The Complexity of Connectivity Problems in Forbidden-Transition Graphs And Edge-Colored Graphs

Thomas Bellitto

Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
tbellitto@mimuw.edu.pl

Shaohua Li

Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
S.Li@mimuw.edu.pl

Karolina Okrasa

Faculty of the Mathematics and Information Science, Warsaw University of Technology, Poland
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
k.okrasa@mini.pw.edu.pl

Marcin Pilipczuk

Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
malcin@mimuw.edu.pl

Manuel Sorge

Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
manuel.sorge@mimuw.edu.pl

Abstract

The notion of *forbidden-transition graphs* allows for a robust generalization of walks in graphs. In a forbidden-transition graph, every pair of edges incident to a common vertex is *permitted* or *forbidden*; a walk is *compatible* if all pairs of consecutive edges on the walk are permitted. Forbidden-transition graphs and related models have found applications in a variety of fields, such as routing in optical telecommunication networks, road networks, and bio-informatics.

We initiate the study of fundamental connectivity problems from the point of view of parameterized complexity, including an in-depth study of tractability with regards to various graph-width parameters. Among several results, we prove that finding a simple compatible path between given endpoints in a forbidden-transition graph is $W[1]$-hard when parameterized by the vertex-deletion distance to a linear forest (so it is also hard when parameterized by pathwidth or treewidth). On the other hand, we show an algebraic trick that yields tractability when parameterized by treewidth of finding a properly colored Hamiltonian cycle in an edge-colored graph; properly colored walks in edge-colored graphs is one of the most studied special cases of compatible walks in forbidden-transition graphs.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Graph algorithms, fixed-parameter tractability, parameterized complexity

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2020.59

Related Version A full version of this work is available on arXiv: `https://arxiv.org/abs/2009.12892`.

Funding This research is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement 714704.

1 Introduction

Graphs have proved to be an extremely useful tool to model routing problems in a very wide range of applications. However, we sometimes need to express constraints on the permitted walks that are stronger than what the standard graph model allows for. For example, in a road network, there can be a crossroad where drivers are not allowed to turn right. In this case, many walks in the underlying graph without transition restrictions would correspond to routes that a driver is not allowed to use. To overcome this limitation, Kotzig introduced forbidden-transition graphs in [31]. Let G be an undirected graph. A *transition* in G is an unordered pair of adjacent edges. Every time a walk in G uses two edges uv and vw consecutively, we say that the walk *uses the transition* $\{uv, vw\}$. A *transition system* of G is a set of transitions in G. A *forbidden-transition graph* is a tuple (G, T) of a graph G together with a transition system T of G.[1] We say that a transition is *permitted* if it is in T and it is *forbidden* otherwise. We say a walk is *compatible* with T or T*-compatible* if all the transitions it uses are permitted, that is, in T. We omit reference to T when it is clear from the context. For notational clarity, it is sometimes useful to refer to the transitions $T(v)$ of a specific vertex $v \in V(G)$, that is, $T(v) = \{\{e, f\} \in T \mid e \cap f = \{v\}\}$.

Since their introduction, forbidden-transition graphs and related models have found applications in a variety of fields, such as routing in optical telecommunication networks [2], road networks [6], and bio-informatics [15]. Problems of routing, connectivity, and robustness in those graphs have received a lot of attention but unfortunately, those problems generally turn out to be algorithmically very difficult, even on very restricted subclasses of graphs. In [36], Szeider famously proved that even determining the existence of a compatible (elementary) path between two given vertices of a forbidden-transition graph is NP-complete. Similarly, many known results about forbidden-transition graphs are proofs of NP-completeness of problems that are polynomially solvable on standard graphs (e.g. [1], [7], [16], [21], [22], [28], [29], [36]).

A very interesting specific case of compatible walks in forbidden-transition graphs are properly colored walks in edge-colored graphs. Here, a graph is given together with a coloring of its edges and we say that a walk is *properly colored* if it does not use consecutively two edges of the same color. These graphs have been introduced by Dorninger in [15] to study chromosome arrangements. They are a powerful generalization of directed graphs (see [5]) and have been studied by many authors since their introduction. The problem of properly colored Hamiltonian cycles was the first problem studied on edge-colored graphs and this problem and its variants (such as longest elementary cycle or spanning trails among many others) are especially well studied in the literature. We refer the reader to [24] or [5] for surveys on these problems and to [4], [12], [13], [25], [32] or [33] for recent developments.

Because of their expressiveness and wide range of applications, the study of forbidden-transition graphs is a fast-emerging field and has been the subject of growing attention in the past decades but we are still very far from understanding them as well as regular graphs. Our aim in this paper is to study the parameterized complexity of some known NP-complete problems, in general forbidden-transition graphs as well as in the specific case of edge-colored graphs. We specifically focus on some problems of great practical interest, such as the existence of an elementary path or the length of a shortest path between given vertices, the problem of Hamiltonian cycles, or linkage problems where we try to connect pairs of vertices

[1] Our notation rather suggests that (G, T) is a *permitted-transition graph* but we use forbidden transitions in keeping with convention in the literature.

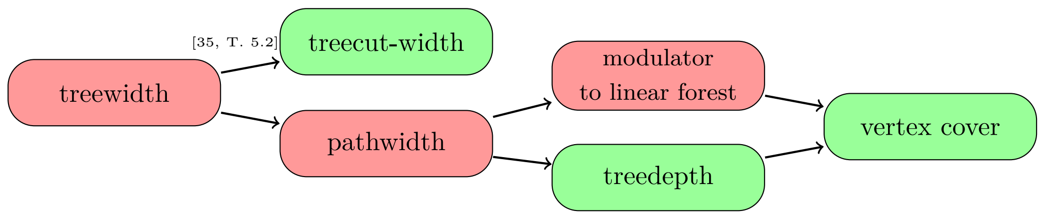

Figure 1 A hierarchy of graph-width parameters considered in this work. An arrow from a to b represents the fact that a bound on parameter b imposes a bound on parameter a, but there exist families of graphs with bounded a and unbounded b. We color a parameter a green if detecting a compatible s-t path is fixed-parameter tractable with respect to a and red if it is W[1]-hard.

by vertex- or edge-disjoint paths. A very rich toolbox already exists to study fixed-parameter tractability in standard graphs (see [14] for example) but the generalization of these concepts to forbidden-transition graphs is widely unexplored and raises many challenges that we hope to see get more attention in the future.

Our results. First, we study the problem of shortest compatible paths between two vertices s and t in a forbidden-transition graph. Recall that determining whether there exists a compatible path between s and t is known to be NP-complete [36]. A simple application of the color-coding technique shows that this problem is fixed-parameter tractable when parameterized by the length of the path. We improve upon this observation by showing that the complexity of finding a shortest compatible path from s to t is actually fixed-parameter tractable when parameterized by the length of the detour that the forbidden transitions impose. In other words, determining whether there exists a compatible path of length at most $d(s,t)+k$ where $d(s,t)$ is the length of the shortest path between s and t in the underlying graph with no forbidden transitions, is fixed-parameter tractable when parameterized by k. Our algorithm follows the main ideas of the algorithm for the Exact Detour problem by Bezáková et al. [10]. The proof is given in the full version [8].

In Section 3, we turn our attention to graph width parameters. The rich ecosystem of relevant graph-width parameters is depicted on Figure 1; see [35, 37, 19] for the corresponding boundedness and unboundedness relations on treecut-width.

First, we focus on the NP-complete problem of determining whether there exists a compatible path between s and t in a forbidden-transition graph. Since the problem is fixed-parameter tractable when parameterized by the length of the path, it is also fixed-parameter tractable when parameterized by the vertex cover number or the treedepth of the graph, as bounding the vertex cover number or the treedepth of the graph by k bounds the length of the longest simple path by $2k$ or $2^k - 1$, respectively. Our main result is a negative one: the problem becomes $W[1]$-hard if one makes one step further to the parameter *modulator to a linear forest*, i.e., the number of vertices one has to remove from the graph to turn it into a union of vertex-disjoint paths. A small tweak of the reduction shows that finding a Hamiltonian cycle is $W[1]$-hard with respect to the size of a modulator to treewidth 2. Our reduction in particular implies hardness for the parameters pathwidth and treewidth (for both the compatible path and Hamiltonian cycle problems).

On the other hand, we show that if one considers parameters based on edge cuts (as opposed to vertex cuts, like in treewidth), one can obtain nontrivial tractability results. *Treecut-width* is a width notion based on edge cuts, introduced by Wollan [37], and playing the role of treewidth in the world of the immersion relation. We prove that the problem of finding a compatible s-t path is fixed-parameter tractable when parameterized by the treecut-width of the graph. More precisely, the problem can be solved in time $k^{\mathcal{O}(k^2)} \cdot n^2 + \mathcal{O}(n^3) + \mathcal{O}\left((4^k \cdot k!)^{\mathcal{O}(3k+1)}\right) \cdot n^2$ where k denotes the treecut-width. The proof is given in the full version [8].

In the light of the hardness in general forbidden-transition graphs of detecting s-t paths, the most fundamental connectivity problem, we move to the special case of properly colored paths in edge-colored graphs. As finding a (simple) properly colored path between given endpoints in an edge-colored graph is polynomial-time solvable, we focus on the problem of finding a Hamiltonian cycle. We introduce an algebraic trick that shows that in edge-colored graphs, finding a properly colored Hamiltonian cycle is fixed-parameter tractable when parameterized by the treewidth of the graph. More specifically, the problem can be solved in time $2^{\mathcal{O}(k)} \cdot (|V(G)| + |V(\mathcal{T})| + \ell)$ where k is the treewidth, $\mathcal{T}$ is the tree of the decomposition and ℓ is the number of different colors the edges can have. The crucial property of the result is that ℓ, the number of colors, is *not* required to be bounded in the parameter and does not appear in the exponential part of the running-time bound.

After discussing graph-width notions, in Section 5, we move to the Disjoint Paths problem. In this problem, we are given a directed graph and a sequence $(s_1, t_1), (s_2, t_2), \ldots, (s_r, t_r)$ of terminal pairs; the goal is to find compatible paths $P_1, P_2, \ldots, P_r$ such that P_i starts in s_i and ends in t_i and the paths P_i are pairwise edge- or vertex-disjoint.

Observe that the problem quickly becomes hard. Even the setting of properly colored paths in edge-colored graphs generalizes directed graphs[2] and the Disjoint Paths problem for $r = 2$ is NP-hard in directed graphs [18]. Furthermore, in general graphs with transitions the case $r = 1$ is NP-hard. Hence, we focus on the specific case where the path P_i is required to be a shortest s_i-t_i path, even in the unrestricted graph. In directed graphs, a tractability result for this problem has been obtained by Bérczi and Kobayashi [9] for $r = 2$. This problem is currently a very active topic and new algorithms have been found very recently for several variants in the case $r = 2$. Polynomial algorithms have been developed by Gottschau et al. [20] and by Kobayashi and Sako [30] for undirected graphs with non-negative weighted edges and by Bang-Jensen et al. [3] in the directed unweighted case where paths do not have to be shortest but have bounded lengths. The complexity of the problem is still open for $r \geq 3$.

Thus, in this work we focus on the case $r = 2$ in directed forbidden-transition graphs. Extending the results of Bérczi and Kobayashi [9], we show that the problem remains polynomial-time solvable both in edge- and vertex-disjoint cases. An overview is presented in Section 5 and full proofs can be found in the full version of the paper.

2 Preliminaries

For each $n \in \mathbb{N}$ we use $[n]$ to denote $\{1, 2, \ldots, n\}$. Unless stated otherwise, all graphs are undirected, without self-loops and parallel edges.

Let G be an undirected graph. By $V(G)$ and $E(G)$ we denote the vertex and edge set of G, respectively. For each $v \in V(G)$ we denote by $E_G(v)$ the set of edges in G that are incident with v in G. We omit the subscript G if it is clear from the context. A *walk* in G is a sequence $(v_1, e_1, v_2, e_2, \ldots, e_\ell, v_{\ell+1})$ where v_is are vertices of G, e_is are edges of G, and for every $1 \leq i \leq \ell$, the vertices v_i and v_{i+1} are the two endpoints of the edge e_i. A walk is *closed* if its first vertex is also its last vertex. The *length* of a walk W equals ℓ, the number of edges in W. A *path* is a walk in which no vertex occurs twice, a *cycle* is a closed walk in which no vertex occurs twice except the first and last vertex.

[2] Consider the reduction that adds an in-neighbor to each s_i and an out-neighbor to each t_i, replaces each terminal by the corresponding in- or out-neighbor and then replaces each directed edge e with two undirected edges with two colors according to the direction of e.

For a graph G, a *tree decomposition* of G is a pair $(\mathcal{T}, \beta)$ where $\mathcal{T}$ is a tree and $\beta : V(\mathcal{T}) \to 2^{V(G)}$ such that the following holds: (i) for every $v \in V(G)$, the set $\{t \in V(\mathcal{T}) \mid v \in \beta(t)\}$ induces a nonempty connected subtree of $\mathcal{T}$, and (ii) for every $uv \in E(G)$, there exists $t \in V(\mathcal{T})$ with $u, v \in \beta(t)$. That is, the function β assigns to every node $t \in V(\mathcal{T})$ a subset $\beta(t) \subseteq V(G)$, often called a *bag*. It is often convenient to root $\mathcal{T}$ at an arbitrary vertex. The width of a tree decomposition $(\mathcal{T}, \beta)$ equals $\max_{t \in V(\mathcal{T})} |\beta(t)| - 1$, and the treewidth of a graph is the minimum possible width of its tree decomposition.

3 Graph Width Parameters: Modulator to Linear Forest

Let G be an undirected graph. A *modulator to a linear forest* of G is a vertex subset $S \subseteq V(G)$ such that $G - S$ is a disjoint union of paths. The *distance k of G to a linear forest* is the minimum size, k, of a modulator to a linear forest. Note that the distance to a linear forest upper bounds the size of a minimum feedback-vertex set and the treewidth and hence W[1]-hardness for these two parameters is implied by W[1]-hardness for k. A *modulator to treewidth two* of G is a vertex subset $S \subseteq V(G)$ such that $G - S$ has treewidth at most two. The *distance of G to treewidth two* is the minimum size of a modulator to treewidth two. Analogously, the distance to treewidth two upper bounds the treewidth and hence W[1]-hardness for treewidth is implied by W[1]-hardness for the distance to treewidth two.

In this section, we first show that finding long paths or cycles is W[1]-hard with respect to the distance k to a linear forest. Moreover, assuming the Exponential Time Hypothesis (ETH), no $f(k) \cdot n^{o(k/\log k)}$-time algorithm can exist. Informally, the ETH states that 3-SAT on n-variable formulas cannot be solved in $2^{o(n)}$ time, see [27, 26]. We obtain the following.

▶ **Theorem 1.** *Let (G, T) be forbidden-transition graph and s, t two vertices in G. Let ℓ be a positive integer and let k be the distance of G to a linear forest. For each of the following, it is W[1]-hard with respect to k to decide and, moreover, an $f(k) \cdot n^{o(k/\log k)}$-time decision algorithm contradicts the ETH:*

- **(i)** *whether G contains a compatible s-t path,*
- **(ii)** *whether G contains a compatible s-t path of length at least ℓ,*
- **(iii)** *whether G contains a compatible cycle, and*
- **(iv)** *whether G contains a compatible cycle of length at least ℓ.*

Proof. We first give a reduction to prove hardness of Item i. Observe that Item ii follows from Item i. We then modify the construction to obtain Item iii and Item iv.

Our reduction is from the Partitioned Subgraph Isomorphism (PSI) problem. Herein, we are given two graphs G and H, where $V(H) = [n_H]$ for some positive integer n_H, and a vertex coloring $\mathsf{col}\colon V(G) \to V(H)$ of the vertices of G with colors that one-to-one correspond to the vertices of H. Moreover, each vertex of H is incident with at least one edge and for each edge $\{u, v\} \in E(G)$ we have $\mathsf{col}(u) \neq \mathsf{col}(v)$. We want to decide whether H is isomorphic to a subgraph of G while respecting the colors, that is, whether there is an injective mapping $\phi\colon V(H) \to V(G)$ such that for all $u \in V(H)$ we have $\mathsf{col}(\phi(u)) = u$ and for all $\{u, v\} \in E(H)$ we have $\{\phi(u), \phi(v)\} \in E(G)$. In that case, we also say that ϕ is a *subgraph isomorphism* from H into G. In the following we let $m_H = |E(H)|$. Observe that $n_H \leq 2m_H$ since each vertex of H is incident with at least one edge. Since PSI contains Multicolored Clique [17] as a special case, PSI is W[1]-hard with respect to m_H. Moreover, Marx [34, Corollary 6.3] observed that an $f(m_H) \cdot n^{o(m_H/\log m_H)}$-time algorithm for PSI would contradict the ETH.

Our construction works as follows: we first build a path from s to a vertex t_1. This path is the concatenation of n_H subpaths $P^1, \ldots, P^{n_H}$ where each subpath is associated with a vertex of H. The subpath P^i contains a vertex for each edge of G incident to a vertex

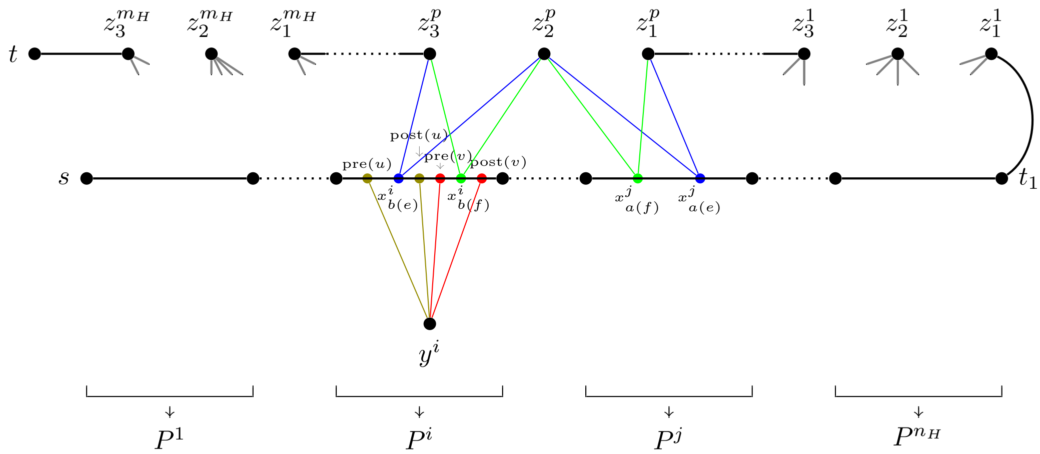

Figure 2 An illustration of our construction. We use colors to denote edges that have to be used consecutively because of the set of permitted transitions. For example, the two dark yellow edges correspond to a vertex u of G in the vertex-selection gadget for vertex i of H and the four blue edges correspond to an edge e of G in the edge-selection gadget for the pth edge of H.

colored i. We then use an extra vertex and an appropriate transition system so that one can choose any vertex v of G with color i and connect the endpoints of P^i with a compatible path that skips the vertices of P^i that denote an edge adjacent to v. This comes down to choosing $\phi(i) = v$. Finally, we connect t_1 to t by a sequence of gadgets each associated with an edge of H. Choosing a path through a gadget comes down to mapping an edge uv of H to an edge wx of G. Our transition system then requires the path in the gadget to visit the two vertices of P that denote the edge wx, which can only be done without repeating vertices if those vertices have been skipped between s and t_1. This means that the endpoints of wx have to be the vertices we chose as $\phi(u)$ and $\phi(v)$. By ensuring that there is an edge between $\phi(u)$ and $\phi(v)$, we prove that ϕ is a subgraph isomorphism.

▶ **Construction 2.** Let (G, H, col) be an instance of PSI, where $V(H) = [n_H]$. For each $i \in [n_H]$ define $V_i = \{v \in V(G) \mid \mathsf{col}(v) = i\}$. For each $i \in [n_H]$ define $E_i = \{e \in E(G) \mid \exists u \in V_i \colon u \in e\}$. We construct a forbidden-transition graph $(G^\star, T)$ as follows, see Figure 2 for an illustration. We begin with $G^\star$ being empty. We will specify T by giving the permitted-transition sets $T(v)$ for the individual vertices $v \in V(G^\star)$. Below, we specify $T(v)$ only for a subset of $V(G^\star)$. For all the remaining vertices v, we put $T(v) = \binom{E(v)}{2}$ (recall that $E(v)$ is the set of edges in $G^\star$ that are incident with v). Introduce new vertices s, t, t_1 into $G^\star$. We construct the vertex-selection gadgets as follows.

Introduce a path P from s to t_1 into $G^\star$; we specify the number of vertices on P indirectly below. For each internal vertex $v \in V(P)$ put $T(v) = \{\{\{u, v\}, \{v, w\}\}\}$ where u and w are the neighbors of v in P. Additional edges and transitions for the vertices on P will be introduced below. Partition P into n_H disjoint paths $P^1, \ldots, P^{n_H}$; we specify the number of vertices in each of these paths in the next step.

For each $i \in [n_H]$, proceed as follows. Let $(e^i_a)_{a \in [r_i]}$ be an ordering of E_i such that, for each $v \in V_i$, the edges in $E(v)$ form a segment in (e^i_a) (observe that such an ordering exists since the endpoints of each edge in $E(G)$ have two different colors). Set the number of vertices in P^i to $r_i + 4$. For each $a \in [r_i]$ denote the $a + 2$-th vertex on P^i by x^i_a. We say that vertex x^i_a *corresponds* to the edge e^i_a of G. (We keep the first two and last two vertices of P^i unnamed.)

Next, introduce a vertex y^i and for each vertex $v \in V_i$ proceed as follows. Let $\mathsf{pre}(v)$ be the vertex in P^i that directly precedes on P^i the first vertex corresponding to an edge in $E_G(v)$. Similarly, let $\mathsf{post}(v)$ be the vertex in P^i that directly succeeds on P^i the last vertex corresponding to an edge in $E_G(v)$. For later, it is useful to observe that all vertices on P^i strictly between $\mathsf{pre}(v)$ and $\mathsf{post}(v)$ correspond to edges in $E_G(v)$. Now add the edges $\{\mathsf{pre}(v), y^i\}$ and $\{y^i, \mathsf{post}(v)\}$ to $G^\star$ and the transition $\{\{\mathsf{pre}(v), y^i\}, \{y^i, \mathsf{post}(v)\}\}$ to $T(y^i)$. Moreover, add the transition $\{\{u, \mathsf{pre}(v)\}, \{\mathsf{pre}(v), y^i\}\}$ to $T(\mathsf{pre}(v))$ where u is the vertex on P^i preceding $\mathsf{pre}(v)$ (if any), and add the transition $\{\{y^i, \mathsf{post}(v)\}, \{\mathsf{post}(v), w\}\}$ to $T(\mathsf{post}(v))$, where w is the vertex on P^i succeeding $\mathsf{post}(v)$. This finishes the construction of the vertex-selection gadgets, but further edges and transitions may be introduced later to the vertices of P.

We now construct the edge-verification gadgets. Let $(e_1, \ldots, e_{m_H})$ be an arbitrary ordering of the edges in $E(H)$. For each $p \in [m_H]$ proceed as follows. Introduce three vertices z_1^p, z_2^p, and z_3^p. Let $\{i, j\} = e_p$ where $i > j$. For each edge $e \in E_i \cap E_j$ of G proceed as follows. Let $a(e)$ be the index of e in the ordering (e_a^i) defined for vertex i when constructing the vertex-selection gadget. Similarly, let $b(e)$ be the index of e in the ordering (e_a^j) defined for j. Introduce the following edges into $G^\star$:

$$\{z_1^p, x_{a(e)}^i\}, \qquad \{x_{a(e)}^i, z_2^p\}, \qquad \{z_2^p, x_{b(e)}^j\}, \qquad \text{and} \qquad \{x_{b(e)}^j, z_3^p\}.$$

Furthermore, add the following transitions:
- $\{\{z_1^p, x_{a(e)}^i\}, \{x_{a(e)}^i, z_2^p\}\}$ to $T(x_{a(e)}^i)$,
- $\{\{z_2^p, x_{b(e)}^j\}, \{x_{b(e)}^j, z_3^p\}\}$ to $T(x_{b(e)}^j)$, and
- $\{\{x_{a(e)}^i, z_2^p\}, \{z_2^p, x_{b(e)}^j\}\}$ to $T(z_2^p)$.

To conclude the construction of the edge-verification gadgets, add the following edges: $\{t_1, z_1^1\}$; for each $p \in [m_H - 1]$ the edge $\{z_3^p, z_1^{p+1}\}$; and $\{z_3^{m_H}, t\}$. This concludes the construction of $G^\star$ and $T(G^\star)$ (recall that for vertices v for which we left $T(v)$ unspecified we put $T(v) = \binom{E(v)}{2}$). ⌟

Observe that Construction 2 can be carried out in polynomial time. We claim that the distance to linear forest of $G^\star$ is at most $n_H + 3m_H \leq 5m_H$. Let $Z = \{z_1^p, z_2^p, z_3^p \mid p \in [m_H]\}$ and $Y = \{y^i \mid i \in [n_H]\}$. Note that the only vertices in $G^\star - (V(P) \cup \{t\})$ are in $Y \cup Z$. Moreover, no edges between two vertices on P have been introduced into $G^\star$. Thus, $Y \cup Z$ is a modulator to a linear forest and $G^\star$ has distance at most $5m_H$ to a linear forest. If Construction 2 is correct, by the properties of PSI it thus follows that deciding whether a graph has a compatible s-t path is W[1]-hard with respect to the distance, k, to a linear forest, and that an $f(k)n^{o(k/\log k)}$-time decision algorithm contradicts the ETH. We next show the correctness of Construction 2.

Correctness. We now show that $(G^\star, T)$ contains a compatible s-t path if and only if there is a subgraph isomorphism from H into G.

Suppose first that there is a subgraph isomorphism ϕ from H into G. Construct an s-t walk $P^\star$ by concatenating the following path segments (observe while reading the construction, that $P^\star$ is compatible):
1. The subpath on P from s to $\mathsf{pre}(\phi(1))$.
2. The three vertices $\mathsf{pre}(\phi(1))$, y^1, $\mathsf{post}(\phi(1))$.
3. For each $i = 2, 3, \ldots, n_H$ take:
   a. The subpath on P from $\mathsf{post}(\phi(i-1))$ to $\mathsf{pre}(\phi(i))$.
   b. The three vertices $\mathsf{pre}(\phi(i))$, y^i, $\mathsf{post}(\phi(i))$.

4. The subpath on P from $\mathsf{post}(\phi(n_H))$ to t_1.
5. For each $p = 1, 2, \ldots, m_H$, let e_p be the pth edge of H according to the ordering of $E(H)$ fixed in Construction 2, let $e_p = \{i, j\}$, where $i > j$, let $e = \{\phi(i), \phi(j)\}$, let $a(e)$ be the index of e in the ordering (e^i_a) and $b(e)$ the index of e in the ordering (e^j_a). Take the vertices z^p_1, $x^i_{a(e)}$, z^p_2, $x^j_{b(e)}$, and z^p_3.
6. The edge $\{z^{m_H}_3, t\}$.

This concludes the construction of $P^\star$. Suppose, for a contradiction, that $P^\star$ is not a path, that is, there is a vertex v in $G^\star$ which is contained twice in $P^\star$. Since $V(G)$ is partitioned into $V(P)$, Y, Z, and $\{t\}$ and each vertex of Y and Z occurs only once in the definition of $P^\star$, we have $v \in V(P)$. Since each segment in the construction of $P^\star$ is a path, the two occurrences must be in different segments. Observe that all segments of $P^\star$ in steps 1 to 4 that are contained in $V(P)$ are pairwise disjoint subpaths of P. Furthermore, all vertices in $V(P)$ used in the segments constructed in step 5 are pairwise distinct. Thus, there is one occurrence of v in steps 1 to 4, and one in step 5. Moreover, v corresponds to some edge e of G. However, according to the steps 1 to 4, vertex v corresponds to some edge which is not incident to a vertex in $\phi(V(H))$ and, according to step 5, vertex v corresponds to some edge which is incident to a vertex in $\phi(V(H))$, a contradiction. Thus, indeed, $P^\star$ is a compatible s-t path, as required.

Now suppose that $(G^\star, T)$ contains a compatible s-t path $P^\star$. Obviously, $P^\star$ starts with a subsegment of P. By construction of the transitions on vertices on P, at each internal vertex of P, the path $P^\star$ may either continue on P or go to some vertex of Y. Moreover, whenever $P^\star$ traverses a vertex of Y, it immediately returns to P with the next vertex. Path $P^\star$ hence begins with a segment which starts at s, alternatingly contains a sequence of vertices on P and a vertex of Y, and ends at t_1. Let $Y' = Y \cap V(P^\star)$ (we show below that $Y' = Y$). Observe that, for each vertex $y^i \in Y'$, there exists $v \in V_i$ such that $P^\star$ contains the edges $\{\mathsf{pre}(v), y^i\}$ and $\{y^i, \mathsf{post}(v)\}$, by the transitions defined for y^i. Define a (partial) function $\phi\colon V(H) \to V(G)$ as follows. For each $i \in [n_H]$ such that $y_i \in Y'$ put $\phi(i) = v$, where v is as defined above. For later, put $P^\star_1$ to be the segment of $P^\star$ from s to t_1 and put $P^\star_2$ to be the segment of $P^\star$ from t_1 to t. Observe that $P^\star_1$ contains precisely all vertices of P except those that correspond to edges in G which are incident to the vertices of $\phi(Y')$.

To show that ϕ is total and that ϕ is a subgraph isomorphism from H into G, we now argue that $P^\star_2$ contains z^p_2 for each $p \in [m_H]$. Since $P^\star_2$ is a path, it starts with the edge $\{t_1, z^1_1\}$. Moreover, by the edges and transitions of the vertices z^p_1, x^i_a, z^p_2, and z^p_3 ($p \in [m_H]$, $i \in [n_H]$, $a \in \mathbb{N}$), whenever $P^\star_2$ traverses a vertex z^p_1, $p \in [m_H]$, it next traverses some vertex x^i_a, then the vertex z^p_2, some vertex x^j_b, and the vertex z^p_3 for some $i, j \in [n_H]$ where $i > j$. Moreover, after z^p_3, path $P^\star_2$ traverses either z^{p+1}_1 (if $p < m_H$) or t (if $p = m_H$) because the only other vertices that $P^\star_2$ may traverse after z^p_3 are vertices $x^j_{a'}$ and, by their transitions, $P^\star_2$ would then have to contain z^p_2 a second time. Concluding, $P^\star_2$ contains z^p_2 for each $p \in [m_H]$.

Let $p \in [m_H]$ and let e_p be the pth edge of H according to the ordering of $E(H)$ fixed in Construction 2. Let $e_p = \{i, j\}$ with $i > j$. As argued above $P^\star_2$ contains z^p_2. Let x^i_a and x^j_b be the vertices that $P^\star_2$ traverses before and after z^p_2. By the transitions of z^p_2, the vertices x^i_a and x^j_b correspond to the same edge of G. Denote this edge by f_p. We now show that the edges f_p, $p \in [m_H]$, ensures that ϕ is total and a subgraph isomorphism.

First, to see that ϕ is total, recall that each vertex $i \in V(H)$ is incident with at least one edge. Say i is incident with edge e_p. Let x^i_a be the vertex that corresponds to an edge in G incident with a vertex of color i and that led to the definition of f_p, that is, $P^\star_2$ traverses x^i_a before or after z^2_p. Now recall that $P^\star_1$ contains all vertices of P except those that correspond to the edges incident with vertices in $\phi(Y')$. Since $P^\star_1$ and $P^\star_2$ are internally vertex-disjoint, $i \in Y'$. It thus follows that ϕ is total.

To see that ϕ is a subgraph isomorphism, take any edge $e_p \in E(H)$. Consider the edge f_p and the two vertices x_a^i and x_b^j that led to the definition of f_p, that is, x_a^i and x_b^j are traversed either before or after z_p^2. By the construction of the edges of z_p^2, we have $e_p = \{i, j\}$. We again use the property that $P_1^\star$ contains all vertices of P except those that correspond to the edges incident with vertices in $\phi(Y')$. Since x_a^i and x_b^j are not in $P_1^\star$, they correspond to an edge incident with both $\phi(i)$ and $\phi(j)$, that is, $f_p = \{\phi(i), \phi(j)\}$. Thus, indeed ϕ is a subgraph isomorphism, as required. This concludes the proof of Theorem 1 Item i. Observe that Item ii is implied by Item i. The remaining parts are proved below.

Cycles. We now adapt Construction 2 to obtain Theorem 1 Item iii. To this end, we simply add the edge $\{s, t\}$ to $G^\star$ (and update the permitted transitions of s and t to allow for combining $\{s, t\}$ with every other edge). Call the resulting graph $G_C^\star$. Observe that $G_C^\star - (Y \cup Z)$ is a path with vertex set $V(P) \cup \{t\}$, and hence $G_C^\star$ has distance to a linear forest at most $5m_H$.

We claim that there is a compatible s-t path in $G^\star$ if and only if there is a compatible cycle in $G_C^\star$. The forward direction is trivial. For the backward direction, let $C^\star$ be a compatible cycle in $G_C^\star$. We show that $C^\star$ contains $\{s, t\}$. For a contradiction, assume it does not. Thus, $C^\star$ is a cycle in $G^\star$. By the transitions of the vertices in P, cycle $C^\star$ does not contain an edge in P nor does it contain a vertex in Y. Let $G_1^\star = (V(G^\star) \setminus Y, E(G^\star) \setminus E(P))$ and observe that $C^\star$ is a cycle in $G_1^\star$. Observe that $V(P)$ is an independent set in $G_1^\star$. Thus each cycle (not necessarily compatible) can be written as $z_2^p, x_a^i, z_1^p, x_b^i, z_2^p$ or $z_2^p, x_a^i, z_3^p, x_b^i, z_2^p$ for the corresponding values of p, i, a, and b. However, by the transitions of z_2^p, none of these cycles is compatible, a contradiction. Thus, $C^\star$ contains $\{s, t\}$. Hence, removing $\{s, t\}$ from $C^\star$ gives an s-t path in $G^\star$, concluding the proof. Observe that Item iv follows from Item iii. ◀

We now adapt Construction 2 to prove that it is W[1]-hard with respect to the distance to treewidth two to check whether there is a compatible Hamiltonian cycle.

▶ **Theorem 3.** *Let G be a graph and k' its distance to treewidth two. It is W[1]-hard with respect to k' to decide whether G contains a compatible Hamiltonian cycle and, moreover, an $f(k') \cdot n^{o(k'/\log k')}$-time decision algorithm contradicts the ETH.*

Proof. To prove this theorem, we use Construction 2 and add a gadget that allows an s-t path in $G^\star$ to collect all so-far untraversed vertices, wherein we use transitions to not disturb the structure of $G^\star$. The basic observation that we use is that the path $P^\star$ we have constructed in the correctness proof for detecting s-t paths above contains all vertices of $G^\star$ except segments of the path P. The idea now is to add a path Q which runs "parallel" to P (like a skewed ladder) and which starts after t and ends in s. Using transitions we allow the solution in each vertex v of Q to either continue to the next vertex of Q or to traverse the vertex parallel to v on P and then immediately return to the next vertex after v on Q. This allows the solution to traverse all vertices it missed on the traversal from s to t. Since Q is parallel to P, removing $Y \cup Z$ will result in a graph of treewidth two.

The formal construction is as follows. Construct a forbidden-transition graph $(G_1^\star, T_1)$ from $(G^\star, T)$ by initially putting $(G_1^\star, T_1) = (G^\star, T)$. Let $n = |V(P)| - 2$. Add a path Q consisting of $n + 1$ vertices to $G_1^\star$ and identify the first and last vertex of Q with s and t, respectively. Let $v_1, v_2, \ldots, v_n$ be the internal vertices of P and $t = u_1, u_2, \ldots, u_{n+1} = s$ the vertices of Q. For each $i \in [n]$ proceed as follows. Add the edges $\{u_i, v_i\}$, and $\{v_i, u_{i+1}\}$. Then, update the transition system T_1 by adding the transitions $\{\{u_i, v_i\}, \{v_i, u_{i+1}\}\}$ to $T_1(v_i)$. This finishes the construction of $G_1^\star$ and its transition system (as before, for all vertices with unspecified transition systems we allow all transitions).

Let $\tilde{G_1^\star} = G_1^\star - (Y \cup Z \cup \{s,t\})$. We claim that $\tilde{G_1^\star}$ has treewidth two. Observe that this graph consists only of the vertices in P and Q except for s and t. Now observe that, by the definition of the edges between P and Q, the following bags give a path decomposition for $\tilde{G_1^\star}$ of width two. Note that we specify a bag containing $t = u_1$ for easier notation:

$$\{u_1, u_2, v_1\}, \{u_2, v_1, v_2\}, \ldots, \{u_i, u_{i+1}, v_i\}, \{u_{i+1}, v_i, v_{i+1}\}, \ldots, \{u_{n-1}, u_n, v_{n-1}\}, \{u_n, v_{n-1}, v_n\}.$$

Thus $Y \cup Z \cup \{s,t\}$ is a modulator of $G_1^\star$ to treewidth two, meaning that $G_1^\star$ has distance to treewidth two at most $5m_H + 2$, as required.

Finally, one can show that $(G_1^\star, T_1)$ contains a compatible Hamiltonian cycle if and only if $(G^\star, T)$ contains a compatible s-t path; the details are given in the full version [8]. ◀

4 Graph Width Parameters: Edge-Colored Graphs and Treewidth

Our main result on properly colored paths and cycles in edge-colored graphs of bounded treewidth is as follows:

▶ **Theorem 4.** *Given an undirected graph G with an edge coloring $\lambda : E(G) \to [\ell]$ and a tree decomposition $(\mathcal{T}, \beta)$ of G of width less than k, one can verify whether G admits a properly colored Hamiltonian Cycle in deterministic time $2^{\mathcal{O}(k)} \cdot \mathcal{O}(|V(G)| + |V(\mathcal{T})| + \ell)$.*

The main highlight of Theorem 4 is the lack of the dependency on ℓ in the exponential part of the running time bound. For sake of simplicity, we do not analyze in detail the base of the exponent in the running time bound of the algorithm of Theorem 4.

To show the main novel ideas of the proof, in this extended abstract we sketch a simpler algorithm with $2^{\mathcal{O}(k \log k)}$ dependency on the parameter in the running time bound. The complete proof of Theorem 4 can be found in the full version of the paper [8].

To this end, consider the naive dynamic programming algorithm for Hamiltonian Cycle in graphs of bounded treewidth, as described e.g. by Ziobro and Pilipczuk [39]. Let G be a graph and $(\mathcal{T}, \beta)$ a tree decomposition of G of width less than k. We treat $\mathcal{T}$ as a rooted tree. Every bag $t \in V(\mathcal{T})$ induces a separation of order at most k between the vertices in bags in the descendants of t and the rest of the graph.

More formally, for every $t \in V(\mathcal{T})$, let G_t be the subgraph of G induced by all vertices appearing in bags $\beta(s)$ for s being descendants of t in $\mathcal{T}$ (including t itself), except for the edges with both endpoints in $\beta(t)$, and $\bar{G}_t$ be the subgraph of G induced by $\beta(t)$ and all vertices of G appearing in bags $\beta(s)$ for s not being a descendant of t. Then, $V(G_t) \cap V(\bar{G}_t) = \beta(t)$ and $E(G) = E(G_t) \uplus E(\bar{G}_t)$, and there are no edges between $V(G_t) \setminus \beta(t)$ and $V(\bar{G}_t) \setminus \beta(t)$.

For a node $t \in V(\mathcal{T})$, a *partial solution* is a family of vertex-disjoint paths in G_t with both endpoints in $\beta(t)$ covering all vertices of G_t (here we allow paths of length 0 consisting of a single vertex in $\beta(t)$). The *trace* of a partial solution P consists of

- a function $f : \beta(t) \to \{0, 1, 2\}$, where $f(v)$ is the degree in P of a vertex in $\beta(t)$, and
- a matching M on $f^{-1}(1)$, pairing up the endpoints of the paths from P.

The crucial observation is that two partial solutions with the same trace are indistinguishable from the point of view of completing them into a Hamiltonian cycle with a set of paths in $\bar{G}_t$, so, to check if G admits a Hamiltonian cycle, the algorithm may only keep a set $\mathcal{A}(t)$ of these traces for which there exists a partial solution in G_t. Observe that there are $2^{\mathcal{O}(k \log k)}$ possible traces. Together with slightly tedious, but straightforward operations between the nodes of the tree decomposition, this gives $2^{\mathcal{O}(k \log k)} \cdot \mathcal{O}(|V(G)| + |V(\mathcal{T})|)$-time algorithm.

Let us now move to the regime of properly colored Hamiltonian cycles. The graph G comes with an edge coloring $\lambda : E(G) \to [\ell]$ and we ask whether G admits a properly colored Hamiltonian cycle. While the notion of a partial solution remains the same, to maintain the property that two partial solutions with the same trace are equivalent, the trace needs to be augmented with the following extra information:

- a function $\zeta : f^{-1}(1) \to [\ell]$, where $\zeta(v)$ is the color of the unique edge of P incident with v.

If we follow the outline of the naive dynamic programming algorithm, we obtain an algorithm with running time $2^{\mathcal{O}(k(\log k+\log \ell))} \cdot \mathcal{O}(|V(G)| + |V(\mathcal{T})| + \ell)$, as there are up to ℓ^k choices for the function ζ.

Fix a node t and let $\mathcal{A}(t)$ be the family of all traces of partial solutions in G_t. For a fixed function f and a matching M let $\mathcal{A}(t, f, M) = \{\zeta \mid (f, M, \zeta) \in \mathcal{A}(t)\}$. The crux of our approach lies in the following lemma:

▶ **Lemma 5.** *Fix a node $t \in V(\mathcal{T})$, a function $f : \beta(t) \to \{0, 1, 2\}$, and a matching M on $f^{-1}(1)$. Let $Z = f^{-1}(1)$. Then one can in time polynomial in $|\mathcal{A}(t, f, M)|$ and $2^{|Z|}$ find a subfamily $\mathcal{A}' \subseteq \mathcal{A}(t, f, M)$ of size at most $2^{|Z|}$ such that if there exists a trace (f, M, ζ) with $\zeta \in \mathcal{A}(t, f, M)$ such that a partial solution with this trace can be completed to a properly colored Hamiltonian cycle, then there exists a trace (f, M, ζ') with $\zeta' \in \mathcal{A}'$ such that a partial solution with this trace can also be completed to a properly colored Hamiltonian cycle.*

If we augment the naive algorithm to shrink every $\mathcal{A}(t, f, M)$ using Lemma 5, we obtain a bound of $2^{\mathcal{O}(k \log k)}$ on the size of the shrunk set $\mathcal{A}'(t)$, leading to $2^{\mathcal{O}(k \log k)} \cdot \mathcal{O}(|V(G)| + |V(\mathcal{T})| + \ell)$-time algorithm, as promised.

Proof of Lemma 5. If $|\mathcal{A}(t, f, M)| \leq 2^{|Z|}$, then there is nothing to do, so assume otherwise.

Let $\mathbb{F}$ be a finite field of size larger than ℓ. Let us identify the color range $[\ell]$ with ℓ non-zero elements of $\mathbb{F}$. Henceforth we assume that λ and all functions of $\mathcal{A}(t, f, M)$ have values in $\mathbb{F} \setminus \{0\}$.

For a function $\zeta : Z \to \mathbb{F}$ we define a function π_ζ that takes a function $\zeta' : Z \to \mathbb{F}$ as an argument and is defined as

$$\pi_\zeta(\zeta') = \prod_{z \in Z} (\zeta(z) - \zeta'(z)).$$

Then, $\pi_\zeta(\zeta') = 0$ if and only if there exists $v \in Z$ with $\zeta(z) = \zeta'(z)$. The function π_ζ can be interpreted as a multilinear polynomial of degree $|Z|$ with $|Z|$ variables, $(x_z)_{z\in Z}$, being the values of the argument ζ'. For every function $\zeta : Z \to \mathbb{F}$, let v_ζ be a vector over $\mathbb{F}$ of length $2^{|Z|}$, indexed by multilinear monomials over $(x_z)_{z\in Z}$ with the coefficients of π_ζ.

Let P_0 be a partial solution with a trace (f, M, ζ_0), $\zeta_0 \in \mathcal{A}(t, f, M)$, such that P_0 can be extended to a properly colored Hamiltonian cycle C and define $Q := E(C) \setminus E(P_0) \subseteq E(\bar{G}_t)$. For every $z \in Z$, the vertex z is incident with exactly one edge of Q; let $\zeta_Q(z)$ be the color of this edge. Since C is properly colored, $\pi_{\zeta_0}(\zeta_Q) \neq 0$.

Let P be a partial solution with a trace (f, M, ζ), $\zeta \in \mathcal{A}(t, f, M)$. Then, $P \cup Q$ is a Hamiltonian cycle, but not necessarily properly colored. Furthermore, it is properly colored if and only if $\pi_\zeta(\zeta_Q) \neq 0$.

Let w_Q be a vector over $\mathbb{F}$ of length $2^{|Z|}$, indexed again by multilinear monomials over $(x_z)_{z\in Z}$, with a value at $I \subseteq Z$ equal to $\prod_{z\in I} \zeta_Q(z)$. Then, $\pi_\zeta(\zeta_Q) = v_\zeta \cdot w_Q$.

Consequently, it suffices to select $\mathcal{A}' \subseteq \mathcal{A}(t, f, M)$ such that $\{v_\zeta \mid \zeta \in \mathcal{A}'\}$ spans the same subspace as $\{v_\zeta \mid \zeta \in \mathcal{A}(t, f, M)\}$. As the vectors v_ζ have dimension $2^{|Z|}$, such $\mathcal{A}'$ of size at most $2^{|Z|}$ can be found by Gaussian elimination. ◀

The proof above contains most of the novel ideas to get Theorem 4. To obtain the promised running-time bound one needs to merge the ideas of the proof of Lemma 5 with the so-called *rank-based approach* [11]. The details are given in the full version of the paper [8].

5 Two Disjoint Shortest Paths

Given a directed graph $G = (V, E)$, a length function $w : E \to \mathbb{R}_{\geq 0}$ and two pairs of vertices (s_1, t_1),(s_2, t_2) in G, the Directed Two Disjoint Shortest Paths Problem (2-DSPP) asks to find two disjoint paths P_1 and P_2 in G (vertex-disjoint or edge-disjoint, depending on the variant) such that P_i is a shortest path from s_i to t_i. Bérczi and Kobayashi showed that this problem is polynomial-time solvable assuming that every dicycle has positive length, in contrast to the NP-hardness of the general 2-DSPP problem [9].

Suppose that our instance graph is now given with a prescribed transition system $T = \{T(v) \mid v \in V(G)\}$. A natural generalization of the previous problem is finding two disjoint (vertex-disjoint or edge-disjoint) paths P_1 and P_2 in G such that for $i = 1, 2$ the path P_i is a shortest s_i-t_i path even in the associated graph G with no transition restrictions, and is also T-compatible. We define this problem to be 2-DSPP with transition restrictions. A reasonable question to ask is whether this generalization remains polynomial-time solvable, assuming that every dicycle has positive length, and we give a positive answer to this question.

▶ **Theorem 6.** *If the length of every directed cycle is positive, 2-DSPP with transition restrictions (both the edge-disjoint and vertex-disjoint variants) is solvable in polynomial time.*

Roughly speaking, we show that transition restrictions are not a barrier for using the strategy of Bérczi and Kobayashi [9].

For the edge-disjoint case, we follow their method, which reduces the problem of Edge Disjoint 2-DSPP to finding a path in a graph $\mathcal{G}$ constructed from the input graph G. The key observation of the algorithm from [9] is that this problem can be further reduced to finding disjoint paths between given pairs of vertices in an acyclic graph. Although finding a T-compatible path between a given pair of vertices in a general graph with a given transition system T is NP-hard [36], we show that finding a path and even two-disjoint paths between given pairs of vertices in a directed acyclic graph with a given transition system is polynomial-time solvable. Based on that, we show that we need to delete edges of $\mathcal{G}$ which correspond to forbidden transitions of G with respect to T and it suffices to find the path in the remaining subgraph of $\mathcal{G}$. We give the details in the full version of the paper [8].

For the vertex-disjoint case, however, we cannot use their method directly since part of the information of transitions will be lost during the procedure. To keep the full information of transitions, we need to introduce parallel edges between two copies v^- and v^+ of every vertex v. For each incoming edge of v, there exists a private parallel edge that "remembers" which outgoing edges can be reached from that incoming edge. We use a strategy similar to the edge-disjoint case but we need to be careful to make sure that the two paths are not only edge-disjoint but also vertex-disjoint. The details are given in the full version of the paper.

6 Conclusion

We would like to conclude our work with posing one open problem that eluded us during this research. The NP-hardness reduction of Szeider [36] for finding a (simple) path between two vertices of a forbidden-transition graph can be easily modified to prove that it is also NP-hard to find a (simple) compatible cycle in a forbidden-transition graph. In contrast, in edge-colored graphs finding any properly colored cycle is polynomial-time solvable [23, 38]. But what about finding a *long* properly colored cycle? More precisely, given an edge-colored graph G and an integer k we ask whether G admits a simple properly colored cycle of length at least k. Is this problem fixed-parameter tractable when parameterized by k? As the notion of properly colored walks in edge-colored graphs generalizes walks in directed graphs, the problem in question is more general than finding a cycle of length at least k in a directed graph.

References

1 Abdelfattah Abouelaoualim, Kinkar Chandra Das, Luérbio Faria, Yannis Manoussakis, Carlos A. J. Martinhon, and Rachid Saad. Paths and trails in edge-colored graphs. *Theoretical Computer Science*, 409(3):497–510, 2008. doi:10.1016/j.tcs.2008.09.021.

2 Mustaq Ahmed and Anna Lubiw. Shortest paths avoiding forbidden subpaths. In *Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science (STACS 2009)*, pages 63–74, 2009. doi:10.4230/LIPIcs.STACS.2009.1831.

3 Jørgen Bang-Jensen, Thomas Bellitto, William Lochet, and Anders Yeo. The directed 2-linkage problem with length constraints. *Theoretical Computer Science*, 814:69–73, 2020. doi:10.1016/j.tcs.2020.01.012.

4 Jørgen Bang-Jensen, Thomas Bellitto, and Anders Yeo. Supereulerian 2-edge-coloured graphs. Technical report, arXiv:2004.01955 [math.CO], 2020. arXiv:2004.01955.

5 Jørgen Bang-Jensen and Gregory Z. Gutin. *Digraphs - Theory, Algorithms and Applications*. Springer, 2009.

6 Thomas Bellitto. Separating codes and traffic monitoring. *Theoretical Computer Science*, 717:73–85, 2018. Selected papers presented at the 11th International Conference on Algorithmic Aspects of Information and Management (AAIM 2016). doi:10.1016/j.tcs.2017.03.044.

7 Thomas Bellitto and Benjamin Bergougnoux. On minimum connecting transition sets in graphs. In Andreas Brandstädt, Ekkehard Köhler, and Klaus Meer, editors, *Proceedings of the 44th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2018)*, volume 11159 of *Lecture Notes in Computer Science*, pages 40–51. Springer, 2018. doi:10.1007/978-3-030-00256-5_4.

8 Thomas Bellitto, Shaohua Li, Karolina Okrasa, Marcin Pilipczuk, and Manuel Sorge. The complexity of connectivity problems in forbidden-transition graphs and edge-colored graphs. Technical report, arXiv:2009.12892 [cs.DS], 2020. arXiv:2009.12892.

9 Kristóf Bérczi and Yusuke Kobayashi. The directed disjoint shortest paths problem. In Kirk Pruhs and Christian Sohler, editors, *Proceedings of the 25th Annual European Symposium on Algorithms (ESA 2017)*, volume 87 of *LIPIcs*, pages 13:1–13:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ESA.2017.13.

10 Ivona Bezáková, Radu Curticapean, Holger Dell, and Fedor V Fomin. Finding detours is fixed-parameter tractable. *SIAM Journal on Discrete Mathematics*, 33(4):2326–2345, 2019.

11 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Information and Computation*, 243:86–111, 2015. doi:10.1016/j.ic.2014.12.008.

12 Alejandro Contreras-Balbuena, Hortensia Galeana-Sánchez, and Ilan A. Goldfeder. A new sufficient condition for the existence of alternating hamiltonian cycles in 2-edge-colored multigraphs. *Discrete Applied Mathematics*, 229:55–63, 2017. doi:10.1016/j.dam.2017.04.033.

13 Alejandro Contreras-Balbuena, Hortensia Galeana-Sánchez, and Ilan A. Goldfeder. Alternating hamiltonian cycles in 2-edge-colored multigraphs. *Discrete Mathematics and Theoretical Computer Science*, 21(1), 2019. URL: http://dmtcs.episciences.org/5564.

14 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

15 Dietmar Dorninger. Hamiltonian circuits determining the order of chromosomes. *Discrete Applied Mathematics*, 50(2):159–168, 1994. doi:10.1016/0166-218X(92)00171-H.

16 Zdenek Dvorák. Two-factors in orientated graphs with forbidden transitions. *Discrete Mathematics*, 309(1):104–112, 2009. doi:10.1016/j.disc.2007.12.050.

17 Michael R. Fellows, Danny Hermelin, Frances Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009. doi:10.1016/j.tcs.2008.09.065.

18 Steven Fortune, John E. Hopcroft, and James Wyllie. The directed subgraph homeomorphism problem. *Theor. Comput. Sci.*, 10:111–121, 1980. doi:10.1016/0304-3975(80)90009-2.

19 Robert Ganian, Eun Jung Kim, and Stefan Szeider. Algorithmic applications of tree-cut width. In Giuseppe F. Italiano, Giovanni Pighizzini, and Donald Sannella, editors, *Proceedings of the 40th International Symposium on Mathematical Foundations of Computer Science (MFCS 2015)*, volume 9235 of *Lecture Notes in Computer Science*, pages 348–360. Springer, 2015. doi:10.1007/978-3-662-48054-0_29.

20 Marinus Gottschau, Marcus Kaiser, and Clara Waldmann. The undirected two disjoint shortest paths problem. *Operations Research Letters*, 47(1):70–75, 2019. doi:10.1016/j.orl.2018.11.011.

21 Laurent Gourvès, Adria Lyra, Carlos A. J. Martinhon, and Jérôme Monnot. Complexity of trails, paths and circuits in arc-colored digraphs. *Discrete Applied Mathematics*, 161(6):819–828, 2013. doi:10.1016/j.dam.2012.10.025.

22 Laurent Gourvès, Adria Lyra, Carlos A. J. Martinhon, Jérôme Monnot, and Fábio Protti. On s-t paths and trails in edge-colored graphs. *Electronic Notes in Discrete Mathematics*, 35:221–226, 2009. doi:10.1016/j.endm.2009.11.037.

23 Jerrold W. Grossman and Roland Häggkvist. Alternating cycles in edge-partitioned graphs. *Journal of Combinatorial Theory, Series B*, 34(1):77–81, 1983. doi:10.1016/0095-8956(83)90008-4.

24 Gregory Gutin and Eun Jung Kim. Properly coloured cycles and paths: Results and open problems. In *Graph Theory, Computational Intelligence and Thought, Essays Dedicated to Martin Charles Golumbic on the Occasion of His 60th Birthday*, pages 200–208, 2009. doi:10.1007/978-3-642-02029-2_19.

25 Gregory Z. Gutin, Mark Jones, Bin Sheng, Magnus Wahlström, and Anders Yeo. Chinese postman problem on edge-colored multigraphs. *Discrete Applied Mathematics*, 217:196–202, 2017. doi:10.1016/j.dam.2016.08.005.

26 R. Impagliazzo and R. Paturi. Complexity of k-SAT. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity (CCC 1999)*, pages 237–240, 1999. doi:10.1109/CCC.1999.766282.

27 R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS 1998)*, pages 653–662, 1998. doi:10.1109/SFCS.1998.743516.

28 Mamadou Moustapha Kanté, Christian Laforest, and Benjamin Momège. Trees in graphs with conflict edges or forbidden transitions. In T.-H. Hubert Chan, Lap Chi Lau, and Luca Trevisan, editors, *Proceedins of the 10th International Conference on Theory and Applications of Models of Computation (TAMC 2013)*, volume 7876 of *Lecture Notes in Computer Science*, pages 343–354. Springer, 2013. doi:10.1007/978-3-642-38236-9_31.

29 Mamadou Moustapha Kanté, Fatima Zahra Moataz, Benjamin Momège, and Nicolas Nisse. Finding paths in grids with forbidden transitions. In Ernst W. Mayr, editor, *Proceedings of the 41st International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2015)*, volume 9224 of *Lecture Notes in Computer Science*, pages 154–168. Springer, 2015. doi:10.1007/978-3-662-53174-7_12.

30 Yusuke Kobayashi and Ryo Sako. Two disjoint shortest paths problem with non-negative edge length. *Operations Research Letters*, 47(1):66–69, 2019. doi:10.1016/j.orl.2018.11.012.

31 Anton Kotzig. Moves without forbidden transitions in a graph. *Matematický časopis*, 18(1):76–80, 1968. URL: http://eudml.org/doc/33972.

32 Ruonan Li, Hajo Broersma, Chuandong Xu, and Shenggui Zhang. Cycle extension in edge-colored complete graphs. *Discret. Math.*, 340(6):1235–1241, 2017. doi:10.1016/j.disc.2017.01.023.

33 Ruonan Li, Hajo Broersma, and Shenggui Zhang. Properly edge-colored theta graphs in edge-colored complete graphs. *Graphs and Combinatorics*, 35(1):261–286, 2019. doi:10.1007/s00373-018-1989-2.

34 Dániel Marx. Can you beat treewidth? *Theory of Computing*, 6(1):85–112, 2010. doi:10.4086/toc.2010.v006a005.

35 Dániel Marx and Paul Wollan. Immersions in highly edge connected graphs. *SIAM Journal on Discrete Mathematics*, 28(1):503–520, 2014. doi:10.1137/130924056.

36 Stefan Szeider. Finding paths in graphs avoiding forbidden transitions. *Discrete Applied Mathematics*, 126(2-3):261–273, 2003. doi:10.1016/S0166-218X(02)00251-2.

37 Paul Wollan. The structure of graphs not admitting a fixed immersion. *Journal of Combinatorial Theory, Series B*, 110:47–66, 2015.

38 Anders Yeo. A note on alternating cycles in edge-coloured graphs. *Journal of Combinatorial Theory, Series B*, 69(2):222–225, 1997. doi:10.1006/jctb.1997.1728.

39 Michal Ziobro and Marcin Pilipczuk. Finding hamiltonian cycle in graphs of bounded treewidth: Experimental evaluation. *ACM Journal of Experimental Algorithmics*, 24(1):2.7:1–2.7:18, 2019. doi:10.1145/3368631.