Computational Complexity of Covering Multigraphs with Semi-Edges: Small Cases

Jan Bok
Computer Science Institute, Faculty of Mathematics and Physics,
Charles University, Prague, Czech Republic

Jiří Fiala
Department of Applied Mathematics, Faculty of Mathematics and Physics,
Charles University, Prague, Czech Republic

Petr Hliněný
Faculty of Informatics, Masaryk University, Brno, Czech Republic

Nikola Jedličková
Department of Applied Mathematics, Faculty of Mathematics and Physics,
Charles University, Prague, Czech Republic

Jan Kratochvíl
Department of Applied Mathematics, Faculty of Mathematics and Physics,
Charles University, Prague, Czech Republic

Abstract

We initiate the study of computational complexity of graph coverings, aka locally bijective graph homomorphisms, for *graphs with semi-edges*. The notion of graph covering is a discretization of coverings between surfaces or topological spaces, a notion well known and deeply studied in classical topology. Graph covers have found applications in discrete mathematics for constructing highly symmetric graphs, and in computer science in the theory of local computations. In 1991, Abello et al. asked for a classification of the computational complexity of deciding if an input graph covers a fixed target graph, in the ordinary setting (of graphs with only edges). Although many general results are known, the full classification is still open. In spite of that, we propose to study the more general case of covering graphs composed of normal edges (including multiedges and loops) and so-called semi-edges. Semi-edges are becoming increasingly popular in modern topological graph theory, as well as in mathematical physics. They also naturally occur in the local computation setting, since they are lifted to matchings in the covering graph. We show that the presence of semi-edges makes the covering problem considerably harder; e.g., it is no longer sufficient to specify the vertex mapping induced by the covering, but one necessarily has to deal with the edge mapping as well. We show some solvable cases and, in particular, completely characterize the complexity of the already very nontrivial problem of covering one- and two-vertex (multi)graphs with semi-edges. Our NP-hardness results are proven for simple input graphs, and in the case of regular two-vertex target graphs, even for bipartite ones. We remark that our new characterization results also strengthen previously known results for covering graphs without semi-edges, and they in turn apply to an infinite class of simple target graphs with at most two vertices of degree more than two. Some of the results are moreover proven in a more general setting (e.g., finding k-tuples of pairwise disjoint perfect matchings in regular graphs, or finding equitable partitions of regular bipartite graphs).

2012 ACM Subject Classification Mathematics of computing → Graph theory

Keywords and phrases graph cover, covering projection, semi-edges, multigraphs, complexity

Digital Object Identifier 10.4230/LIPIcs.MFCS.2021.21

Related Version The full proofs of *-marked statements can be found in the extended version of this paper.
Extended Version: `https://arxiv.org/abs/2103.15214`

46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021).
Editors: Filippo Bonchi and Simon J. Puglisi; Article No. 21; pp. 21:1–21:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Funding *Jan Bok*: Supported by research grant GAČR 20-15576S of the Czech Science Foundation and by SVV–2020–260578. The author was also partially supported by GAUK 1580119.
Jiří Fiala: Supported by research grant GAČR 20-15576S of the Czech Science Foundation.
Petr Hliněný: Supported by research grant GAČR 20-04567S of the Czech Science Foundation.
Nikola Jedličková: Supported by research grant GAČR 20-15576S of the Czech Science Foundation and by SVV–2020–260578.
Jan Kratochvíl: Supported by research grant GAČR 20-15576S of the Czech Science Foundation.

1 Introduction

1.1 Graph coverings and complexity

The notion of a *graph covering* is a discretization of coverings between surfaces or topological spaces, a notion well known and deeply studied in classical topology. Graph coverings have found many applications. Primarily as a tool for construction of highly symmetric graphs [5, 15, 24, 27], or for embedding complete graphs in surfaces of higher genus [48].

Graph coverings attracted attention of computer scientists as well. Angluin [2] exploited graph covers when introducing models of local computations, namely by showing that a graph and its cover cannot be distinguished by local computations. Later, Litovsky et al. [39] proved that planar graphs and series-parallel graphs cannot be recognized by local computations, and Courcelle and Metivier [14] showed that in fact no nontrivial minor-closed class of graphs can. In both of these results, graph coverings were used as the main tool, as well as in more recent papers of Chalopin et al. [8, 9]. Here, the authors presented a model for distributed computations and addressed the algorithmic complexity of problems associated with such a model. To this end, they used the existing results on NP-completeness of the covering problem to provide their hardness results. In [10], the authors study a close relation of packing bipartite graphs to a special variant of graph coverings called *pseudo-coverings*.

Another connection to algorithmic theory comes through the notions of the *degree partition* and the *degree refinement matrix* of a graph. These notions were introduced by Corneill [12, 13] in hope of solving the graph isomorphism problem efficiently. It can be easily seen that a graph and all of its covers have the same degree refinement matrix. Motivated by this observation, Angluin and Gardiner [3] proved that any two finite regular graphs of the same valency have a finite common cover, and conjectured the same for every two finite graphs with the same degree refinement matrix, which was proved by Leighton [37].

The stress on finiteness of the common cover is natural. For every matrix, there exists a universal cover, an infinite tree, that covers all graphs with this degree refinement matrix. Trees are planar graphs, and this inspired an at first sight innocent question of which graphs allow a finite planar cover. Negami observed that projective planar graphs do (in fact, their double planar covers characterize their projective embedding), and conjectured that these two classes actually coincide [46]. Despite a serious effort of numerous authors, the problem is still open, although the scope for possible failure of Negami's conjecture has been significantly reduced [4, 28, 29].

A natural computational complexity question is how difficult is to decide, given two graphs, if one covers the other one. This question is obviously at least as difficult as the graph isomorphism problem (consider two given graphs on the same number of vertices). It was proven NP-complete by Bodlaender[7] (in the case of both graphs being part of the input). Abello et al. [1] initiated the study of the computational complexity of the H-cover problem for a fixed target graph H by showing that deciding if an input graph covers the dumbbell graph $W(0,1,1,1,0)$ (in our notation from Section 4) is NP-complete (note that the dumbbell graph has loops, and they also allowed the input graph to contain loops). Furthermore, they asked for a complete characterization of the computational complexity, depending on the parameter graphs H. Such a line of research was picked by Kratochvíl, Proskurowski and Telle, who completely characterized the complexity for simple target graphs with at most 6 vertices [33], and then noted that in order to fully characterize the complexity of the H-cover problem for simple target graphs, it is sufficient (but also necessary) to classify it for mixed colored multigraphs with minimum degree at least three [30]. The latter result gives a hope for a more concise description of the characterization, but is also in line with the original motivation from topological graph theory, where loops and multiedges are widely considered.

The complexity of covering 2-vertex multigraphs was fully characterized in [30], the characterization for 3-vertex undirected multigraphs can be found in [34]. The most general NP-hardness result known so far is the hardness of covering simple regular graphs of valency at least three [32, 17]. More recently, Bílka et al. [6] proved that covering several concrete small graphs (including the complete graphs K_4, K_5 and K_6) remains NP-hard for planar inputs. This shows that planarity does not help in graph covering problems in general, yet the conjecture that the H-Cover problem restricted to planar inputs is at least as difficult as for general inputs, provided H itself has a finite planar cover, remains still open. Planar graphs have also been considered by Fiala et al. [19] who showed that for planar input graphs, H-RegularCover is in FPT when parameterized by H. This is in fact the first and only paper on the complexity of regular covers, i.e., covering projections determined by a regular action of a group of automorphisms on the covering graph.

Graph coverings were also extensively studied under a unifying umbrella of *locally constrained homomorphisms*. In these relaxations, homomorphisms can be either locally injective or locally surjective and not necessarily locally bijective. The computational complexity of locally surjective homomorphisms has been classified completely, with respect to the fixed target graph [22]. Though the complete classification of the complexity of locally injective homomorphisms is still out of sight, it has been proved for its list variant [16]. The problem is also interesting for its applied motivation – a locally injective homomorphism into the complement of a path of length k corresponds to an $L(2,1)$-labeling of span k, an intensively studied notion stemming from the theory of frequency assignment. Further generalizations include the notion of $H(p,q)$-coloring, a homomorphism into a fixed target graph H with additional rules on the neighborhoods of the vertices [18, 35]. To find more about locally injective homomorphisms, see e.g. [41, 11] or [21]. For every fixed graph H, the existence of a locally injective homomorphism to H is provably at least as hard as the H-cover problem. In this sense our hardness results extend the state of the art also for the problem of existence of locally injective homomorphisms.

1.2 Graphs with semi-edges

The notion of *semi-edges* has been introduced in the modern topological graph theory and it is becoming more and more frequently used (the terminology has not yet stabilized; semi-edges are often called half-edges, and sometimes fins). Mednykh and Nedela recently wrote a monograph [44] in which they summarize and survey the ambitions and efforts behind generalizing the notion of graph coverings to the graphs with semi-edges. This generalization, as the authors pinpoint, is not artificial as such graphs emerge "in the situation of taking quotients of simple graphs by groups of automorphisms which are semiregular on vertices and darts (arcs) and which may fix edges". As the authors put it: "A problem arises when one wants to consider quotients of such graphs (graphs embedded to surfaces) by an involution fixing an edge e but transposing the two incident vertices. The edge e is halved and mapped to a semiedge – an edge with one free end.” This direction of research proved to be very fruitful and provided many applications and generalizations to various parts of algebraic graph theory. For example, Malnič et al. [42] considered semi-edges during their study of abelian covers and as they write “...in order to have a broader range of applications we allow graphs to have semiedges.” To highlight a few other contributions, the reader is invited to consult [45, 43], the surveys [36] and (aforementioned) [44], and finally for more recent results the series of papers [19, 23, 20]. It is also worth noting that the concept of graphs with semi-edges was introduced independently and naturally in mathematical physics by Getzler and Karpanov [26].

In the view of the theory of local computations, semi-edges and their covers prove very natural, too, and it is even surprising that they have not been considered before in the context. If a computer network is constructed as a cover of a small template, the preimages of normal edges in the covering projection are matchings completely connecting nodes of two types (the end-vertices of the covered edge). Preimages of loops are disjoint cycles with nodes of the same type. And preimages of semi-edges are matchings on vertices of the same type. The role of semi-edges was spotted by Woodhouse et. al. [51, 49] who have generalized the fundamental theorem of Leighton on finite common covers of graphs with the same degree refinement matrix to graphs with semi-edges.

Our goal is to initiate the study of the computational complexity of covering graphs with semi-edges, and the current paper is opening the door in this direction.

1.3 Formal definitions

In this subsection we formally define what we call *graphs*. A graph has a set of vertices and a set of edges (also referred to as links). As it is standard in topological graph theory, we automatically allow multiple edges and loops. Every ordinary edge is connecting two vertices, every loop is incident with only one vertex. On top of these, we also allow *semi-edges*. Each semi-edge is also incident with only one vertex. The difference between loops and semi-edges is that a loop contributes two to the degree of its vertex, while a semi-edge only one. A very elegant description of ordinary edges, loops and semi-edges through the concept of *darts* is used in more algebraic-based papers on covers. The following formal definition is a reformulation of the one given in [44].

▶ **Definition 1.** *A* graph *is a triple* (D, V, Λ)*, where* D *is a set of* darts*, and* V *and* Λ *are each a partition of* D *into disjoint sets. Moreover, all sets in* Λ *have size one or two.*

With this definition, the *vertices* of a graph (D, V, Λ) are the sets of V (note that empty sets correspond to isolated vertices, and since we are interested in covers of connected graphs by connected ones, we assume that all sets of V are nonempty). The sets of Λ are referred to as *links*, and they are of three types – loops (2-element sets with both darts from the same set of V), (ordinary) edges (2-element sets intersecting two different sets of V), and semi-edges (1-element sets). After this explanation it should be clear that this definition is equivalent to a definition of multigraphs which is standard in the graph theory community:

▶ **Definition 2.** *A* graph *is an ordered triple* (V, Λ, ι)*, for* $\Lambda = E \cup L \cup S$*, where* ι *is the* incidence mapping $\iota : \Lambda \longrightarrow V \cup \binom{V}{2}$ *such that* $\iota(e) \in V$ *for all* $e \in L \cup S$ *and* $\iota(e) \in \binom{V}{2}$ *for all* $s \in E$*.*

For a comparison of Definitions 1 and 2, see Figure 1. Since we consider multiple edges of the same type incident with the same vertex (or with the same pair of vertices), the edges are given by their names and the incidence mapping ι expresses which vertex (or vertices) “belong” to a particular edge. The degree of a vertex is then defined as follows.

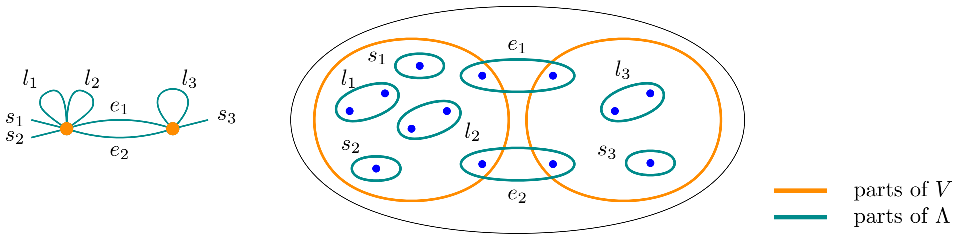

Figure 1 An example of a graph presented in a usual graph-theoretical way (left) and using the dart-based Definition 1 (right).

▶ **Definition 3.** *For a graph $G = (V, \Lambda = E \cup L \cup S, \iota)$, the degree of a vertex $u \in V$ is defined as*

$$deg_G(u) = p_S(u) + p_E(u) + 2p_L(u),$$

where $p_S(u)$ ($p_L(u)$) is the number of semi-edges $e \in S$ (of loops $e \in L$) such that $\iota(e) = u$, and $p_E(u)$ is the number of ordinary edges $e \in E$ such that $u \in \iota(e)$.

We call a graph G *simple* if $p_S(u) = p_L(u) = 0$ for every vertex $u \in V(G)$ (the graph has no loops or semi-edges) and $\iota(e) \neq \iota(e')$ for every two distinct $e, e' \in E$ (the graph has no multiple (ordinary) edges). We call G *semi-simple* if $p_S(u) \leq 1$ and $p_L(u) = 0$ for every vertex $u \in V(G)$ and $\iota(e) \neq \iota(e')$ for every two distinct $e, e' \in E$.

Note that in the language of Definition 1, the degree of a vertex $v \in V$ is simply $|v|$. And in this language, the main object of our study, a *graph cover* (or equivalently a *covering projection*), is defined as follows.

▶ **Definition 4.** *We say that a graph $G = (D_G, V_G, \Lambda_G)$* covers *a connected graph $H = (D_H, V_H, \Lambda_H)$ (denoted as $G \longrightarrow H$) if there exists a map $f : D_G \to D_H$ such that:*

- *For every $u \in V_G$, there is a $u' \in V_H$ such that the restriction of f onto u is a bijection between u and u'.*
- *For every $e \in \Lambda_G$, there is an $e' \in \Lambda_H$ such that $f(e) = e'$.*

The map f is called graph cover *(or* covering projection*).*

One must appreciate how compact and elegant this definition is after translating it into the language of Definition 2 in Proposition 5, which otherwise is the definition of (multi)graph covering in the standard language of Definition 2.

▶ **Proposition 5.** *A graph G covers a graph H if and only if G allows a pair of mappings $f_V : V(G) \longrightarrow V(H)$ and $f_\Lambda : \Lambda(G) \longrightarrow \Lambda(H)$ such that*

1. *$f_\Lambda(e) \in L(H)$ for every $e \in L(G)$ and $f_\Lambda(e) \in S(H)$ for every $e \in S(G)$,*
2. *$\iota(f_\Lambda(e)) = f_V(\iota(e))$ for every $e \in L(G) \cup S(G)$,*
3. *for every link $e \in \Lambda(G)$ such that $f_\Lambda(e) \in S(H) \cup L(H)$ and $\iota(e) = \{u, v\}$, we have $\iota(f_\Lambda(e)) = f_V(u) = f_V(v)$,*
4. *for every link $e \in \Lambda(G)$ such that $f_\Lambda(e) \in E(H)$ and $\iota(e) = \{u, v\}$ (note that it must be $f_V(u) \neq f_V(v)$), we have $\iota(f_\Lambda(e)) = \{f_V(u), f_V(v)\}$,*
5. *for every loop $e \in L(H)$, $f^{-1}(e)$ is a disjoint union of loops and cycles spanning all vertices $u \in V(G)$ such that $f_V(u) = \iota(e)$,*
6. *for every semi-edge $e \in S(H)$, $f^{-1}(e)$ is a disjoint union of edges and semi-edges spanning all vertices $u \in V(G)$ such that $f_V(u) = \iota(e)$, and*
7. *for every edge $e \in E(H)$, $f^{-1}(e)$ is a disjoint union of edges (i.e., a matching) spanning all vertices $u \in V(G)$ such that $f_V(u) \in \iota(e)$.*

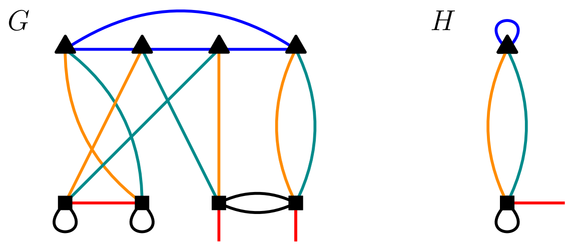

Figure 2 An example of a covering. The vertex mapping of the covering from G to H is determined by the shape of the vertices, the edge mapping by the colors of the edges.

See an example of a covering projection in Fig. 2. Conditions 1–4. express the fact that f_V and f_E commute with ι, i.e., that f is a homomorphism from G to H. Conditions 5–7 express that this homomorphism is locally bijective (for every ordinary edge e incident with $f_V(u)$ in H, there is exactly one ordinary edge of G which is incident with u and mapped to e by f_E; for every semi-edge e incident to $f_V(u)$ in H, there is exactly one semi-edge, or exactly one ordinary edge (but not both) in G incident with u and mapped to e by f_E; and for every loop e incident with $f_V(u)$ in H, there is exactly one loop or exactly two ordinary edges (but not both) of G which are incident with u and mapped to e by f_E).

Even though the aforementioned definitions of graphs and graph covers through darts are compact and elegant, in the rest of the paper we shall work with the standard definition of graphs and the equivalent description of graph covers given by Proposition 5, because they are better suited for describing the reductions and understanding the illustrative figures.

It is clear that a covering projection (more precisely, its vertex mapping) preserves degrees. One may ask when (or if) a degree preserving vertex mapping can be extended to a covering projection. An obvious necessary condition is described by the following definition.

▶ **Definition 6.** *A vertex mapping $f_V : V(G) \longrightarrow V(H)$ between graphs G and H is called* degree-obedient *if*

1. *for any two distinct vertices $u, v \in V(H)$ and any vertex $x \in f_V^{-1}(u)$, the number of ordinary edges e of H such that $\iota(e) = \{u, v\}$ equals the number of ordinary edges of G with one end-vertex x and the other one in $f_V^{-1}(v)$, and*
2. *for every vertex $u \in V(H)$ and any vertex $x \in f_V^{-1}(u)$, the value $p_{S(H)}(u) + 2p_{L(H)}(u)$ equals $p_{S(G)}(x) + 2p_{L(G)}(x) + r$, where r is the number of edges of G with one end-vertex x and the other one from $f_V^{-1}(u) \setminus \{x\}$,*
3. *for every vertex $u \in V(H)$ and any vertex $x \in f_V^{-1}(u)$, $p_{S(G)}(x) \leq p_{S(H)}(u)$.*

Finally, let us recall that the *product* $G \times H$ of graphs G and H is defined as the graph with the vertex set being the Cartesian product $V(G) \times V(H)$ and with vertices (u, v) and (u', v') being adjacent in $G \times H$ if and only if u is adjacent to u', and v is adjacent to v'.

1.4 Overview of our results

The first major difference between graphs with and without semi-edges is that for target graphs without semi-edges, every degree-obedient vertex mapping to it can be extended to a covering. This is not true anymore when semi-edges are allowed (consider a one-vertex graph with three semi-edges, every 3-regular graph allows a degree-obedient mapping onto it, but only the 3-edge-colorable ones are covering it). In Section 2 we show that the situation is not as bad if the source graph is bipartite. In Theorem 10 we prove that if the source graph is bipartite and has no semi-edges, then every degree-obedient vertex mapping can be extended to a covering, while if semi-edges are allowed in the bipartite source graph, it can at least be decided in polynomial time if a degree-obedient mapping can be extended to a covering.

All other results concern the complexity of the following decision problem

Problem:	H-Cover
Input:	A graph G.
Question:	Does G cover H?

In order to present our results in the strongest possible form, we aim at proving the hardness results for restricted classes of input graphs, while the polynomial ones for the most general inputs. In particular, we only allow simple graphs as inputs when we prove NP-hardness, and on the other hand, we allow loops, multiple edges as well as semi-edges when we present polynomial-time algorithms.

The first NP-hardness result is proven in Theorem 11, namely that covering semi-simple regular graphs of valency at least 3 is NP-hard even for simple bipartite input graphs. In Sections 3 and 4 we give a complete classification of the computational complexity of covering graphs with one and two vertices. This extends the main result of [30] to graphs with semi-edges. Moreover, we strengthen the hardness results of [30] considerably by showing that all NP-hard cases of covering regular two-vertex graphs (even those without semi-edges) remain NP-hard for simple *bipartite* input graphs. It must be noted that through the reduction from [31], our results on the complexity of covering one- or two-vertex graphs provide characterization results on infinitely many simple graphs which contain at most two vertices of degrees greater than 2.

All considered computational problems are clearly in the class NP, and thus we only concentrate on the NP-hardness proofs in the NP-completeness results. We restrict our attention to connected target graphs, in which case it suffices to consider only connected input graphs. In this case every cover is a k-fold cover for some k, which means that the preimage of every vertex has the same size.

2 The impact of semi-edges

In this section we demonstrate the huge difference between covering graphs with and without semi-edges. First, we discuss the necessity of specifying the edge mapping in a covering projection. In other words, we discuss when a degree mapping can always be extended to a covering, and when this question can be decided efficiently. The following proposition follows straightforwardly from the definitions.

▶ **Proposition 7.** *For every graph covering projection between two graphs, the vertex mapping induced by this projection is degree-obedient.*

▶ **Proposition * 8.** *If H has no semi-edges, then for any graph G, any degree-obedient mapping from the vertex set of G onto the vertex set of H can be extended to a graph covering projection of G to H.*

Proof sketch. For simple graphs G, this is proved already in [30]. If multiple edges and loops are allowed, we use a similar approach. The key point is that Petersen theorem [47] about 2-factorization of regular graphs of even valence is true for multigraphs without semi-edges as well, and the same holds true for König-Hall theorem [40] on 1-factorization of regular bipartite multigraphs. ◀

As we will see soon, the presence of semi-edges changes the situation a lot. Even for simple graphs, degree-obedient vertex mappings to a graph with semi-edges may not extend to a graph covering projection, and the possibility of such an extension may even be NP-complete.

▶ **Observation 9.** *Let $F(3,0)$ be the graph with one vertex and three semi-edges pending on this vertex. Then a graph covers $F(3,0)$ if and only if it is 3-regular and 3-edge-colorable. Testing 3-edge-colorability is well known to be NP-hard even for simple graphs.*

However, if the input graph is bipartite, the situation gets much easier.

▶ **Theorem * 10.** *If a graph G is bipartite, then for any graph H, it can be decided in polynomial time whether a degree-obedient mapping from the vertex set of G onto the vertex set of H can be extended to a graph covering projection of G to H. In particular, if G has no semi-edges and is bipartite, then every degree-obedient mapping from the vertex set of G onto the vertex set of H can be extended to a graph covering projection of G to H.*

Proof sketch. To prove this statement, it is enough to analyze the edges of H and their preimages in G according to the following classification:

- For each vertex pair $x \neq y \in V(H)$ inducing $k \geq 0$ parallel edges in H, their preimage forms a k-regular subgraph $G_{x,y}$ of bipartite G, and hence $G_{x,y}$ is k-edge colorable which immediately gives a covering projection for these edges.
- For each vertex $x \in V(H)$ with $b \geq 0$ semi-edges and $c \geq 0$ loops incident to x in H, these semi-edges and loops lift to a $(b+2c)$-regular subgraph $\widetilde{G_x}$ of G. The algorithmic task now is to decide whether $\widetilde{G_x}$ admits a factor projecting onto the semi-edges incident to x (this is efficiently solvable, e.g., by network flows since $\widetilde{G_x}$ is again bipartite). If the answer is true, a projection of the remaining edges onto the loops incident to x always exists by Petersen theorem. ◀

Now we prove the first general hardness result, namely that covering semi-simple regular graphs is always NP-complete (this is the case when every vertex of the target graph is incident with at most one semi-edge, and the graph has no multiple edges nor loops). See Fig. 3 for examples of semi-simple graphs H defining such hard cases.

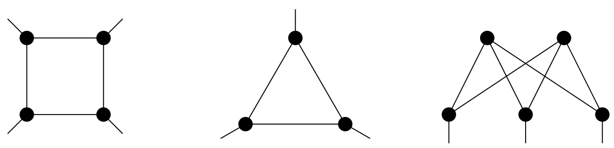

Figure 3 Examples of small semi-simple graphs which define NP-complete covering problems.

▶ **Theorem 11.** *Let H be a semi-simple k-regular graph, with $k \geq 3$. Then the H-*Cover *problem is NP-complete even for simple bipartite input graphs.*

Proof. Consider $H' = H \times K_2$. This graph is simple, k-regular and bipartite, hence the H'-Cover problem is NP-complete by [32]. Given an input k-regular graph G, it is easy to see that G covers H' if and only it is bipartite and covers H. Since bipartiteness can be checked in polynomial time, the claim follows. ◀

3 One-vertex target graphs

We start the section by proving a slightly more general hardness result, which may be of interest on its own. In particular, it implies that for every $d \geq 3$, it is NP-complete to decide if a simple d-regular graph contains an even 2-factor, i.e., a spanning 2-regular subgraph whose every cycle has even length.

▶ **Theorem * 12.** *For every $k \geq 2$ and every $d \geq k+1$, it is NP-complete to decide if a simple d-regular graph contains k pairwise disjoint perfect matchings.*

Proof sketch. The complement of the union of k pairwise disjoint perfect matchings in a $(k+1)$-regular graph is a perfect matching as well, and thus a $(k+1)$-regular graph contains k pairwise disjoint perfect matchings if and only if it is $(k+1)$-edge colorable. Hence for $d = k+1$, the claim follows from the NP-completeness of d-edge colorability of d-regular graphs which has been proven by Leven and Galil [38].

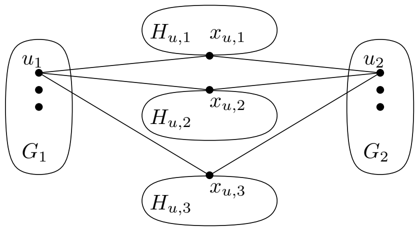

Figure 4 An illustration to the construction of the graph G' in the proof of Theorem 12.

For $d \geq k+2$, we reduce from the previous case, as we sketch next. If G is a $(k+1)$-regular instance (of the k disjoint perfect matchings problem), we construct an equivalent d-regular instance G' starting from two copies G_1 and G_2 of G, as shown in Fig. 4. Then for each vertex u of G, we connect its two copies u_1, u_2 (in G_1, G_2) by $d-k-1$ paths of length 2, and add copies of a suitable gadget H to the middle vertices of those paths. The purpose of this gadget is two-fold – it raises all degrees to d, and it prevents edges of the incident path from being used in a perfect matching since the gadget is of an even order. It is routine to finish the construction and to show that G contains k disjoint perfect matchings if and only if G' does so. ◀

Now we are ready to prove a dichotomy theorem on the complexity of covering one-vertex graphs. Let us denote by $F(b, c)$ the one-vertex graph with b semi-edges and c loops.

▶ **Theorem * 13.** *The $F(b, c)$-Cover problem is polynomial-time solvable if $b \leq 1$, or $b = 2$ and $c = 0$, and it is NP-complete otherwise, even for simple graphs.*

Proof sketch. The high-level idea is similar to the second point of the proof of Theorem 10: The input graph G possibly covering $F(b, c)$ should better be $(b+2c)$-regular (which can be easily checked), and it remains to argue that such G covers $F(b, c)$ if and only if it contains b pairwise disjoint perfect matchings (then a covering projection onto the c loops follows easily). The cases of $b = 0$, $b = 1$, or $b = 2$ and $c = 0$ can be efficiently solved using standard tools, while the remaining cases are hard from Theorem 12 by setting $k = b$ and $d = b + 2c$. ◀

4 Two-vertex target graphs

Let $W(k, m, \ell, p, q)$ be the two-vertex graph with k semi-edges and m loops at one vertex, p loops and q semi-edges at the other one, and $\ell > 0$ multiple edges connecting the two vertices (these edges are referred to as *bars*). In other words, $W(k, m, \ell, p, q)$ is obtained from the disjoint union of $F(k, m)$ and $F(q, p)$ by connecting their vertices by ℓ parallel edges. For an example see the graph H from Fig. 2 which is isomorphic to both $W(1, 1, 2, 1, 0)$ and $W(0, 1, 2, 1, 1)$.

▶ **Theorem 14.** *The $W(k,m,\ell,p,q)$-*Cover *problem is solvable in polynomial time in the following cases*

1. $k+2m \neq 2p+q$ *and* ($k \leq 1$ *or* $k=2$ *and* $m=0$) *and* ($q \leq 1$ *or* $q=2$ *and* $p=0$)
2. $k+2m = 2p+q$ *and* $\ell=1$ *and* $k=q\leq 1$ *and* $m=p=0$
3. $k+2m = 2p+q$ *and* $\ell>1$ *and* $k=m=p=q=0$

and it is NP-complete otherwise.

Note that case 1 applies to non-regular target graph W, while cases 2 and 3 apply to regular graphs W, i.e., they cover all cases when $k+2m+\ell = 2p+q+\ell$.

We will refer to the vertex with k semi-edges as *blue* and the vertex with q semi-edges as *red*. In a covering projection $f=(f_V,f_E)$ from a graph G onto $W(k,m,\ell,p,q)$, we view the restricted vertex mapping f_V as a coloring of $V(G)$. We call a vertex $u \in V(G)$ blue (red) if f_V maps u onto the blue (red, respectively) vertex of $W(k,m,\ell,p,q)$. In order to keep the text clear and understandable, we divide the proof into a sequence of claims in separate subsections. This will also allow us to state several hardness results in a stronger form.

4.1 Polynomial parts of Theorem 14

We follow the case-distinction from the statement of Theorem 14:

1. If $k+2m \neq 2p+q$, then the two vertex degrees of $W(k,m,\ell,p,q)$ are different, and the vertex restricted mapping is uniquely defined for any possible graph covering projection from the input graph G to $W(k,m,\ell,p,q)$. For this coloring of G, if it exists, we check if it is degree-obedient. If not, then G does not cover $W(k,m,\ell,p,q)$. If yes, we check using Theorem 12 whether the blue subgraph of G covers $F(k,m)$ and whether the red subgraph of G covers $F(q,p)$. If any one of them does not, then G does not cover $W(k,m,\ell,p,q)$. If both of them do, then G covers $W(k,m,\ell,p,q)$, since the "remaining" subgraph of G formed by edges with one end-vertex red and the other one blue is ℓ-regular and bipartite, thus covering the ℓ parallel edges of $W(k,m,\ell,p,q)$ (Proposition 8).
2. In case 2, the input graph G covers $W(1,0,1,0,1)$ only if G is 2-regular. If this holds, then G is a disjoint union of cycles, and it is easy to see that a cycle covers $W(1,0,1,0,1)$ if and only if it length is divisible by 4. For the subcase of $k=q=0$, see the next point.
3. The input graph G covers $W(0,0,\ell,0,0)$ only if it is a bipartite ℓ-regular graph without semi-edges, but in that case it does cover $W(0,0,\ell,0,0)$, as follows from Proposition 8.

4.2 NP-hardness for non-regular target graphs

▶ **Proposition * 15.** *Let the parameters k,m,p,q be such that $k+2m \neq 2p+q$, and (($k \geq 3$ or $k=2$ and $m \geq 1$), or ($q \geq 3$ or $q=2$ and $p \geq 1$)). Then the $W(k,m,\ell,p,q)$-*Cover *problem is NP-complete.*

Proof sketch. The proof essentially relies on the reductions from the preceding section. The parameters ensure that after deleting the ℓ ordinary edges from the target graph, we end up with two graphs, $F(k,m)$ and $F(p,q)$, where at least one of them identifies one of the hard cases of covering one-vertex graphs. We then utilize a special gadget by which we connect the vertices of instances of $F(k,m)$-Cover and $F(q,p)$-Cover to get a graph G'. We further claim that we can decide both of these instances if and only if G' covers $W(k,m,\ell,p,q)$. The argument is significantly simplified by the determination of images of vertices due to the different degrees of vertices in the target graph. ◀

4.3 NP-hardness for connected regular target graphs

The aim of this subsection is to conclude the proof of Theorem 14 by showing the NP-hardness for the case of $\ell \geq 1$ and $k + 2m = 2p + q$. We will actually prove a result which is more general in two directions. Firstly, we formulate the result in the language of colorings of vertices, and secondly, we prove the hardness for bipartite inputs. This might seem surprising, as we have seen in Section 2 that bipartite graphs can make things easier. Moreover, this strengthening in fact allows us to prove the result in a unified, and hence simpler, way.

Note that the following definition of a relaxation of usual proper 2-coloring resembles the so-called *defective 2-coloring* (see survey of Wood [50]). Hoverer, the definitions are not equivalent.

▶ **Definition 16.** *A* (b, c)-coloring *of a graph is a 2-coloring of its vertices such that every vertex has* b *neighbors of its own color and* c *neighbors of the other color.*

▶ **Observation 17.** *For any parameters* k, m, ℓ, p, q *such that* $k+2m = 2p+q$*, a bipartite graph* G *with no semi-edges covers* $W(k, m, \ell, p, q)$ *if and only if it allows a* $(k + 2m, \ell)$*-coloring.*

Proof. On one hand, any graph covering projection from G to $W(k, m, \ell, p, q)$ induces a $(k+2m, \ell)$-coloring of G, provided $k+2m = 2p+q$. On the other hand, a $(k+2m, \ell)$-coloring of G is a degree-obedient vertex mapping from G to $W(k, m, \ell, p, q)$, again provided that $k + 2m = 2p + q$. If G is bipartite and has no semi-edges, then this mapping can be extended to a graph covering projection by Theorem 10. ◀

In view of the previous observation, we will be proving the NP-hardness results for the problem (b, c)-Coloring which takes a graph G on input and asks if G allows a (b, c)-coloring.

▶ **Theorem * 18.** *For every pair of positive integers* b, c *such that* $b + c \geq 3$*, the* (b, c)*-*Coloring *problem is NP-complete even for simple bipartite graphs.*

Proof sketch. First observe that the (b, c)-Coloring and (c, b)-Coloring problems are polynomially equivalent on bipartite graphs, as the colorings are mutually interchangeable by switching the colors in one class of the bi-partition. Thus we may consider only $b \geq c$.

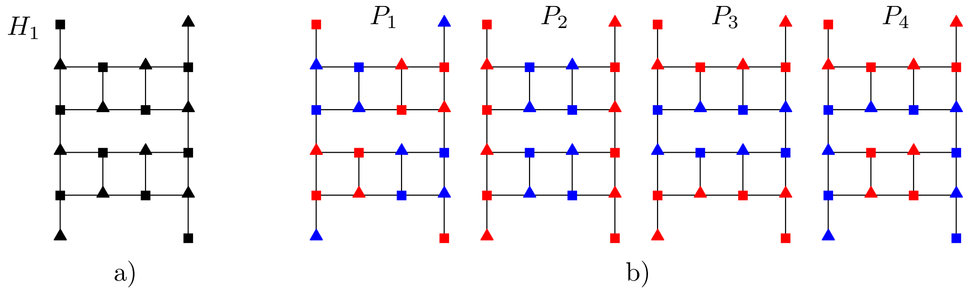

Figure 5 A 20-vertex auxiliary graph H_1, used in the first part of the proof of Theorem 18, and its possible partial $(2, 1)$-colorings.

NP-hardness of the $(2, 1)$-Coloring is proved by a reduction from NAE-3-SAT [25] by using three kinds of building blocks: a clause gadget (here $K_{1,3}$), a vertex gadget enforcing the same color on selected subset of vertices, and a garbage collection that allows to complete the coloring to a cubic graph, that as a part contains the vertex and clause gadgets linked together to represent a given instance of NAE-3-SAT. This reduction is the actual core of the proof, and is briefly sketched in Figures 5 and 6. The former one shows a special gadget

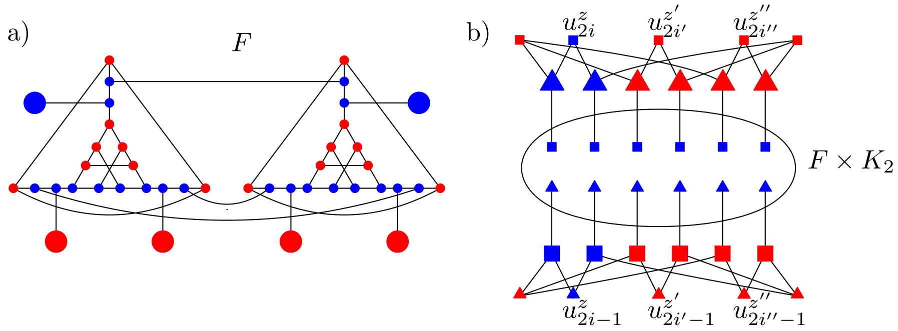

Figure 6 Garbage collection and the overall construction for the first part of Theorem 18. Clause gadgets are in the corners of the figure b).

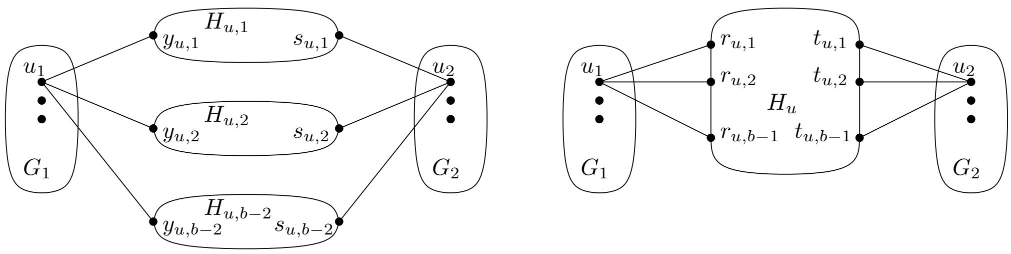

Figure 7 An illustration of the constructions used in the proof of Theorem 18; a reduction to $(b, 1)$-Coloring on the left, and a reduction to (b, c)-Coloring on the right.

H_1 used in the color-enforcing constructions of this reduction. For every variable, copies of H_1 are concatenated into a chain, whose one side is connected to ensure that the coloring P_2 (or its inverse) is the only admissible (2,1)-coloring, the other side transfers this information as the truth valuation of the variable to the clause gadgets of clauses containing it. The latter figure sketches the garbage collection and the overall construction of the reduction.

The result on $(2, 1)$-Coloring, in particular, implies that $W(0, 2, 1, 2, 0)$-Cover is NP-complete for simple bipartite input graphs, whereas the semi-edgeless dumbbell graph $W(0, 2, 1, 2, 0)$ is the smallest semi-edgeless graph whose covering is NP-complete.

Further on, we reduce $(2, 1)$-Coloring to $(b, 1)$-Coloring by using two copies of the instance of $(2, 1)$-Coloring and linking them together by suitable graphs called bridges, that enforce replication of colors for the desired coloring. See a brief sketch in Fig. 7 (left). In view of the initial observation, at this point we know that $(b, 1)$-Coloring and $(1, b)$-Coloring are NP-complete on bipartite inputs for all $b \geq 2$.

Then we reduce $(1, c)$-Coloring to (b, c)-Coloring with $b > c$. Again we take two copies of an instance of $(1, c)$-Coloring, say a $(1 + c)$-regular graph G. As sketched in Fig. 7 (right), we construct an auxiliary graph H with two vertices of degree $b - 1$ (called the "connector" vertices), all other vertices being of degree $b + c$ (these are called the "inner" vertices). This bridge graph is such that in every two-coloring of its vertices, such that all inner vertices have exactly b neighbors of their own color and exactly c neighbors of the opposite color, while the connector vertices have at most b neighbors of their own color and at most c neighbors of the opposite color, in every such a coloring the connector vertices and their neighbors always get the same color. And, moreover, such a coloring exists. We then take two copies of G and for every vertex of G, identify its copies with the connector vertices of a copy of the bridge graph (thus we have as many copies of the bridge graph as is the number of vertices of G). The above stated properties of the bridge graph guarantee that the new graph allows a (b, c)-coloring if and only if G allows a $(1, c)$-coloring.

It is worth mentioning that we have provided two different constructions of the bridge gadget. A general one for the case of $b \geq c+2$ and a specific one for the case of $b = c+1$. It is a bit surprising that the case analysis needed to prove the properties of the bridge graph is much more involved for the specific construction in the case of $b = c+1$.

Finally, for (b, b)-Coloring with $b \geq 2$ we establish a completely different reduction from a special variant of satisfiability (k-in-$2k$)-SAT$_q$, a generalization of NAE-3-SAT. ◀

Theorem 18 and Observation 17 imply the following proposition, which concludes the proof of Theorem 14.

▶ **Proposition 19.** *The $W(k, m, \ell, p, q)$-Cover problem is NP-complete for simple bipartite input graphs for all parameter sets such that $k + 2m = 2p + q \geq 1$, $\ell \geq 1$, and $k + 2m + \ell \geq 3$.*

5 Conclusion

The main goal of this paper is to initiate the study of the computational complexity of covering graphs with semi-edges. We have exhibited a new level of difficulty that semi-edges bring to coverings by showing a connection to edge-colorings. We have presented a complete classification of the computational complexity of covering graphs with at most two vertices, which is already a quite nontrivial task. In the case of one-vertex target graphs, the problem becomes polynomial-time solvable if the input graph is bipartite, while in the case of two-vertex target graphs, bipartiteness of the input graphs does not help. This provides a strengthening of known results of covering two-vertex graphs without semi-edges.

It is worth noting that the classification in [30] concerns a more general class of *colored mixed* (multi)graphs. I.e., graphs which may have both directed and undirected edges and whose edges come with assigned colors which must be preserved by the covering projections. It turns out that covering a two-vertex (multi)graph is NP-hard if and only if it is NP-hard for at least one of its maximal monochromatic subgraphs. It can be shown that the same holds true when semi-edges are allowed (note that all semi-edges must be undirected only).

We end up with an intriguing open problem.

▶ **Problem.** *Do there exist graphs H_1 and H_2, both without semi-edges, such that H_1 covers H_2, and such that the H_1-Cover is polynomial-time solvable and H_2-Cover is NP-complete?*

If semi-edges are allowed, then $H_1 = W(0, 0, 3, 0, 0)$ and $H_2 = F(3, 0)$ is such a pair. All further examples that we can obtain generalize this observation. They are unique in the sense that NP-completeness of H_2-Cover follows from the NP-completeness of the edge-colorability problem of general graphs which becomes polynomialy solvable for bipartite instances.

References

1 James Abello, Michael R. Fellows, and John C. Stillwell. On the complexity and combinatorics of covering finite complexes. *Australian Journal of Combinatorics*, 4:103–112, 1991.

2 Dana Angluin. Local and global properties in networks of processors. *Proceedings of the 12th ACM Symposium on Theory of Computing*, pages 82–93, 1980.

3 Dana Angluin and A. Gardiner. Finite common coverings of pairs of regular graphs. *Journal of Combinatorial Theory B*, 30:184–187, 1981.

4 Dan Archdeacon. Two graphs without planar covers. *Journal of Graph Theory*, 41(4):318–326, 2002.

5 Norman Biggs. *Algebraic Graph Theory.* Cambridge University Press, 1974.

6 Ondřej Bílka, Jozef Jirásek, Pavel Klavík, Martin Tancer, and Jan Volec. On the complexity of planar covering of small graphs. In Petr Kolman and Jan Kratochvíl, editors, *Graph-Theoretic Concepts in Computer Science*, volume 6986 of *Lecture Notes in Computer Science*, pages 83–94. Springer, 2011.

7 Hans L. Bodlaender. The classification of coverings of processor networks. *Journal of Parallel Distributed Computing*, 6:166–182, 1989.

8 Jérémie Chalopin, Yves Métivier, and Wiesław Zielonka. Local computations in graphs: the case of cellular edge local computations. *Fundamenta Informaticae*, 74(1):85–114, 2006.

9 Jérémie Chalopin and Daniël Paulusma. Graph labelings derived from models in distributed computing: A complete complexity classification. *Networks*, 58(3):207–231, 2011.

10 Jérémie Chalopin and Daniël Paulusma. Packing bipartite graphs with covers of complete bipartite graphs. *Discrete Applied Mathematics*, 168:40–50, 2014.

11 Steven Chaplick, Jiří Fiala, Pim van 't Hof, Daniël Paulusma, and Marek Tesař. Locally constrained homomorphisms on graphs of bounded treewidth and bounded degree. *Theoretical Computer Science*, 590:86–95, 2015.

12 Derek G. Corneil. *Graph Isomorphism.* PhD thesis, University of Toronto, 1968.

13 Derek G. Corneil and Calvin C. Gotlieb. An efficient algorithm for graph isomorphism. *Journal of the Association for Computing Machinery*, 17:51–64, 1970.

14 Bruno Courcelle and Yves Métivier. Coverings and minors: Applications to local computations in graphs. *European Journal of Combinatorics*, 15:127–138, 1994.

15 Dragomir Ž. Djoković. Automorphisms of graphs and coverings. *Journal of Combinatorial Theory B*, 16:243–247, 1974.

16 Jiří Fiala and Jan Kratochvíl. Locally injective graph homomorphism: Lists guarantee dichotomy. In Fedor V. Fomin, editor, *WG*, volume 4271 of *Lecture Notes in Computer Science*, pages 15–26. Springer, 2006.

17 Jiří Fiala. *Locally injective homomorphisms.* PhD thesis, Charles University, Prague, 2000.

18 Jiří Fiala, Pinar Heggernes, Petter Kristiansen, and Jan Arne Telle. Generalized H-coloring and H-covering of trees. *Nordic Journal of Computing*, 10(3):206–224, 2003.

19 Jiří Fiala, Pavel Klavík, Jan Kratochvíl, and Roman Nedela. Algorithmic aspects of regular graph covers with applications to planar graphs. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *ICALP (1)*, volume 8572 of *Lecture Notes in Computer Science*, pages 489–501. Springer, 2014.

20 Jiří Fiala, Pavel Klavík, Jan Kratochvíl, and Roman Nedela. 3-connected reduction for regular graph covers. *European Journal of Combinatorics*, 73:170–210, 2018.

21 Jiří Fiala and Jan Kratochvíl. Locally constrained graph homomorphisms — structure, complexity, and applications. *Computer Science Review*, 2(2):97–111, 2008.

22 Jiří Fiala and Daniël Paulusma. A complete complexity classification of the role assignment problem. *Theoretical Computer Science*, 1(349):67–81, 2005.

23 Jiří Fiala, Pavel Klavík, Jan Kratochvíl, and Roman Nedela. Algorithmic aspects of regular graph covers, 2016. `arXiv:1609.03013`.

24 Anthony Gardiner. Antipodal covering graphs. *Journal of Combinatorial Theory B*, 16:255–273, 1974.

25 Michael R. Garey and David S. Johnson. *Computers and Intractability.* W. H. Freeman and Co., New York, 1979.

26 Ezra Getzler and Mikhail M Kapranov. Modular operads. *Compositio Mathematica*, 110(1):65–125, 1998.

27 Jonathan L. Gross and Thomas W. Tucker. Generating all graph coverings by permutation voltage assignments. *Discrete Mathematics*, 18:273–283, 1977.

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

29 Petr Hliněný and Robin Thomas. On possible counterexamples to Negami's planar cover conjecture. *Journal of Graph Theory*, 46(3):183–206, 2004.

30 Jan Kratochvíl, Andrzej Proskurowski, and Jan Arne Telle. Covering directed multigraphs I. colored directed multigraphs. In Rolf H. Möhring, editor, *WG*, volume 1335 of *Lecture Notes in Computer Science*, pages 242–257. Springer, 1997.

31 Jan Kratochvíl, Andrzej Proskurowski, and Jan Arne Telle. Covering directed multigraphs II. when 2-SAT helps. Technical Report 1997–354, KAM-DIMATIA Preprint Series, 1997.

32 Jan Kratochvíl, Andrzej Proskurowski, and Jan Arne Telle. Covering regular graphs. *Journal of Combinatorial Theory, Series B*, 71(1):1–16, 1997.

33 Jan Kratochvíl, Andrzej Proskurowski, and Jan Arne Telle. Complexity of graph covering problems. *Nordic Journal of Computing*, 5:173–195, 1998.

34 Jan Kratochvíl, Jan Arne Telle, and Marek Tesař. Computational complexity of covering three-vertex multigraphs. *Theoretical Computer Science*, 609:104–117, 2016.

35 Petter Kristiansen and Jan Arne Telle. Generalized H-coloring of graphs. In D. T. Lee and Shang-Hua Teng, editors, *ISAAC*, volume 1969 of *Lecture Notes in Computer Science*, pages 456–466. Springer, 2000.

36 Jin Ho Kwak and Roman Nedela. Graphs and their coverings. *Lecture Notes Series*, 17, 2007.

37 Frank Thomas Leighton. Finite common coverings of graphs. *Journal of Combinatorial Theory B*, 33:231–238, 1982.

38 Daniel Leven and Zvi Galil. NP completeness of finding the chromatic index of regular graphs. *Journal of Algorithms*, 4:35–44, 1983.

39 Igor Litovsky, Yves Métivier, and Wiesław Zielonka. The power and the limitations of local computations on graphs. In Ernst W. Mayr, editor, *WG*, volume 657 of *Lecture Notes in Computer Science*, pages 333–345. Springer, 1992.

40 Laszló Lovász and Michael. D. Plummer. *Matching Theory*. Akadémiai Kiadó, Budapest, 1986.

41 Gary MacGillivray and Jacobus Swarts. The complexity of locally injective homomorphisms. *Discrete Mathematics*, 310(20):2685–2696, 2010. Graph Theory — Dedicated to Carsten Thomassen on his 60th Birthday.

42 Aleksander Malnič, Dragan Marušič, and Primož Potočnik. Elementary abelian covers of graphs. *Journal of Algebraic Combinatorics*, 20(1):71–97, 2004.

43 Aleksander Malnič, Roman Nedela, and Martin Škoviera. Lifting graph automorphisms by voltage assignments. *European Journal of Combinatorics*, 21(7):927–947, 2000.

44 Alexander D. Mednykh and Roman Nedela. *Harmonic Morphisms of Graphs: Part I: Graph Coverings*. Vydavatelstvo Univerzity Mateja Bela v Banskej Bystrici, 1st edition, 2015.

45 Roman Nedela and Martin Škoviera. Regular embeddings of canonical double coverings of graphs. *Journal of Combinatorial Theory, Series B*, 67(2):249–277, 1996.

46 Seiya Negami. Graphs which have no planar covering. *Bulletin of the Institute of Mathematics, Academia Sinica*, 16(4):377–384, 1988.

47 Julius Petersen. Die theorie der regulären graphs. *Acta Mathematica*, 15:193–220, 1891.

48 Gerhard Ringel. *Map color theorem*, volume 209. Springer, Berlin, 1974.

49 Sam Shepherd, Giles Gardam, and Daniel J. Woodhouse. Two generalisations of Leighton's theorem, 2019. arXiv:1908.00830.

50 David R. Wood. Defective and clustered graph colouring. *Electronic Journal of Combinatorics*, 2018. doi:10.37236/7406.

51 Daniel J. Woodhouse. Revisiting Leighton's theorem with the Haar measure, 2018. arXiv:1806.08196.