An Algorithmic Framework for Locally Constrained Homomorphisms*

Laurent Bulteau[1][0000−0003−1645−9345],
Konrad K. Dabrowski[2]✉[0000−0001−9515−6945],
Noleen Köhler[3][0000−0002−1023−6530], Sebastian Ordyniak[4][0000−0002−1825−0097],
and Daniël Paulusma[5][0000−0001−5945−9287]

[1] LIGM, CNRS, Université Gustave Eiffel, France laurent.bulteau@univ-eiffel.fr
[2] School of Computing, Newcastle University, UK
konrad.dabrowski@newcastle.ac.uk
[3] LAMSADE, CNRS, Université Paris-Dauphine, PSL University, France
noleen.kohler@dauphine.psl.eu
[4] School of Computing, University of Leeds, UK sordyniak@gmail.com
[5] Department of Computer Science, University of Durham, UK
daniel.paulusma@durham.ac.uk

Abstract. A homomorphism ϕ from a guest graph G to a host graph H is locally bijective, injective or surjective if for every $u \in V(G)$, the restriction of ϕ to the neighbourhood of u is bijective, injective or surjective, respectively. The corresponding decision problems, LBHom, LIHom and LSHom, are well studied both on general graphs and on special graph classes. We prove a number of new FPT, W[1]-hard and para-NP-complete results by considering a hierarchy of parameters of the guest graph G. For our FPT results, we do this through the development of a new algorithmic framework that involves a general ILP model. To illustrate the applicability of the new framework, we also use it to prove FPT results for the Role Assignment problem, which originates from social network theory and is closely related to locally surjective homomorphisms.

Keywords: (locally constrained) graph homomorphism · parameterized complexity · fracture number

1 Introduction

A *homomorphism* from a graph G to a graph H is a mapping $\phi : V(G) \to V(H)$ such that $\phi(u)\phi(v) \in E(H)$ for every $uv \in E(G)$. Graph homomorphisms generalise graph colourings (using a complete graph for H) and have been intensively studied over a long period of time. We refer to the textbook of Hell and Nešetřil [34] for a further introduction.

We write $G \to H$ if there exists a homomorphism from G to H; here, G is called the *guest graph* and H is the *host graph*. We denote the corresponding

* The second and fourth authors acknowledge support from the Engineering and Physical Sciences Research Council (EPSRC, project EP/V00252X/1).

decision problem by Hom, and if H is fixed, that is, not part of the input, we write H-Hom. The renowned Hell-Nešetřil dichotomy [33] states that H-Hom is polynomial-time solvable if H is bipartite, and NP-complete otherwise. We denote the vertices of H by $1, \ldots, |V(H)|$ and call them *colours*.

Instead of fixing the host graph H, one can also restrict the structure of the guest graph G by bounding some graph parameter. Here, it is known that if $\mathsf{FPT} \neq \mathsf{W}[1]$, then Hom can be solved in polynomial time if and only if the so-called core of the guest graph has bounded treewidth [31].

Locally Constrained Homomorphisms. We are interested in three well-studied variants of graph homomorphisms that occur after placing constraints on the neighbourhoods of the vertices of the guest graph G. Consider a homomorphism ϕ from a graph G to a graph H. We say that ϕ is locally injective, locally bijective or locally surjective for $u \in V(G)$ if restricting ϕ to a function $\phi_u : N_G(u) \to N_H(\phi(u))$ is injective, bijective or surjective, respectively. Here, $N_G(u) = \{v \mid uv \in E(G)\}$ denotes the (open) neighbourhood of a vertex u in a graph G. We say that ϕ is *locally injective*, *locally bijective* or *locally surjective* if it is locally injective, locally bijective or locally surjective for every $u \in V(G)$. We denote existence of these *locally constrained* homomorphisms by $G \xrightarrow{B} H$, $G \xrightarrow{I} H$ and $G \xrightarrow{S} H$, respectively.

Locally injective homomorphisms are also known as *partial graph coverings* and are used in telecommunications [23], in distance constrained labelling [22] and as indicators of the existence of homomorphisms of derivative graphs [46]. Locally bijective homomorphisms originate from topological graph theory [4,45] and are more commonly known as *graph coverings*. They are used in distributed computing [2,3,7] and in constructing highly transitive regular graphs [5]. Locally surjective homomorphisms are sometimes called *colour dominations* [41]. They have applications in distributed computing [11,12] and in social science [20,50,53,54]. In the latter context they are known as *role assignments*.

Let LBHom, LIHom and LSHom be the three problems of deciding, for two graphs G and H, whether $G \xrightarrow{B} H$, $G \xrightarrow{I} H$ or $G \xrightarrow{S} H$ holds, respectively. As before, we write H-LBHom, H-LIHom and H-LSHom in the case when the host graph H is fixed. Out of the three problems, only the complexity of H-LSHom has been completely classified, both for general graphs and bipartite graphs [26]. We refer to a series of papers [1,6,23,25,38,39,44] for polynomial-time solvable and NP-complete cases of H-LBHom and H-LIHom; see also the survey by Fiala and Kratochvíl [24]. Some more recent results include sub-exponential algorithms for H-LBHom, H-LIHom and H-LSHom on string graphs [48] and complexity results for H-LBHom for host graphs H that are multigraphs [40] or that have semi-edges [9].

In our paper we assume that both G and H are part of the input. We note a fundamental difference between locally injective homomorphisms on the one hand and locally bijective and surjective homomorphisms on the other. Namely, for connected graphs G and H, we must have $|V(G)| \geq |V(H)|$ if $G \xrightarrow{B} H$ or $G \xrightarrow{S} H$ (this is a consequence of Observation 1), whereas H might be arbitrarily larger than G if $G \xrightarrow{I} H$ holds. For example, if we let G be a complete graph and

H be a graph without self-loops, then $G \xrightarrow{I} H$ holds if and only if H contains a clique on at least $|V(G)|$ vertices.

The above difference is also reflected in the complexity results for the three problems under input restrictions. In fact, LIHom is closely related to the Subgraph Isomorphism problem and is usually the hardest problem. For example, LBHom is Graph Isomorphism-complete on chordal guest graphs, but polynomial-time solvable on interval guest graphs and LSHom is NP-complete on chordal guest graphs, but polynomial-time solvable on proper interval guest graphs [32]. In contrast, LIHom is NP-complete even on complete guest graphs G, which follows from a reduction from the Clique problem via the aforementioned equivalence: $G \xrightarrow{I} H$ holds if and only if H contains a clique on at least $|V(G)|$ vertices.

The aforementioned polynomial-time result on Hom for guest graphs G with a core of bounded treewidth [15,30] does not carry over to any of the three locally constrained homomorphism problems. Indeed, LBHom, LSHom and LIHom are NP-complete for guest graphs G of path-width at most 5, 4 and 2, respectively [14] (all three problems are polynomial-time solvable if G is a tree [14,27]). It is also known that LBHom [37], LSHom [41] and LIHom [23] are NP-complete even if G is cubic and H is the complete graph K_4 on four vertices, but polynomial-time solvable if G has bounded treewidth and one of the two graphs G or H has bounded maximum degree [14].

An Application. Locally surjective homomorphisms from a graph G to a graph H are known as H-role assignments in social network theory. Role assignments were introduced by White and Reitz [54]. A connected graph G has an h-role assignment if and only if $G \xrightarrow{S} H$ for some connected graph H with $|V(H)| = h$, as long as we allow H to have self-loops (while we assume that G is a graph with no self-loops). The Role Assignment problem is to decide, for a graph G and an integer h, whether G has an h-role assignment. If h is fixed, we denote the problem h-Role Assignment. h-Role Assignment is NP-complete for planar graphs ($h \geq 2$) [51], cubic graphs ($h \geq 2$) [52], bipartite graphs ($h \geq 3$) [49], chordal graphs ($h \geq 3$) [35] and split graphs ($h \geq 4$) [16].

Our Focus. We continue the line of study in [14] and focus on the following research question: *For which parameters of the guest graph do* LBHom*,* LSHom *and* LIHom *become fixed-parameter tractable?*

We will also apply our new techniques towards answering this question for the Role Assignment problem. In order to address our research question, we need some additional terminology. A graph parameter p *dominates* a parameter q if there is a function f such that $p(G) \leq f(q(G))$ for every graph G. If p dominates q but q does not dominate p, then p is *more powerful* than q. We denote this by $p \rhd q$. If neither p dominates q nor q dominates p, then p and q are *incomparable (orthogonal)*. Given the para-NP-hardness results on LBHom, LSHom and LIHom for graph classes of bounded path-width [14], we will consider a range of graph parameters that are less powerful than path-width. In this way we aim to increase our understanding of the (parameterized) complexity of LBHom, LSHom and LIHom.

For an integer $c \geq 1$, a *c-deletion set* of a graph G is a subset $S \subseteq V(G)$ such that every connected component of $G \setminus S$ has at most c vertices. The *c-deletion set number* $\mathrm{ds}_c(G)$ of a graph G is the minimum size of a c-deletion set in G. If $c = 1$ we obtain the *vertex cover number* $\mathrm{vc}(G)$ of G. The c-deletion set number is closely related to the *fracture number* $\mathrm{fr}(G)$, introduced in [19], which is the minimum k such that G has a k-deletion set on at most k vertices. Both these parameters are also closely related to *vertex integrity* [18]. Note that $\mathrm{fr}(G) \leq \max\{c, \mathrm{ds}_c(G)\}$ holds for every integer c. The *feedback vertex set number* $\mathrm{fv}(G)$ of a graph G is the size of a smallest set S such that $G \setminus S$ is a forest. We write $\mathrm{tw}(G)$, $\mathrm{pw}(G)$ and $\mathrm{td}(G)$ for the treewidth, path-width and tree-depth of a graph G, respectively; see [47] for more information. It is known that $\mathrm{tw}(G) \rhd \mathrm{pw}(G) \rhd \mathrm{td}(G) \rhd \mathrm{fr}(G) \rhd \mathrm{ds}_c(G)(\text{fixed } c) \rhd \mathrm{vc}(G) \rhd |V(G)|$, where the second relationship is proven in [8] and the others follow immediately from their definitions (see also Section 2). It is readily seen that $\mathrm{tw}(G) \rhd \mathrm{fv}(G) \rhd \mathrm{ds}_2(G)$ and that $\mathrm{fv}(G)$ is incomparable with the parameters $\mathrm{pw}(G)$, $\mathrm{td}(G)$, $\mathrm{fr}(G)$ and $\mathrm{ds}_c(G)$ for every fixed $c \geq 3$ (consider e.g. a tree of large path-width and the disjoint union of many triangles).

Our Results. We prove a number of new parameterized complexity results for LBHom, LSHom and LIHom by considering some property of the guest graph G as the parameter. In particular, we consider the graph parameters above. Our two main results, which are proven in Section 4, show that LBHom and LSHom are fixed-parameter tractable parameterized by the fracture number of G. These two results cannot be strengthened to the tree-depth of the guest graph, for which we prove para-NP-completeness. Note that the latter results imply the known para-NP-completeness results for path-width of the guest graph [14]. We also prove that LBHom and LSHom are para-NP-complete when parameterized by the feedback vertex set number of the guest graph. This result and the para-NP-hardness for tree-depth motivated us to consider the fracture number as a natural remaining graph parameter for obtaining an fpt algorithm.

Concerning LIHom, we prove that it is in XP and W[1]-hard when parameterized by the vertex cover number, or equivalently, the c-deletion set number for $c = 1$. We then show that the XP-result for LIHom cannot be generalised to hold for $c \geq 2$. In fact, in Section 4, we will determine the complexity of LIHom on graphs with c-deletion set number at most k for every fixed pair of integers c and k. Our results for LBHom, LSHom and LIHom are summarised, together with the known results, in Table 1.

Algorithmic Framework. The fpt algorithms for LBHom and LSHom are proven via a new algorithmic framework (described in detail in Section 3) that involves a reduction to an integer linear program (ILP) that has a wider applicability. To illustrate this, in Section 4 we also use our general framework to prove that Role Assignment is in FPT when parameterized by $c + \mathrm{ds}_c$, or equivalently by fracture number.

Techniques. The main ideas behind our algorithmic ILP framework are as follows. Let G and H be the guest and host graphs, respectively. First, we observe that if G has a c-deletion of size at most k and there is a locally surjective

Table 1. Table of results. The results marked with a ($\star$) are the new results in this paper. The results in black are either known results, some of which are now also implied by our new results, or follow immediately from other results in the table; in particular, for a graph G, $\mathrm{ds}_c(G) \geq \mathrm{fr}(G)$ if $c \leq \mathrm{fr}(G) - 1$, and $\mathrm{ds}_c(G) \leq \mathrm{fr}(G)$ if $c \geq \mathrm{fr}(G)$. Also note that LIHom is W[1]-hard when parameterized by $|V(G)|$, as Clique is W[1]-hard when parameterized by the clique number [17], so as before, we can let G be the complete graph in this case.

guest graph parameter	LIHom	LBHom	LSHom
$\|V(G)\|$	XP, W[1]-hard [17]	FPT	FPT
vertex cover number	XP ($\star$), W[1]-hard	FPT	FPT
c-deletion set number (fixed c)	para-NP-c ($c \geq 2$) ($\star$)	FPT	FPT
fracture number	para-NP-c	FPT (Theorem 4) ($\star$)	FPT (Theorem 4) ($\star$)
tree-depth	para-NP-c	para-NP-c ($\star$)	para-NP-c ($\star$)
path-width	para-NP-c [14]	para-NP-c [14]	para-NP-c [14]
treewidth	para-NP-c	para-NP-c	para-NP-c
maximum degree	para-NP-c [23]	para-NP-c [37]	para-NP-c [41]
treewidth plus maximum degree	XP, W[1]-hard	XP [14]	XP [14]
feedback vertex set number	para-NP-c	para-NP-c ($\star$)	para-NP-c ($\star$)

homomorphism from G to H, then H must also have a c-deletion set of size at most k. However it does not suffice to compute c-deletion sets D_G and D_H for G and H, guess a partial homomorphism h from D_G to D_H, and use the structural properties of c-deletion sets to decide whether h can be extended to a desired homomorphism from G to H. This is because a homomorphism from G to H does not necessarily map D_G to D_H. Moreover, even if it did, vertices in $G \setminus D_G$ can still be mapped to vertices in D_H. Consequently, components of $G \setminus D_G$ can still be mapped to more than one component of $H \setminus D_H$. This makes it difficult to decompose the homomorphism from G to H into small independent parts. To overcome this challenge, we prove that there are small sets D_G and D_H of vertices in G and H, respectively, such that every locally surjective homomorphism from G to H satisfies:

1. the pre-image of D_H is a subset of D_G,
2. D_H is a c'-deletion set for H for some c' bounded in terms of only $c + k$, and
3. all but at most k components of $G \setminus D_G$ have at most c vertices and, while the remaining components can be arbitrarily large, their treewidth is bounded in terms of $c + k$.

As D_G and D_H are small, we can enumerate all possible homomorphisms from some subset of D_G to D_H. Condition 2 allows us to show that any locally surjective homomorphism from G to H can be decomposed into locally surjective homomorphisms from a small set of components of $G \setminus D_G$ (plus D_G) to one component of $H \setminus D_H$ (plus D_H). This enables us to formulate the question of whether a homomorphism from a subset of D_G to D_H can be extended to a desired homomorphism from G to H in terms of an ILP. Finally, Condition 3 allows us to efficiently compute the possible parts of the decomposition, that is, which (small) sets of components of $G \setminus D_G$ can be mapped to which components of $H \setminus D_H$.

2 Preliminaries

Let G be a graph. We denote the vertex set and edge set of G by $V(G)$ and $E(G)$, respectively. Let $X \subseteq V(G)$ be a set of vertices of G. The *subgraph of G induced by X*, denoted $G[X]$, is the graph with vertex set X and edge set $\{uv \in E(G) \mid u, v \in X\}$. When the underlying graph is clear from the context, we will sometimes refer to an induced subgraph simply by its set of vertices. We use $G \setminus X$ to denote the subgraph of G induced by $V(G) \setminus X$. Similarly, for $Y \subseteq E(G)$ we let $G \setminus Y$ be the subgraph of G obtained by deleting all edges in Y from G. For a graph G and a vertex $u \in V(G)$, we let $N_G(u) = \{v \mid uv \in E(G)\}$ and $N_G[v] = N_G(v) \cup \{v\}$ denote the open and closed neighbourhood of v in G, respectively. Recall that we assume that the guest graph G does not contain self-loops, while the host graph H is permitted to have self-loops. In this case, by definition, $u \in N_H(u)$ if $uu \in E(H)$. We need the following well-known fact:

Proposition 1 ([42]). *Let G be a graph and let k and c be natural numbers. Then, deciding whether G has a c-deletion set of size at most k is fixed-parameter tractable parameterized by $k + c$.*

A (k, c)*-extended deletion set* for G is a set $D \subseteq V(G)$ such that: (1) every component of $G \setminus D$ either has at most c vertices or has a c-deletion set of size at most k and (2) at most k components of $G \setminus D$ have more than c vertices.

Locally Constrained Homomorphisms. Here we show some basic properties of locally constrained homomorphisms.

Observation 1. *Let G and H be non-empty connected graphs and let ϕ be a locally surjective homomorphism from G to H. Then ϕ is surjective.*

Observation 2. *Let G and H be graphs, let $D \subseteq V(G)$, and let ϕ be a homomorphism from G to H. Then, for every component C_G of $G \setminus D$ such that $\phi(C_G) \cap \phi(D) = \emptyset$, there is a component C_H of $H \setminus \phi(D)$ such that $\phi(C_G) \subseteq C_H$. Moreover, if ϕ is locally injective/surjective/bijective, then $\phi|_{D \cup C_G}$ is a homomorphism from $G' = G[D \cup C_G]$ to $H' = H[\phi(D) \cup C_H]$ that is locally injective/surjective/bijective for every $v \in V(C_G)$.*

Lemma 1. *Let G and H be non-empty connected graphs, let $D \subseteq V(G)$ be a c-deletion set for G, and let ϕ be a locally surjective homomorphism from G to H. Then $\phi(D)$ is a c-deletion set for H.*

Integer Linear Programming. Given a set $\mathcal{X}$ of variables and a set $\mathcal{C}$ of linear constraints (i.e. inequalities) over the variables in $\mathcal{X}$ with integer coefficients, the task in the feasibility variant of *integer linear programming* (ILP) is to decide whether there is an assignment $\alpha : \mathcal{X} \to \mathbb{Z}$ of the variables satisfying all constraints in $\mathcal{C}$. We will use the following well-known result by Lenstra [43].

Proposition 2 ([21,29,36,43]). ILP *is fpt parameterized by the number of variables.*

3 Our Algorithmic Framework

Here we present our main algorithmic framework that will allow us to show that LSHom, LBHom and Role Assignment are fpt parameterized by $k + c$ when the guest graph has c-deletion set number at most k. To illustrate the main ideas behind our framework, let us first explain these ideas for the examples of LSHom and LBHom. In this case we are given G and H and we know that G has a c-deletion set of size at most k. Because of Lemma 1, it then follows that if (G, H) is a yes-instance of LSHom or LBHom, then H also has a c-deletion set of size at most k. Informally, our next step is to compute a small set Φ of partial locally surjective homomorphisms such that (1) every locally surjective homomorphism from G to H augments some $\phi_P \in \Phi$ and (2) for every $\phi_P \in \Phi$, the domain of ϕ_P is a (k, c)-extended deletion set of G and the co-domain of ϕ_P is a c'-deletion set of H, where c' is bounded by a function of $k + c$. Here and in what follows, we say that a function $\phi : V(G) \to V(H)$ *augments* (or is an *augmentation* of) a partial function $\phi_P : V_G \to V_H$, where $V_G \subseteq V(G)$ and $V_H \subseteq V(H)$ if $v \in V_G \Leftrightarrow \phi(v) \in V_H$ and $\phi|_{V_G} = \phi_P$. This allows us to reduce our problems to (boundedly many) subproblems of the following form: Given a (k, c)-extended deletion set D_G for G, a c'-deletion set D_H for H, and a locally surjective (respectively bijective) homomorphism ϕ_P from D_G to D_H, find a locally surjective homomorphism ϕ from G to H that augments ϕ_P. We will then show how to formulate this subproblem as an integer linear program and how this program can be solved efficiently. Importantly, our ILP formulation will allow us to solve a much more general problem, where the host graph H is not explicitly given, but defined in terms of a set of linear constraints, which will allow us to solve the Role Assignment problem.

Partial Homomorphisms for the Deletion Set. For a graph G and $m \in \mathbb{N}$ we let $D_G^m := \{v \in V(G) \mid \deg_G(v) \geq m\}$. We will show in Lemma 4 that there is a small set Φ of partial homomorphisms such that every locally surjective (respectively bijective) homomorphism from G to H augments some $\phi_P \in \Phi$ and, for every $\phi_P \in \Phi$, the domain of ϕ_P is a (k, c)-extended deletion set for G of size at most k and its co-domain is a c'-deletion set of size at most k for H. The main idea behind finding this set Φ is to consider the set of high degree vertices in G and H, i.e. the sets D_G^{k+c} and D_H^{k+c}. As it turns out (see Lemma 2), for every subset $D \subseteq D_G^{k+c}$, D is a $(k - |D|, c)$-extended deletion set for G of size at most k and D_H^{k+c} is a c'-deletion set for H of size at most k, where $c' = kc(k + c)$. Moreover, as we will show in Lemma 3, every locally surjective (respectively bijective) homomorphism from G to H has to augment a locally surjective (respectively bijective) homomorphism from some induced subgraph of $G[D_G^{k+c}]$ to $D_H = D_H^{k+c}$. Intuitively, this holds because for every locally surjective homomorphism, only vertices of high degree in G can be mapped to a vertex of high degree in H and every vertex in H must have a pre-image in G.

Lemma 2. *Let G be a graph. If G has a c-deletion set of size at most k, then the set D_G^{k+c} is a $kc(k + c)$-deletion set of size at most k. Furthermore, every subset $D \subseteq D_G^{k+c}$ is a $(k - |D|, c)$-extended deletion set of G.*

Lemma 3. *Let G and H be non-empty connected graphs such that G has a c-deletion set of size at most k. If there is a locally surjective homomorphism ϕ from G to H, then there is a set $D \subseteq D_G^{k+c}$ and a locally surjective homomorphism ϕ_P from $G[D]$ to $H[D_H^{k+c}]$ such that ϕ augments ϕ_P. If ϕ is locally bijective, then $D = D_G^{k+c}$ and ϕ_P is a locally bijective homomorphism.*

Proof. By Lemma 2, D_G^{k+c} is a $kc(k+c)$-deletion set of size at most k. Furthermore, observe that for a locally surjective homomorphism ϕ from G to H, the inequality $\deg_G(v) \geq \deg_H(\phi(v))$ holds for every $v \in V(G)$ ($\deg_G(v) = \deg_H(\phi(v))$ holds in the locally bijective case). Since ϕ is surjective by Observation 1, this implies that $\phi(D_G^{k+c}) \supseteq D_H^{k+c}$ (and if ϕ is locally bijective, then $\phi(D_G^{k+c}) = D_H^{k+c}$). By Lemma 1, $\phi(D_G^{k+c})$ is a $kc(k+c)$-deletion set for H. Let $D = \phi^{-1}(D_H^{k+c})$, so $D \subseteq D_G^{k+c}$ (note that $D = D_G^{k+c}$ if ϕ is locally bijective). Now $\phi|_D$ is a surjective map from D to D_H^{k+c}. Furthermore, $\phi(D_G^{k+c} \setminus D) \cap \phi(D) = \phi(D_G^{k+c} \setminus D) \cap D_H^{k+c} = \emptyset$. Moreover, for every $v \in V(G) \setminus D_G^{k+c}$, $\phi(v) \notin D_H^{k+c} = \phi|_D(D)$, since $\deg_G(v) \geq \deg_H(\phi(v))$. Furthermore, $\phi|_D$ is a homomorphism from $G[D]$ to $H[D_H^{k+c}]$ because ϕ is a homomorphism. We argue that $\phi|_D$ is locally surjective (respectively bijective) by contradiction. Suppose $\phi|_D$ is not locally surjective. Then there is a vertex $u \in D$ and a neighbour $v \in D_H^{k+c}$ of $\phi|_D(u)$ such that $v \notin \phi|_D(N_G(u) \cap D)$. Since ϕ is locally surjective, there must be $w \in N_G(u) \setminus D$ such that $\phi(w) = v$. This contradicts the fact that $\phi(V(G) \setminus D) \cap D_H^{k+c} = \emptyset$. Hence $\phi|_D$ is a locally surjective homomorphism. In the bijective case we just need to additionally observe that $\phi|_D$ restricted to the neighbourhood of any vertex $v \in D$ must be injective. This completes the proof. ◻

Lemma 4. *Let G and H be non-empty connected graphs and let k, c be non-negative integers. For any $D \subseteq D_G^{k+c}$, we can compute the set Φ_D of all locally surjective (respectively bijective) homomorphisms ϕ_P from $G[D]$ to $H[D_H^{k+c}]$ in $\mathcal{O}(|D|^{|D|+2})$ time. Furthermore, $|\Phi_D| \leq |D|^{|D|}$.*

ILP Formulation. We will show how to formulate the subproblem obtained in the previous subsection in terms of an ILP instance. More specifically, we will show that the following problem can be formulated in terms of an ILP: given a partial locally surjective (respectively bijective) homomorphism ϕ_P from some induced subgraph D_G of G to some induced subgraph D_H of H, can this be augmented to a locally surjective (respectively bijective) homomorphism from G to H? Moreover, we will actually show that for this to work, the host graph H does not need to be given explicitly, but can instead be defined by a certain system of linear constraints.

The main ideas behind our translation to ILP are as follows. Suppose that there is a locally surjective (respectively bijective) homomorphism ϕ from G to H that augments ϕ_P. Because ϕ augments ϕ_P, Observation 2 implies that ϕ maps every component C_G of $G \setminus V(D_G)$ entirely to some component C_H of $H \setminus V(D_H)$, moreover, $\phi|_{V(D_G) \cup V(C_G)}$ is already locally surjective (respectively bijective) for every vertex $v \in V(C_G)$. Our aim now is to describe ϕ in terms of its parts consisting of locally surjective (respectively bijective) homomorphisms from *extensions* of D_G in G, i.e. sets of components of $G \setminus D_G$ plus D_G, to *simple extensions* of D_H in H, i.e. single components of $H \setminus D_H$ plus D_H. Note that the main difficulty comes from the fact that we need to ensure that ϕ is locally surjective (respectively bijective) for every $d \in D_G$ and not only for the vertices within the components of $G \setminus D_G$. This is why we need to describe the parts of ϕ using sets of components of $G \setminus D_G$ and not just single components. However, as we will show, it will suffice to consider only minimal extensions of D_G in G, where an extension is minimal if no subset of it allows for a locally surjective (respectively bijective) homomorphism from it to some simple extension of D_H in H. The fact that we only need to consider minimal extensions is important for showing that we can compute the set of all possible parts of ϕ efficiently. Having shown this, we can create an ILP that has one variable $x_{\text{Ext}_G\text{Ext}_H}$ for every minimal extension Ext_G and every simple extension Ext_H such that there is a locally surjective (respectively bijective) homomorphism from Ext_G to Ext_H that augments ϕ_P. The value of the variable $x_{\text{Ext}_G\text{Ext}_H}$ now corresponds to the number of parts used by ϕ that map minimal extensions isomorphic to Ext_G to simple extensions isomorphic to Ext_H that augment ϕ_P. We can then use linear constraints on these variables to ensure that:

(SB2') H contains exactly the right number of extensions isomorphic to Ext_H required by the assignment for $x_{\text{Ext}_G\text{Ext}_H}$,
(B1') G contains exactly the right number of minimal extensions isomorphic to Ext_G required by the assignment for $x_{\text{Ext}_G\text{Ext}_H}$ (if ϕ is locally bijective),
(S1') G contains at least the number of minimal extensions isomorphic to Ext_G required by the assignment for $x_{\text{Ext}_G\text{Ext}_H}$ (if ϕ is locally surjective),
(S3') for every simple extension Ext_G of G that is not yet used in any part of ϕ, there is a homomorphism from Ext_G to some simple extension of D_H in H that augments ϕ_P and is locally surjective for every vertex in $\text{Ext}_G \setminus D_G$ (if ϕ is locally surjective).

Together, these constraints ensure that there is a locally surjective (respectively bijective) homomorphism ϕ from G to H that augments ϕ_P. To do so, we need the following additional notation.

Given a graph D, an *extension* for D is a graph E containing D as an induced subgraph. It is *simple* if $E \setminus D$ is connected, and *complex* in general. Given two extensions $\text{Ext}_1, \text{Ext}_2$ of D, we write $\text{Ext}_1 \sim_D \text{Ext}_2$ if there is an isomorphism τ from Ext_1 to Ext_2 with $\tau(d) = d$ for every $d \in D$. Then $\sim_D$ is an equivalence relation. Let the *types* of D, denoted $\mathcal{T}_D$, be the set of equivalence classes of $\sim_D$ of simple extensions of D. We write $\mathcal{T}_D^c$ to denote the set of types of D of size at most $|D| + c$, so $|\mathcal{T}_D^c| \leq (|D| + c)2^{\binom{|D|+c}{2}}$.

Given a complex extension E of D, let C be a connected component of $E \setminus D$. Then C has type $T \in \mathcal{T}_D$ if $E[D \cup C] \sim_D T$ (depending on the context, we also say that the extension $E[D \cup C]$ has type T). The *type-count* of E is the function $\text{tc}_E : \mathcal{T}_D \to \mathbb{N}$ such that $\text{tc}_E(T)$ for $T \in \mathcal{T}_D$ is the number of connected components of $E \setminus D$ with type T (in particular if E is simple, the type-count is 1 for E and 0 for other types). Note that two extensions are equivalent if and only if they have the same type-counts; this then also implies that there is an isomorphism τ between the two extensions satisfying $\tau(d) = d$ for every $d \in D$. We write $E \preceq E'$ if $\mathrm{tc}_E(T) \leq \mathrm{tc}_{E'}(T)$ for all types $T \in \mathcal{T}_D$. If E is an extension of D, we write $\mathcal{T}_D(E) = \{T \in \mathcal{T}_D \mid \mathrm{tc}_E(T) \geq 1\}$ for the *set of types of* E and $\mathcal{E}_D(E)$ for the set of simple extensions of E. Moreover, for $T \in \mathcal{T}_D$, we write $\mathcal{E}_D(E,T)$ for the set of simple extensions in E having type T.

A *target description* is a tuple (D_H, c, CH) where D_H is a graph, c is an integer and CH is a set of linear constraints over variables x_T, $T \in \mathcal{T}^c_{D_H}$. A type-count for D_H is an integer assignment of the variables x_T. A graph H satisfies the target description (D_H, c, CH) if it is an extension of D_H, $\mathrm{tc}_H(T) = 0$ for $T \notin \mathcal{T}^c_{D_H}$, and setting $x_T = \mathrm{tc}_H(T)$ for all $T \in \mathcal{T}^c_{D_H}$ satisfies all constraints in CH.

In what follows, we assume that the following are given: the graphs D_G, D_H, an extension G of D_G, a target description $\mathcal{D} = (D_H, c, \mathrm{CH})$, and a locally surjective (respectively bijective) homomorphism $\phi_P : D_G \to D_H$. Let Ext_G be an extension of D_G with $\mathrm{Ext}_G \preceq G$ and let $T_H \in \mathcal{T}^c_{D_H}$; note that we only consider $T_H \in \mathcal{T}^c_{D_H}$, because we assume that T_H is a type of a simple extension of a graph H that satisfies the target description $\mathcal{D}$. We say Ext_G can be *weakly* ϕ_P*-S-mapped* to a type T_H if there exists an augmentation $\phi : \mathrm{Ext}_G \to T_H$ of ϕ_P such that ϕ is locally surjective for every $v \in \mathrm{Ext}_G \setminus D_G$. We say that Ext_G can be ϕ_P*-S-mapped* (respectively ϕ_P*-B-mapped*) to a type T_H if there exists an augmentation $\phi : \mathrm{Ext}_G \to T_H$ of ϕ_P such that ϕ is locally surjective (respectively locally bijective). Furthermore, Ext_G can be *minimally* ϕ_P*-S-mapped* (respectively *minimally* ϕ_P*-B-mapped*) to T_H if Ext_G can be ϕ_P-S-mapped (respectively ϕ_P-B-mapped) to T_H and no other extension Ext'_G with $\mathrm{Ext}'_G \preceq \mathrm{Ext}_G$ can be ϕ_P-S-mapped (respectively ϕ_P-B-mapped) to T_H. Let $\mathrm{wSM}(G, D_G, \mathcal{D}, \phi_P)$ be the set of all pairs (T_G, T_H) such that $T_G \in \mathcal{T}_{D_G}(G)$ can be weakly ϕ_P-S-mapped to T_H. Let $\mathrm{SM}(G, D_G, \mathcal{D}, \phi_P)$ be the set of all pairs (Ext_G, T_H) with $\mathrm{Ext}_G \preceq G$, $T_H \in \mathcal{T}^c_{D_H}$ such that Ext_G can be minimally ϕ_P-S-mapped to T_H and let $\mathrm{BM}(G, D_G, \mathcal{D}, \phi_P)$ be the set of all pairs (Ext_G, T_H) with $\mathrm{Ext}_G \preceq G$, $T_H \in \mathcal{T}^c_{D_H}$ such that Ext_G can be minimally ϕ_P-B-mapped to T_H.

We now build a set of linear constraints. To this end, besides variables x_T for $T \in T_H$, we introduce variables $x_{\mathrm{Ext}_G T_H}$ for each $(\mathrm{Ext}_G, T_H) \in \mathrm{SM}$ (respectively BM), where here and in what follows $\mathrm{wSM} = \mathrm{wSM}(G, D_G, \mathcal{D}, \phi_P)$, $\mathrm{SM} = \mathrm{SM}(G, D_G, \mathcal{D}, \phi_P)$ and $\mathrm{BM} = \mathrm{BM}(G, D_G, \mathcal{D}, \phi_P)$.

(S1) $\sum_{(\mathrm{Ext}_G, T_H) \in \mathrm{SM}} \mathrm{tc}_{\mathrm{Ext}_G}(T_G) * x_{\mathrm{Ext}_G T_H} \leq \mathrm{tc}_G(T_G)$ for every $T_G \in \mathcal{T}_{D_G}(G)$,
(B1) $\sum_{(\mathrm{Ext}_G, T_H) \in \mathrm{BM}} \mathrm{tc}_{\mathrm{Ext}_G}(T_G) * x_{\mathrm{Ext}_G T_H} = \mathrm{tc}_G(T_G)$ for every $T_G \in \mathcal{T}_{D_G}(G)$,
(S2) $\sum_{\mathrm{Ext}_G : (\mathrm{Ext}_G, T_H) \in \mathrm{SM}} x_{\mathrm{Ext}_G, T_H} = x_{T_H}$ for every $T_H \in \mathcal{T}_{D_H}$,
(B2) $\sum_{\mathrm{Ext}_G : (\mathrm{Ext}_G, T_H) \in \mathrm{BM}} x_{\mathrm{Ext}_G, T_H} = x_{T_H}$ for every $T_H \in \mathcal{T}_{D_H}$,
(S3) $\sum_{(T_G, T_H) \in \mathrm{wSM}} x_{T_H} \geq 1$ for every $T_G \in \mathcal{T}_{D_G}(G)$.

Lemma 5. *Let D_G and D_H be graphs, let G be an extension of D_G and let $\mathcal{D} = (D_H, c, \mathrm{CH})$ be a target description. Moreover, let $\phi_P : V(D_G) \to V(D_H)$ be a locally surjective (respectively bijective) homomorphism from D_G to D_H. There exists a graph H satisfying $\mathcal{D}$ and a locally surjective (respectively bijective) homomorphism ϕ augmenting ϕ_P if and only if the equation system* (CH, S1, S2, S3) *(respectively* (CH, B1, B2)*) admits a solution.*

Constructing and Solving the ILP. We show the following theorem.

Theorem 3. *Let G be a graph, let D_G be a (k,c)-extended deletion set (respectively a c-deletion set) of size at most k for G, let $\mathcal{D} = (D_H, c', \mathrm{CH})$ be a target description and let $\phi_P : D_G \to D_H$ be a locally surjective (respectively bijective) homomorphism from D_G to D_H. Then, deciding whether there is a locally surjective (respectively bijective) homomorphism that augments ϕ_P from G to any graph satisfying* CH *is fpt parameterized by $k + c + c'$.*

To prove Theorem 3, we need to show that we can construct and solve the ILP instance given in the previous section. The main ingredient for the proof of Theorem 3 is Lemma 7, which shows that we can efficiently compute the sets wSM, SM, and BM. A crucial insight for its proof is that if $(\mathrm{Ext}_G, \mathrm{Ext}_H) \in \mathrm{SM}$ (or $(\mathrm{Ext}_G, \mathrm{Ext}_H) \in \mathrm{BM}$), then Ext_G consists of only boundedly many (in terms of some function of the parameters) components, which will allow us to enumerate all possibilities for Ext_G in fpt-time. We start by showing that the set $\mathcal{T}_{D_G}(G)$ can be computed efficiently and has small size.

Lemma 6. *Let G be a graph and let D_G be a (k,c)-extended deletion set of size at most k for G. Then, $\mathcal{T}_{D_G}(G)$ has size at most $k + (|D_G| + c)2^{\binom{|D_G|+c}{2}}$ and computing $\mathcal{T}_{D_G}(G)$ and tc_G is fpt parameterized by $|D_G| + k + c$.*

Lemma 7. *Let G be a graph, let D_G be a (k,c)-extended deletion set (respectively a c-deletion set) of size at most k for G, let $\mathcal{D} = (D_H, c', \mathrm{CH})$ be a target description and let ϕ_P be a locally surjective (respectively bijective) homomorphism from D_G to D_H. Then, the sets $\mathrm{wSM} = \mathrm{wSM}(G, D_G, \mathcal{D}, \phi_P)$ and $\mathrm{SM} = \mathrm{SM}(G, D_G, \mathcal{D}, \phi_P)$ (respectively the set $\mathrm{BM} = \mathrm{BM}(G, D_G, \mathcal{D}, \phi_P)$) can be computed in fpt-time parameterized by $k + c + c'$ and $|\mathrm{SM}|$ (respectively $|\mathrm{BM}|$) is bounded by a function depending only on $k + c + c'$. Moreover, the number of variables in the equation system (CH, S1, S2, S3) (respectively (CH, B1, B2)) is bounded by a function depending only on $k + c + c'$.*

4 Applications of Our Algorithmic Framework

Here we show the main results of our paper, which are simple applications of our framework from the previous section. Our first result implies that LSHom and LBHom are fpt parameterized by the fracture number of the guest graph.

Theorem 4. LSHom *and* LBHom *are fpt parameterized by $k+c$, where k and c are such that the guest graph G has a c-deletion set of size at most k.*

Proof. Let G and H be non-empty connected graphs such that G has a c-deletion set of size at most k. Let $D_H = H[D_H^{k+c}]$. We first verify whether H has a c-deletion set of size at most k using Proposition 1. Because of Lemma 1, we can return that there is no locally surjective (and therefore also no bijective) homomorphism from G to H if this is not the case. Therefore, we can assume in what follows that H also has a c-deletion set of size at most k, which together with Lemma 2 implies that $V(D_H)$ is a $kc(k+c)$-deletion set of size at most k for H. Therefore, using Lemma 6, we can compute tc_H in fpt-time parameterized by $k+c$. This now allows us to obtain a target description $\mathcal{D} = (D_H, c', \text{CH})$ with $c' = kc(k+c)$ for H, i.e. $\mathcal{D}$ is satisfied only by the graph H, by adding the constraint $x_T = \text{tc}_H(T_H)$ to CH for every simple extension type $T_H \in \mathcal{T}^{c'}_{D_H}$; note that $\mathcal{T}^{c'}_{D_H}$ can be computed in fpt-time parameterized by $k+c$ by Lemma 6.

Because of Lemma 3, we obtain that there is a locally surjective (respectively bijective) homomorphism ϕ from G to H if and only if there is a set $D \subseteq D^{k+c}_G$ and a locally surjective (respectively bijective) homomorphism ϕ_P from $D_G = G[D]$ to D_H such that ϕ augments ϕ_P. Therefore, we can solve LSHom by checking, for every $D \subseteq D^{k+c}_G$ and every locally surjective homomorphism ϕ_P from $D_G = G[D]$ to D_H, whether there is a locally surjective homomorphism from G to H that augments ϕ_P. Note that there are at most 2^k subsets D and because of Lemma 4, we can compute the set Φ_D for every such subset in $\mathcal{O}(k^{k+2})$ time. Furthermore, due to Lemma 2, D is a $(k-|D|, c)$-extended deletion set of size at most k for G. Therefore, for every $D \subseteq D^{k+c}_G$ and $\phi_p \in \Phi_D$, we can use Theorem 3 to decide in fpt-time parameterized by $k+c$ (because $c' = kc(k+c)$), if there is a locally surjective (resp. bijective) homomorphism from G to a graph satisfying $\mathcal{D}$ that augments ϕ_P. As H is the only graph satisfying $\mathcal{D}$, we proved the theorem. □

The proof of our next theorem is similar to that of Theorem 4. The difference is that H is not given. Instead, we use Theorem 3 for a selected set of target descriptions. Each target description enforces that graphs satisfying it have to be connected and have precisely h vertices, where h is part of the input for Role Assignment. We ensure that every graph H satisfying the requirements of Role Assignment satisfies at least one of the selected target descriptions. The size of the set of considered target descriptions depends only on c and k, as it suffices to consider any small graph D_H and types of small simple extensions of D_H.

Theorem 5. Role Assignment *is fpt parameterized by $k+c$, where k and c are such that G has a c-deletion set of size at most k.*

We also obtain the following dichotomy, where the $c=1$, $k \geq 1$ case (vertex cover number case) follows from our ILP framework: we first find, in XP time, a partial mapping from a vertex cover of the host graph G to the guest graph H and then use our ILP framework to map the remaining vertices in FPT-time.

Theorem 6. *Let $c, k \geq 1$. Then* LIHom *is polynomial-time solvable on guest graphs with a c-deletion set of size at most k if either $c = 1$ and $k \geq 1$ or $c = 2$ and $k = 1$; otherwise, it is* NP*-complete.*

5 Conclusions

We aim to extend our ILP-based framework. If successful, this will then also enable us to address the parameterized complexity of other graph homomorphism variants such as quasi-covers [28] and pseudo-covers [10,12,13]. We also recall the open problem from [14]: are LBHom and LSHom in FPT when parameterized by the treewidth of the guest graph plus the maximum degree of the guest graph?

References

1. Abello, J., Fellows, M.R., Stillwell, J.: On the complexity and combinatorics of covering finite complexes. Australasian Journal of Combinatorics **4**, 103–112 (1991)
2. Angluin, D.: Local and global properties in networks of processors (extended abstract). Proc. STOC 1980 pp. 82–93 (1980)
3. Angluin, D., Gardiner, A.: Finite common coverings of pairs of regular graphs. Journal of Combinatorial Theory, Series B **30**, 184–187 (1981)
4. Biggs, N.J.: Algebraic Graph Theory. Cambridge University Press (1974)
5. Biggs, N.J.: Constructing 5-arc transitive cubic graphs. Journal of the London Mathematical Society II **26**, 193–200 (1982)
6. Bílka, O., Lidický, B., Tesar, M.: Locally injective homomorphism to the simple weight graphs. Proc. TAMC 2011, LNCS **6648**, 471–482 (2011)
7. Bodlaender, H.L.: The classification of coverings of processor networks. Journal of Parallel and Distributed Computing **6**, 166–182 (1989)
8. Bodlaender, H.L., Gilbert, J.R., Hafsteinsson, H., Kloks, T.: Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. Journal of Algorithms **18**, 238–255 (1995)
9. Bok, J., Fiala, J., Hlinený, P., Jedlicková, N., Kratochvíl, J.: Computational complexity of covering multigraphs with semi-edges: Small cases. Proc. MFCS 2021, LIPIcs **202**, 21:1–21:15 (2021)
10. Chalopin, J.: Local computations on closed unlabelled edges: The election problem and the naming problem. Proc. SOFSEM 2005, LNCS **3381**, 82–91 (2005)
11. Chalopin, J., Métivier, Y., Zielonka, W.: Local computations in graphs: The case of cellular edge local computations. Fundamenta Informaticae **74**, 85–114 (2006)
12. Chalopin, J., Paulusma, D.: Graph labelings derived from models in distributed computing: A complete complexity classification. Networks **58**, 207–231 (2011)
13. Chalopin, J., Paulusma, D.: Packing bipartite graphs with covers of complete bipartite graphs. Discrete Applied Mathematics **168**, 40–50 (2014)
14. Chaplick, S., Fiala, J., van 't Hof, P., Paulusma, D., Tesar, M.: Locally constrained homomorphisms on graphs of bounded treewidth and bounded degree. Theoretical Computer Science **590**, 86–95 (2015)
15. Chekuri, C., Rajaraman, A.: Conjunctive query containment revisited. Theoretical Computer Science **239**, 211–229 (2000)
16. Dourado, M.C.: Computing role assignments of split graphs. Theoretical Computer Science **635**, 74–84 (2016)
17. Downey, R.G., Fellows, M.R.: Fixed-parameter tractability and completeness II: on completeness for W[1]. Theoretical Computer Science **141**, 109–131 (1995)
18. Drange, P.G., Dregi, M.S., van 't Hof, P.: On the computational complexity of vertex integrity and component order connectivity. Algorithmica **76**, 1181–1202 (2016)
19. Dvorák, P., Eiben, E., Ganian, R., Knop, D., Ordyniak, S.: Solving integer linear programs with a small number of global variables and constraints. Proc. IJCAI 2017 pp. 607–613 (2017)
20. Everett, M.G., Borgatti, S.P.: Role colouring a graph. Mathematical Social Sciences **21**, 183–188 (1991)
21. Fellows, M.R., Lokshtanov, D., Misra, N., Rosamond, F.A., Saurabh, S.: Graph layout problems parameterized by vertex cover. In: ISAAC. pp. 294–305. Lecture Notes in Computer Science, Springer (2008)

22. Fiala, J., Kloks, T., Kratochvíl, J.: Fixed-parameter complexity of lambda-labelings. Discrete Applied Mathematics **113**, 59–72 (2001)
23. Fiala, J., Kratochvíl, J.: Partial covers of graphs. Discussiones Mathematicae Graph Theory **22**, 89–99 (2002)
24. Fiala, J., Kratochvíl, J.: Locally constrained graph homomorphisms - structure, complexity, and applications. Computer Science Review **2**, 97–111 (2008)
25. Fiala, J., Kratochvíl, J., Pór, A.: On the computational complexity of partial covers of theta graphs. Discrete Applied Mathematics **156**, 1143–1149 (2008)
26. Fiala, J., Paulusma, D.: A complete complexity classification of the role assignment problem. Theoretical Computer Science **349**, 67–81 (2005)
27. Fiala, J., Paulusma, D.: Comparing universal covers in polynomial time. Theory of Computing Systems **46**, 620–635 (2010)
28. Fiala, J., Tesar, M.: Dichotomy of the H-Quasi-Cover problem. Proc. CSR 2013, LNCS **7913**, 310–321 (2013)
29. Frank, A., Tardos, É.: An application of simultaneous diophantine approximation in combinatorial optimization. Combinatorica **7**(1), 49–65 (1987)
30. Freuder, E.C.: Complexity of k-tree structured constraint satisfaction problems. Proc. AAAI 1990 pp. 4–9 (1990)
31. Grohe, M.: The complexity of homomorphism and constraint satisfaction problems seen from the other side. Journal of the ACM **54**, 1:1–1:24 (2007)
32. Heggernes, P., van 't Hof, P., Paulusma, D.: Computing role assignments of proper interval graphs in polynomial time. Journal of Discrete Algorithms **14**, 173–188 (2012)
33. Hell, P., Nešetřil, J.: On the complexity of H-coloring. Journal of Combinatorial Theory, Series B **48**, 92–110 (1990)
34. Hell, P., Nešetřil, J.: Graphs and Homomorphisms. Oxford University Press (2004)
35. van 't Hof, P., Paulusma, D., van Rooij, J.M.M.: Computing role assignments of chordal graphs. Theoretical Computer Science **411**, 3601–3613 (2010)
36. Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Oper. Res. **12**(3), 415–440 (1987)
37. Kratochvíl, J.: Regular codes in regular graphs are difficult. Discrete Mathematics **133**, 191–205 (1994)
38. Kratochvíl, J., Proskurowski, A., Telle, J.A.: Covering regular graphs. Journal of Combinatorial Theory, Series B **71**, 1–16 (1997)
39. Kratochvíl, J., Proskurowski, A., Telle, J.A.: On the complexity of graph covering problems. Nordic Journal of Computing **5**, 173–195 (1998)
40. Kratochvíl, J., Telle, J.A., Tesar, M.: Computational complexity of covering three-vertex multigraphs. Theoretical Computer Science **609**, 104–117 (2016)
41. Kristiansen, P., Telle, J.A.: Generalized H-coloring of graphs. Proc. ISAAC 2000, LNCS **1969**, 456–466 (2000)
42. Kronegger, M., Ordyniak, S., Pfandler, A.: Backdoors to planning. Artif. Intell. **269**, 49–75 (2019)
43. Lenstra Jr., H.W.: Integer programming with a fixed number of variables. Math. Oper. Res. **8**(4), 538–548 (1983)
44. Lidický, B., Tesar, M.: Complexity of locally injective homomorphism to the theta graphs. Proc. IWOCA 2010, LNCS **6460**, 326–336 (2010)
45. Massey, W.S.: Algebraic Topology: An Introduction. Harcourt, Brace and World (1967)
46. Nešetřil, J.: Homomorphisms of derivative graphs. Discrete Mathematics **1**, 257–268 (1971)

47. Nešetřil, J., Ossona de Mendez, P.: Sparsity: Graphs, Structures, and Algorithms, Algorithms and Combinatorics, vol. 28. Springer (2012)
48. Okrasa, K., Rzążewski, P.: Subexponential algorithms for variants of the homomorphism problem in string graphs. Journal of Computer and System Sciences **109**, 126–144 (2020)
49. Pandey, S., Sahlot, V.: Role coloring bipartite graphs. CoRR **abs/2102.01124** (2021)
50. Pekeč, A., Roberts, F.S.: The role assignment model nearly fits most social networks. Mathematical Social Sciences **41**, 275–293 (2001)
51. Purcell, C., Rombach, M.P.: On the complexity of role colouring planar graphs, trees and cographs. Journal of Discrete Algorithms **35**, 1–8 (2015)
52. Purcell, C., Rombach, M.P.: Role colouring graphs in hereditary classes. Theoretical Computer Science **876**, 12–24 (2021)
53. Roberts, F.S., Sheng, L.: How hard is it to determine if a graph has a 2-role assignment? Networks **37**, 67–73 (2001)
54. White, D.R., Reitz, K.P.: Graph and semigroup homomorphisms on networks of relations. Social Networks **5**, 193–235 (1983)