The "Art of Trellis Decoding" is NP-Hard*

Navin Kashyap

Dept. Mathematics and Statistics,
Queen's University, Kingston, ON, K7L 3N6, Canada.
Email: nkashyap@mast.queensu.ca

Abstract. Given a linear code $\mathcal{C}$, the fundamental problem of trellis decoding is to find a coordinate permutation of $\mathcal{C}$ that yields a code $\mathcal{C}'$ whose minimal trellis has the least state-complexity among all codes obtainable by permuting the coordinates of $\mathcal{C}$. By reducing from the problem of computing the pathwidth of a graph, we show that the problem of finding such a coordinate permutation is NP-hard, thus settling a long-standing conjecture.

1 Introduction

Maximum-likelihood (ML) decoding of a linear code can be implemented using the Viterbi algorithm on a trellis representation of the code. The run-time complexity of such an implementation depends on the complexity (size) of the trellis representation, and so it is desirable to find, for a given code $\mathcal{C}$, a low-complexity trellis representing $\mathcal{C}$. The theory of trellis representations of a linear code is well understood, and we refer the reader to the review by Vardy [11] for an excellent survey of this theory. A fundamental result of this theory is that a linear code has a unique minimal trellis that simultaneously minimizes several important measures of trellis complexity, including the number of states, the number of edges, and the so-called state-complexity of the trellis. There are several efficient algorithms known for determining the minimal trellis for a given linear code (again, see [11] and the references therein).

It is a somewhat surprising fact that permuting the coordinates of a code can result in a drastic change in the complexity of the minimal trellis. To be precise, if $\mathcal{C}'$ is a code obtained by permuting the coordinates of $\mathcal{C}$, then the minimal trellises of $\mathcal{C}$ and $\mathcal{C}'$ may have very different sizes. However, the simple action of coordinate permutation does not affect the performance of the code from an error-correction viewpoint. Therefore, given a code $\mathcal{C}$, one may as well use the code $\mathcal{C}'$ obtained by permuting the coordinates of $\mathcal{C}$, such that the minimal trellis of $\mathcal{C}'$ has the least complexity among the minimal trellises of codes obtained from $\mathcal{C}$ via coordinate permutations. The problem of determining the coordinate permutation of $\mathcal{C}$ that minimizes the complexity of the resulting minimal trellis has been termed the "art of trellis decoding" by Massey [8].

* This work was supported in part by a research grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada.

It now matters which measure of trellis complexity is to be minimized, as the coordinate permutation of $\mathcal{C}$ that yields a minimal trellis with, say, the least state-complexity need not be the same as the coordinate permutation that yields a minimal trellis with the smallest number of states. The prior literature has most often focused on the problem of finding the coordinate permutation of a given code that minimizes the state-complexity of the resulting minimal trellis, and it has repeatedly been conjectured that this problem is NP-hard [5], [6], [11, Section 5]. To put it another way, the following decision problem was conjectured to be NP-complete:

Problem: Trellis State-Complexity

Let $\mathbb{F}_q$ be a fixed finite field.

Instance: An $m \times n$ generator matrix for a linear code $\mathcal{C}$ over $\mathbb{F}_q$, and an integer $w > 0$.

Question: Is there a coordinate permutation of $\mathcal{C}$ that yields a code $\mathcal{C}'$ whose minimal trellis has state-complexity at most w?

This decision problem was called "Maximum Partition Rank Permutation" in [5], and "Maximum Width" in [6]. Forney [4] has referred to the resolution of the aforementioned conjecture as the "only significant open problem" in the context of trellis representations.

In this paper, we settle the conjecture in the affirmative. We show that, for any fixed finite field $\mathbb{F}_q$, given an arbitrary code $\mathcal{C}$ over $\mathbb{F}_q$, the problem of finding the coordinate permutation of $\mathcal{C}$ that yields a minimal trellis with the least possible state-complexity is indeed NP-hard. Thus, Trellis State-Complexity is NP-complete. Our proof is by reduction from the problem of computing the pathwidth of a graph, which is known to be NP-hard [1],[2].

The rest of the paper is organized as follows. In Section 2, we lay down the definitions and notation necessary for our development. In Section 3, we sketch out a proof of the fact that for any fixed finite field $\mathbb{F}_q$, Trellis State-Complexity is NP-complete. We have had to omit some of the details of the proof due to space limitations; the complete proof can be found in our full paper [7]. We make some concluding remarks in Section 4.

2 Preliminaries

A *trellis* T for a length-n linear code $\mathcal{C}$ over a finite field $\mathbb{F}_q$ is an edge-labelled directed acyclic graph with certain properties. The vertex set, V, of T can be partitioned into $n+1$ disjoint subsets $V_0, V_1, \ldots, V_n$, such that each (directed) edge of T starts at V_i and ends at V_{i+1} for some $i \in \{0, 1, \ldots, n-1\}$. The set V_i is called the set of *states* at time index i. The set V_0 consists of a unique initial state v_0, and the set V_n consists of a unique terminal state v_n. It is further required that each state $v \in V$ lie on some (directed) path from v_0 to v_n. Note that each path from v_0 to v_n is of length exactly n. The edges of T are given labels from $\mathbb{F}_q$ in such a way that the set of all label sequences associated with paths from v_0 to v_n is precisely the code $\mathcal{C}$.

It turns out that if T is the minimal trellis for a linear code $\mathcal{C}$, then the cardinalities of the sets V_i are all powers of q. It is thus convenient to define the *state-complexity profile* of T to be the $(n+1)$-tuple $\mathbf{s} = (s_0, s_1, \ldots, s_n)$, where $s_i = \log_q(|V_i|)$. The *state-complexity* of T is then defined as $s_{\max} = \max_i s_i$. When T is the minimal trellis of $\mathcal{C}$, there is an explicit expression known for the s_i's. We will find it convenient to give this expression in terms of the connectivity function of $\mathcal{C}$, as defined below.

The set $[n] \overset{\text{def}}{=} \{1, 2, \ldots, n\}$ is taken to be the coordinate set of the length-n code $\mathcal{C}$. Given a subset $J \subset [n]$, we let $\mathcal{C}|_J$ denote the restriction of $\mathcal{C}$ to the coordinates with labels in J. In other words, $\mathcal{C}|_J$ is the code obtained by puncturing the coordinates in $J^c = [n] - J$. The *connectivity function* of the code $\mathcal{C}$ is the function $\lambda_{\mathcal{C}} : 2^{[n]} \to \mathbb{Z}$ defined by

$$\lambda_{\mathcal{C}}(J) = \dim(\mathcal{C}|_J) + \dim(\mathcal{C}|_{J^c}) - \dim(\mathcal{C}), \tag{1}$$

for each $J \subset [n]$. It is obvious that for any $J \subset [n]$, we have $\lambda_{\mathcal{C}}(J) \geq 0$ and $\lambda_{\mathcal{C}}(J) = \lambda_{\mathcal{C}}(J^c)$. Observe also that $\lambda_{\mathcal{C}}(\emptyset) = \lambda_{\mathcal{C}}([n]) = 0$. Furthermore, some elementary linear algebra suffices to verify that $\lambda_{\mathcal{C}}(J) = \lambda_{\mathcal{C}^\perp}(J)$ for any $J \subset [n]$.

The state-complexity profile of the minimal trellis of $\mathcal{C}$ can now be expressed as $\mathbf{s}(\mathcal{C}) = (s_0(\mathcal{C}), s_1(\mathcal{C}), \ldots, s_n(\mathcal{C}))$, where $s_0(\mathcal{C}) = s_n(\mathcal{C}) = 0$, and for $1 \leq i \leq n-1$,

$$s_i(\mathcal{C}) = \lambda_{\mathcal{C}}(\{1, 2, \ldots, i\}). \tag{2}$$

Thus, the state-complexity of the minimal trellis of $\mathcal{C}$ is given by $s_{\max}(\mathcal{C}) = \max_{i \in [n]} s_i(\mathcal{C})$. Note that since $\lambda_{\mathcal{C}}(J) = \lambda_{\mathcal{C}^\perp}(J)$ for any $J \subset [n]$, we have $\mathbf{s}(\mathcal{C}) = \mathbf{s}(\mathcal{C}^\perp)$, and hence, $s_{\max}(\mathcal{C}) = s_{\max}(\mathcal{C}^\perp)$.

As mentioned in Section 1, different coordinate permutations of the same code may result in codes with minimal trellises of very different complexities [11, Example 5.1]. Therefore, letting $[\mathcal{C}]$ denote the set of all codes that can be obtained from a code $\mathcal{C}$ by means of coordinate permutations, it is of interest to define the *trellis-width* of the family $[\mathcal{C}]$ as follows:

$$\operatorname{tw}[\mathcal{C}] = \min_{\mathcal{C}' \in [\mathcal{C}]} s_{\max}(\mathcal{C}') = \min_{\mathcal{C}' \in [\mathcal{C}]} \max_{i \in [n]} s_i(\mathcal{C}'). \tag{3}$$

The main aim of this paper is to show that, given a code $\mathcal{C}$, the problem of computing the trellis-width of $[\mathcal{C}]$ is NP-hard. We accomplish this by reduction from the known NP-hard problem of computing the pathwidth of a graph.

3 NP-Hardness of Trellis-Width

The notion of graph pathwidth was introduced by Robertson and Seymour in [10]. Let $\mathcal{G}$ be a graph with vertex set V. An ordered collection $\mathcal{V} = (V_1, \ldots, V_t)$, $t \geq 1$, of subsets of V is called a *path-decomposition* of $\mathcal{G}$, if

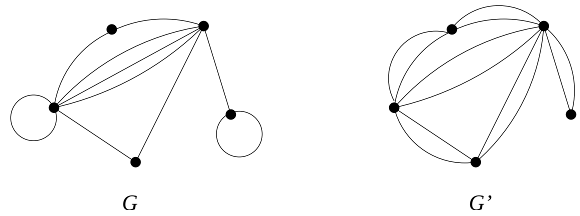

Fig. 1. Construction of $\mathcal{G}'$ from $\mathcal{G}$.

(i) $\bigcup_{i=1}^{t} V_i = V$;
(ii) for each pair of adjacent vertices $u, v \in V$, we have $\{u, v\} \subset V_i$ for some $i \in [t]$; and
(iii) for $1 \leq i < j < k \leq t$, $V_i \cap V_k \subset V_j$.

The *width* of such a path-decomposition $\mathcal{V}$ is defined to be $w_{\mathcal{G}}(\mathcal{V}) = \max_{i\in[t]} |V_i| - 1$. The *pathwidth* of $\mathcal{G}$, denoted by $\text{pw}(\mathcal{G})$, is the minimum among the widths of all its path-decompositions. A path-decomposition $\mathcal{V}$ such that $w_{\mathcal{G}}(\mathcal{V}) = \text{pw}(\mathcal{G})$ is called an *optimal* path-decomposition of $\mathcal{G}$.

Let $\mathbb{F}_q$ be an arbitrary finite field. Given a graph $\mathcal{G}$ with vertex set V, our aim is to produce, in time polynomial in $|V|$, a matrix A that generates a code $\mathcal{C}$ over $\mathbb{F}_q$ such that $\text{pw}(\mathcal{G})$ can be directly computed from $\text{tw}[\mathcal{C}]$. The NP-hardness of computing graph pathwidth then implies the NP-hardness of computing the trellis-width of $[\mathcal{C}]$ for an arbitrary code $\mathcal{C}$ over F_q. We now describe our construction of the matrix A.

Let $\mathcal{G}'$ be a graph defined on the same vertex set, V, as $\mathcal{G}$, having the following properties (see Figure 1):

(P1) $\mathcal{G}'$ is loopless;
(P2) a pair of distinct vertices is adjacent in $\mathcal{G}'$ iff it is adjacent in $\mathcal{G}$; and
(P3) in $\mathcal{G}'$, there are exactly two edges between each pair of adjacent vertices.

It is evident from the definition that $(V_1, \ldots, V_t)$ is a path-decomposition of $\mathcal{G}$ iff it is a path-decomposition of $\mathcal{G}'$. Therefore, $\text{pw}(\mathcal{G}') = \text{pw}(\mathcal{G})$.

Define $\overline{\mathcal{G}}$ to be the graph obtained by adding an extra vertex, henceforth denoted by x, to $\mathcal{G}'$, along with a pair of parallel edges from x to each $v \in V$ (see Figure 2). We will denote by $\overline{V}$ and $\overline{E}$ the vertex and edge sets, respectively, of $\overline{\mathcal{G}}$. Clearly, $\overline{\mathcal{G}}$ is constructible directly from $\mathcal{G}$ in $O(|V|^2)$ time. But more importantly, the desired matrix A can be readily obtained from the graph $\overline{\mathcal{G}}$. Indeed, letting $D(\overline{\mathcal{G}})$ be any directed graph obtained by arbitrarily assigning orientations to the edges of $\overline{\mathcal{G}}$, we simply take A to be the vertex-edge incidence matrix of $D(\overline{\mathcal{G}})$. This is the $|\overline{V}| \times |\overline{E}|$ matrix whose rows and columns are indexed by the vertices and directed edges, respectively, of $D(\overline{\mathcal{G}})$, and whose (i, j)th entry, $a_{i,j}$, is determined as follows:

$$a_{i,j} = \begin{cases} 1 & \text{if vertex } i \text{ is the tail of non-loop edge } j \\ -1 & \text{if vertex } i \text{ is the head of non-loop edge } j \\ 0 & \text{otherwise.} \end{cases}$$

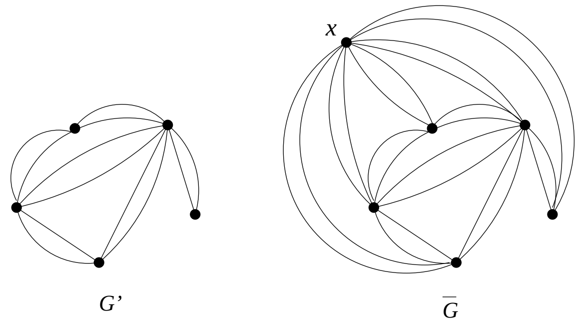

Fig. 2. Construction of $\overline{\mathcal{G}}$ from $\mathcal{G}'$.

Denote by $\overline{\mathcal{C}}$ the linear code over $\mathbb{F}_q$ generated by the matrix A. The trellis-width of $[\overline{\mathcal{C}}]$ relates very simply to the pathwidth of the original graph $\mathcal{G}$, as made precise by the following proposition.

Proposition 1. *$tw[\overline{\mathcal{C}}] = pw(\mathcal{G}) + 1$.*

Before proving the above proposition, we observe that it yields the desired NP-hardness result. Indeed, it is easily checked that the matrix A can be constructed directly from $\mathcal{G}$ in $O(|V|^3)$ time. Now, suppose that there were a polynomial-time algorithm for computing the trellis-width of $[\mathcal{C}]$ for an arbitrary code $\mathcal{C}$ over $\mathbb{F}_q$, the code $\mathcal{C}$ being specified by some generator matrix. Then, given any graph $\mathcal{G}$, we can construct the matrix A, and then compute the trellis-width of $[\overline{\mathcal{C}}]$, all in polynomial time. Therefore, by Proposition 1, we have a polynomial-time algorithm to compute the pathwidth of $\mathcal{G}$. However, the graph pathwidth problem is NP-hard [1],[2]. So, if there exists a polynomial-time algorithm for it, then we must have $P = NP$. This implies our main result.

Theorem 2. *Let $\mathbb{F}_q$ be a fixed finite field. The problem of computing the trellis-width of an arbitrary linear code over $\mathbb{F}_q$, specified by any of its generator matrices, is NP-hard.*

Corollary 3. *For any fixed finite field $\mathbb{F}_q$, the decision problem* TRELLIS STATE-COMPLEXITY *is NP-complete.*

The remainder of this section is devoted to the proof of Proposition 1. Since $\mathrm{pw}(\mathcal{G}') = \mathrm{pw}(\mathcal{G})$, for the purpose of our proof, we may assume that $\mathcal{G}' = \mathcal{G}$. Thus, from now until the end of this section, we take $\mathcal{G}$ to be a loopless graph satisfying property (P3) above. Note that $\overline{\mathcal{G}}$ also satisfies (P3). For each pair of adjacent vertices u, v in $\mathcal{G}$ or $\overline{\mathcal{G}}$, we denote by l_{uv} and r_{uv} the two edges between u and v. Recall that V and E denote the sets of vertices and edges, respectively, of $\mathcal{G}$, and that $\overline{V}$ and $\overline{E}$ denote the corresponding sets of $\overline{\mathcal{G}}$. We thus have $\overline{V} = V \,\dot{\cup}\, \{x\}$, and $\overline{E} = E \,\dot{\cup}\, \left(\bigcup_{v \in V} \{l_{xv}, r_{xv}\}\right)$.

We will make much use of a basic fact, stated next, about the $|\overline{V}| \times |\overline{E}|$ matrix A whose construction was described above. For any $J \subset \overline{E}$, if $A|_J$ denotes the matrix obtained by restricting A to the columns indexed by the edges in J, then

$$\text{rank}(A|_J) = \dim(\overline{\mathcal{C}}|_J) = r(J), \tag{4}$$

where rank and dim above are computed over the field $\mathbb{F}_q$, and $r(J)$ denotes the number of edges in any spanning forest of the subgraph of $\overline{\mathcal{G}}$ induced by J. To be precise, letting $\overline{\mathcal{G}}[J]$ denote the subgraph of $\overline{\mathcal{G}}$ induced by J, we have $r(J) = |V(\overline{\mathcal{G}}[J])| - \omega(\overline{\mathcal{G}}[J])$, where $\omega(\overline{\mathcal{G}}[J])$ is the number of connected components of $\overline{\mathcal{G}}[J]$. Equation (4) can be inferred from [9, Proposition 5.1.2].

We shall identify the set $\overline{E}$ with the coordinate set of the code $\overline{\mathcal{C}}$ generated by A. Given an ordering $\pi = (e_1, e_2, \ldots, e_n)$ of the elements of $\overline{E}$, we will denote by $\overline{\mathcal{C}}_\pi$ the code obtained by putting the coordinates of $\overline{\mathcal{C}}$ in the order specified by π. For any $J \subset \overline{E}$, and any ordering, π, of $\overline{E}$, we have by virtue of (4),

$$\begin{aligned}\lambda_{\overline{\mathcal{C}}_\pi}(J) = \lambda_{\overline{\mathcal{C}}}(J) &= r(J) + r(\overline{E} - J) - r(\overline{E}) \\ &= r(J) + r(\overline{E} - J) - |V|, \end{aligned} \tag{5}$$

the last equality above following from the fact that $\omega(\overline{\mathcal{G}}) = 1$ since $\overline{\mathcal{G}}$ is connected (each $v \in V$ is adjacent to x), so that $r(\overline{E}) = |\overline{V}| - 1 = |V|$.

We are now in a position to begin the proof of Proposition 1. We will first prove that $\text{tw}[\overline{\mathcal{C}}] \leq \text{pw}(\mathcal{G}) + 1$. Let $\mathcal{V} = (V_1, \ldots, V_t)$ be a path-decomposition of $\mathcal{G}$. We need the following fact about $\mathcal{V}$: for each $j \in [t]$,

$$\bigcup_{i \leq j} V_i \;\cap\; \bigcup_{k \geq j} V_k \;=\; V_j. \tag{6}$$

The above equality follows from the fact that a path-decomposition, by definition, has the property that for $1 \leq i < j < k \leq t$, $V_i \cap V_k \subset V_j$.

For $j \in [t]$, let F_j be the set of edges of $\mathcal{G}$ that have both their end-points in V_j. By condition (ii) in the definition of path-decomposition, $\bigcup_{j=1}^t F_j = E$. Now, let $\overline{F_j} = F_j \cup \left(\bigcup_{v \in V_j} \{l_{xv}, r_{xv}\}\right)$, so that $\bigcup_{j=1}^t \overline{F_j} = \overline{E}$.

Definition 4. *An ordering $(e_1, \ldots, e_n)$ of the edges of $\overline{\mathcal{G}}$ is said to* induce *an ordered partition $(E_1, \ldots, E_t)$ of $\overline{E}$ if for each $j \in [t]$, $\{e_{n_{j-1}+1}, e_{n_{j-1}+2}, \ldots, e_{n_j}\} = E_j$, where $n_j = \left|\bigcup_{i \leq j} E_i\right|$ (and $n_0 = 0$).*

Let $\pi = (e_1, \ldots, e_n)$ be any ordering of $\overline{E}$ that induces the ordered partition $(E_1, E_2, \ldots, E_t)$, where for each $j \in [t]$, $E_j = \overline{F_j} - \bigcup_{i<j} \overline{F_i}$. We claim that the state-complexity of the minimal trellis of $\overline{\mathcal{C}}_\pi$ is at most one more than the width of the path-decomposition $\mathcal{V}$.

Lemma 5. $s_{\max}(\overline{\mathcal{C}}_\pi) \leq w_{\mathcal{G}}(\mathcal{V}) + 1.$

Proof. Observe first that

$$\begin{aligned} s_{\max}(\overline{\mathcal{C}}_\pi) &= \max_{j\in[t]} \max_{1\le k\le n_j-n_{j-1}} \lambda_{\overline{\mathcal{C}}_\pi}\left(\bigcup_{i<j} E_i \cup \{e_{n_{j-1}+1},\ldots,e_{n_{j-1}+k}\}\right) \\ &\le \max_{j\in[t]} \max_{E'\subset E_j} \lambda_{\overline{\mathcal{C}}_\pi}\left(\bigcup_{i<j} E_i \cup E'\right). \end{aligned} \tag{7}$$

Let $X = \bigcup_{i<j} E_i \cup E'$ for some $j \in [t]$ and $E' \subset E_j$. By (5), $\lambda_{\overline{\mathcal{C}}_\pi}(X) = r(X) + r(\overline{E} - X) - |V|$. If v is a vertex of $\overline{\mathcal{G}}$ incident with an edge in X, then $v \in \bigcup_{i\le j} V_i \,\dot\cup\, \{x\}$. So, the subgraph of $\overline{\mathcal{G}}$ induced by X has its vertices contained in $\bigcup_{i\le j} V_i \,\dot\cup\, \{x\}$. Therefore, $r(X) \le \left|\bigcup_{i\le j} V_i \,\dot\cup\, \{x\}\right| - 1 = \left|\bigcup_{i\le j} V_i\right|$.

Next, consider $\overline{E} - X = (\bigcup_{k>j} E_k) \cup (E_j - E')$. Reasoning as above, the subgraph of $\overline{\mathcal{G}}$ induced by $\overline{E} - X$ has its vertices contained in $\bigcup_{k\ge j} V_k \,\dot\cup\, \{x\}$. Hence, $r(\overline{E} - X) \le \left|\bigcup_{k\ge j} V_k\right|$.

Therefore, we have

$$\lambda_{\overline{\mathcal{C}}_\pi}(X) \;\le\; \left|\bigcup_{i\le j} V_i\right| + \left|\bigcup_{k\ge j} V_k\right| - |V| \;=\; \left|\bigcup_{i\le j} V_i \cap \bigcup_{k\ge j} V_k\right| \;=\; |V_j|,$$

the last equality arising from (6). Hence, carrying on from (7),

$$s_{\max}(\overline{\mathcal{C}}_\pi) \le \max_{j\in[t]} |V_j| = w_{\mathcal{G}}(\mathcal{V}) + 1,$$

as desired. ∎

The fact that $\mathrm{tw}[\mathcal{C}] \le \mathrm{pw}(\mathcal{G}) + 1$ easily follows from the above lemma. Indeed, we may choose $\mathcal{V}$ to be an optimal path-decomposition of $\mathcal{G}$. Then, by Lemma 5, there exists an ordering π of $\overline{E}$ such that $s_{\max}(\overline{\mathcal{C}}_\pi) \le \mathrm{pw}(\mathcal{G}) + 1$. Hence, $\mathrm{pw}(M) \le s_{\max}(\overline{\mathcal{C}}_\pi) \le \mathrm{pw}(\mathcal{G}) + 1$.

We prove the reverse inequality in two steps, first showing that $\mathrm{pw}(\overline{\mathcal{G}}) = \mathrm{pw}(\mathcal{G}) + 1$, and then showing that $\mathrm{tw}[\mathcal{C}] \ge \mathrm{pw}(\overline{\mathcal{G}})$.

Lemma 6. *$pw(\overline{\mathcal{G}}) = pw(\mathcal{G}) + 1$.*

Proof. Clearly, if $\mathcal{V} = (V_1, \ldots, V_t)$ is a path-decomposition of $\mathcal{G}$, then $\overline{\mathcal{V}} = (V_1 \cup \{x\}, \ldots, V_t \cup \{x\})$ is a path-decomposition of $\overline{\mathcal{G}}$. Hence, choosing $\mathcal{V}$ to be an optimal path-decomposition of $\mathcal{G}$, we have that $\mathrm{pw}(\overline{\mathcal{G}}) \le w_{\overline{\mathcal{G}}}(\overline{\mathcal{V}}) = w_{\mathcal{G}}(\mathcal{V}) + 1 = \mathrm{pw}(\mathcal{G}) + 1$.

For the inequality in the other direction, we will show that there exists an optimal path-decomposition, $\widetilde{\mathcal{V}} = (\widetilde{V}_1, \ldots, \widetilde{V}_s)$, of $\overline{\mathcal{G}}$ such that $x \in \widetilde{V}_i$ for all

$i \in [s]$. We then have $\mathcal{V} = (\widetilde{V}_1 - \{x\}, \ldots, \widetilde{V}_s - \{x\})$ being a path-decomposition of $\mathcal{G}$, and hence, $\mathrm{pw}(\mathcal{G}) \leq w_{\mathcal{G}}(\mathcal{V}) = w_{\overline{\mathcal{G}}}(\widetilde{\mathcal{V}}) - 1 = \mathrm{pw}(\overline{\mathcal{G}}) - 1$.

Let $\overline{\mathcal{V}} = (\overline{V}_1, \ldots, \overline{V}_t)$ be an optimal path-decomposition of $\overline{\mathcal{G}}$, and let $i_0 = \min\{i : x \in \overline{V}_i\}$ and $i_1 = \max\{i : x \in \overline{V}_i\}$. Since $\overline{V}_i \cap \overline{V}_k \subset \overline{V}_j$ for $i < j < k$, we must have $x \in \overline{V}_i$ for each $i \in [i_0, i_1]$.

We claim that $(\overline{V}_{i_0}, \overline{V}_{i_0+1}, \ldots, \overline{V}_{i_1})$ is a path-decomposition of $\overline{\mathcal{G}}$. We only have to show that $\bigcup_{i=i_0}^{i_1} \overline{V}_i = \overline{V}$, and that for each pair of adjacent vertices $u, v \in \overline{V}$, $\{u, v\} \subset \overline{V}_i$ for some $i \in [i_0, i_1]$. To see why the first assertion is true, consider any $v \in \overline{V}$, $v \neq x$. Since x is adjacent to v, and $\overline{\mathcal{V}}$ is a path-decomposition of $\overline{\mathcal{G}}$, $\{x, v\} \subset \overline{V}_i$ for some $i \in [t]$. However, $x \in \overline{V}_i$ iff $i \in [i_0, i_1]$, and so, $\{x, v\} \subset \overline{V}_i$ for some $i \in [i_0, i_1]$. In particular, $v \in \overline{V}_i$ for some $i \in [i_0, i_1]$.

For the second assertion, suppose that u, v is a pair of vertices adjacent in $\overline{\mathcal{G}}$. Obviously, $\{u, v\} \subset \overline{V}_j$ for some $j \in [t]$. Suppose that $j \notin [i_0, i_1]$. We consider the case when $j > i_1$; the case when $j < i_0$ is similar. As $\bigcup_{i=i_0}^{i_1} \overline{V}_i = \overline{V}$, there exist $i_2, i_3 \in [i_0, i_1]$ such that $u \in \overline{V}_{i_2}$ and $v \in \overline{V}_{i_3}$. Without loss of generality (WLOG), $i_2 \leq i_3$. If $i_2 = i_3$, then there exists $i \in [i_0, i_1]$ such that $\{u, v\} \subset \overline{V}_i$. If $i_2 < i_3$, we have $u \in \overline{V}_{i_2} \cap \overline{V}_j$ and $i_2 < i_3 < j$. Hence, $u \in \overline{V}_{i_3}$ as well, and so once again, we have an $i \in [i_0, i_1]$ such that $\{u, v\} \in \overline{V}_i$.

Thus, $(\overline{V}_{i_0}, \overline{V}_{i_0+1}, \ldots, \overline{V}_{i_1})$ is a path-decomposition of $\overline{\mathcal{G}}$, with the property that $x \in \overline{V}_i$ for all $i \in [i_0, i_1]$. It must be an optimal path-decomposition, since it is a subsequence of the optimal path-decomposition $\overline{\mathcal{V}}$. ■

To complete the proof of Proposition 1, it remains to show that $\mathrm{tw}[\overline{\mathcal{C}}] \geq \mathrm{pw}(\overline{\mathcal{G}})$. We introduce some notation at this point. Recall that the two edges between a pair of adjacent vertices u and v in $\overline{\mathcal{G}}$ (or $\mathcal{G}$) are denoted by l_{uv} and r_{uv}. We define

$$L_{\mathcal{G}} = \{l_{uv} : u, v \text{ are adjacent vertices in } \mathcal{G}\},$$
$$R_{\mathcal{G}} = \{r_{uv} : u, v \text{ are adjacent vertices in } \mathcal{G}\},$$

$L_x = \bigcup_{v \in V} \{l_{xv}\}$ and $R_x = \bigcup_{v \in V} \{r_{xv}\}$, where x is the distinguished vertex in $\overline{V} - V$. For $L \subset L_x$, define the *closure* of L to be the set $\mathrm{cl}(L) = L \cup \{r_{xu} :\ l_{xu} \in L\} \cup \{l_{uv}, r_{uv} :\ l_{xu}, l_{xv} \in L\}$. Note that $\mathrm{cl}(L_x) = \overline{E}$.

Our argument rests on the next lemma, whose somewhat technical proof we omit here. We refer the reader instead to the proof given in [7].

Lemma 7. *There exists an ordering $\pi = (e_1, \ldots, e_n)$ of $\overline{E}$ with the following properties:*

(a) $s_{\max}(\overline{\mathcal{C}}_\pi) = tw[\overline{\mathcal{C}}]$.

(b) The ordering π induces an ordered partition of $\overline{E}$ of the form

$$(L_1, A_1, B_1, R_1, L_2, A_2, B_2, R_2, \ldots, L_t, A_t, B_t, R_t),$$

where for each $j \in [t]$, $L_j \subset L_x$, $A_j \subset L_{\mathcal{G}}$, $B_j \subset R_{\mathcal{G}}$ and $R_j \subset R_x$. Moreover, for each $u, v \in \overline{V}$, $l_{uv} \in L_j \cup A_j$ iff $r_{uv} \in B_j \cup R_j$.

(c) For the ordered partition in (b), we have for each $j \in [t]$,

$$A_j \cup B_j \subset cl(\bigcup_{i\le j} L_i) - cl(\bigcup_{i<j} L_i).$$

We can now furnish the last remaining piece of the proof of Proposition 1.

Lemma 8. *$tw[\overline{\mathcal{C}}] \ge pw(\overline{\mathcal{G}})$.*

Proof. Let $\pi = (e_1, \ldots, e_n)$ be an ordering of $\overline{E}$ having the properties guaranteed by Lemma 7. This ordering induces an ordered partition $(L_1, A_1, B_1, R_1, \ldots, L_t, A_t, B_t, R_t)$ of $\overline{E}$, as in Lemma 7(b). For $j \in [t]$, define $Y_j = \bigcup_{i<j}(L_i \cup A_i \cup B_i \cup R_i) \cup (L_j \cup A_j)$, and $Y'_j = \overline{E} - Y_j \ = \bigcup_{i>j}(L_i \cup A_i \cup B_i \cup R_i) \cup (B_j \cup R_j)$. Letting $\overline{\mathcal{G}}[Y_j]$ and $\overline{\mathcal{G}}[Y'_j]$ denote the subgraphs of $\overline{\mathcal{G}}$ induced by Y_j and Y'_j, respectively, set $V_j = V(\overline{\mathcal{G}}[Y_j]) \cap V(\overline{\mathcal{G}}[Y'_j])$. In other words, V_j is the set of vertices common to both $\overline{\mathcal{G}}[Y_j]$ and $\overline{\mathcal{G}}[Y'_j]$. It is easily checked that $\mathcal{V} = (V_1, \ldots, V_t)$ is a path-decomposition of $\overline{\mathcal{G}}$. Note that

$$|V_j| = |V(\overline{\mathcal{G}}[Y_j])| + |V(\overline{\mathcal{G}}[Y'_j])| - |\overline{V}|.$$

We next observe that $\overline{\mathcal{G}}[Y_j]$ and $\overline{\mathcal{G}}[Y'_j]$ are connected graphs. From Lemma 7(c), we have that $Y_j \subset \text{cl}(\bigcup_{i\le j} L_i)$. Therefore, for any edge l_{uv} (or r_{uv}) in $Y_j - \bigcup_{i\le j} L_i$, both l_{xu} and l_{xv} must be in some L_i, $i \le j$. Thus, in $\overline{\mathcal{G}}[Y_j]$, each vertex $v \ne x$ is adjacent to x, which shows that $\overline{\mathcal{G}}[Y_j]$ is connected.

Consider any vertex $v \ne x$ in $\overline{\mathcal{G}}[Y'_j]$, such that $r_{xv} \notin Y'_j$. Then, $r_{uv} \in Y'_j$ for some $u \ne x$. So, $r_{uv} \in B_k$ for some $k \ge j$. By Lemma 7(c), $r_{uv} \in \text{cl}(\bigcup_{i\le k} L_i) - \text{cl}(\bigcup_{i<k} L_i)$. This implies that either $l_{xu} \in L_k$ or $l_{xv} \in L_k$. Hence, either $r_{xu} \in R_k$ or $r_{xv} \in R_k$. However, r_{xv} cannot be in R_k, since $r_{xv} \notin Y'_j$, and so, $r_{xu} \in R_k$. Thus, (r_{xu}, r_{uv}) forms a path in $\overline{\mathcal{G}}[Y'_j]$ from x to v. It follows that $\overline{\mathcal{G}}[Y'_j]$ is connected.

Therefore, by (5),

$$\begin{aligned}\lambda_{\overline{\mathcal{C}}_\pi}(Y_j) &= r(Y_j) + r(Y'_j) - |V| \\ &= (|V(\overline{\mathcal{G}}[Y_j])| - 1) + (|V(\overline{\mathcal{G}}[Y'_j])| - 1) - (|\overline{V}| - 1) = |V_j| - 1.\end{aligned}$$

Hence, from Lemma 7(a),

$$\text{tw}[\overline{\mathcal{C}}] = s_{\max}(\overline{\mathcal{C}}_\pi) \ge \max_{j\in[t]} \lambda_{\overline{\mathcal{C}}_\pi}(Y_j) = \max_{j\in[t]} |V_j| - 1 = w_{\overline{\mathcal{G}}}(\mathcal{V}) \ge \text{pw}(\overline{\mathcal{G}}),$$

which proves the lemma. ∎

The proof of Proposition 1 is now complete.

4 Concluding Remarks

The main contribution of this paper was to show that the decision problem TRELLIS STATE-COMPLEXITY is NP-complete, thus settling a long-standing conjecture. Now, the situation is rather different if we consider a variation of the problem in which the integer w is *not* taken to be a part of the input to the problem. In other words, consider the following problem:

Problem: WEAK TRELLIS STATE-COMPLEXITY
Let $\mathbb{F}_q$ be a fixed finite field, and let w be a fixed positive integer.

Instance: An $m \times n$ generator matrix for a linear code $\mathcal{C}$ over $\mathbb{F}_q$.
Question: Is there a coordinate permutation of $\mathcal{C}$ that yields a code $\mathcal{C}'$ whose minimal trellis has state-complexity at most w?

There is good reason to believe that this problem is solvable in polynomial time. We again refer the reader to our full paper [7] for evidence in support of this belief.

Acknowledgment

The author would like to thank Jim Geelen for contributing some of his ideas to this paper.

References

1. S. Arnborg, D.G. Corneil and A. Proskurowski, "Complexity of finding embeddings in a k-tree," *SIAM J. Alg. Disc. Meth.*, vol. 8, pp. 277–284, 1987.
2. H.L. Bodlaender, "A tourist guide through treewidth," *Acta Cybernetica*, vol. 11, pp. 1–23, 1993.
3. G.D. Forney Jr., "Dimension/length profiles and trellis complexity of linear block codes," *IEEE Trans. Inform. Theory*, vol. 40, no. 6, pp. 1741–1752, Nov. 1994.
4. G.D. Forney Jr., "Codes on Graphs: Constraint Complexity of Cycle-Free Realizations of Linear Codes," *IEEE Trans. Inform. Theory*, vol. 49, no. 7, pp. 1597–1610, July 2003.
5. G.B. Horn and F.R. Kschischang, "On the intractability of permuting a block code to minimize trellis complexity," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 2042–2048, Nov. 1996.
6. K. Jain, I. Măndoiu and V.V. Vazirani, "The 'art of trellis decoding' is computationally hard — for large fields," *IEEE. Trans. Inform. Theory*, vol. 44, no. 3, pp. 1211–1214, May 1998.
7. N. Kashyap, "Matroid pathwidth and code trellis complexity," to appear in *SIAM J. Discrete Math.* ArXiv e-print 0705.1384.
8. J.L. Massey, "Foundation and methods of channel encoding," in *Proc. Int. Conf. Inform. Theory and Systems*, vol. 65, Berlin, Germany: NTG-Fachberichte, 1978.
9. J.G. Oxley, *Matroid Theory*, Oxford University Press, Oxford, UK, 1992.
10. N. Robertson and P.D. Seymour, "Graph minors. I. Excluding a forest," *J. Combin. Theory, Ser. B*, vol. 35, pp. 39–61, 1983.
11. A. Vardy, "Trellis Structure of Codes," in *Handbook of Coding Theory*, R. Brualdi, C. Huffman and V. Pless, Eds., Amsterdam, The Netherlands: Elsevier, 1998.