is a bipartite graph. Bipartite graphs have a good characterization:

Theorem 1.3 *The graph G is bipartite if an only if it doesn't contain an odd cycle as a subgraph.*

The *cyclic k-coloring* is a homomorphism $G \to C_k$.

The *proper edge coloring* of a graph G using k colors is a labeling of edges of G with numbers from $[k]$ s.t. edges which share a common vertex get different labels. The minimum number of colors that are necessary for the existence of an edge coloring of G is called the *chromatic index*, and we indicate it by $\chi'(G)$. Observe that the chromatic index of a graph is equal to the chromatic number of its line graph: $\chi'(G) = \chi(L(G))$.

Theorem 1.4 (Vizing)
The chromatic index of any graph G is bounded by terms of its maximum degree $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$.

A set of pairwise disjoint edges is called a *matching.* A matching that contains all vertices of G is called a *perfect matching.*

A perfect matching in a graph can be computed in polynomial time by Edmond's algorithm.

The *factor* of G is a subgraph of G on the same vertex set. A k-regular factor we call the *k-factor*. Thus, the perfect matching is a synonym for the 1-factor.

We say that a graph G is *k-factorable*, if it can be split into a set of k-factors with pairwise disjoint edge sets.

It follows that a 1-factorable graph is regular and its chromatic index is equal to the degree of an arbitrary vertex.

The following application of the König–Hall marriage theorem shows that all bipartite regular graphs are easily decomposable into a set of perfect matchings.

Theorem 1.5 *All bipartite regular graphs are* 1*-factorable.*

Moreover, the theorem implies that all bipartite graphs have the chromatic index equal to the maximal degree, and that an edge coloring using $\Delta(G)$ colors can be found in polynomial time.

The following theorems show that any $2k$-regular graph can be easily decomposed into 2-factors or k-factors, as well.

Theorem 1.6 (Petersen)
Every $2k$-regular graph is 2-factorable, and the k disjoint 2-factors can be found in polynomial time.

Theorem 1.7 *Each $2k$-regular graph having an even number of edges in each component can be split into two disjoint k-factors in polynomial time.*

1.1.6 Covers, partial covers

Let us denote the maximum star in G with the vertex u as the center by $S_G(u)$, i.e., the subgraph of G on $N[u]$ induced by the edges incident with u.

If G and H are simple graphs, then a homomorphism $f : G \to H$ is called a *local isomorphism* or a *covering projection* of H by G, if the mapping f restricted to any $S_G(u)$ is an isomorphism to $S_H(f(u))$.

If the mapping f in the above definition is not isomorphic but only injective on $S_H(f(u))$, we call the homomorphism f the *partial covering projection.*

If any (partial) covering projection $G \to H$ exists, we also say that G (partially) covers H or that G is a (partial) cover of H.

For the purpose of this thesis, we need a more precise definition of covers and partial covers, since we will deal also with colored multigraphs.

Definition *Let G and H be multigraphs where vertex and edge sets are split into disjoint classes (colors) $V(G) = V_{G,1}\dot{\cup}...\dot{\cup}V_{G,j}$, $E(G) = E_{G,1}\dot{\cup}...\dot{\cup}E_{G,k}$, $\vec{E}(G) = \vec{E}_{G,1}\dot{\cup}...\dot{\cup}\vec{E}_{G,l}$ (and similarly for H).*

A mapping $f : (V(G) \cup E(G) \cup \vec{E}(G)) \to (V(H) \cup E(H) \cup \vec{E}(H))$ is called the covering projection on multigraphs*, if the following conditions are satisfied:*

1. $\forall u \in V_{G,i} : f(u) \in V_{H,i}$,
2. $\forall (u,v) \in E_{G,i} : f((u,v)) \in E_{H,i} \wedge f((u,v)) = (f(u), f(v))$,
3. $\forall [u,v] \in \vec{E}_{G,i} : f([u,v]) \in \vec{E}_{H,i} \wedge f([u,v]) = [f(u), f(v)]$,
4. $\forall (u,v),(u,w) \in E(G) \wedge (u,v) \neq (u,w) : f((u,v)) \neq f((u,w))$,
5. $\forall [u,v],[u,w] \in \vec{E}(G) \wedge [u,v] \neq [u,w] : f([u,v]) \neq f([u,w])$,
6. $\forall [v,u],[w,u] \in \vec{E}(G) \wedge [v,u] \neq [w,u] : f([v,u]) \neq f([w,u])$,
7. $\forall u \in V(G) : deg_G(u) = deg_H(f(u))$,
8. $\forall u \in V(G) : outdeg_G(u) = outdeg_H(f(u))$, *and*
9. $\forall u \in V(G) : indeg_G(u) = indeg_H(f(u))$.

If f satisfies only the first six conditions, then it is called the partial covering projection on multigraphs.

Note that the variables v and w in items 4,5, and 6 may refer to the same vertex. In addition, one or both of them can be equal to u when u is incident with a loop.

See Fig. 1.1 for an example of a covering projection $G \to H$. Various shapes of vertices and edges correspond to the different color classes. The covering projection is indicated by numbers on vertices and edges.

1.2 Computational Complexity

In this section, we outline several basic definitions from complexity theory. For a more detailed description, see the monograph [21].

In the thesis we will describe all algorithms as a sequence of steps written in the natural language or we use algol-like structures. However, we suppose

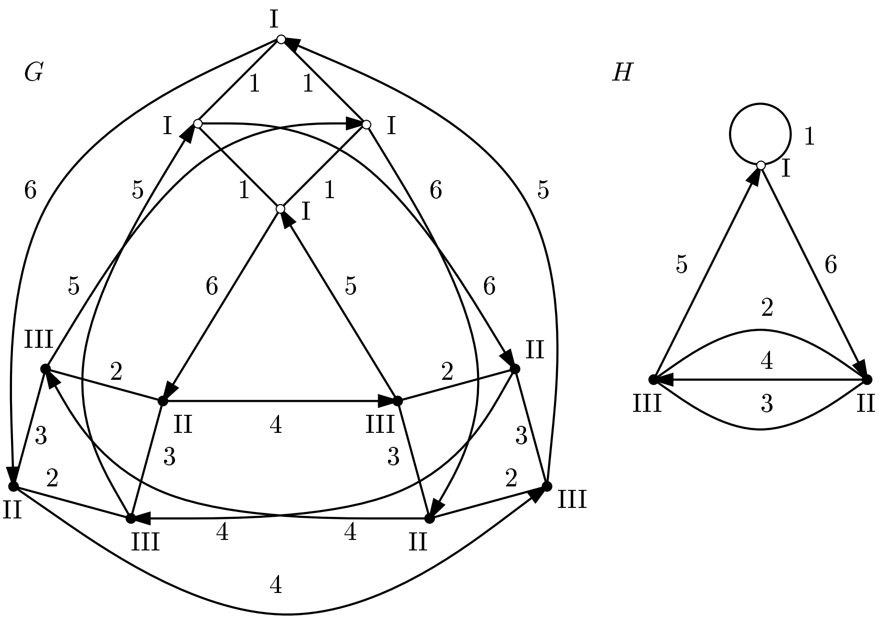

Figure 1.1: An example of a covering projection $G \to H$

that our algorithm will be executed by a Turing machine with the program embedded into the transition function.

An input instance is an object on which the computation is performed. Size of an input instance is a number of symbols (or bits) written on the input tape, that describes the input instance.

The *running time* is the number of steps needed for processing the algorithm. It is expressed as a function of the size of the input instance.

For easier expression of the running time of an algorithm, we use the O-notation. This helps us to concentrate to the most important factor that affects the running time. We describe the running time function f as $O(g)$ if there exist constants c and n_0, such that $\forall n > n_0 : f(n) < cg(n)$.

For example, if the running time of an algorithm is bounded by a polynomial of degree k, we write that it runs in time $O(n^k)$.

Objects we deal with are graphs and are represented by a list of vertices and adjacent edges. For simplicity, we suppose that a constant space is sufficient to describe the vertex and edge labels (or a logarithmic factor appears both at the size of the input instance and also in the running time function). Typical input size parameters are the number of vertices n and the number of edges m.

Our representation allows us to test adjacency of a pair of vertices in the constant time $O(1)$, process all vertices or neighbors of a fixed vertex in a linear time $O(n)$ and process all edges in $O(m)$ time.

A problem is *polynomially solvable* if there exists a Turing machine which, for any input instance written on the input tape, writes a correct output instance (solution) and stops in the time which is bounded by a polynomial in the size of the input instance. A Turing machine answers a decision problem if it accepts or rejects the input.

The class of polynomially solvable problems is denoted by P.

There exists a wide range of graph-oriented problems that are polynomially solvable, for example:

- Is G planar?
- Is G bipartite?
- Does G contain a perfect matching?
- What is the size of a maximum matching?
- Does G contain a C_5 as an induced subgraph?
- What is the chromatic number of a perfect graph G?

A problem R is *polynomially reducible* to a problem S if there exists a Turing machine running in a polynomial time which converts any input instance i_R to an input instance i_S and any output instance o_S to an output instance o_R, such that the output instance o_R is a correct solution for i_R if and only if the solution o_S is correct for i_S

If two problems are polynomially reducible to each other, we say that they are *polynomially equivalent.*

For example, the searching for a k-factor can be reduced to the searching for a perfect matching by the following reduction:

Theorem 1.8 *Searching for a perfect matching and searching for a k-factors are polynomially equivalent problems.*

Proof: One reduction is trivial since every perfect matching is also a 1-factor.

On the other direction, perform the following construction of the graph G' in $O(n + m)$ steps: Split each vertex u into $d = deg(u)$ independent

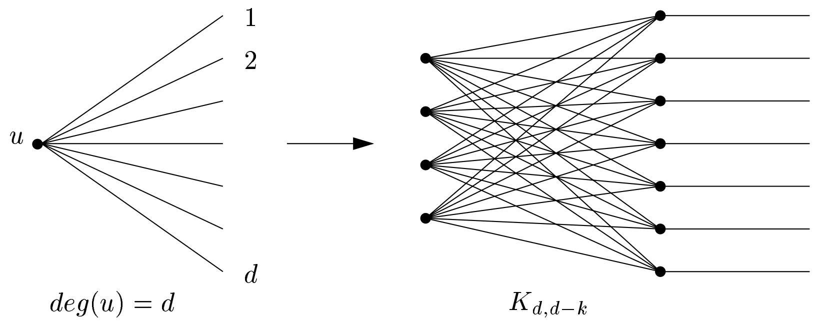

Figure 1.2: Reduction of k-factor to perfect matching

vertices $u_1, ..., u_d$, s.t. each edge is incident with exactly one vertex u_i. Then add $d - k$ extra new vertices and connect them to vertices $u_1, ..., u_d$. by the graph $K_{d-k,d}$, where vertices $u_1, ..., u_d$ form one partition.

Every perfect matching in G' uses exactly $d-k$ edges inside each complete subgraph $K_{d-k,d}$. Therefore, exactly k edges are connecting every $K_{d-k,d}$ with the rest of the graph G', and the corresponding edges form a k-factor in G.

The graph G' contains a perfect matching if and only if the graph G has a k-factor, since any k edges leaving $K_{d-k,d}$ can be completed to a perfect matching inside $K_{d-k,d}$. Hence, any matching of G' that selects k vertices in each $K_{d-k,d}$ can be extended to a perfect matching of the entire graph G'. □

The class of problems which can be verified in polynomial time is denoted by *NP*. Verified means that there exists a Turing machine that reads the input instance and proof of the output instance and decides whether the solution is correct to the input or not.

Class *P* is trivially a subclass of *NP*but, at the moment, there is no proof of whether these classes are equivalent or whether *P* is strictly smaller than *NP*. Throughout the thesis we use the commonly accepted assumption that *P*≠*NP*. However, this implies the existence of a dense distribution of classes in between *P* and *NP* [41].

Several problems are unknown to belong to *NP*, e.g. "What is the chromatic number of G?", since there is no known polynomial verifier which, for the output $\chi(G) = k$, proves that there is no coloring of G using at most $k - 1$ colors.

The class of *NP*-complete problems (class *NPc*) is a subset of *NP* s.t.

each problem from *NP* is polynomially reducible to an arbitrary problem from *NPc*. The importance of *NP*-complete problems is derived from the fact that the existence of a polynomial algorithm for a single *NP*-complete problem proves the equality *P*=*NP*.

There are many famous graph related *NP*-complete problems, e.g.:

- Does a graph G contain a Hamiltonian cycle?
- Is a graph G k-colorable?
- Does there exist a homomorphism from G to C_5?

In this thesis, we will often show a reduction to a H-coloring problem.

Problem: *H-coloring*
Input: *A graph G*
Question: *Does there exists a homomorphism from G to H?*

The computational complexity of the class of H-coloring problems was fully characterized in [28].

Theorem 1.9 (Hell-Nešetřil)
The H-coloring problem is NP-complete if and only if H contains an odd cycle, and is polynomially solvable otherwise.

In particular, the classical k-coloring problem is equivalent to the H-coloring problem when selecting $H = K_k$.

Another problem that is frequently used in this thesis asks for a specific bicoloring of a given regular graph:

Problem: $BW(k, j)$
Input: *A $(k + j)$-regular graph G*
Question: *Does there exists a coloring of $V(G)$ with black and white colors s.t. each vertex has adjacent exactly k vertices of the same color?*

When k or l is equal to zero, the problem is trivially satisfied, but all other cases are *NP*-complete: The $BW(2,1)$ problem was proven to be *NP*-complete in [34, 35]. For the *NP*-completeness of the case of an even $k \geq 2$ and an arbitrary $l \geq 1$, see [37]. The remaining case of an odd k can be treated similarly [17].

Chapter 2

Covers

2.1 Structural behavior of covers and partial covers

2.1.1 Introduction

We introduce two observations for the better description of the behavior of a covering projection.

We already expressed that the covering projection $f : G \to H$ is a homomorphism whose restriction to $N[u]$ of an arbitrary vertex $u \in V(G)$ is an isomorphism to $N[f(u)]$.

We also defined the covering projection for multigraphs, which — rewritten to the case of simple graphs — satisfies the following conditions:

1. f is a homomorphism, i.e. $\forall (u,v) \in E(G) : (f(u), f(v)) \in E(H)$,
2. f is locally injective, i.e. $\forall u, v \in V(G), dist(u,v) = 2 : f(u) \neq f(v)$, and
3. f is degree-preserving, i.e. $\forall u \in V(G) : deg_G(u) = deg_H(f(u))$.

It is easy to check that the local injectivity (no pair of edges can be merged into a single edge) and degree preserving (the target doesn't have incident more edges than the source) are necessary and sufficient conditions for the local isomorphism and, therefore, the above definitions are equivalent.

Observe that every isomorphism $G \to H$ is a covering projection.

The set of all covering projections is closed under a composition. In other words, if $f : G \to H$ and $g : H \to F$ are covering projections, then $f \circ g$ is a covering projection from G to F. More especially any covering projection $f : G \to H$ composed with a nontrivial automorphism of H is another covering projection from the graph G to H.

In the literature, a covering projection $f : G \to H$ is sometimes mentioned with the adjective *k-fold*. This means that the number of vertices that map onto a fixed vertex u is constant, i.e., there exists a positive integer k, such that $|f^{-1}(u)| = k$ for all vertices $u \in V(H)$.

Observation 2.1 *If the graph H is connected, every covering projection into H is k-fold for some k.*

Proof: Suppose that k is size of $f^{-1}(u)$ for a particular vertex u of H, and that $e = (u, v)$ is an arbitrary edge incident with u. Since the covering projection f is a local isomorphism, it means that $|f^{-1}(e)| = k$, and the constant is the same for both ends of e, i.e. $|f^{-1}(u)| = |f^{-1}(v)|$.

Due to the connectedness of the graph H, we get the equality for all vertices $u \in V(H)$. $\square$

The observation immediately implies that whenever G covers a connected graph H, then the size of the vertex set of G is a multiple of the number of vertices in H. In particular, every covering projection $G \to G$ of a connected graph G is an automorphisms of G.

For disconnected graphs, this multiplicity principle holds between pairs of blocks of G and H, however, the constants may vary in different cases.

In the construction of graphs of special properties, we will use the following extension lemma:

Lemma 2.2 *If a graph G is a partial cover of a graph H, then there exists a graph $G' \supseteq G$ that fully covers H.*

Proof: Denote by g the partial covering projection $G \to H$.

Enlarge the vertex set $V(G)$ by introducing extra new vertices into the set $V(G')$ and extend the mapping g into $V(G')$ such that $\forall v, v' \in V(H) :$ $|g^{-1}(v)| = |g^{-1}(v')|$.

For each edge $e = (v, v')$ of H, find sets $A = g^{-1}(v), B = g^{-1}(v')$ and, if necessary, insert into G' new edges, s.t. the sets A and B are connected by a perfect matching. The mapping g is locally isomorphic. Hence, G' is a full cover of H. $\square$

2.1.2 Degree refinement

Any full covering projection maintains the degree of a vertex. Therefore only vertices of the same degree might be mapped onto the same target.

Looking on the neighborhood of a vertex, we extend our observation that also all neighbors of our candidates should be matched into pairs of the same degree. In this section, we construct a partition of the vertex set into classes in a way that generalizes this property of a full covering projection.

Definition *The* degree refinement *of a graph G is a partition of the vertex set $V(G)$ into the minimum number of disjoint sets $R_1, ..., R_k$, such that:*

- *vertices in the same set have the same degree,*
- *if u and v belong to the same set then, for each R_i, the number of neighbors of u in R_i is equal to the number of neighbors of v in R_i.*

Note that the degree refinement of a regular graph consists of only a single set containing all vertices of the graph.

The degree refinement can be computed in $O(n^3)$ time by the following procedure:

1. Split vertices into sets $R'_1, ..., R'_{k'}$ by their degree, and order sets in the descending degree.
2. For each vertex, compute the *degree vector* whose i-th entry is the number of neighbors in the set R'_i.
3. Stop if all vertices have the same degree vector in each set.
4. Otherwise refine the partition, s.t. sets contain vertices with the same vector. Then refine the set order by the lexicographic ordering of the corresponding vectors and continue with step 2.

The above algorithm gives us also the unique ordering of sets of the degree refinement. The square matrix whose rows are degree vectors of the final refinement is called the *degree refinement matrix M*.

Observe that the degree refinement matrix is filled by non-negative integers, and is weakly symmetric with respect to zero. If $(M)_{ij} = 0$, then $(M)_{ji} = 0$.

The following theorems glue together the shape of the degree refinement matrix and the existence of a full covering projection.

Theorem 2.3 *[2, 44]*
If a graph G covers a connected graph H, then their degree refinement matrices are equal.

Theorem 2.4 *[16, 17]*
If connected graphs G and H have the same degree refinement matrix, then any partial covering projection from G to H is also a full covering projection.

Proof: (Sketch) Perform the degree refinement procedure simultaneously on graphs G and H, and sort the sets $R_{G,i}$ and $R_{H,i}$ lexicographically. Put $t_x(u) = i$ for each $u \in R_{x,i}$ where x stands for G or H respectively. Note that vertices in the corresponding sets in G and H have the same degree vectors during the execution of the degree refinement procedure.

Now, suppose that a partial covering projection $f : G \to H$ exists. We show that, for any vertex $u \in V(G)$, $t_G(u) = t_H(f(u))$. Hence f is degree-preserving, i.e., a full covering projection.

By way of contradiction suppose $t_G(u) < t_H(f(u))$. Then the vertices u and $f(u)$ have the same degree and belong to the corresponding classes $R'_{G,i}$ and $R'_{H,i}$ of the initial degree distribution.

Suppose that the degree vectors of u and $f(u)$ became different during the k-th round of the degree refinement algorithm. This implies that, among neighbors of u, there exists a vertex u', s.t. $t_G(u') < t_H(f(u'))$ and u' gets a different degree vector from $f(u')$ earlier than the vertex u. We repeat this argument for u' to get u'', u''' etc. and, after at most k iterations, we obtain a vertex $u^{(l)}$ satisfying $u^{(l)} \in R'_{G,i}$, while $f(u^{(l)}) \in R'_{H,i'}$ and $i < i'$, a contradiction.

To exclude the opposite inequality, suppose that there exists a vertex u satisfying $t_G(u) > t_H(f(u))$. Select a vertex $v \in R_{G,1}$ arbitrarily, and note that $t_G(v) = t_H(f(v))$. Since the graph G is connected, there exists a path from u to v and on the path there is a edge (v', u') such that $t_G(v') = t_H(f(v'))$ but $t_G(u') > t_H(f(u'))$. This yields the existence of a neighbor u'' of v', s.t. $t_G(u'') < t_H(f(u''))$, what is impossible. $\square$

The last theorem implies the result of Nešetřil and Pultr [53] claiming that every partial cover $G \to G$ of a connected graph G is its automorphism.

Call a graph *ground* if the degree refinement matrix is equal to its adjacency matrix.

Ground graphs have exactly one vertex in each class of degree refinement. They may cover only itself, since there is no possibility to map two vertices on the same target. On the other hand, it is easy to test whether an input graph covers a ground graph, since any covering projection is uniquely identified by the classes of degree refinement.

This approach was extended in [38], where an polynomial algorithm was given which tested the existence of a full covering projection for simple graphs which have at most two vertices in each class of the degree refinement. The algorithm used a reduction to the 2-SAT problem.

2.1.3 Marked products

Let $G = (V, E_1, ..., E_k)$ be a k-regular k-edge-chromatic graph, where sets $E_1, ..., E_k$ are color classes of an proper edge coloring. In this section, we call this structure *marked graph.* For a given k, let the class $\mathcal{M}_k$ contains all marked graphs, where the edge-coloring is represented by the color classes.

Note that each E_i is a perfect matching in G, and that each ordering of color classes represents a different object in $\mathcal{M}_k$.

For two marked graphs $G = (V, E_1, ..., E_k), G' = (V', E'_1, ..., E'_k) \in \mathcal{M}_k$, we define the product $G \times G' = (V \times V', E_1 \times E'_1, ..., E_k \times E'_k)$.

It is obvious that projections $\pi : G \times G' \to G$ and $\pi' : G \times G' \to G'$ defined as $\pi((u, v)) = u, \pi'((u, v)) = v$ are covers, since they are locally isomorphic. Each vertex of $G \times G'$ has exactly k adjacent edges where each of them belong to a different color class, and the same holds the graphs G and G'.

The product $G \times G'$ satisfies the categorical property with respect to covers.

Lemma 2.5 *Whenever there is a marked graph $\tilde{G}$ that covers both G and G', and both covering projections f, f' maintain the color classes, i.e., $f(\tilde{E}_i) = E_i$ and $f'(\tilde{E}_i) = E'_i$, then there exists an unique covering projection $\tilde{f} : \tilde{G} \to G \times G'$, such that $\tilde{f}$ commutes with projections π, π', i.e., $f = \tilde{f} \circ \pi$ and $f' = \tilde{f} \circ \pi'$.*

Proof: The projection $\tilde{f}$ is uniquely defined as $\tilde{f}(u) = (f(u), f'(u))$. It follows directly from the definition of projections π, π', that $\tilde{f}$ commutes with projection. We have to check that $\tilde{f}$ is a covering projection. If an edge (u, v) belongs to the class $\tilde{E}_i$, then $(f(u), f(v)) \in E_i$ and $(f'(u), f'(v)) \in E'_i$. Therefore, $(\tilde{f}(u), \tilde{f}(v)) \in (E_i \times E'_i)$, and $\tilde{f}$ is locally injective and respects color classes. But both $\tilde{B}$ and $G \times G'$ are from $\mathcal{M}_k$ and, hence, are k-regular graphs, so $\tilde{f}$ is a local isomorphism. $\square$

We extend the class of marked k-regular graphs to the class $\mathcal{M}'_k$ of all k-edge chromatic graphs to deal with general graphs, as well. Elements of $\mathcal{M}'_k$ are graphs together with the color classes $G = (V, E_1, ..., E_k)$. Now, each color class E_i is a matching in G.

For marked graphs $G, G' \in \mathcal{M}'_k$, let the product $G \times G'$ and projections π, π' be defined by the same formula as for k-regular marked graphs: $G \times$

$G' = (V \times V', E_1 \times E'_1, ..., E_k \times E'_k)$, $\pi((u,v)) = u$, $\pi'((u,v)) = v$.

We get a similar result as in the previous paragraph. The only difference is that all projections we deal with in $\mathcal{M}'_k$ are partial covers.

Lemma 2.6 *Suppose $G, G', \tilde{G} \in \mathcal{M}'_k$, the graph $\tilde{G}$ partially covers both G and G' with respect to the color classes, and g, g' are desired partial covering projections. Then there exist a unique partial covering projection $\tilde{g} : \tilde{G} \to G \times G'$ s.t. $g = \tilde{g} \circ \pi$ and $g' = \tilde{g} \circ \pi'$.*

Proof: Set $\tilde{g}(u) = (g(u), g'(u))$. Then $\tilde{g}$ commutes, and and edge $(u,v) \in \tilde{E}_i$ has its mirrors both in E_i and E'_i, and in $(E_i \times E'_i)$ too. Thus, $\tilde{g}$ is locally injective, since all color classes are matchings. In other words, each color class locally contains at most one edge and the product of corresponding color classes has locally at most one edge, as well. $\square$

One cannot expect that Lemma 2.6 will hold also for full covers of general graphs and on the same vertex set $V \times V'$, because whenever there are two vertices $u \in G, u' \in G'$ of different degree, then the vertex (u, u') has the same degree as u and u', which is impossible. But if more structure is achieved, a particular component of $G \times G'$ covers fully both G and G'.

Call a graph $G = (V_1, ..., V_l, E_1, ..., E_k)$ *well marked* with respect to degree refinement $V_1, ..., V_k$ if E_i are classes of edge coloring and each E_i is a perfect matching either inside one class V_k or between two classes $V_{k'}$ and $V_{k''}$.

Definition *Let $G = (V_1, ..., V_l, E_1, ..., E_k), G' = (V'_1, ..., V'_l, E'_1, ..., E'_k)$ be two well marked graphs with the same degree refinement, let classes V_i and V'_i correspond to each other in the refinement, and let E_j acts on partitions in G with the same indices as E'_j in G'. Then, we define $G \otimes G' = (V_1 \times V'_1, ..., V_l \times V'_l, E_1 \times E'_1, ..., E_k \times E'_k)$ as a* product of well-marked graphs *G and G'.*

Theorem 2.7 *If G, G' are well marked graphs and $\tilde{G}$ fully covers both G and G' by g and g' with respect to the edge color classes, then there is a unique* full covering projection *$\tilde{G} \to G \otimes G'$, satisfying $g = \tilde{g} \circ \pi, g' = \tilde{g} \circ \pi'$.*

Proof: It follows from the definition of well marked graphs, that if $u, v \in V_i$ and $u' \in V'_i$, then the neighborhoods of u, v, and u' are mutually isomorphic with respect to the color classes. Therefore, each of them is also isomorphic to the neighborhood of (u, u') in $G \otimes G'$, because all edges are achieved: If (u, v) is an edge between V_i and $V_{i'}$ and belongs to E_j, then an edge

$(w,x) \in E'_j$ connects a vertex (say x) from V'_i with a vertex from $V'_{i'}$. Then $((u,x),(v,w))$ is edge of $G \otimes G'$, and belongs to the color class $E_j \times E'_j$.

Hence, $\tilde{g}$ commutes, and π, π' are full covers, too. It is necessary to check that $\tilde{g}$ is a full covering projection. Due to Lemma 2.6, the mapping g is a partial covering projection. Moreover, g maintains the degree of every vertex. Hence, g is a full covering projection. □

It follows directly from the definition that all well marked graphs have a symmetric matrix of degree refinement.

2.1.4 Common covers

Angluin in 1980 explored in her paper [2] the power of distributed computing in a network of processors and defined the class DAA as the class of sets of graphs that might be recognized by a deterministic distributed computation with a uniform initial and final configuration.

She defined the *universal cover* $U(G)$ of a simple graph G as an possibly infinite tree with vertex set consisting of all walks started in a fixed vertex v_0 that do not traverse the same edge in two consecutive steps. Two walks w, w' are adjacent if w' is a one edge extension of w or vice versa.

The universal cover $U(G)$ is an infinite structure whenever G has a cycle. For an easier and practical recognition of graphs, she proved that $U(G) = U(H)$ if and only if G nd H have the same degree refinement matrix.

In the same paper, she proved that the classes $\mathcal{C}_G = \{H : U(H) = U(G)\}$ belong to DAA. In other words, for each G there exists a finite set of processor types that, when it is assigned to vertices of any H, it computes whether H has the same degree refinement as G or not. From a practical point of view, it may be implemented as a hardware test whether a certain network of processors is feasible for the processing of an distributed algorithm that assigns to the processors at vertices from the same class of degree refinement the same task.

It is proved there that two graphs G, G' are indistinguishable by DDA computations if there exists a finite graph H called the *finite common cover* that covers both G and G'.

To establish the full characterization of whether two graphs are recognized by a DAA computation, she conjectured that G and H have the same degree refinement matrix if and only if G and H have a finite common cover.

This conjecture is closely connected to the categorical product of two marked k-regular graphs $G \times G'$, since we already proved that this product is also a common cover.

Unfortunately, the structure of several k-regular graphs G cannot be extended to a marked graph, since these graphs are not k-edge colorable. For example, the Petersen graph is cubic, but its chromatic index is equal to four.

The following technique was used to prove the Angluin's conjecture for a restricted class of graphs in [3].

Observe the graph $\tilde{G} = G \times K_2$, called the *Kronecker double cover* [2] of G. Since G is k-regular, and K_2 is bipartite, $\tilde{G}$ is bipartite k-regular, and due to Theorem 1.5, it is k-edge colorable. It follows that $\tilde{G}$ can be marked. In addition, $\tilde{G}$ fully covers G by the canonical projection $\pi((u, v)) = u$.

Corollary 2.8 *For each k-regular graphs G, G' there exists a graph $\tilde{G}$ that fully covers each of them.*

Proof: If G is not k-edge chromatic, use $\tilde{G}$ instead of G and produce $\tilde{G} \times G'$. Then $\tilde{G} \times G'$ covers $\tilde{G}$ that covers G. Similarly for G'. $\square$

For a general graph G, there is a question of whether there exists a graph $\tilde{G}$ that can be well-marked and that fully covers G. The question is positively answered for graphs whose degree refinement matrix is symmetric. The graph $\tilde{G} = G \times K_2$ is regular and bipartite inside each $V_k \times K_2$. The bipartite graphs between $V_{k'}$ and $V_{k''}$ are regular due to the symmetry of the degree refinement matrix. This property remains after multiplication by K_2.

The conjecture was proven by Leighton two years later in 1982 [44]:

Theorem 2.9 *Let G, G' be two finite connected simple graphs. Then the following statements are equivalent:*

- *G and G' share a common finite cover;*
- *G and G' have the same universal cover;*
- *G and G' share a common (possibly infinite) cover; and*
- *G and G' have the same degree refinement matrix.*

Proof: (Sketch) Denote by G_{ij} the subgraph of G induced by edges connecting sets R_i and R_j. Suppose that for all i, j, we are able to construct graphs H_{ij} that cover G_{ij} as well as G'_{ij}. By the induction hypothesis, suppose that we are able to construct a common cover of graphs $H_{\overline{ij}}$ that covers both $(G \setminus E(G_{ij}))$ and $(G' \setminus E(G'_{ij}))$ for some $(M)_{ij} \neq 0$. Then, we use multiple copies of H_{ij} and $H_{\overline{ij}}$ until sets $R_i(H_{ij})$ and $R_i(H_{\overline{ij}})$ have the same size.

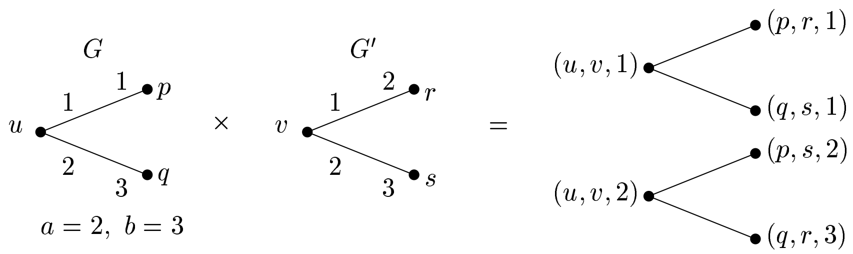

Figure 2.1: Detail of Leighton's construction

(Immediately, the equality holds also for sets R_j.) Gluing together vertices $u \in R_i(H_{ij})$ and $v \in R_i(H_{\overline{ij}})$ that have the same image in G and G' (and in the same way, for index j), we get a common cover of both G and G'.

Now we show a construction of graphs H_{ij}. Due to Corollary 2.8, we suppose that $i \neq j$. For each vertex $u \in V(G_{ij})$, fix an injective labeling of its incident edges by numbers from $[deg(u)]$. We use the symbol $c(u, e)$ for the label of edge e incident to vertex u. We perform the same procedure for the graph G'_{ij}.

Let $a = (M)_{ij}, b = (M)_{ji}$, and put
$V(H_{ij}) = (V(R_i(G) \times V(R_i(G') \times [a]) \cup (V(R_j(G) \times V(R_j(G') \times [b])$.

Two vertices (u, u', k) and (v, v', l) are adjacent if and only if

$$\begin{aligned}
&(u, v) \in E(G), \quad (u', v') \in E(G'),\\
&k \equiv c(u, (u, v)) - c'(u', (u', v')) \quad (mod\ a), \text{ and}\\
&l \equiv c(v, (u, v)) - c'(v', (u', v')) \quad (mod\ b).
\end{aligned}$$

Projections $f : (u, u', k) \to u$ and $f' : (u, u', k) \to u'$ are covering projections, which prove that H_{ij} is a common cover of both G and G'. $\square$

See Fig. 2.1 for an example of the product of two vertices and their adjacent edges.

Call the graph constructed by the above theorem *Leighton cover*.

Theorem 2.9 directly implies the following corollary [2]:

Corollary 2.10 *If graphs G and G' cover the same connected graph F, then they have a finite common cover.*

2.1.5 Colored directed multigraphs

Lemma 2.11 *Any tree T fully covers only an isomorphic tree.*

Proof: If the covering projection maps two vertices u and u' of T on the same vertex in H, then the path joining u and u' is mapped onto a cycle in H. Going around this cycle in H, we get a sequence of vertices in T, which form a cycle or an infinite path. Both cases are impossible since T is a finite tree.

We proved that the covering projection is injective. Any tree is also connected graph, and there exists only 1-fold cover. Hence, it is an isomorphism by Observation 2.1. □

If a general graph contains a cutvertex whose removal gives a tree as a one block, then the mirror of that cutvertex has the same property under any covering projection, and the trees are isomorphic. If more trees appear by the removal of the vertex, then they should be arranged into isomorphic pairs.

This observation gives us an idea of how to concentrate our attention only to graphs without leaves: If there is a leaf in the graph, remove it and maintain a code that the leaf was removed together with the code of the leaf. This gives us a graph without leaves, where some vertices are labeled. For simplicity, we will view different labels as different vertex colors.

At the second step, we remove all vertices of degree two from the graph: Consider a path connecting two vertices of degree at least three, whose all internal vertices have degree two. We replace the path by a single edge and maintain the code of the number, order and colors of vertices of the replaced path. Due to a similar reason, we call the code of the path the edge color. Since the removed path is not necessarily symmetric, we assign an orientation to the edge. Note, that we can uniquely reconstruct the original path from the color and the orientation of the edge.

It is possible, that the path replacement create a multigraph with loops and multiple edges.

Now, we are able to represent each graph by a directed (edge and vertex) colored multigraph with minimum degree 3, and with the following property: If there exists a full covering projection between two directed colored multigraphs (due to the definition the covering projection maintains edge direction and both edge and vertex colors), then there exists also a full covering projection between the transformed multigraphs [37].

In the paragraph 2.1.2, we constructed a degree refinement and the matrix of degree refinement as a tool, that allows us to partially determine the image of a vertex under a covering projection between simple graphs. A similar procedure can be performed for a colored directed multigraph G:

First fix an ordering of all edge and vertex colors — this is necessary for the unique definition of the degree refinement matrix. Suppose that the undirected edges are colored by $1, .., p$, while directed edges by $1', ..., q'$.

Denote by $\vec{d}(u)$ the degree vector having the following structure: $\vec{d}(u) = (c_v(u), deg_1(u), .., deg_p(u), ind_{1'}(u), outd_{1'}(u), ..., ind_{q'}(u), outd_{q'}(u))$, where $c_v(u)$ is the vertex color of u, $deg_i(u)$ is the number of edges of color i incident to u, and the symbols $ind_{i'}(u)$ and $outd_{i'}(u)$ have a similar meaning — the number of oriented edges of color i' incident to u.

The first step of the degree refinement procedure consists of splitting $V(G)$ into sets R_i', such that vertices in the same set have the same degree vector. Sort sets by the lexicographical order of their representatives.

Then refine the partition, as shown in paragraph 2.1.2 until all vertices from the same set have the same number in neighbors in each set R_i.

Note that having the degree partition, we can distinguish between edges of the same color, that connects different pairs of blocks. Therefore, we may assume that without lost of generality, the edge colors used inside a single block or colors of edges that connect a pair of blocks are unique, and that they are not used elsewhere in the multigraph G. For this purpose, we introduce extra new colors to distinguish these edge sets.

Using the same argument, we separate oriented edges leaving a block from the incoming edges. Hence, we assume that the oriented edges appear only inside blocks of the degree refinement.

2.2 Computational complexity of the H-COVER problem

The computational point of view states a question of whether for given graphs G and H there exists a (partial) covering projection from G to H. If both graphs are part of the input, then the problem is trivially *NP*-complete. By selecting $H = K_4$ we can test the existence of a proper 4-coloring of a cubic graph G, such that on the closed neighborhood of every vertex, all four colors are used [35].

We use a similar approach, as is used for the testing the existence of a graph homomorphism (i.e., the H-coloring problem) and define a class of H-cover problems, where each problem corresponds to a specific graph H:

Problem: *H-cover*
Input: *A graph G*
Question: *Does there exists a covering projection mapping the graph G onto the graph H?*

The same technique is used for partial covers:

Problem: *H-partial cover*
Input: *A graph G*
Question: *Does there exists a partial covering projection from G to H?*

Without lost of generality, we suppose that the input graph G is connected, since each block of connectivity of G have to (partially) cover the graph H, if and only if the entire graph G covers H.

2.2.1 Complexity of covering sparse graphs

Both H-cover and H-partial cover problems are polynomially solvable for trees, even if H became a part of the input. Then, the H-cover problem is equivalent to the tree-isomorphism problem. If the tree H is fixed, then the tree isomorphism testing is solvable in constant time.

The H-partial cover problem is solvable in constant time too, because we can ask whether an input graph G is a subtree of H. If H is fixed, then it has only finitely many subtrees, and we can try each case separately. On the other hand, quite sophisticated algorithms running in polynomial time exist for testing subtree isomorphism, see [21], problem GT48 or [54].

In addition, both problems are solvable in polynomial time for graphs H that have only one cycle. If a connected graph G covers unicyclic H, then the graph G has exactly one cycle, and its length is multiple of $girth(H)$. We denote the multiplicity by k, and build an connected k-fold cover of H. Since H has one cycle, this k-fold cover H_k is uniquely determined. Finally, the test, whether two unicyclic graphs are isomorphic, can by done by a slight modification of the tree isomorphism algorithm:

1. Compare $|V(G)|$ and $\Delta(G)$ with $|V(H_k)|$ and $\Delta(H_k)$. If there are different numbers, the graphs cannot be isomorphic.

2. Add into G two disjoint copies of the graph $S_{\Delta(G)+1}$. Now, G has three components. Denote these two stars by S and S'

3. Select an edge $e = (u, u') \in E(G)$ that lies on the cycle and remove it from G. Unify u with the centre of S, and u' with the centre of S'. Denote the resulting tree by T_G.

4. Perform steps 2 and 3 on the graph H_k, and test for the tree isomorphism between T_G and T_{H_k}. If the test succeeds, claim that G and H_k are isomorphic.

5. If the test fails, repeat step 4 at most $girth(H)$ times, and each time select one of the $girth(H)$ consecutive edges along the cycle in H_k. If all these tests fail, then G is not isomorphic to H_k.

To check that the algorithm is correct, observe the following facts.

- Since u and u' have maximal degree, they must be mapped onto the corresponding vertices in H_k, and after the reconstruction of the original graph they will be connected by an edge on both sides.
- We have checked all possible non-isomorphic splitting of the graph H_k into a tree, and if G was isomorphic to H_k, at least one of these trees was isomorphic to T_G.

The H-partial cover problem for unicyclic graphs can be solved by a similar procedure. There are only two differences:

1. If G contains a cycle, we prepare a k-fold cover H_k as above, otherwise we select k, such that $girth(H_k) \geq diam(G)$.
2. Test for the subtree isomorphism, instead of the tree isomorphism.

Note that both methods use the (sub)tree-isomorphism routines in the way, that graphs T_G and T_{H_k} form the input instance.

Corollary 2.12 *The H-cover and H-partial cover problems are solvable in polynomial time for every graph H with at most one cycle in each component of connectivity.*

We will see in the following section that two cycles in H may cause that the H-cover problem become *NP*-complete.

2.2.2 Results review

We already claimed in Theorem 2.3 that, if any covering projection $G \to H$ exist, then G and H share the same degree refinement matrix. The computation of the degree refinement matrix can be done in polynomial time, hence we get the following corollary.

Corollary 2.13 *The H-cover problem is solvable in polynomial time for ground graphs H.*

The fact, that the adjacency and degree refinement matrices are equal, is equivalent to the formulation, that every block of degree refinement of H contains only one vertex.

The H-cover problem first appeared in [3], and even in this pioneer paper, examples of both polynomially solvable and *NP*-complete instances were shown.

The last corollary was extended by Kratochvíl, Proskurowski and Telle [39] for graphs with more complicated cycle structure than unicyclic graphs. We first mention the case of simple graphs.

Theorem 2.14 *If all sets of the degree partition of a simple graph H have at most two vertices, then the H-cover problem is solvable in polynomial time.*

Proof: (Sketch) Let G be the input graph. Ask, whether G has the same degree refinement matrix, and continue, only if the question is answered positively.

If any covering projection $f : G \to H$ exists, then all vertices of G, that corresponds to one-vertex sets in H, have uniquely determined image under f. Therefore, the "hard" task is to define the mapping f on vertices that corresponds to the two vertex-sets $B_i(H) = \{a_i, b_i\}$.

For each vertex $u \in B_i(G)$, introduce a boolean variable x_u, which will be assigned the truth value, when u is mapped onto a_i, and x_u is set to false, whenever $f(u) = b_i$.

We construct a 2-SAT formula Φ, such that each its satisfying assignment corresponds to a proper covering $G \xrightarrow{f} H$.

- If two distinct vertices u and v belongs to the same block $B_i(G)$ and if they are adjacent or have a common neighbor, then let Φ contains $(x_u \vee x_v) \wedge (\neg x_u \vee \neg x_v)$ as a subformula.

- If $(a_i, a_j), (b_i, b_j)$ are the only edges between $B_i(H)$ and $B_j(H)$, then let Φ include conjunction $(x_u \vee \neg x_v) \wedge (\neg x_u \vee x_v)$ as a subformula, for all pairs of vertices $u \in B_i(G), v \in B_j(G)$.

- If $(a_i, b_j), (b_i, a_j)$ are the only edges that connects $B_i(H)$ and $B_j(H)$, then let Φ contains $(x_u \vee x_v) \wedge (\neg x_u \vee \neg x_v)$, for all $u \in B_i(G), v \in B_j(G)$.

These three types of clauses in Φ force, that whenever a satisfying assignment for Φ exists, then the corresponding covering projection is locally injective. In the other direction, every covering projection $f : G \to H$ can be transformed to a satisfying assignment of Φ.

We proved that for all graphs H, which blocks of degree partition have at most two vertices, the H-cover problem is polynomially reducible to the 2-SAT problem, which is known to be polynomially solvable. $\square$

Note, that trees or unicyclic graphs may have more than two vertices in a block of degree partition and, therefore, none of the above theorems give a complete characterization of the polynomially solvable cases.

The paper [39] includes the complete catalogue of H-cover instances of all simple graphs H with at most six vertices, where 36 cases of 208 are *NP*-complete, and a non-trivial polynomial reduction is shown for about 100 graphs.

In the sequel paper [37], Kratochvíl et al. introduced the colored directed multigraphs as a structure, that exclude vertices of degree at most two, and they gave a complete characterization for cdm-graphs with at most two vertices.

The proof technique of Theorem 2.14 was extended in [36] to the case of colored directed multigraphs as follows:

Proposition 2.15 *The H-cover is a polynomially solvable problem if H is a colored directed multigraph, whose classes of degree refinement (with respect to vertex and edge color) have 1, 2 or 4 vertices, and further two conditions are satisfied:*

- *Each block of degree refinement restricted to the edges of the same color is one the following type:*
  - *a disjoint union of (directed) loops or (directed) multiple edges,*
  - *the graph depicted in Fig. 2.2 or two disjoint copies of this graph,*
  - *the cycle C_4,*
  - *C_4 whose all edges are replaced by a multiple directed edges, all in the same direction and with the same multiplicity,*
  - *C_4 whose all edges are replaced by a directed C_2.*
- *Moreover, the edges of the same color, that join a pair of distinct blocks, induce a undirected subgraph of one of the following type:*
  - *a disjoint union of multiple edges,*
  - *$K_{2,1}$ or a disjoint union of two $K_{2,1}$,*
  - *$K_{2,2}$ or a disjoint union of two $K_{2,2}$.*

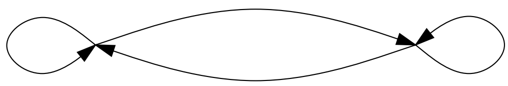

Figure 2.2: One of the polynomially solvable block types

Recall, that vertices forming a block of degree refinement have the same degree, indegree and outdegree with respect to an arbitrary edge color. If a block is a disjoint union of more components (the fist and the second case), then the degree is the same, for all vertices from the block of the degree refinement.

As a particular result, the proposition states that the following cases are polynomially solvable problems:

- The $F(j_1,..,j_k)$-cover, where F denotes the flower graph,
- the banana $B(j_1,..,j_k)$-cover,
- the 4-starfish-cover.

Now, we focus our attention to the *NP*-complete instances of the H-cover problem. In [39], Kratochvíl, Proskurowski and Telle proved that:

- The $W(1,3,3)$-cover problem is *NP*-complete.
- The H-cover problem is *NP*-complete for all k-starfish graphs, for odd k greater to three.

The crucial role of the weight graph $W(1,1,1)$, in the characterization of *NP*-complete cases, was explored in [37]:

Theorem 2.16 *Let H be a colored directed multigraph on two vertices. The H-cover problem is NP-complete, if and only if H has only one block of degree refinement, and there exists a color class E_i, such that $H|_{E_i}$ contains $W(1,1,1)$ as a subgraph or its directed clone with indegree and outdegree greater or equal to three.*

See Fig. 2.3, for the three smallest *NP*-complete cases.

2.2.3 Covers of regular graphs

In this section, we consider a regular graph H, as the underlying graph for the H-cover problem. Its matrix of the degree refinement has only one

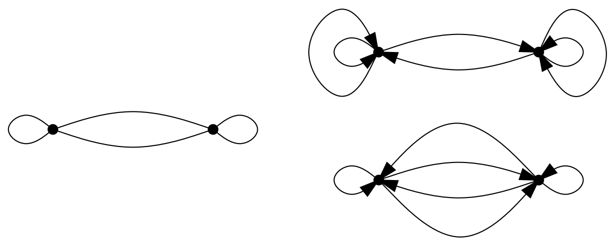

Figure 2.3: The three smallest *NP*-complete cases

element, that is equal to the degree of an arbitrary vertex. The methods, that were used to prove the existence of a polynomial-time algorithm for the H-cover problem, were based on a fine distribution of vertices into classes of the degree distribution. No algorithm has been constructed yet for the polynomial testing of the existence of a H-cover, when the graph H has at least two cycles and contains a block of degree refinement with at least 5 vertices. Therefore, it was generally expected that for all k-regular graphs with $k \geq 3$ the H-cover problem is *NP*-complete.

Here we prove this conjecture.

The result of Abello et al. [1] stated, that there are many graphs H, such that the H-cover problem is *NP*-complete, even if the construction was based on highly symmetric graphs, i.e., graphs with a rich group of automorphisms.

The question was open for the class of rigid graphs [1], and was solved positively by using the following "multicover" approach [38].

Definition *A graph G is called a* multicover *of H, if for any pair of vertices $u \in V(G)$ and $v \in V(H)$, every isomorphism $S_G(u) \to S_H(v)$ can be extended to a covering projection $G \to H$.*

The multicover always exists for all regular graph H, and can be obtained by a Cayley-like construction, even if it requires exponentially large number of vertices with respect to the size of H.

We need one more definition for the proof of the *NP*-completeness of covering problem for k-regular graphs.

Definition *A graph H is* solid *[16] (or* good *[38]), if for any vertex $u \in V(H)$, the graph H_u, that arises by the splitting the vertex u of degree d into leaves $u_1, .., u_d$, involves only partial covers $H_u \to H$, that became au-*

tomorphisms of H after unifying all vertices u_i into the original vertex u.

The multicover and solid graphs were used in a construction of an gadget for a polynomial reduction from the hypergraph colorability.

Theorem 2.17 *[38] The H-cover problem is NP-complete for solid graphs.*

In addition two large classes of graphs were shown to be solid.

Theorem 2.18 *[38] All k-edge colorable k-regular graphs and all $\lceil\frac{k+2}{2}\rceil$-edge-connected k-regular graphs are solid.*

The characterization is tight in the sense that in [16] there was given an example of a $\lceil\frac{k+1}{2}\rceil$-edge-connected k-regular graph, that is not solid.

We finish the classification by extending the result of Theorems 2.17 and 2.18 into the class of all regular graphs.

Theorem 2.19 *[14] The H-cover problem is NP-complete for all k-regular graphs H of $k \geq 3$.*

Proof: Without loss of generality, we assume that H is connected, and that H is not a solid graph. In particular, we assume that H is not bipartite, since bipartite k-regular graphs are k-edge colorable (cf. Theorem 1.5) and are solid.

The Kronecker double cover $\tilde{H} = H \times K_2$ is k-edge colorable k-regular connected graph, and the $\tilde{H}$-cover problem is *NP*-complete, due to Theorems 2.17 and 2.18.

We show a reduction of the $\tilde{H}$-cover problem to the H-cover problem. Consider a graph G, whose covering projection $G \to \tilde{H}$ is questioned. We claim that G covers $\tilde{H}$, if and only if G is bipartite, and G covers H.

The only if statement is trivial, since $\tilde{H}$ is bipartite, and only bipartite graphs can cover a bipartite graph (this holds even for a general graph homomorphism). Moreover, any covering projection $G \to \tilde{H}$ can be extended to H by a composition with a covering projection $\tilde{H} \to H$.

In the other direction, assume that $f : G \to H$ is a covering projection, that G is bipartite, and that its proper bicoloring using black and white is given. For each vertex u of H, denote by u_b and u_w its two copies in $u \times K_2 \subset H \times K_2 = \tilde{H}$. We define a mapping $\tilde{f} : G \to \tilde{H}$ by

$$\tilde{f}(v) = \begin{cases} u_w & \text{if } f(v) = u \text{ and } v \text{ is white,} \\ u_b & \text{if } f(v) = u \text{ and } v \text{ is black.} \end{cases}$$

Since each vertex has all neighbors colored by the complementary color, the above mentioned mapping satisfies all properties of a covering projection. □

As a consequence of Corollary 2.12 and Theorem 2.19, we get that the computational complexity of the class of all regular graphs is fully classified.

2.2.4 Covers of cyclic graphs

In the last section, we show the complexity characterization for the k-starfish-cover problems.

We have already mentioned in Proposition 2.15 that the k-starfish-cover problem is polynomially solvable for $k \in \{1, 2, 4\}$, even if the first two instances are multigraphs and do not correspond to a simple k-starfish graph.

Here we prove that for all other k, the problem is NP-complete.

Consider a k-starfish graph. By the method described in the paragraph 2.1.5, it can be transferred into a colored multigraph, which consists of two cycles glued together. Each of those cycles has a different color, for simplicity, call them red and green. By the same technique we rebuild the input graph into a colored multigraph, whose edges are colored by the same color set. Call the modification of the k-starfish graph H_k. For the future use, we denote the vertex set of $V(H_k) = \{v_1, .., v_k\}$

Observe that each vertex of H_k has adjacent exactly two red and exactly two green edges, and the same assumption holds for any graph that covers H_k.

Definition *Suppose that a connected graph G has its edges colored by red and green, and that each color class induces a set of disjoint cycles and isolated vertices. We define the k-*filling graph* $\widehat{G}$ as k copies of G connected by extra new edges, such that:*

- *All k copies of the same vertex of degree two are connected by a cycle C_k.*
- *The newly introduced edges connect the i-th copy of a vertex to its $(i+1)$-th copy (counted modulo k).*
- *The new cycles are colored, such that every vertex is incident with two red and two green edges.*

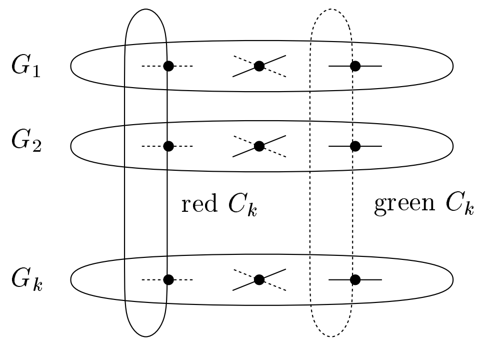

Figure 2.4: The k-filling of a graph G

Observation 2.20 *The k-filling graph $\widehat{G}$ fully covers the graph H_k, if and only if G partially covers H_k.*

Proof: Since $G \subseteq \widehat{G}$ then one implication is trivial.

Let $[u, i]$ be the i-th copy of the vertex u. If g is a partial covering projection $G \to H_k$, then we define $f : \widehat{G} \to H$ as $f([u, i]) = v_{x+i}$ if $g(u) = v_x$, where the addition is done modulo k. Then the mapping f is a full covering projection. $\square$

Lemma 2.21 *If the k-starfish-cover problem is NPcomplete, then the (ck)-starfish-cover problem is also NP-complete, for any positive integer c.*

Proof: Let G be the input graph for the H_k-cover problem. If $c = 1$, then there is nothing to do, otherwise subdivide every edge by $c - 1$ extra new vertices and call the new graph G'. Then G' partially covers H_{ck}, if and only if G covers H_k. Use $\widehat{G'}$ as the input graph for the (ck)-starfish-cover problem, and note, that $\widehat{G'}$ covers H_{ck}, if and only if G covers H_k. $\square$

Lemma 2.22 *The k-starfish-cover problem is NP-complete for all odd k.*

Proof: Let G be an input graph for the C_k-color problem. We show a polynomial time reduction from the C_k-coloring problem, that is *NP*-complete for all odd k due to to Theorem 1.9.

Replace each vertex u of degree d by a red cycle of length dk. Denote by $u_1, ..., u_{dk}$ the vertices of the cycle corresponding to the vertex u. Note, that u_i and u_{i+k} are mapped onto the same vertex under any (partial) covering into H_k.

For each edge incident to u, select a distinct representative among vertices $u_k, u_{2k}, ..., u_{dk}$. If u_i and u'_j are representatives of the edge $e = (u, u') \in E(G)$, then connect u_i and u'_j by an extra new green cycle C_k, such that u_i and u'_j are adjacent. Perform the last step for all edges of G, and call the new colored graph G'.

It is clear that under any partial covering $g : G' \to H_k$, each set $u_{ki}, 1 \leq i \leq deg(u)$ are mapped onto the same vertex. We define the homomorphism $c : G \to C_k$ by $c(u) = g(u_k)$, where C_k is the cycle of H_k spanned by the green edges. Moreover, if $(u, u') \in E(G)$, then there are vertices u_i, u'_j, such that $(g(u_i), g(u'_j))$ is a green edge of H_k, and $(c(u), c(u')) \in C_k$. The application of the "filling" Observation 2.20 finishes the first implication of the polynomial reduction from the C_k-coloring problem.

In the other direction, if a C_k-coloring of G is given, we embed C_k as the green cycle into H_k. Define a mapping on vertices u_{ik} as the color of u, and find its extension onto vertices of degree two lying on the red and green cycles. Due to Observation 2.20 it is always possible extend the partial cover $g : G' \to H_k$ into a full covering projection $f : \widehat{G}' \to H_k$. $\square$

For the complete the characterization of the computational complexity of the k-starfish cover problem, we need results for k being at least third power of 2.

Lemma 2.23 *[13] The 8-starfish-cover problem is NP-complete.*

Proof: We will reduce the traditional fourcolorability problem. Let G be a graph whose coloring using at most four colors is questioned. Replace each vertex u of degree d by an extra red cycle C_{8k}, and select d representatives of incident edges as described in Lemma 2.22.

For every edge (u, u') and its representatives u_i and u'_j, use the gadget consisting of two green C_8 and one red C_{16} depicted in Fig. 2.5. Call the new graph G'.

Without lost of generality suppose, that the vertex u_i is mapped onto v_l under a partial covering $g : G' \to H_8$. Then a and b are mapped either on v_{l+1} or v_{l+7}, and it forces that u'_j maps either on v_{l+2}, v_{l+4} or v_{l+6}. All representatives of the original vertices are mapped onto v_l with the same parity of the index l. Assume that even indices are exposed. Then, we can use vertices v_2, v_4, v_6, and v_8 as indicators for the four distinct colors. We define coloring of G as $c(u) = g(u_i)$. The construction of the edge gadget ensures that adjacent vertices get different colors, and all pairs of distinct colors might be used on the coloring of any edge of G.

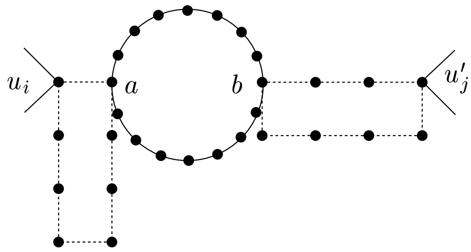

Figure 2.5: The edge gadget for the 8-starfish-cover problem (solid edges are red, dotted are green)

Finally use Observation 2.20 to create the graph $\widehat{G}'$ and ask for a full covering projections instead of a partial covering projection of G'. $\square$

Let us summarize the complexity results for the k-starfish problem.

Corollary 2.24 *The k-starfish problem is solvable in polynomial time, if and only if $k \in \{1, 2, 4\}$. The problem is NP-complete in all other cases.*

Chapter 3

The H-partial cover problem

We have defined the partial covering projection as a locally injective graph homomorphism. In the previous chapter, we have also showed several complexity results for the H-cover problem, which asks for a more restricted — locally isomorphic homomorphism.

Recall, that the computational complexity of the graph homomorphism (H-coloring) problem was fully characterized by Hell and Nešetřil theorem (see Theorem 1.9).

We would like to establish the closest relation of the complexity characterization of H-partial cover problems to the complexity classes of the H-cover problem and the H-coloring problem, respectively.

Any full covering projection can be viewed as a partial covering projection. Conversely, Theorem 2.4 implies that whenever G and H have the same degree refinement matrix, then every partial covering projection is locally isomorphic, i.e., a full covering projection. Hence, the following result follows:

Theorem 3.1 *[17] Let H be a connected graph. If the H-cover problem is NP-complete, then the H-partial cover problem is NP-complete, as well.*

Proof: Let G be a graph, for which the existence of a full covering projection to H is questioned. Compute the degree refinement matrices M_G and M_H. If these matrices are different, reject the input, since due to Theorem 2.3 it is a necessary condition for the existence of a full covering projection. If $M_G = M_H$, then ask for a partial covering projection $f : G \to H$. When f exists, then due to Theorem 2.4 the mapping f is a full covering projection, too. When no partial covering projection is found, then obviously no full covering projection exists. □

Observe that the above theorem and Theorem 2.17 prove the existence of bipartite graphs, such that the H-partial cover is *NP*-complete.

Due to Theorem 3.1, we focus our attention to graphs H, whose H-cover problem is solvable in polynomial time, since there is still a possibility that the corresponding H-partial cover problem is *NP*-complete. In particular, due to Proposition 2.15, we will consider the classes of flower, banana and weight graphs, and we show that there are large subclasses of polynomially solvable problems, as well as *NP*-complete instances.

Recall Corollary 2.12, which states that the H-partial cover problem is polynomially solvable for graphs with at most one cycle in each component of connectivity.

3.1 Proof techniques

In the complexity discussion we will use three different techniques that help us to determine the computational complexity of a particular H-partial cover problem.

These methods use various approaches from the graph optimization problems, and some of these results might be interesting on their own. We devote the forthcoming section to them.

3.1.1 Subset of halfedges

In several cases the question of the existence of a partial covering projection can be solved by the finding of a feasible mapping on the neighborhood of vertices of higher degree. The halfedge object, that is defined below, helps us to find a mapping on a vertex and its adjacent edges.

Definition *A multiset of* halfedges $E^{\frac{1}{2}}(G)$ *of a graph* G *consists of all ordered pairs of vertices and edges that are mutually incident.*

More formally, $E^{\frac{1}{2}}(G) = \{[u,e],[v,e] \in V(G) \times E(G), e = [u,v]\}$.

Observe that the element $[u,e]$ appears at least twice in $E^{\frac{1}{2}}$, when e is a multiple edge or a loop incident with the vertex u.

We introduce two lemmas which shows that it is possible to fing a specific subset of halfedges using the matching algorithm.

First we describe the case of an unoriented graph.

Lemma 3.2 *Let* G *be a graph where every vertex* $u \in V(G)$ *has assigned a nonempty interval* $I_u = [a_u, b_u] \subseteq [0, deg_G(u)]$*, and where every edge* e *of*

G has given a subset $J_e \subseteq \{0,1,2\}$. The question of whether there exists a subset of halfedges $S \subseteq E^{\frac{1}{2}}(G)$ satisfying

$$\forall u \in V(G) : |\{[u,e] \in S : e \in E(G)\}| \in I_v$$

$$\forall e \in E(G) : |\{[u,e] \in S : u \in V(G)\}| \in J_e$$

is solvable in polynomial time.

Proof: Subdivide each edge e of G, such that $J_e \neq \{0,2\}$, by an extra new vertex u_e, and put $I_{u_e} = J_e$. It is clear that the set S exists in G, if and only if the new graph G' contains a factor F, s.t. $deg_F(u) \in I_u$ for all vertices $u \in V(G')$.

The above factor problem can be solved by a matching procedure [48], (Exercise 10.2.2). For the self-consistence we sketch the proof. Use a similar construction as in the proof of Theorem 1.8. For each vertex $u \in V(G')$:

1. Split its incident edges into $deg(u)$ independent vertices forming the set A_u,
2. connect the set A_u by a complete bipartite graph to a newly introduced set B_u of $deg(u) - a_u$ independent vertices, and
3. insert an extra set C_u on $b_u - a_u$ vertices and join it by a complete bipartite graph to the set B_u.

Finally form a clique on the set $\cup_{u \in V(G')} C_u$, and, if the total number of vertices in the new graph is odd, add an extra new vertex to the clique. Call the new graph G''.

Suppose, that a perfect matching in G'' exists. The construction of the tripartite gadget forces that each A_u is incident to at most b_u and at least a_u edges of the matching. The same edges incident to the vertex u form the wanted factor in the original graph G .

In the opposite direction, any factor of G' can be easily transformed to a perfect matching of G''. $\square$

The statement and the proof of the directed case is similar. Now, we can more precisely specify constraints for both endvertices of an edge.

Lemma 3.3 *Let D be an orientation of a multigraph G, and for every vertex u an interval $I_u = [a_u, b_u] \subseteq [0, deg_G(u)]$ is given. Moreover, let each edge $\vec{e} = [u,v] \in \vec{E}(D)$ has assigned a set of pairs $J_{\vec{e}} \subseteq (\{0,1\} \times \{0,1\})$. The question whether there exists a subset of halfedges $S \subseteq E^{\frac{1}{2}}(G)$ satisfying*

$$\forall u \in V(G) : |[u,e] \in S : e \in E(G))| \in I_u$$

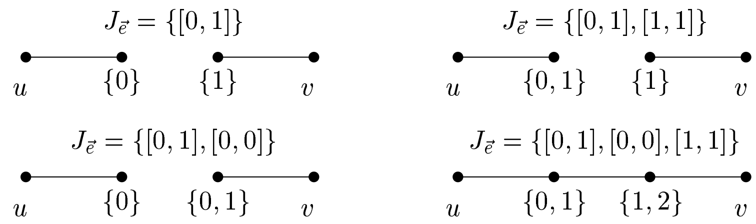

Figure 3.1: The four replacement cases

$$\forall \vec{e} = [u,v] \in \vec{E}(D) : [|[u,e] \in S|, |[v,e] \in S|] \in J_{\vec{e}}$$

is solvable in polynomial time.

Proof: We describe a replacement procedure, that should be performed on all edges of the graph G. If the set $J_{\vec{e}}$ contains none or both of unsymmetric pairs $[0,1]$ and $[1,0]$, then the orientation of e is not decisive and we can modify e as showed in Lemma 3.2.

We show a replacement procedure for an edge that contains exactly one unsymmetric pair in $J_{\vec{e}}$. W.l.o.g. suppose, that $[0,1] \in J_{\vec{e}}$. There are four possible cases for the set $J_{\vec{e}}$, and the corresponding replacement gadgets of $\vec{e} = [u,v]$ together with the definition of the intervals I for extra new vertices are depicted in Fig. 3.1.

Perform the replacement for all edges $\vec{e}$ of D and obtain the graph G'. Then, a factor F in G' satisfying the degree constraints I_v exists, if and only if D has a proper subset of halfedges S. $\square$

3.1.2 Edge precoloring extension

The edge coloring theory is one of the most developed part of the graph theory. We shall mention the Vizing theorem (see Theorem 1.4) and Holyer's result on the *NP*-completeness of the existence of a edge 3-coloring [32].

Kratochvíl and Sebö showed in [40] that the precoloring extension is *NP*-complete for the class of perfect graphs, when at least three distinct colors are used in the precoloring, or when the graph is precolored by two distinct colors and each of these two colors is used on at least two vertices.

The line graphs of bipartite graphs form a subclass of the class of perfect graphs, (see Theorem 1.5) and we prove that the hardness result of the precoloring extension holds also for this reduced class of graphs.

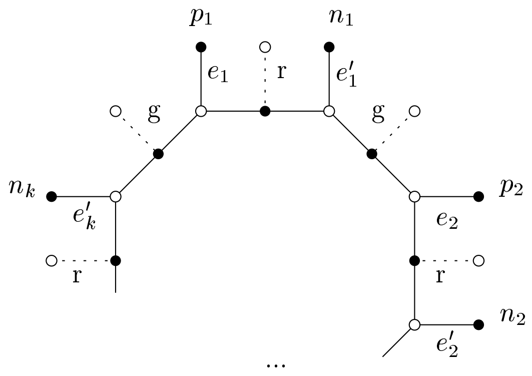

Figure 3.2: The variable gadget V

Theorem 3.4 *[15] The question, whether there exists a proper edge 3-coloring of a bipartite graph extending a given precoloring is a NP-complete problem.*

Proof: We show a reduction from the Not-All-Equal 3-SAT [21], problem LO3.

Let Φ be a formula in the normal form, and let each clause has three (not necessarily distinct) literals. We construct a graph G and define a coloring f on a subset of $E(G)$, s.t. f allows an extension to the entire graph G, if and only if Φ has an satisfying assignment, s.t. each clause contains a false valued literal.

We denote the three colors used in the edge-coloring of G by r, g and b and call them red, green and blue.

Assume that every variable in Φ has at most k positive and at most k negative occurrences. For each variable x, put into G an extra copy V^x of the graph depicted in Fig. 3.2.

For each clause $z = (L_1^z \wedge L_2^z \wedge L_3^z)$ of the formula Φ, put into G an extra copy of the graph C^z depicted in Fig. 3.3. We finish the construction of the graph G by series of unifications:

For every variable x and each literal L_i^z equal to x, unify the corresponding vertex l_i^z with an unique p_j^x that is not used by other literals. For each $L_i^z = \neg x$ unify l_i^z with an unique $n_{j'}^x$.

The graph G is bipartite. The classes of bipartition are indicated by white and black vertex color.

Define the precoloring f on the dotted edges of G, as depicted in Figures 3.2 and 3.3.

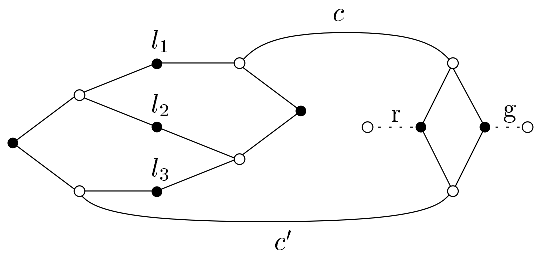

Figure 3.3: The clause gadget C

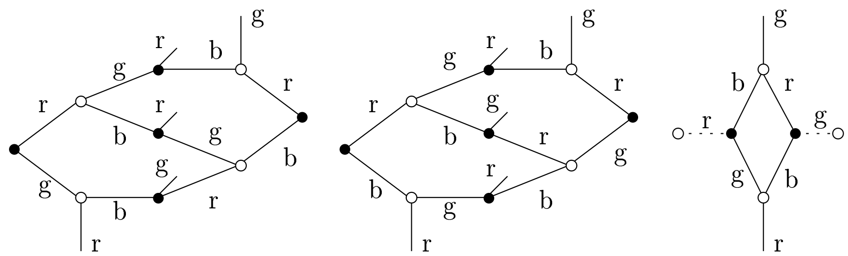

Figure 3.4: Coloring of the clause gadget C

Consider a proper edge coloring g of the graph G, that extends f. On every copy V^x, the edges $e_1, .., e_k$ and $e'_1, .., e'_k$ are colored red or green, and $g(e_1) = g(e_2) = ... = g(e_k) \neq g(e'_1) = ... = g(e'_k)$.

For each variable x, we assign x the true value if $g(e^x_1) = r$, and the false value otherwise.

We show that under the above assignment, each clause contains both positively and negatively valued literals. For a contradiction assume, that a clause z has all three literals positively valued. In the corresponding graph C^z, all three edges connecting vertices l^z_1, l^z_2 and l^z_3 to the variable graphs are colored red. The coloring cannot be extended to the entire graph C^z, since edges c and c' should be colored by the same color. This coloring cannot be extended to the right part of the clause gadget, a contradiction. The same argument proves the impossibility of the occurrence of three negative valued literals in C^z.

In the opposite direction, consider a proper assignment of variables of the formula Φ. For each variable x, color the edge e^x_1 red, if the variable x has assigned the true value, and color it green otherwise. Then each gadget V^x has an unique extension of the above coloring. Moreover, each clause gadget is connected to the rest of the graph by three edges, such that at least one is red and at least one is green. Fig. 3.4 shows that the coloring can be extended to the entire edge gadget C^z. The four remaining cases are obtained due to the symmetry of the graph and by the exchange of red and green color. $\square$

Corollary 3.5 *The edge precoloring extension problem is NP-complete for the class of cubic bipartite graphs.*

Proof: Use two copies of the graph G constructed in the previous proof, and merge each pair of the corresponding edges ending in a vertex of degree one into a single edge. In addition, join every pair of the corresponding vertices of degree two by an extra new edge. The new graph G' is cubic, and the formula Φ has a solution, if and only if both copies allow an edge precoloring extension. On the other hand, when a coloring of a single copy of G is given, it can be extended to the entire graph G'. $\square$

3.2 Results

We have already pointed that graphs whose (full) covering problem can be solved in polynomial time can bring a nontrivial characterization for the corresponding H-partial cover problem. In this section, we consider the flower $F(a_1, ..., a_k)$, banana $B(a_1, ..., a_k)$ and weight $W(a, b, c)$ graphs with various parameters.

Lemma 3.6 *Let H be a graph, and t be a positive integer. Denote by $H^{:t}$ the graph that arise from H by subdividing each edge by $t-1$ extra new vertices. Then the H-partial cover and $H^{:t}$-partial cover problems are polynomially equivalent.*

Proof: If f is a partial covering projection $G \to H$, it is easy to extend f into a partial covering projection $G^{:t} \to H^{:t}$.

In the opposite direction, let G be the graph, whose partial covering projection to $H^{:t}$ is questioned. Without lost of generality, we assume that G is connected, otherwise we can test each component of G separately.

Suppose $G = C_k$. The length of an arbitrary cycle in H is divisible by t, hence, G can cover $H^{:t}$, if and only if k is divisible by t, and $C_{k/t}$ covers H.

Now consider $G \neq C_k$. Call a path *maximal subpath* of G, if all its inner vertices are of degree two in G, and its endpoints have degree different from two. In G, replace every maximal subpath of length k by a path of length $\lceil k/t \rceil$. Call the new graph G'.

Every pair of vertices of $H^{:t}$ of degree different from two are at distance that is a multiple of t. Then the same property holds for vertices of degree at least three in G, or G cannot cover $H^{:t}$. Hence, any maximal subpath of length k in G' between vertices of degree at least three corresponds to a path of length kt in G. A similar argument shows that any maximal path of G' of length k, that ends with a leaf, corresponds to q maximal subpath of length at most kt in G.

Then, G partially covers $H^{:t}$, if and only if G' covers H, and every pair of vertices of G that are of degree at least three, has distance divisible by a. $\square$

Corollary 3.7 *The $W(ta, tb, tc)$-partial cover problem is polynomially equivalent to the $W(a, b, c)$-partial cover problem. Similarly, the $F(ta_1, ..., ta_k)$-partial cover, and the $B(ta_1, ..., ta_k)$-partial cover problem are polynomially equivalent to the $F(a_1, ..., a_k)$-partial cover and to the $B(a_1, ..., a_k)$-partial cover, respectively.*

3.2.1 Two distinct parameters

We first consider the situation, when at most two distinct parameters a, b appear in the specification of flower, banana and weight graphs. Instead of $F(a, a, .., a, b, ..., b)$, we write $F(a^i, b^j)$, where i and j denote the multiplicity of the parameters a and b, respectively. For simplicity, we drop the zero exponent term in our notation, i.e., $F(a^i) = F(a^i, b^0) = F(b^0, a^i)$. The same notation we use for banana graphs.

Due to Corollary 3.7, we suppose through this section that a and b are relatively prime.

Theorem 3.8 *The $F(a^i, b^j)$-partial covering problem is solvable in polynomial time for all a and b and every i, j.*

Proof: Let G be the graph whose partial covering to $F = F(a^i, b^j)$ is questioned. In the proof, the order of parameters a, b does not matter, and we assume without lost of generality that $i \geq 1$.

Assume that G is connected, otherwise we perform the following computation separately on each component of G. If G is a cycle, then it covers F if and only if its length is a nonnegative linear combination of a and b (when $i, j \geq 1$) or a multiple of a (when $j = 0$). This question can be easily tested in constant time.

Now, assume G is not a cycle, and denote v the central vertex of F. By the local injectivity, every vertex of G of degree at least three must be mapped onto v under any partial covering projection. It remains to decide where vertices of degree at most two of G will be mapped. Consider a maximal subpath of length l in G with both endpoints of degree at least three. We decide whether none, one or both terminal edges of the path can be mapped into a cycle of length a in F. This decision can be done in fixed time, since for $l > ab$ all three cases are possible. Denote the set of all possible cases by $J(l)$, more formally, put $0 \in J(l)$, if the equation $l = pa + qb$ allows a nonnegative integer solution with $q \geq 2$, let $1 \in J(l)$, when $p, q \geq 1$, and finally $2 \in J(l)$, if $p \geq 2$.

In G, replace each maximal subpath of length l by a single edge e, and put $J_e = \{0, 1, 2\}$, when e ends in a vertex of degree one, and put $J_e = J(l)$ otherwise. Call the new graph G'.

Assign $I_u = [\max(deg(u) - 2j, 0), \min(deg(u), 2i)]$ to every vertex u of G' and ask whether a proper subset of halfedges S for G' exists, with respect to the sets I_u and J_e. Due to Lemma 3.2 the question can be tested in polynomial time. If the result is negative, then G cannot partially cover $F(a^i, b^j)$, since the existence of the set S is necessary.

Suppose that the set of halfedges S exists. There is a natural correspondence of the set of halfedges $E^{\frac{1}{2}}(G')$ and the set $E(G'^{:2})$. Denote G'_a the bipartite subgraph of $E(G'^{:2})$ restricted to the halfedges of S. The upper bound of each interval I_v shows that $\Delta(G'_a) \leq 2i$. Hence, edges of G'_a can be properly colored by at most $2i$ colors in polynomial time. We use these $2i$ colors to distinguish between $2i$ starting segments of cycles of length a in the graph $F(a^i, b^j)$.

Repeat the above coloring procedure also for the set $\overline{S} = E^{\frac{1}{2}}(G') \setminus S$, and get a similar coloring of halfedges that will be mapped in cycles of length b. Now the coloring uses at most $2j$ colors, different from the $2i$ colors reserved for a-cycles.

Let us summarize what we have computed so far. We have constructed a graph G' and colored its halfedges by at most $2(i+j)$ colors that correspond to $2(i+j)$ halfedges incident with the central vertex v. Now, in the graph G, partially cover every maximal subpath $u_0, .., u_l$ joining two vertices of degree at least three into F, such that both endvertices u_0 and u_l are mapped onto the central vertex v, and vertices u_1, u_{l-1} are mapped into the cycles of F, that are used as colors on halfedges $(u_0, (u_0, u_l))$ and $(u_l, (u_0, u_l))$. Similarly, find a partial covering of the other maximal subpaths of G, but remember that only the path ends of degree at least three needs to be mapped onto the central vertex. □

In contrary to the previous theorem, the banana graphs with two distinct parameters allows both polynomially solvable and *NP*-complete instances.

Theorem 3.9 *The $B(a^i, b^j)$-partial covering problem is solvable in polynomial time, if a and b are both odd, or i or j are equal to zero.*

Proof: Note that due to Corollary 3.7 the $B(a^i)$-partial covering problem is equal to the $B(1^k)$-partial covering problem, that is equal to the edge coloring of bipartite graphs, and can be solved in polynomial time (see Theorem 1.5).

The proof is based on a similar argument like proof of Theorem 3.8. We expose the differences from the previous proof.

Now, assume $i, j \geq 1$, and since the proof is independent on the relative size of a and b, we also assume $i \geq 2$.

Let G be the input graph for the B-partial covering problem, and as above, we assume that G is connected. If G is a cycle, its length l is a nonnegative linear combination $ap + bq$ with $p + q$ even and $q = 0$ when $j = 0$, and with $q \leq p$ when $j = 1$ respectively. This test can be performed in fixed time.

Denote by v and w the two vertices of $B = B(a^i, b^j)$ that have degree at least three. Observe that the graph B is bipartite, and that the vertices v and w belong to the different classes of the bi-partition.

The graph G is bipartite and any pair vertices of degree at least three maps onto the same target (v or w), whenever they belong to the same class of bi-partition, or no partial covering projection exists. We fix one of the two possible mappings on vertices of degree at least three and denote it by f.

Create the graph G', and compute sets $J(l)$. When $j = 1$ then only linear combinations with parameters $p - 1 \geq q$ are allowed. Assign sets J_e and $I_u = [\max(deg(u) - j, 0), \min(deg(u), i)]$, and ask for a subset of halfedges S. As above the existence of S is the necessary and sufficient condition for the existence of a partial covering projection $f : G \to B$.

Consider the graph G'_a induced by halfedges from S and determine a proper edge coloring using at most i colors. This is always possible since G'_a is bipartite and $\Delta(G'_a) \leq i$. This coloring helps us to extend the mapping f onto beginning segments of maximal subpaths G that maps onto a-paths of B. Finally perform the same procedure for the complement of S, and extend f onto the entire graph G. □

Theorem 3.10 *The $B(a^i, b^j)$-partial covering problem is NP-complete whenever $|a - b|$ is odd, and $i, j \geq 1$.*

Proof: For $i \geq 2$ we show a reduction from the $BW(i,j)$ problem, and we reduce the $BW(j,i)$ problem in the case $i = 1$.

Assume a is odd, b is even, and both parameters are relatively prime. We discuss the case $i, j \geq 2$ first. Let G be the $i + j$-regular graph whose black and white coloring is questioned. We replace each edge of G by a path of length $l = ab$.

We claim that the new graph G' partially covers $B = B(a^i, b^j)$, if and only if a proper $BW(i,j)$-coloring of G exists. Consider a partial covering projection $f : G' \to B$. All original vertices are mapped onto v or w, the vertices of degree at least three in B. Color a vertex $u \in V(G)$ black, if $f(u) = v$ and color it white otherwise. There are only two ways how to express $l = ab$ as nonnegative linear combination $ap + bq$: either $p = b, q = 0$ or $p = 0, q = a$. Thus a maximal subpath, that is covered only onto b-paths of B, has ends mapped onto distinct vertices of B, whereas both ends are mapped onto the same target, if a a-pattern is used. Due to the local injectivity of the partial covering projection and the fact that every vertex of G has degree $i + j$, exactly i neighbors of any vertex of degree at least three are mapped into an a-path, and exactly j neighbors are mapped into a b-path of B. Obviously, the black and white coloring derived from the partial covering is a proper $BW(i,j)$ coloring.

For the opposite direction, consider any $BW(i,j)$ coloring of the graph G. The subgraph of G spanned by the edges connecting vertices with the same color is i-regular and we denote it by G_s. The graph $G_s^{:2}$ is bipartite with maximum degree i, and due to Theorem 1.5 its proper edge-coloring using i colors always exists. This edge coloring determines the mapping from G' into a-paths of B as follows: i different colors represent i different a-paths of B. Since the beginning segments on any maximal subpath connecting vertices with the same color should be mapped onto different a-paths, such mapping always exists (remember that b is even, $j \geq 2$).

Similarly, subgraph of G', spanned by the edges interconnecting sets of white and black vertices, is bipartite and j-regular, and can be colored using j colors. These edge colors represent different b-paths of B, and for every edge color c, we use a partial covering pattern, that starts and ends inside the c-th b-path to cover all maximal subpaths, that correspond to c-colored edges of G'.

The mapping defined above is locally injective on the neighborhood of every vertex of G', hence, it is a partial covering projection $G' \to B$.

Now, consider the $B(a^i, b)$ and $B(a, b^i)$-partial covering problems. We show a reduction to the $BW(i, 1)$ problem. The base idea and several arguments are inherited from the previous case. Let G be the $(i+1)$-regular graph whose black and white coloring is questioned. Replace every edge of G by a path of length l where

- $l = ab + (a-1)a$ for the reduction to the $B(a^i, b)$-partial covering problem,
- $l = ab + (b-1)b$ for the $B(a, b^i)$-partial cover.

Suppose, that the new graph $G' = G^{:l}$ partially covers $B(a^i, b)$. There are only two possibilities to cover a path of length $l = ap + bq$ with both ends mapped onto vertices v and w, namely, $p = a + b - 1, q = 0$ and $p = a - 1, q = a$. The corresponding patterns are $l = a + a + \cdots + a$ and $l = b + a + b + a + \cdots + b$, and in the first case both ends of the path are mapped onto the same target, while at the second case, one end is mapped on v and the other onto w. Note, that it is impossible to use two b-paths consecutively, since it violates the local injectivity around vertices v or w. As in the above case, the existence of a partial covering gives us a proper $BW(i, 1)$ coloring. When a $BW(i, 1)$ exists, it is possible to find a partial covering by the (half)edge coloring argument.

Finally consider a partial covering projection $G' \to B(a, b^i)$. The equation $l = ap + bq$ allows only the following solutions: $p = 0, q = a + b - 1$ and $p = b, q = b - 1$ that corresponds to partial covering of a maximal subpath of length l, namely by patterns $l = b + b + \cdots + b$ and $l = a + b + a + b + \cdots + a$. The only difference from the previous case is that the covering pattern that starts with a b-path corresponds to an edge connecting two vertices with the same color (observe that the number of summands is even), while the pattern with the a-path corresponds to a edge in G that connects white and black vertex. The already presented edge coloring argument shows that a partial covering projection $G' \to B(a, b^i)$ exists, whenever a proper $BW(i, 1)$ coloring is given. $\square$

Combining together Theorems 3.9 and 3.10 and Corollary 3.7 we get the following complete classification:

Corollary 3.11 *The $B(a^i, b^j)$-partial covering problem is polynomially solvable if a and b are divisible by the same power of two, and is NP-complete otherwise*

Now, we focus our attention to the third class of "simple" graphs, namely to the class of weight graphs. Recall, that due to Theorems 2.16, 3.1 and

Corollary 3.7 the $W(a,a,a)$-partial cover problem is *NP*-complete. Surprisingly there are parameters a and b that the weight partial covering problem allows a tractable — polynomial time algorithm.

Theorem 3.12 *The $W(a,b,b)$-partial covering problem is polynomially solvable, when the parameter a is odd, and b is even.*

Proof: Observe that the graph $W = W(a,b,b)$ is bipartite, hence, only bipartite graphs G can partially cover W, and classes of bi-partition of G determine the mapping f on vertices of degree three, as in the proof of Theorem 3.9. Denote by v, w the two vertices of degree three in W, and color a vertex $u \in V(G)$ of degree three black, if $f(u) = v$, and color it white when $f(u) = w$. Thus, the "hard" problem is to determine the mapping on vertices of degree at most two, and it can be solved by a simple procedure: For each maximal subpath of length l connecting two vertices of the same color, determine whether $l = ap + bq$ allows a nonnegative solution with p even and $q \geq p/2 - 1$. Any maximal subpath connecting vertices of different colors can cover W, when $l = p$, or if $l = ap + bq$ has a solution satisfying $q \geq (p-1)/2 - 1$ and p is odd and greater or equal to three.

The local injectivity on vertices of degree three — namely the decision which initial segments will be mapped onto a-paths — can be tested by the halfedge coloring procedure described in the proof of Theorem 3.9. $\Box$

Theorem 3.13 *The $W(a,b,b)$-partial covering problem is NP-complete, if the parameter b is odd.*

Proof: We show a reduction from the $BW(2,1)$ problem. Let G be a cubic graph whose black and white vertex coloring is questioned.

Replace each edge of G by a path of length $l = ab + (b-1)b$ to obtain the graph $G' = G^{:l}$ and suppose that a partial covering projection $f : G' \to W = W(a,b,b)$ exists. Color vertices of G, such that a vertex u of degree three gets black color, if $f(u) = v$, and is colored white when $f(u) = w$. The length l can be expressed either as $a + a + \cdots + a$, or $b + b + \cdots + b$. Hence, each vertex has two neighbors of the same color (when the b-pattern is used in G' along the corresponding path), and exactly one vertex of the opposite color: note, that the number of summands equal to a in the expression $l = a + b + a \cdots + b + a$ is odd.

In the opposite direction assume that G allows a $BW(2,1)$-coloring. The maximal subpaths of G' can be partially covered into W exactly by the same way, as was shown in the proof of $B(a,b,b)$ problem. $\Box$

Theorems 3.12 and 3.13 gives us also the full characterization for the class of $W(a,b,b)$-partial covering problems, i.e., the problem of testing the existence of a partial covering of weight graphs with both cycles of the same length.

Corollary 3.14 *The $W(a,b,b)$-partial covering problem is polynomially solvable, whenever b is divisible by a strictly higher power of two than a, and is NP-complete otherwise.*

3.2.2 Three parameters

One of the necessary conditions of the existence of an partial covering projection states that the mapping of a selected maximal path allows only a restricted set of patterns. We perform the first classification of the maximal subpaths by the length of the path.

For this purposes we introduce an argument based on the solving of a equation in natural numbers with special requirements:

Definition *Let $J = \{j_1, ..., j_k\}$ be a set of distinct positive integers. We say that m has a* path covering pattern *with respect to J of type (a,b) and length l, if there exist integers $x_i, 1 \le i \le l$ satisfying*

- $m = x_1 + \cdots + x_l$
- $x_i \in J,\ 1 \le i \le l$
- $x_1 = a,\ x_l = b,$
- $x_{p-1} \ne x_p \ne x_{p+1}$ *whenever* x_{p-1} *or* x_{p+1} *are defined.*

Note that whenever m has a solution of type (a,b), then it can be transformed into a solution of type (b,a) and the same length. Hence, the type of a solution will be always expressed by an unordered pair.

Now, we focus our attention to “simple” graphs with three different parameters a, b and c. Although we cannot give a complete characterization, there still appear several *NP*-complete instances of the H-partial covering problem as well as polynomially solvable cases. We start the classification by banana graphs.

Lemma 3.15 *The $B(a,b,c)$-partial cover problem is NP-complete whenever there exists m, such that m has a path covering pattern of type (c,c) of an odd length, and a pattern of type (a,b) of an even length, and no other covering patterns exist with respect to $J = \{a,b,c\}$.*

Proof: We show a reduction from the $BW(2,1)$-coloring problem. Let G be a cubic graph, whose black and white coloring is questioned. We replace each edge of G by a path of length m, and show that the new graph $G' = G^{:m}$ allows a partial covering to $B = B(a,b,c)$, if and only if G has a proper $BW(2,1)$-coloring.

Denote by v, w the two vertices of degree three in the graph B, and assume that a partial covering projection $f : G' \to B$ exists. Then every vertex of degree three in G' is mapped either on v or w. Color each vertex $u \in V(G)$ black, if $f(u) = v$, and color it white otherwise. The mapping f is locally injective on neighborhood of all u in G', hence, one of the incident edges (u, u') is mapped into a a-path. The maximal subpath of length m that starts by the exposed edge can be covered only by the pattern of type (c, c). The odd length of the path covering pattern implies that the opposite end of the maximal subpath will be mapped onto the other vertex of degree three in B, causing that u' gets a different color from the color of u.

By the same argument we can show that the even length of the path covering pattern of type (a, b) implies, that every vertex of G has two neighbors colored by the same color.

In the opposite direction, assume a $BW(2,1)$-coloring of the graph G. The partial covering projection can be found by the technique already described in the proof of *NP*-completeness of $B(a,b,b)$-partial covering problem. □

Corollary 3.16 *The NP-completeness of the $B(a,b,c)$-partial covering problem is maintained, even if at most one of the following cases occurs:*

- *m has a covering pattern of type (a,a) or (b,b) of an even length, or*
- *m allows a pattern of type (a,c) of any length, or*
- *m has a pattern of type (b,c) of any length,*

in addition to the mandatory covering patterns of type (a,b) and (c,c) described in Lemma 3.15.

Proof: The existence of the new covering patterns does not influent the fact that any black and white coloring of graph G can be transformed into a covering projection $f : G' \to B$.

We shall prove that the opposite implication is still valid. Any pattern described in the first case makes no contradiction, since the paths which covering starts by the a or b-segment, yield a vertex of the same color.

One could be more careful when discussing the existence of a pattern of type (a, c). Every vertex of degree three in G' has assigned three maximal paths, and the covering projection of each of these three maximal subpaths starts by a different path of B, namely a, b and c-path. The number of appearance of the a-path as a stating segment is the same as the number for b-path or c-path.

On the other hand, even a single use of the covering pattern of type (a, c) in the covering projection breaks the equality (the a-path is used more frequently than the b-path), and hence the pattern of type (a, c) cannot appear in the covering projection $G' \to B$.

Due to the symmetry between a and b, we get the same argument for the third case considering covering patterns of type (b, c). □

Theorem 3.17 *[17] The $B(a, b, c)$-partial cover problem is NP-complete whenever $a + b$ divides c.*

Proof: We apply Lemma 3.15 for m equal to c. The only covering patterns are $m = c$ of type (c, c) (odd length) and $m = a + b + a + b + \cdots + a + b$ of type (a, b) (even length). □

The above approach yields the complete characterization of the computational complexity of the H-partial covering problem for banana graphs with parameters $1, 2$ and c.

Theorem 3.18 *[17] The $B(1, 2, c)$-partial cover problem is NP-complete for all $c \geq 3$.*

Proof: If $c = 3k$, then the result follows directly from Theorem 3.17.

When $c = 3k + 1$, then putting $m = c + 1$, we get the following covering patterns $m = 2 + 1 + 2 + 1 + 2 \cdots + 2$ and $m = c + 1$. Similarly, for $c = 3k + 2$ we select $m = c + 2$, and get patterns $m = 1 + 2 + \cdots + 1 = 1 + c + 1$ and $m = c + 2$. □

The smallest triple of parameters for the $B(a, b, c)$-partial cover problem which is not tractable by the above technique is $1, 3$ and 5. It is a direct consequence of the fact that the banana graph $B(1, 3, 5)$ is bipartite, hence the distribution of vertices of degree three of an input graph G into the classes of the same target under a partial covering projection to $B(1, 3, 5)$ follows from the bipartition of the graph G, and can be solved in polynomial time.

We use the result on the edge precoloring extension to prove, that finding a partial covering projection to $B(1, 3, 5)$ is a *NP*-complete problem.

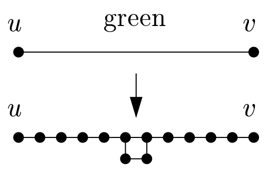

Figure 3.5: The green gadget

Proposition 3.19 *The $B(1,3,5)$-partial covering problem is NP-complete.*

Proof: We show a reduction from the edge precoloring extension problem described in Theorem 3.4. Assume G is a cubic bipartite graph, whose edges are either blank or precolored by red, green or blue.

We replace each blank edge by a path of length eleven, each blue edge by a path of length three, and each green edge by the gadget depicted in Fig. 3.5. Denote the new graph by G'.

Consider a partial covering projection $f : G' \to B(1,3,5)$. The graph G' has maximal subpaths of lengths 1, 3, 5 and 11 and by exploring all possible cases, we get that all covering patterns are of type $(1,1)$, $(3,3)$ or $(5,5)$ of an odd length, namely the trivial patterns, $1 = 1$, $3 = 3$, $5 = 5$ and four nontrivial patterns $5 = 1+3+1$ and $11 = 1+5+1+3+1 = 3+5+3 = 3+1+3+1+3 = 5+1+5$. We color every edge of G red if the covering of the corresponding maximal subpath in G' starts with a 1-path of $B = B(1,3,5)$, we color it blue if the 3-path is used as the starting segment, and green for the 5-paths, respectively.

Obviously, the new coloring of the graph G is an extension of the given precoloring (note that the pattern 1+3+1 is never used due to the construction of the green gadget), and the local injectivity of the partial covering projection implies that every vertex of G has all three incident edges colored by pairwise distinct colors.

Now, consider a proper extension of the precoloring of G. It can be simply transformed to a partial covering projection $G' \to B$, by using the covering patterns described above. □

The covering projection to a weight graph has a more complicated structure. We present a polynomial reduction from the $BW(2,1)$-coloring problem to show that there are several *NP*-complete instances, but for this purposes we require different properties of the covering patterns.

Assume a maximal subpath of length m (with both ends of degree three) in a graph G, that partially covers $W = W(a,b,c)$. Then m can be expressed as a sum $x_1 + \cdots + x_l$ satisfying:

- $x_i \in \{a, b, c\}$ $1 \leq i \leq l$,
- if $x_i = a$ then $x_{i-1} \neq a$ and $x_{i+1} \neq a$, whenever x_{i-1} or x_{i+1} are defined,
- if $x_i = x_j \in \{b, c\}, i < j$, then the number of summands among $x_{i+1}, ..., x_{j-1}$ equal to a is even.

Observe, that the above properties also imply, that whenever $x_i = b$ and $x_j = c,\ i < j$, then the number of a elements among $x_{i+1}, ..., x_{j-1}$ is odd.

Definition *Call the expression $m = x_1 + \cdots + x_l$ the* weight covering pattern *of type (x_1, x_l) if all three properties defined in the previous paragraph are satisfied. Define the parity of the pattern as the number of elements from the sum, that are equal to a.*

Lemma 3.20 *The $W(a, b, c)$-partial covering problem is NP-complete, whenever there exists an integer m, such that the only weight covering patterns with respect to (a, b, c) are of type (a, a) and the odd parity, and of type (b, b) and (c, c) (of an even parity due to the definition) and that for each allowed type and parity at least one covering pattern exists.*

Proof: The reduction from the $BW(2, 1)$ problem is straightforward, and is done by the same method as in Lemma 3.15. Recall that the partial covering of graph $G' = G^{:m}$ onto $W = W(a, b, c)$ defines a proper black and white coloring, where the odd parity along the pattern of (a, a) type forces distinct colors of vertices incident to the corresponding edge in G, while the patterns of an even parity connect vertices of the same color.

The opposite direction is even simpler, since each color class of vertex color of G use patterns with both ends colored by the fixed color. $\square$

Theorem 3.21 *The $W(a, b, c)$-partial cover problem is NP-complete, whenever a is a common multiple of b and c.*

Proof: Put $m = a$. The only possible covering patterns are $m = a = b + b + \cdots + b = c + c + \cdots + c$, that are required for the application of Lemma 3.20. $\square$

Corollary 3.22 *The $W(a, 1, 2)$-partial covering problem is NP-complete for all even $a \geq 2$.*

We conclude the section by showing that the seemingly harder class of $W(a,b,c)$-partial covering problem surprisingly allows a polynomially solvable instance.

Proposition 3.23 *The $W(1,2,4)$-partial cover problem is solvable in a polynomial time.*

Proof: Let G be the graph whose partial covering to $W = W(1,2,4)$ is questioned. Call $v \in V(W)$ the vertex of degree three belonging to the cycle C_4, and call w the other vertex of degree three. We refer the edge (v,w) as the central edge of W.

The multigraph W is bipartite, hence we can assume that G is bipartite too, otherwise no partial covering exists. If G is a cycle, then a partial covering exists, if and only if G is an even cycle. In the following, we assume G has at least one vertex of degree three. Find the bipartition of $V(G) = A \cup B$, and denote A_3, B_3 the vertices from the set A of degree three, or from the set B respectively.

We show a test of the existence of a partial covering projection $f : G \to W$ satisfying $f(A_3) = v, f(B_3) = w$. By the symmetry of sets A and B, we can perform the same test with sets A and B interchanged. If both tests fail, then no partial covering projection exists.

Let f be a mapping on vertices of degree three, which we want to extend into a partial covering of the entire graph G. Consider a maximal subpath of length l in G with both endpoints u, u' of degree three. According to the length l and mapping of its endvertices, it can be decided in constant time, whether there exists a partial covering of the maximal subpath extending f, and having none, single or both initial edges mapped onto the central edge of W. Denote the set of all possibilities $J(l, f(u), f(u')) \subseteq \{0,1\}^2$. Note that unsymmetric pairs $[0,1]$ and $[1,0]$ can occur in $J(l,v,w)$.

We build a graph G' by replacing each maximal subpath of length l connecting vertices u and u' by a single edge, and put $J_{[u,u']} = J(l, f(u), f(u'))$ whenever both u and u' are of degree three and $J_{[u,u']} = \{0,1\}^2$ otherwise.

In addition put $I_u = \{1\}$ for all vertices of G', and ask whether there is a proper subset of halfedges $S \subseteq HE(G')$, satisfying oriented constraints given by sets I_u and $J_{[u,u']}$. Due to Lemma 3.3 this problem can be solved in polynomial time.

The existence of the set S is a necessary condition for any partial covering, and we show that it is also a sufficient condition. By the definition of sets $J_{[u,u']}$ there always exists a weighted covering pattern of the corresponding maximal subpath connecting vertices u and u', mapping only those initial segments on the central edge, that are selected by the subset of halfedges S, and mapping the other initial segments into cycles in W. Therefore, the subset of halfedges S can be transformed in polynomial time into a partial covering projection $G \to W(1,2,4)$. □

3.3 Further research

We showed that the complete characterization of computational complexity of H-partial covering problems is close related to the complexity characterization of the class of H-covering problems.

An open instance of the H-partial covering problem is rather expected to be *NP*-complete, not only for the strict inclusion of the *NP*-complete instances, but also due to the fact that the algorithm based on the classes of degree refinement does not work for partial covers. We expect that many large ground graphs are *NP*-complete instances of the H-partial covering problem.

It seems, that the variety of the cycle structure of the graph is essential for the *NP*-completeness of the H-partial covering problem, and although we are far away to prove the statement, we offer the following conjecture.

Conjecture 3.24 *The H-partial covering problem is NP-complete whenever H contains K_4 as a minor.*

We have illustrated the diversity of complexity results of the H-partial covering problem on "simple" graphs whose H-covering problem is known to be polynomially solvable, and showed that many of them turn to be *NP*-complete instances when asking for partial covers. Moreover, the cases that remain polynomially solvable require a more complex technique, in this thesis usually based on the matching algorithm.

We shall finally remark that bipartite targets require a special approach, since the partial covering projection on vertices of higher degree is partially determined by the classes of bipartition, that might substantially simplify the computational complexity of the corresponding H-partial covering problem.

Chapter 4

The λ-labeling problem

4.1 Motivation

We start this chapter by a practical motivation for the graph theoretic model.

The telecommunication industry uses the modulation of electromagnetic waves for signal transfer, e.g., in the television or radio broadcasting, or in mobile telephony networks. The radio transmitters generate and receive such a signal, and are distributed onto earth surface in a network, that covers as largest area as possible. Each transmitter uses one or several frequencies, so that any device that is close enough and has tuned the same frequency can establish the communication.

When two transmitters are close enough and have assigned the same or almost the same frequency, then their simultaneous broadcasting causes that these waves interfere so no reception is possible at their neighborhood.

The range of possible wavelengths is limited due to the physical and organizational reasons, so there is a natural motivation to reuse the same frequencies on distant transmitters, and cover the largest area by the shortest range of frequencies, while maintaining the necessary difference between close transmitters.

There are several models considering various aspects — global or local interference, static or dynamic system, frequency separation or distance separation, see [43, 56] for an overview of possible models.

We present a simple graph theoretic model for the above optimization problem. Transmitters are represented as vertices of a graph, where edges connect vertices in a close distance. We suppose, that each transmitter (vertex) needs to assign only one frequency (i.e., a nonnegative real number), that the difference between two frequencies is simply calculated by the subtraction, and that this difference should be greater than some prescribed constant c. A practical implementation should demand a proportional frequency difference, but after applying the logarithmic transformation of frequencies we get the "subtraction" difference.

The next simplification step allows us consider only natural-valued frequencies, because instead of assigning real numbers, it is possible to use multiples of c [25] and perform all calculation and comparison modulo c.

We have translated the problem of selecting a suitable set of frequencies to the graph coloring problem, because we ask for a minimal numbering of vertices of a given graph, such that adjacent vertices get different numbers.

We have used several results from the the graph coloring theory in the chapter devoted to graph covers, because this discipline is one of the oldest and the best developed part of graph theory and many structural and complexity results were already discovered.

A more sophisticated model of the channel assignment problem considers also interference of transmitters at a quite longer distance. In the new model, we ask for an assignment of nonnegative integers to the vertices of a graph, and we demand that vertices at distance i have assigned numbers that differ by at least p_i, where $(p_1, p_2, .., p_k)$ is a fixed non increasing sequence of natural numbers.

More formally, we define:

Definition *A function* $c : V(G) \to \mathbb{N}_0$ *is called a* $L_{(p_1,..,p_k)}$-labeling, *if*

$$dist(u, v) = i \leq k \Rightarrow |c(u) - c(v)| \geq p_i.$$

Let λ *be a positive integer. If all vertices of a* $L_{(p_1,..,p_k)}$*-labeling have labels less or equal to* λ*, then we call the labelling a* $\lambda_{(p_1,..,p_k)}$-labeling.

The minimum λ *such that a graph* G *admits a* $\lambda_{(p_1,..,p_k)}$*-labeling is denoted by* $\lambda_{(p_1,..,p_k)}(G)$.

We are interested in the computational complexity of the determining the optimal $\lambda_{(p_1,..,p_k)}$-labeling. Hence we define the two following classes of decision problems:

Problem: $\lambda(p_1, .., p_k)$*-labeling problem*
Input: A graph G
Question: Is $\lambda_{(p_1,..,p_k)}(G) \leq \lambda$*?*

And more generally:

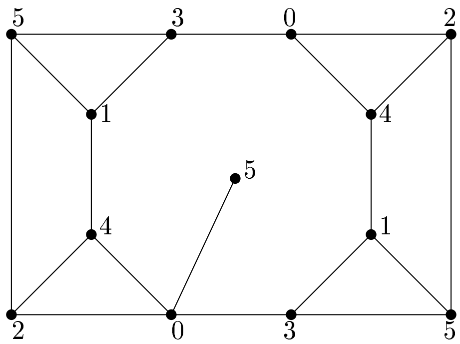

Figure 4.1: An example of a $5_{(2,1)}$-labeling of a graph

Problem: *$L(p_1, .., p_k)$-problem*
Input: A graph G, nonnegative integer λ
Question: Does G allow a $\lambda_{(p_1,..,p_k)}$-labeling?

The $L(p_1, .., p_k)$ decision problem generalizes the class of all $\lambda(p_1, .., p_k)$-labeling problems, because, if for a certain λ the $\lambda(p_1, .., p_k)$-labeling problem is *NP*-complete for a certain class of graphs, then for the same class of graphs, the $L(p_1, .., p_k)$-problem is *NP*-complete, as well.

While studying properties of a $L_{(p_1,...,p_k)}$-labeling, we can consider only labelings such that parameters $p_1, ..., p_k$ have no common divisor.

Lemma 4.1 *All graphs G and all constants $a \in \mathbb{N}$ satisfy:*

$$\lambda_{(ap_1,ap_2,..,ap_k)}(G) = a \cdot \lambda_{(p_1,p_2,..,p_k)}(G)$$

.

Proof: The $\leq$ inequality is obvious, since by multiplying each label of an optimal $\lambda_{(p_1,p_2,..,p_k)}$-labeling by a, we got a $\lambda_{(ap_1,ap_2,..,ap_k)}$-labeling of G. On the other way when a $\lambda_{(ap_1,ap_2,..,ap_k)}$-labeling of G is given, we replace each label c by $\lfloor c/a \rfloor$. We obtain a $\lambda_{(p_1,p_2,..,p_k)}$-labeling of G, because, if $|c - c'| \geq ap_i$, then $|a\lfloor c/a \rfloor - a\lfloor c'/a \rfloor| \geq ap_i$ and $|\lfloor c/a \rfloor - \lfloor c'/a \rfloor| \geq p_i$. $\square$

In the rest of the thesis we will deal with labelings with two parameters p and q, and due to the above lemma we suppose, that without lost of generality, p and q are relatively prime. Also note, that any optimal $\lambda_{(p)}$-labeling is equivalent to a $\lambda_{(1)}$-labeling with all labels multiplied by p. Every $\lambda_{(1)}$-labeling of a graph is a ordinary coloring using at most $\lambda + 1$ colors.

We can express the minimum number of necessary labels as $\lambda_{(p)}(G) = p \cdot (\chi(G) - 1)$.

Corollary 4.2 *The $\lambda(p)$-labeling problem is NP-complete for $\lambda \geq 2p$ with respect to the class of all graphs.*

If parameters p and q are equal, then due to Lemma 4.1 it is sufficient to explore properties of the $L_{(1,1)}$-labeling of a given graph G. This labeling uses distinct labels along each edge, and, moreover, if two vertices share a common neighbor, then they have different labels too. In the other words, the labeling is locally injective to the set of labels, and when we model the labels $0, 1, .., \lambda$ as the vertices of complete graph $K_{\lambda+1}$, then each $\lambda_{(1,1)}$-labeling of G corresponds to a partial covering projection $G \to K_{\lambda+1}$.

We already showed by Theorems 2.17, 2.18 and 3.1 that the K_λ-coloring problem is *NP*-complete whenever $\lambda \geq 3$, and is polynomially solvable otherwise (see Corollary 2.12).

Corollary 4.3 *The $\lambda(p,p)$-labeling problem is NP-complete, for $\lambda \geq 3p$ and the class of all graphs.*

4.2 The $\lambda_{(2,1)}$-labeling problem

The first non-trivial parameters of the $L(p,q)$-labeling problem are parameters $p = 2, q = 1$. Historically this is the original form of a graph labeling problem with a condition at distance two [58, 25]. In this section, we review results on calculating $\lambda_{(2,1)}(G)$, and on the computational complexity of the $\lambda(2,1)$-labeling problem.

Griggs and Yeh showed that the number $\lambda_{(2,1)}(G)$ can be easily determined for paths, cycles and wheels.

Proposition 4.4 *[25, 58] Let P_n be a path on n vertices. Then $\lambda_{(2,1)}(P_2) = 2$, $\lambda_{(2,1)}(P_3) = \lambda_{(2,1)}(P_4) = 3$, and $\lambda_{(2,1)}(P_n) = 4$ for all $n \geq 5$.*

Proposition 4.5 *[25, 58] Let C_n be a cycle of length n. Then $\lambda_{(2,1)}(C_n) = 4$ for all $n \geq 5$.*

Proposition 4.6 *[58] Let W_n denote the* wheel *graph on $n+1 \geq 4$ vertices formed from a cycle C_n and a star S_n by unifying each vertex of the cycle with a unique vertex of degree one of S_n. Then $\lambda_{(2,1)}(W_n) = n + 1$ for all $n \geq 3$.*

The number $\lambda_{(2,1)}$ for trees can not be stated explicitly, but only two cases can occur.

Theorem 4.7 *[25] Let T be a tree with the maximum degree $\Delta(T) \geq 1$. Then $\lambda_{(2,1)}(T) \in \{\Delta+1, \Delta+2\}$.*

However Griggs and Yeh [25] conjectured that the $L(2,1)$-problem for trees is *NP*-complete, Chang and Kuo [8] gave a polynomial time algorithm. In Section 4.2.3 we show a polynomial time algorithm solving the $L(2,1)$-problem for the class of k-almost trees with fixed k.

Now consider the class of all graphs. Then the following upperbound holds:

Observation 4.8 *[25, 58] $\lambda_{(2,1)}(G) \leq \Delta(G)^2 + 2\Delta(G)$.*

Proof: We prove the observation by induction on the number of vertices. If G has one vertex, then the statement is trivially satisfied.

Select a vertex $v \in V(G)$ arbitrarily and label the graph $G \setminus \{v\}$. By induction hypothesis, there exists a labeling with the maximum label at most $\Delta(G)^2 + 2\Delta(G)$. We show that among numbers $[0, \Delta(G)^2 + 2\Delta(G)]$ there is at least one suitable for the label of the vertex v. The vertex v has at most $\Delta(G)$ neighbors, and the label of each of them blocks at most three possible labels for v. In addition there are at most $\Delta(G)^2 - \Delta(G)$ vertices at distance two from v, and their labels also can not be used as the label of v. This gives us at most $\Delta(G)^2 + 2\Delta(G)$ forbidden labels for v, and at lest one number remains in the interval $[0, \Delta(G)^2 + 2\Delta(G)]$ as a suitable label of the vertex v. $\square$

The above upperbound on $\lambda(G)$ was improved to $\Delta(G)^2 + \Delta(G)$ in [8].

It was conjectured by Griggs and Yeh [25], that all graphs with maximum degree $\Delta \geq 2$ allow a $(\Delta(G)^2)_{(2,1)}$-labeling. The conjecture is still open, even if it was proven for restricted classes of graphs like chordal graphs [55], or graphs of diameter two [25].

4.2.1 Partial covers and generalized $L(2,1)$-labeling

The general implementation of the channel assignment problem considers also spaces with non-linear metrics. For example, by using this approach, it is possible to describe an interference between a frequency and its multiples.

We will describe this setting as a graph-homomorphism model.

Definition *Let H be a simple graph. A* $H_{(2,1)}$-labeling *of a graph G is a mapping $f : V(G) \to V(H)$ satisfying:*

1. *if* $(u,v) \in E(G)$, *then* $dist_H(f(u), f(v)) \geq 2$,
2. *when* $dist_G(u,v) = 2$, *then* $f(u) \neq f(v)$.

In other words, any $H_{(2,1)}$-labeling of G satisfies the homogeneous condition: $dist_G(u,v) + dist_H(f(u), f(v)) \geq 3$.

In this concept, every $\lambda_{(2,1)}$-labeling is equivalent to a $(P_{\lambda+1})_{(2,1)}$-labeling.

For example, a labeling satisfying constraints $(2,1)$ with circular metric was considered by Leese, van den Heuvel and Shepherd in [56, 43, 42], and is equivalent to the $(C_n)_{(2,1)}$-labeling.

Proposition 4.9 *A graph G allows a $H_{(2,1)}$-labeling, if and only if G partially covers $\overline{H}$.*

Proof: Consider a $H_{(2,1)}$-labeling $f : V(G) \to V(H)$. The first condition is equivalent to the statement that f is a homomorphism to the complement of H, since the condition $dist_H(f(u), f(v)) \geq 2$ implies $(f(u), f(v)) \in E(\overline{H})$. The second condition expresses that c is a locally injective mapping. $\square$

The decision problem, parameterized by the graph H, which asks whether an input graph G admits a $H_{(2,1)}$-labeling, will be called the $H_{(2,1)}$*-labeling problem.*

Corollary 4.10 *The computational complexity of the $H(2,1)$-labeling problem is equivalent to the computational complexity of the $\overline{H}$-partial cover problem.*

In view of Theorems 2.17 and 2.18, and Corollary 4.10, we get the following statement:

Theorem 4.11 *The $C_n(2,1)$-labeling problem is NP-complete, if and only if $n \geq 5$.*

4.2.2 Computational complexity of the $\lambda(2,1)$-labeling problem

The concept of the $L_{(p_1,\ldots,p_k)}$-labeling of a graph is derived from the traditional graph coloring theory, and we have already shown that any proper graph coloring is equivalent to the $L_{(1)}$-labeling. Since the problem of decide whether a graph can be colored with k colors is *NP*-complete for every $k \geq 3$, we expect that every $L(p_1, \ldots, p_k)$-problem is *NP*-complete. We will

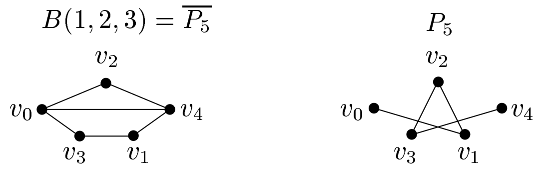

Figure 4.2: Graphs $B(1,2,3)$ and P_5

prove the above conjecture for all labelings with two parameters in Theorem 4.16.

In particular the *NP*-completeness of the $L(2,1)$-problem was explored by Griggs and Yeh in the following special form: They proved that the $L(2,1)$-problem is *NP*-complete, for graphs of diameter two and $\lambda = |V(G)|$ [25, 58].

Using a proof technique based on partial covers, we improve the above result to a full complexity characterization of the class of $\lambda(2,1)$-labeling problems.

Observation 4.12 *The* $4(2,1)$*-labeling problem is NP-complete.*

Proof: The banana graph $B(1,2,3)$ is the complement of the graph P_5, see Fig. 4.2. The *NP*-completeness of $B(1,2,3)$-partial cover follows from Theorem 3.17 and we get the *NP*-completeness of the $4(2,1)$-labeling problem due to Corollary 4.10. □

Our proof technique uses frequently the following lemma, which shows that on a certain set of vertices we can effectively reduce the set of possible labels.

Lemma 4.13 *Every vertex of degree* $\lambda - 1$ *is labelled either by 0 or by* λ*, under any* $\lambda_{(2,1)}$*-labeling.*

Proof: Due to Proposition 4.9 we can investigate the partial covering to $\overline{P_{\lambda+1}}$, instead of the $\lambda_{(2,1)}$-labeling. Any partial covering f is locally injective, and, therefore, $deg_G(u) \leq deg_{\overline{P_{\lambda+1}}}(f(u))$. The biggest degree of graph $\overline{P_{\lambda+1}}$ is $\lambda - 1$, and only two vertices, namely v_0 and v_λ, reach that maximal degree. Hence, every vertex $u \in V(G)$ of degree $\lambda - 1$ maps either onto v_0 or v_λ. □

Theorem 4.14 *[17] The* $\lambda(2,1)$*-labeling problem is NP-complete for all* $\lambda \geq 4$*, and is polynomially solvable otherwise.*

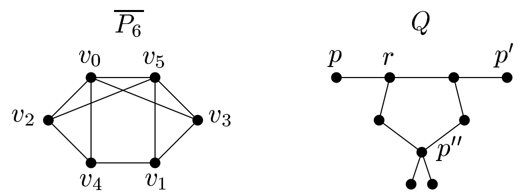

Figure 4.3: Graph $\overline{P_6}$ and the replacement graph Q

Proof: The case $\lambda < 4$ follows from Corollary 2.12. All graphs $\overline{P_1}$, $\overline{P_2}$, $\overline{P_3}$ and $\overline{P_4}$ have at most one cycle and hence the corresponding partial covering problems are solvable in polynomial time.

We prove the statement by the induction by two on λ. The case $\lambda = 4$ was proven by Observation 4.12. For $\lambda = 5$ we show a reduction from the $BW(2,2)$-coloring problem.

Let G be a fourregular graph, whose $BW(2,2)$-coloring is questioned. Replace each edge (u, u') of G by a graph Q depicted in the Fig. 4.3, such that u corresponds to p and vice versa for u'. Call the new graph G' and we show that G' covers $\overline{P_6}$ (see Fig. 4.3), if and only if G admits a $BW(2,2)$-coloring.

Due to Lemma 4.13, all vertices of degree four in G' are mapped onto v_0 or v_5. We define a coloring of G as follows: color a vertex u black, if its mirror in G' is mapped onto v_0, and color it white otherwise.

Note that all vertices $p, p', p'' \in V(Q)$ are mapped onto v_0 or v_5 under any partial covering to $\overline{P_6}$. The case study shows, that there are only six partial covering projections $f : Q \to \overline{P_6}$ satisfying $f(p), f(p') \in \{v_0, v_5\}$. All six cases are depicted in Fig. 4.4.

Consider a covering $f : G' \to \overline{P_6}$ and suppose that a vertex u of G is black. Denote its mirror in G' by u'. The vertex u' is mapped onto v_0 and its neighbors are bijectively mapped onto the vertices v_2, v_3, v_4 and v_5.

Suppose u' is identified with p in Q. When its neighbor r is mapped onto v_5 or v_2 then due to the case study the vertex p' is mapped onto v_0 and hence two neighbors of u have the same — black color. If $f(r) \in \{v_3, v_4\}$ then $f(p') = v_5$ and on the other two neighbors of u are white.

Colors of the neighbors of a white vertex can be discussed by the same argument.

For the opposite direction consider a $BW(2,2)$-coloring of the graph G. First consider a subgraph of G spanned by edges with both ends black. This graph is 2-regular, i.e., a set of disjoint cycles. We cover the corresponding

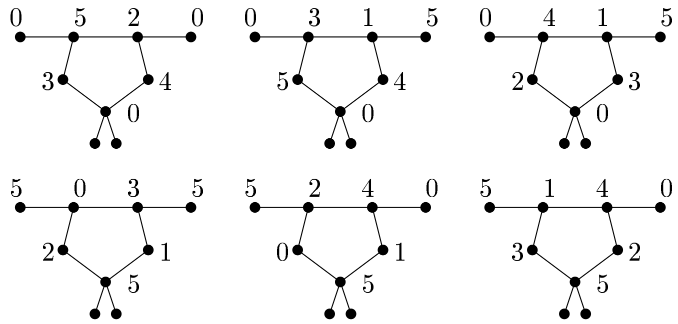

Figure 4.4: The six cases of partial covering $Q \to \overline{P_6}$

subgraph in G', s.t. on the cycle the pattern $v_0, v_2, v_5, v_0, v_2, v_5, v_0, \ldots$ is used and the original vertices are mapped onto v_0. By a similar argument, cover the subgraph of G' corresponding to the white cycles by the pattern $v_5, v_3, v_0, v_5, v_3, v_0, v_5, \ldots$.

The remaining edges connect the set of white vertices to the set of black vertices. These edges form a bipartite 2-regular factor of G, i.e. a set of even cycles. For the corresponding subgraph of G' we use pattern $v_0, v_3, v_1, v_5, v_2, v_4, v_0, \ldots$.

We defined an injective mapping on the neighborhood of the original vertices, since two neighbors of a mirror of a black vertex (mapped to v_0) are mapped onto v_2 and v_5 (in the "black" subgraph), while the other two neighbors are mapped onto v_3 and v_4 (in the "black–white" factor).

Induction step: We show that the $\overline{P_{\lambda+3}}$-partial covering problem is *NP*-complete, when $\lambda + 2 \geq 6$. We reduce the problem from the $\overline{P_{\lambda+1}}$-partial covering problem, which we assume to be *NP*-complete by the induction hypothesis.

Let G be an input graph for the $\overline{P_{\lambda+1}}$-partial covering problem. Form a binary tree T with at least $|V(G)|$ leaves, all of them in the same distance from the root. Denote by L_i the set of vertices of T at distance i from the root, and suppose that the layer L_{k-1} contains all leaves. Add into T a layer L_k with $|L_{k-1}|$ vertices, and connect it by a perfect matching to the vertices of layer L_{k-1}. Subdivide each edge of the tree by an extra new vertex, and join every vertex of G by an edge to a unique vertex from L_k, see Fig. 4.5. Finally introduce extra new leaves to increase the degree of vertices in all layers $L_i, i \leq k$ upto $\lambda + 1$. Call the new graph G'.

We show that G' partially covers $\overline{P_{\lambda+3}}$, if and only if G covers $\overline{P_{\lambda+1}}$. Consider a covering $f : G' \to \overline{P_{\lambda+3}}$. By Lemma 4.13 all vertices in layers

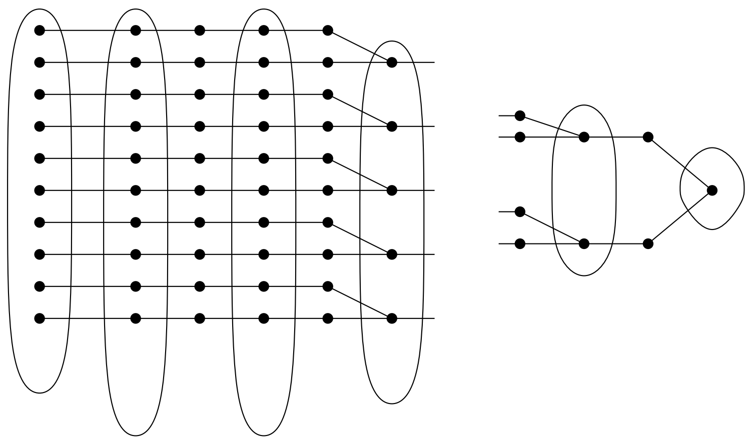

Figure 4.5: The construction of the graph G' (leaves are not included)

$L_i, i \leq k$ are mapped onto v_0 or $v_{\lambda+2}$. All vertices in layers with even subscripts are mapped onto the same vertex (and vice versa for the odd subscripts), since there cannot appear two vertices form two consecutive layers with the same image under f.

W.l.o.g., assume that all vertices in L_k map onto $v_{\lambda+2}$. Then no vertex from G can map either on $v_{\lambda+2}$ nor $v_{\lambda+1}$. Hence, G covers $\overline{P_{\lambda+3}} \setminus \{v_{\lambda+2}, v_{\lambda+1}\} = \overline{P_{\lambda+1}}$.

In the opposite direction, suppose, that G covers $\overline{P_{\lambda+1}}$. Extend the partial covering to the entire graph G' as follows: Vertices from layers L_i, where $i \equiv k \pmod 2$, map onto $v_{\lambda+2}$. Vertices from $L_i : i \not\equiv k \pmod 2$ map onto v_0. Starting from the root of the tree T select a feasible image for every vertex between layers L_{i-1} and L_i, $i = 1, \dots k$. Note, that the above selection can be done also for every vertex u between layers L_k and L_{k-1}, since its two neighbors are mapped onto v_0 and $v_{\lambda+2}$, and it must be mapped into set $\{v_2, \dots, v_\lambda\}$. Moreover, there are two vertices at distance two from u and $f(u)$ has to be different from their images. Since $\lambda \geq 4$, at least one feasible image for u remains, see Fig. 4.6 showing the forbidden labels.

Finally find a feasible image for all vertices of degree one adjacent to all layers L_i. $\square$

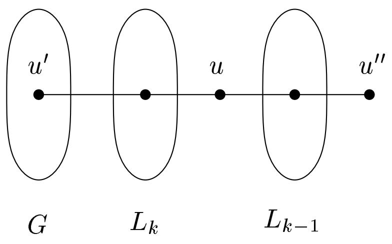

Figure 4.6: Forbidden images for u are $v_0, v_1, v_{\lambda+1}, v_{\lambda+2}, f(u'), f(u'')$

4.2.3 The $\lambda_{(2,1)}$-labeling of sparse graphs

We conclude the section by exposing a class of graphs, for which the existence of a $\lambda_{(2,1)}$-labeling can be tested in polynomial time. The base algorithm was designed for trees by Chang and Kuo [8], but we show its extension to the class of k-almost trees and prove that the $\lambda(2,1)$-labeling problem is solvable in linear time when the parameter k is fixed.

Definition *A* k-almost tree *is a connected graph on* n *vertices with* $n+k-1$ *edges.*

Theorem 4.15 *[18] The* $\lambda(2,1)$*-labeling problem is solvable in* $O(\lambda^{2k+9/2}n)$ *time for the class of* k*-almost trees on* n *vertices.*

Proof: Let G be a k-almost tree and consider its spanning tree. There are exactly k edges that are out of the spanning tree. We denote them by $e_1 = (u_1, u_1'), ..., e_k$.

There are at most $O(\lambda^{2k})$ $\lambda_{(2,1)}$-labelings of the vertices $\{u_i, u_i' : i = 1, ..., k\}$ and we show a polynomial-time algorithm that tests whether such a labeling f can be extended to the entire graph G.

For every edge e_i add into G two extra new vertices v_i and v_i', remove the edge e_i and replace it by the two edges (u_i, v_i) and (u_i', v_i'). This operation interrupts all cycles in G, hence the new graph T is a tree. For each $i = 1, ..., k$, set $f(v_i) = f(u_i')$ and $f(v_i') = f(u_i)$.

The labeling f can be extended to G, if and only if f allows an extension to T. The "only if" implication is obvious, for the other direction we note, that any path including the edge e_i, and connecting a neighbor x of u_i to a neighbor x' of u_i', has length three. Hence, no combination of labels $f(x)$ and $f(x')$ can violate the properties of the $\lambda_{(2,1)}$-labeling, when re-creating the edge e_i.

We show a modification of the algorithm of Chang and Kuo [8] that tests whether a tree allows a $\lambda_{(2,1)}$-labeling, and the tree contains pre-labeled leaves.

Suppose that T is rooted in the vertex v_1. Use dynamic programming, and for each edge $e = (x, y)$ of T, where x is parent of y, determine the set S_e of all $\lambda_{(2,1)}$-labelings of e, that contain the labeling f and can be extended to the subtree of T, consisting of x, y, and all descendants of y.

When e is a leaf, then the set S_e is explicitly defined, even if one or both vertices of e are already labeled by f.

Consider a edge $e = (x, y)$, and assume that sets $S_{(y,z)}$ are known for all descendants z of y. Then the set S_e includes a pair (a, b), if and only if $|a - b| \geq 2$, and for each z, we can select a unique representative from the set $\{c : (b, c) \in S_z, c \neq a\}$. The system of distinct representatives can be found in $O(\lambda^{5/2})$ time by the matching algorithm.

The tree T allows a $\lambda_{(2,1)}$-labeling, if and only if the pair $(f(v_1), f(u_1))$ appears in the set S_{v_1,u_1}. $\square$

Note that the $L(2,1)$-problem is polynomially solvable for trees, because due to Theorem 4.7, we have to test only the case $\lambda = \Delta(T) + 1$.

4.3 The general $\lambda(p, q)$-labeling problem

However, the direct relation between partial covers and the channel assignment problem with general parameters p and q disappears, when q is at least two, we can prove the *NP*-completeness of several $\lambda(p, q)$-labeling problems by the similar proof methods, like those we used for complexity characterization of (partial) covering problems.

We prove, that when parameters p and q are fixed, then at least one instance is *NP*-complete in the class of $\lambda(p, q)$-labeling problems.

In addition, under slightly stronger assumption $p > 2q$, the class of $\lambda(p, q)$-labeling problems has a finite number of polynomially solvable instances (here we assume $P \neq NP$, as well).

Theorem 4.16 *[18] For every fixed $p, q : p > q \geq 1$ the $\lambda(p, q)$-labeling problem is NP-complete for $\lambda = p + q\lceil \frac{p}{q} \rceil$.*

Proof: We show a reduction from the $BW(2, k)$-coloring problem for $k = \lceil \frac{p}{q} \rceil - 1$.

Let G be a $(k + 2)$-regular graph, whose feasible $BW(2, k)$-coloring is questioned. Replace each edge of G by a path of length three, and call the new graph G'. We show that there exist a $\lambda_{(p,q)}$-labeling of G', if and only if G allows a proper $BW(2,k)$-coloring.

Suppose, that a $\lambda_{(p,q)}$-labeling f of G' exists. The vertices of degree $k+2$ are labelled either by 0 or by λ, because when another label is used, there is no sufficient space to label all its $k+2$ neighbors.

Consider a vertex u_0 of degree $k+2$ in G', that is labeled by 0. Then its neighbors are labeled by $p, p+q, p+2q, ..., \lambda-q$ and λ, and no two neighbors have the same label. Consider a path of length three $P = (u_0, u_1, u_2, u_3)$. When u_1 is labeled by p or by λ, then the vertex u_3 (of degree $k+2$) is also labeled by 0 and the pattern $(0, p, \lambda, 0)$ is used on P. If u_1 is labeled by one of $p+q, ..., \lambda-q$ then the vertex u_3 is labelled by λ, because the label of u_2 can attain the number $\lambda - p - q < p$. Hence, among $k+2$ vertices at distance three from u_0, k of them are labelled by λ and the remaining two have assigned label 0. Due to the symmetry of any $\lambda_{(p,q)}$-labeling, the vice versa holds for vertices that are labeled by λ.

We define a $BW(2,k)$-coloring of G as follows: Color a vertex black, if the corresponding vertex in G' is labelled by 0, and color it white otherwise.

In the opposite direction, suppose that a $BW(2,k)$-coloring of G is given. The edges connecting white vertices induce a 2-regular subgraph, and on the corresponding paths in G' we use the pattern $(0, p, \lambda, 0)$ cyclically. By symmetry we use the sequence $(\lambda, 0, p, \lambda)$ between "black" vertices. The edges connecting the sets of white and black vertices form a bipartite k-factor of the graph G. Due to Theorem 1.5 these edges can be split into k disjoint 1-factors. Then in G', use pattern $(0, p+iq, iq, \lambda)$ on paths corresponding to edges from the i-th 1-factor of G. □

Theorem 4.17 *For each $p, q : p > 2q$, the $\lambda(p,q)$-labeling problem is NP-complete whenever $\lambda \geq 9pq + 2p + q + 1$.*

Proof: In order to prove the statement, we will reduce the K_4-cover problem to the $\lambda(p,q)$-labeling problem.

We first discuss properties of a special graph F, that will be used later in the construction.

Express λ as a linear combination $ap + bq$, where $a > 4q+2, b > 4p+1$. Coefficients a and b always exists, since the number $\lambda - 8p - 2p - q$ is greater to pq and can be expressed as a positive linear combination $(a-4q-2)p + (b-4p-1)q$. Denote by $c = (a-1)(\lfloor \frac{p}{q} \rfloor - 1) + b - 1$. Put the vertex set

$$V(F) = \{z = x_0^\circ, x_1^\circ, x_2^\circ, ..., x_b^\circ, x_1^\bullet, ..., x_a^\bullet, x_1^-, ..., x_c^-\}$$

. The edge set consists of three types of edges $E(F) = E^\circ \cup E^\bullet \cup E^-$:

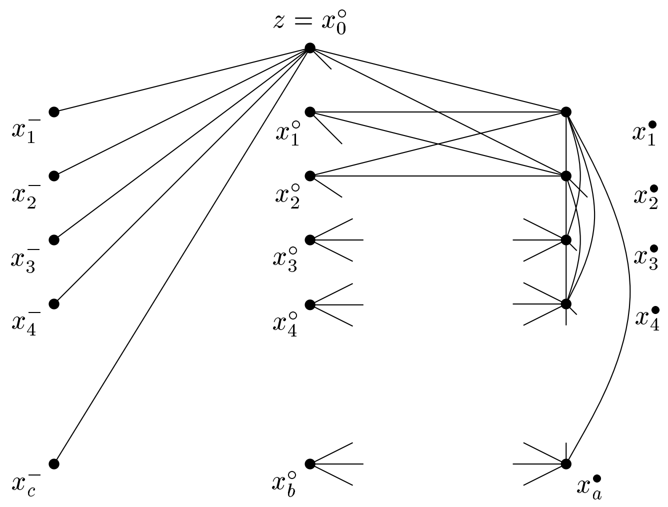

Figure 4.7: The construction of the auxiliary graph F

- $E^\circ = \{(x_i^\circ, x_j^\bullet) : 0 \le i \le b, 1 \le j \le a\}$ ... edges forming a complete bipartite graph between $\circ$ and $\bullet$ vertices.
- $E^\bullet = \{(x_i^\bullet, x_j^\bullet) : 1 \le i < j \le a\}$ ... a clique on $\bullet$ vertices.
- $E^- = \{(z, x_i^-) : 1 \le i \le c\}$ ... c leaves added to the vertex $z = x_0^\circ$.

Claim: *The vertex z will be labeled either by $0, q, \lambda - q$ or by λ, under any $\lambda_{(p,q)}$-labeling of the graph F.*

Consider a $\lambda_{(p,q)}$-labeling f of the graph F. Vertices $x_i^\bullet$ induce a clique, hence, their labels are pairwise at least p apart. Every pair of vertices x_j° and $x_{j'}^\circ$ has a common neighbor, so their labels differ by at least q. Moreover, the set E° forms a complete bipartite subgraph, so labels of any $x_i^\bullet$ differs from label of any x_j° by at least p. The only two possibilities of labeling these vertices are: Either use labels $\{0, q, ..., bq\}$ on $V^\circ = \{x_j^\circ, 0 \le j \le b\}$ and $\{p + bq, 2p + bq, ..., ap + bq\}$ on $V^\bullet = \{x_i^\bullet, 1 \le i \le a\}$, or reverse the labeling, s.t. $f(V^\circ) = \{ap + bq, ..., ap + q, ap\}$ and $f(V^\bullet) = \{(a-1)p, ..., p, 0\}$.

Without lost of generality suppose that z is labelled by iq, where $0 \le i \le b$. Observe that at most $\lfloor \frac{p}{q} - 1 \rfloor$ vertices of the set $V^- = \{x_j^-, 1 \le j \le c\}$ can use labels from the interval $[bq, bq + p]$. The same holds for any interval of form $[bq + kp, bq + (k+1)p]$ for any $k \le a - 1$. In total at most $(a-1)\lfloor \frac{p}{q} - 1 \rfloor$ vertices of V^- vertices can be labeled using labels greater than bq. In V^-, there remain at least $b - 1 = c - (a-1)\lfloor \frac{p}{q} - 1 \rfloor$ unlabelled vertices. Hence,

$f(z) \in \{0, q\}$, because any other label of u does not leave sufficient space in $[0, bq]$ for labels of the remaining $b-1$ vertices from V^-.

The case of $f(z) = \lambda$ or $\lambda - q$ follows due to symmetry of the labeling, i.e., the case $f(z) \in f(V^\circ) = \{ap + bq, ..., ap + q, ap\}$.

Now, we are ready for the reduction from the K_4-covering problem. Let G be a cubic graph whose covering to K_4 is questioned. Form the graph H as follows: for each vertex $u \in V(G)$ insert into H a disjoint copy of the graph F, and denote it by F_u. In each F_u rename the vertices $z, x_1^-, x_2^-, x_3^-, x_4^-$ by $z_u, u', u_v, u_{v'}, u_{v''}$ where v, v' and v'' are the three neighbors of the vertex u in G.

The last step in the construction of the graph H glues together the graphs F_u: For each $u \in V(G)$, unify vertices u', v_u, v'_u, v''_u, and call the new vertex s_u.

We show that any $\lambda(p, q)$-labeling f of H induces a covering projection $g : G \to K_4$. Let $V(K_4) = 0, q, \lambda - q, \lambda$, and put $g(u) = f(z_u)$.

If v, v' and v'' are neighbors of u in G, then the vertices z_u, z_v, $z_{v'}$ and $z_{v''}$ get distinct labels, because they share a common neighbor s_u. Immediately g is locally injective, i.e., a covering projection.

In the opposite direction, suppose that the graph G covers K_4 via g. In H label all z_u by a label from $0, q, \lambda - q, \lambda$, such that z_u and $z_{u'}$ gets the same label, if and only if $g(u) = g(u')$.

In each F_u, extend the labeling to vertex sets V° and $V^\bullet$, as claimed earlier.

In the next step we label the set $S = \{s_u, u \in V(G)\}$. Every s_u has at most nine neighbors at distance two in S, hence, ten labels, that differs by at lest q, are sufficient for a labeling of S. Use $p + q, p + (p + 1)q, p + (2p + 1)q, ..., p + (4p + 1)q, \lambda - (p + (4p + 1)q), ..., \lambda - (p + q)$ as these labels. Fig. 4.8 shows that there is no conflict with the other labels in H. (The first row exhibits labels of V° and $V^\bullet$ when z is labeled 0 or q. The last row describes labels of these sets, when z is labeled by λ or $\lambda - q$. The middle row shows possible labels of the the vertex z and the set S.)

Now, the only unlabelled vertices of H are of degree one. Extend the labeling f to these leaves, as was shown earlier when we discussed properties of a labeling of the graph F. □

4.3.1 Complexity of the $\lambda(p, 1)$-labeling problem

In contrary to the $\lambda(p, q)$-labeling problem with parameter q being at least two, the condition $q = 1$ transforms to a searching for a labeling, where vertices with a common neighbor gets distinct labels, but not *distant* labels.

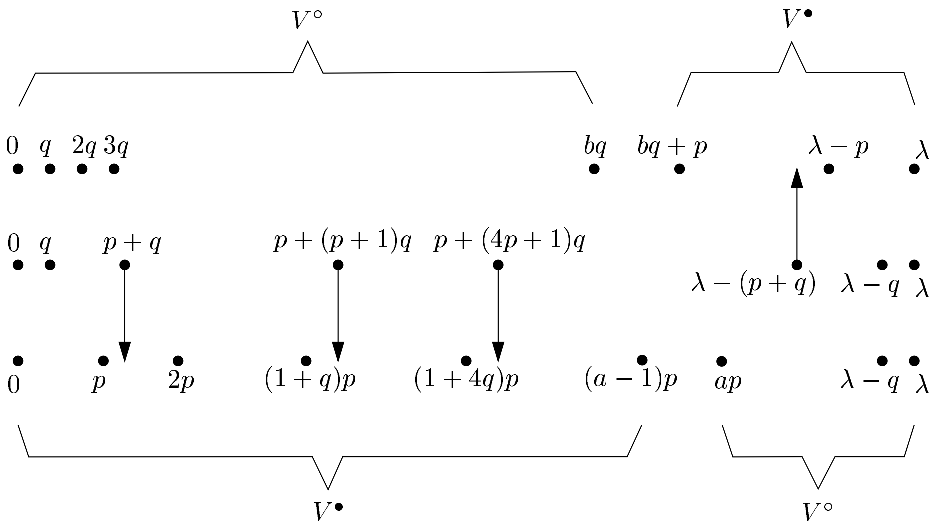

Figure 4.8: Ten labels are sufficient for the set S (comments in the text)

This allows us to state the equivalence between a $\lambda_{(p,1)}$-labeling of a graph G and a partial covering $G \to H_{\lambda,p}$, where the graph $H_{\lambda,p}$ is defined as follows: $V(H_{\lambda,p}) = \{v_0, ..., v_\lambda\}$, $E(H_{\lambda,p}) = \{(v_i, v_j) : |i-j| \geq p\}$. The complete characterization of the class of H-partial covering problems gives the complete characterization of the class of $L(p,1)$-labeling problems. As a immediate consequence of Corollary 2.12, we get the lower bound on the transition between polynomially solvable and *NP*-complete cases.

Corollary 4.18 *The $\lambda(p,1)$-labeling problem is polynomially solvable for every $\lambda \leq p+2, p \geq 3$.*

In [18], there is presented a reduction, showing that the $\lambda(p,1)$-labeling problem is *NP*-complete whenever $\lambda \geq p+5, p \geq 3$.

In addition the matching algorithm used in the proof of Theorem 4.15 works also for parameters $p \geq 2, q = 1$, however, it has been proven *NP*-complete when $q \geq 2$ [18].

Corollary 4.19 *For every fixed k, the $\lambda(p,1)$-labeling problem is solvable in polynomial time for the class of k-almost trees on n vertices.*

Chapter 5

Conclusion

Let us summarize what we have presented so far.

We started exploring properties of graph covering projections by exposing the degree refinement, i.e., a factorization of the vertex set, that restricts the image of a vertex under a possible covering projection. This structure was essential during the polynomial reduction from the H-partial covering problem to the H-covering problem.

The product of two graphs with respect to covers might be interesting for its relation to the theory of categories, as well as for its application in distributed computing. We showed that the structure of degree refinement allows an extension using the factorization of edges into perfect matchings, and that this extension achieves categorical properties of the product. On the other hand, we have mentioned results of Angluin, Gardiner and Leighton on the existence of a common cover. Both constructions might be interesting in the emulation concept as a minimal universal networks for a certain class of parallel algorithms.

We have reviewed the recent results on the computational complexity of the H-covering problem, and showed several instances that are polynomially solvable, as well as NP-complete instances.

In Theorem 2.19 we have extended the result of Kratochvíl, Proskurowski and Telle, proving that all k-regular graphs H of $k \geq 3$ are NP-complete instances for the H-cover problem.

The class of all H-covering problems is not fully characterized yet and, furthermore, no conjecture has been suggested for the boundary (if exists) between tractable instances (those which allow a polynomial-time algorithm) and difficult (NP-complete) cases.

A similar situation holds on the class of H-partial covering problems,

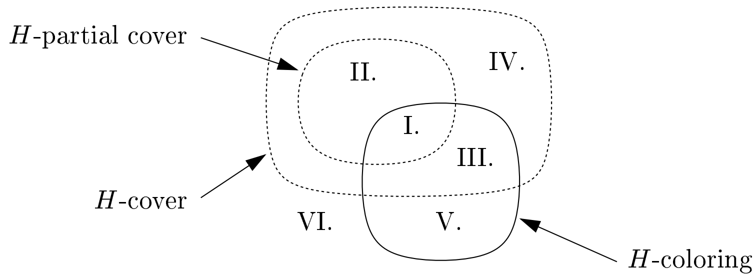

Figure 5.1: Diagram on polynomially solvable and *NP*-complete instances

even though there is a direct connection to the characterization of computational complexity of H-covering problems. Namely the class of *NP*-complete instances of the H-partial covering problem is a superset of *NP*-complete elements related to full covers.

We concentrated on graphs H, where it is known that the H-covering problem is polynomially solvable, and we investigated whether this is still valid when asking for a partial covering projection. Both *NP*-complete and polynomially solvable instances were found. It deserves interest to find out the extent to which matching and halfedge coloring methods are applicable, as well as the question of why bipartite graphs are more frequently tractable than the others (e.g., several bipartite banana graphs). The last question relate homomorphisms and locally injective homomorphisms on bipartite graphs, and would show, in which moment, the local injectivity causes the H-partial covering problem to be hard.

Both H-cover and H-partial cover problems belong to so-called constraint satisfaction problems ($CSP \subseteq NP$). Assuming $P \neq NP$, it is expected that the class CSP has a strict boundary separating *NP*-complete and polynomially solvable problems, i.e., each problem belonging to CSP is either polynomially solvable or *NP*-complete [41, 12].

We try to illustrate the relation of polynomially solvable and *NP*-complete instances of H-coloring, H-cover and H-partial cover problems in Fig. 5.1. The outer region corresponds to the *NP*-complete instances and the inner area contains graphs for which an algorithm running in polynomial time is known. The dotted boundary means that there exist instances where computational complexity is still undecided.

The following list of examples shows that no region is empty:

I. $W(1, 2, 4)$ or $B(a^k, b^l)$, a and b odd,

II. even cycles C_{2k+1},

III. $B(1,3,5)$,

IV. $B(a^k, b^j)$, $a \not\equiv b \pmod 2$,

V. $K_n^{:2}$, the cube graph $(K_2)^3$,

VI. complete graphs K_n.

In the last chapter, we have described a simple graph theoretic model for the channel assignment problem, and have showed that several cases can be reduced to partial covering projections. The same argument glues together the characterization of the computational complexity of several classes of the $\lambda(p_1, ..., p_k)$-labeling problem with corresponding classes of the H-partial covering problem, even on both sides, there are problems requiring their own approach. Similarly, no full complexity characterization is known yet for the $\lambda(p_1, ..., p_k)$-labeling problem, and even two parameters p, q expose a variety of non-trivial reductions.

We have concentrated on the $L(p, q)$-labeling problem of almost trees, and showed that there exists a fixed parameter tractable algorithm when the parameter q is at most one. As far as we know, the $L(p, q)$-labeling problem is open for higher values of q.

We hope that we presented several interesting aspects connecting algebraic nature of graph homomorphism with the combinatorial optimization methods that might find an application in the channel assignment industry.

Index

$Aut(G)$, 13
$H_{(2,1)}$-labeling, 66
$H(2,1)$-labeling problem, 67
$L_{(p_1,..,p_k)}$-labeling, 63
$\Delta(G)$, 9
$\alpha(G)$, 10
$\chi'(G)$, 13
$\chi(G)$, 13
$\lambda_{(p_1,..,p_k)}$-labeling, 63
$\lambda_{(p_1,..,p_k)}(G)$, 63
$\omega(G)$, 10
k-almost tree, 72
k-factor, 14
k-fold cover, 21

adjacency matrix, 9
adjacent vertices, 8
articulation, 12
automorphism, 13

banana graph, 11
bipartite graph, 13
block, 12
bridge, 12

Cartesian product, 7
center of star, 11
chromatic index, 13
chromatic number, 13
class NP, 18
class NPc, 19
class P, 17
clique, 10
clique number, 10
clique, maximum, 10
closed neighborhood, 9
coloring, 13
common cover, 26
complement, 8
complete bipartite graph, 10
complete graph, 10
component of connectivity, 11
connected graph, 11
contraction, 10
covering projection, 14
cubic graph, 9
cutset, 12
cutvertex, 12
cycle, 10
cyclic coloring, 13

degree, 9
degree refinement, 22
degree refinement matrix, 22
degree vector, 22
degree, in multigraph, 9
diameter, 12
digraph, 8
directed graph, 8
distance, 12
domain, 7

edge, 8
edge k-connectivity, 12
edge coloring, 13

edge cutset, 12
edge homomorphism, 12
endomorphism, 13

face, 11
factor, 14
factorable graphs, 14
filling graph, 38
flower graph, 11
function, 7

girth, 10
graph, 8
graph with loops, 8
graph, empty, 10
ground graph, 23

halfedge, 43
Hamiltonian cycle, 10
head, 8
homomorphism, 12

identity, 7, 13
image, 7
indegree, 9
independence number, 10
independent set, 10
induced minor, 10
injective mapping, 7
input instance, 16
inverse function, 7
isomorphism, 13

Kronecker double cover, 27

leaf, 9
Leighton's cover, 28
line graph, 9
local isomorphism, 14
loop, 8

mapping, 7
marked graph, 24
matching, 13
maximal subpath, 48
maximum degree, 9
minor, 10
minor closed class, 10
multicover, 36
multigraph, 8

neighborhood, 9

outdegree, 9
outerface, 11
outerplanar graph, 11

partial covering, 14
path, 10
path covering pattern, 55
path, length, 10
perfect graph, 13
perfect matching, 13
planar drawing, 11
planar graph, 11
polynomial reduction, 17
polynomially equivalent problems, 17
product of well marked graphs, 25
projective planar graph, 11

range, 7
ray, 11
regular graph, 9
removing of edge, 10
rigid graph, 13
running time, 16

simple graph, 8
solid graph, 36
star, 11
starfish graph, 11
subgraph, 9

subgraph, induced, 9
surjective mapping, 7

tail, 8
tour, 10
tree, 10

universal cover, 26

vertex, 8
vertex k-connectivity, 12
vertex coloring, 13

walk, 10
weight covering pattern, 59
weight graph, 11
well marked graph, 25
wheel, 65

Bibliography

[1] ABELLO, J., FELLOWS, M. R., AND STILLWELL, J. C. On the complexity and combinatorics of covering finite complexes. *Australian Journal of Combinatorics 4* (1991), 103–112.

[2] ANGLUIN, D. Local and global properties in networks of processors. *Proceedings of the 12th ACM Symposium on Theory of Computing* (1980), 82–93.

[3] ANGLUIN, D., AND GARDINER, A. Finite common coverings of pairs of regular graphs. *Journal of Combinatorial Theory B 30* (1981), 184–187.

[4] BIGGS, N. *Algebraic Graph Theory.* Cambridge University Press, 1974.

[5] BIGGS, N. Covering biplanes. In *The theory and applications of graphs, Fourth International Conference, Kalamazoo* (1981), John Wiley & Sons., pp. 73–79.

[6] BIGGS, N. Homological coverings of graphs. *Journal of London Mathematical Society II. 30* (1984), 1–14.

[7] BODLAENDER, H. L. The classification of coverings of processor networks. *Journal of Parallel Distributed Computing 6* (1989), 166–182.

[8] CHANG, G. J., AND KUO, D. The $L(2,1)$-labeling problem on graphs. *SIAM Journal of Discrete Mathematics 9* (1996), 309–316.

[9] CLARKE, F. W., THOMAS, A. D., AND WALLER, D. A. Embeddings of covering projections of graphs. *Journal of Combinatorial Theory B 28* (1980), 10–17.

[10] COURCELLE, B., AND MÉTIVIER, Y. Coverings and minors: Applications to local computations in graphs. *European Journal of Combinatorics 15* (1994), 127–138.

[11] Djoković, D. Ž. Automorphisms of graphs and coverings. *Journal of Combinatorial Theory B 16* (1974), 243–247.

[12] Feder, T., and Vardi, M. Y. The computational structure of momotone monadic SNP and constraint satisfaction: A sudy through datalog and group theory. *SIAM Journal of Computing 1* (1998), 57–104.

[13] Fiala, J. Cmputational complexity of covering cyclic graphs. accepted.

[14] Fiala, J. Note on the computational complexity of covering regular graphs. submitted.

[15] Fiala, J. The NP completeness of the edge precoloring extension problem on bipartite graphs. submitted.

[16] Fiala, J. Graph covers. Master's thesis, Charles University, 1997. in czech.

[17] Fiala, J., and Kratochvíl, J. Partial covers of graphs. submitted.

[18] Fiala, J., Kratochvíl, J., and Kloks, T. Fixed-parameter tractability of λ-colorings. In *Graph-Theoretical Concepts in Computer Science, 25th WG '99, Ascona* (1999), no. 1665 in Lecture Notes in Computer Science, Springer Verlag, pp. 350–363.

[19] Fishburn, P. C., and Roberts, F. S. Full color theorems for $L(2,1)$-colorings. manuscript.

[20] Gardiner, A. Antipodal covering graphs. *Journal of Combinatorial Theory B 16* (1974), 255–273.

[21] Garey, M. R., and Johnson, D. S. *Computers and Intractability.* W. H. Freeman and Co., New York, 1979.

[22] Georges, J. P., and Mauro, D. W. On the size of graphs labeled with a condition at distance two. *Journal of Graph Theory 22* (1996), 47–57.

[23] Georges, J. P., Mauro, D. W., and Whittlesey, M. A. Relating path coverings to vertex labellings with a condition at distance two. *Discrete Mathematics 135* (1994), 103–111.

[24] Giaro, K. The complexity of consecutive Δ-coloring in bipartite graphs: 4 is easy, 5 is hard. *Ars Combinatoria 47* (1997), 287–298.

[25] Griggs, J. R., and Yeh, R. K. Labelling graphs with a condition at distance 2. *SIAM Journal of Discrete Mathematics 5* (1992), 586–595.

[26] Gross, J. L., and Tucker, T. W. Generating all graph coverings by permutation voltage assignments. *Discrete Mathematics 18* (1977), 273–283.

[27] Gross, J. L., and Tucker, T. W. *Topological Graph Theory.* J. Wiley and Sons, 1987.

[28] Hell, P., and Nešetřil, J. On the complexity of H-colouring. *Journal of Combinatorial Theory B 48* (1990), 92–110.

[29] Hliněný, P. $K_{4,4} - e$ has no finite planar cover. *Journal of Graph Theory 21*, 1 (1998), 51–60.

[30] Hliněný, P., and Thomas, R. On possible counterexamples to negami's planar cover conjecture. manuscript, 1999.

[31] Hofmeister, M. Isomorphisms and automorphisms of graph coverings. *Discrete Mathematics 98* (1991), 175–183.

[32] Holyer, I. The NP-completeness of edge coloring. *SIAM Journal of Computing 10*, 4 (1981), 718–720.

[33] Jonas, K. *Graph colorings analogues with a condition at distance two: $L(2,1)$-labelings and list λ-labelings.* PhD thesis, University of South Carolina, 1993.

[34] Kratochvíl, J. *Perfect codes in general graphs.* Academia Praha, 1991.

[35] Kratochvíl, J. Regular codes in regular graphs are difficult. *Discrete Mathematics 133* (1994), 191–205.

[36] Kratochvíl, J., Proskurowski, A., and Telle, J. A. Covering directed multigraphs II. when 2-SAT helps. submitted.

[37] Kratochvíl, J., Proskurowski, A., and Telle, J. A. Covering directed multigraphs I. colored directed multigraphs. In *Graph-Theoretical Concepts in Computer Science, 23rd WG '97, Berlin* (1997), no. 1335 in Lecture Notes in Computer Science, Springer Verlag, pp. 242–257.

[38] Kratochvíl, J., Proskurowski, A., and Telle, J. A. Covering regular graphs. *Journal of Combinatorial Theory B 71* (1997), 1–16.

[39] Kratochvíl, J., Proskurowski, A., and Telle, J. A. Complexity of graph covering problems. *Nordic Journal of Computing 5* (1998), 173–195.

[40] Kratochvíl, J., and Sebö, A. Coloring precolored perfect graphs. *Journal of Graph Theory 25* (1997), 207–215.

[41] Ladner, R. E. On the structure of polynomial time reducibility. *Journal of the Association for Computing Machinery 22* (1975), 155–171.

[42] Leese, R. A. A linear programming approach to radio channel assignment problem in heavily loaded, evolving networks. NATO Symposium in Aalborg, Denmark, 1998.

[43] Leese, R. A. Radio spectrum: a raw material for the telecommunications industry. 10th Conference of the European Consortium for Mathematics in Industry, Gteborg, 1998.

[44] Leighton, F. T. Finite common coverings of graphs. *Journal of Combinatorial Theory B 33* (1982), 231–238.

[45] Litovsky, I., Métivier, Y., and Zielonka, W. The power and the limitations of local computations on graphs and networks. In *Graph-Theoretical Concepts in Computer Science, 18th WG '92, Berlin* (1993), no. 657 in Lecture Notes in Computer Science, Springer Verlag, pp. 333–345.

[46] Liu, D. D.-F., and Yeh, R. K. On distance two labellings of graphs. *Ars Combinatoria 47* (1997), 13–22.

[47] Lovász, L. *Combinatorial Problems and Exercises.* Akadémiai Kiadó, Budapest, 1979.

[48] Lovász, L., and Plummer, M. D. *Matching Theory.* Akadémiai Kiadó, Budapest, 1986.

[49] Massey, W. *Algebraic Topology: An Introduction.* Harcourt, Brace and World, 1967.

[50] Matoušek, J., and Nešetřil, J. *Invitation to Discrete Mathematics.* Oxford University Press, 1998.

[51] Nedela, R., and Škoviera, M. Cayley snarks and almost simple groups. manuscript, 2000.

[52] Negami, S. The spherical genus and virtually planar graphs. *Discrete Mathematics 70*, 2 (1988), 159–168.

[53] Nešetřil, J. Homomorphisms of derivative graphs. *Discrete Mathematics 1*, 3 (1971), 257–268.

[54] Reyner, S. W. An analysis of a good algorithm for the subtree problem. *SIAM Journal of Computing 6* (1977), 730–732.

[55] Sakai, D. Labeling chordal graphs: distance two condition. *SIAM Journal of Discrete Mathematics 7* (1994), 133–140.

[56] van den Heuvel, J., Leese, R. A., and Shepherd, M. A. Graph labeling and radio channel assignment. *Journal of Graph Theory 29*, 4 (1998), 263–283.

[57] Whittlesey, M. A., Georgess, J. P., and Mauro, D. W. On the lambda-coloring of Q_n and related graphs. *SIAM Journal of Discrete Mathematics 8* (1995), 499–506.

[58] Yeh, K.-C. *Labeling graphs with a condition at distance two*. PhD thesis, University of South Carolina, 1990.