Computational Geometry: Theory and Applications 105–106 (2022) 101879

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

On interval decomposability of 2D persistence modules

Hideto Asashiba [a], Mickaël Buchet [b,*], Emerson G. Escolar [c,d], Ken Nakashima [d], Michio Yoshiwaki [d,e]

[a] *Faculty of Science, Shizuoka University, Shizuoka, Japan*
[b] *Institute of Geometry, TU Graz, Graz, Austria*
[c] *Graduate School of Human Development and Environment, Kobe University, Kobe, Japan*
[d] *Center for Advanced Intelligence Project, RIKEN, Tokyo, Japan*
[e] *Osaka City University Advanced Mathematical Institute, Osaka, Japan*

ARTICLE INFO

Article history:
Received 10 May 2021
Received in revised form 4 February 2022
Accepted 20 March 2022
Available online 24 March 2022

Keywords:
Multidimensional persistence
Interval representations
Representation theory

ABSTRACT

In the persistent homology of filtrations, the indecomposable decompositions provide the persistence diagrams. However, in almost all cases of multidimensional persistence, the classification of all indecomposable modules is known to be a wild problem. One direction is to consider the subclass of interval-decomposable persistence modules, which are direct sums of interval representations. We introduce the definition of pre-interval representations, a more natural algebraic definition, and study the relationships between pre-interval, interval, and thin indecomposable representations. We show that over the "equioriented" commutative 2D grid, these concepts are equivalent. Moreover, we provide a criterion for determining whether or not an nD persistence module is interval/pre-interval/thin-decomposable without having to explicitly compute decompositions. For 2D persistence modules, we provide an algorithm for determining interval-decomposability, together with a worst-case complexity analysis that uses the total number of intervals in an equioriented commutative 2D grid. We also propose several heuristics to speed up the computation.

1. Introduction

In recent years, the use of topological data analysis to understand the shape of data has become popular, with persistent homology [1] as one of its leading tools. Persistent homology is used to study the persistence – the lifetime – of topological features such as holes, voids, etc, in a filtration – a one parameter increasing family of spaces. These features are summarized in a persistence diagram, which shows the birth and death parameter values of the topological features. This is enabled by the algebraic result of being able to decompose any 1D persistence module into intervals [2,3]. The endpoints of these intervals are precisely the birth and death values of topological features.

The focus on one parameter families is a limitation of the current theory. While there is a need for practical tools applying the ideas of persistence to multiparametric data, multidimensional persistence [4] is known to be difficult to apply practically and in full generality. More precisely, there does not exist a complete discrete invariant that captures all the indecomposable modules in this setting. This is unlike the 1D case, where all indecomposables are guaranteed to be

* Corresponding author.
E-mail addresses: asashiba.hideto@shizuoka.ac.jp (H. Asashiba), buchet@tugraz.at (M. Buchet), e.g.escolar@people.kobe-u.ac.jp (E.G. Escolar), ken.nakashima@riken.jp (K. Nakashima), yosiwaki@sci.osaka-cu.ac.jp, michio.yoshiwaki@a.riken.jp (M. Yoshiwaki).

https://doi.org/10.1016/j.comgeo.2022.101879

intervals and where the persistence diagram is a complete invariant. In terms of representation theory, this difficulty can be expressed by the fact that the commutative nD grid is of wild type (see [5, Definition 6.4]) for $n \geq 2$ (and grid large enough).

One way to avoid these difficulties is to consider persistence modules that decompose into indecomposables contained in a restricted set. A promising candidate is the class of interval-decomposable persistence modules, which decompose into the so-called interval representations (see Definition 3). For example, the paper [6] provides a polynomial-time algorithm for computing the bottleneck distance between two 2D interval-decomposable persistence modules. The paper [7] studies stability for certain subclasses of interval-decomposable modules.

In this work, we focus not only on interval representations, but also study some other related classes of indecomposable persistence modules. One reason is that the definition of interval representations used in the literature [7,8,6] depends on a choice of bases and seems to be overly restrictive. For example, being an interval representation is not closed under isomorphisms (see Remark 11). This is unsatisfying from an algebraic/category-theoretic point of view. We review the definition of thin representations, introduce the new notion of pre-interval representations, and study the relationship among thin, pre-interval, and interval representations.

As one contribution of this work, we answer the following question in Section 3: Given an nD persistence module M, is there a way to determine, without explicitly computing its indecomposable decomposition, whether or not it is (pre)interval-decomposable or thin-decomposable? More generally, given some set $\mathscr{S}$ of indecomposable persistence modules, we provide in Theorem 18 equivalent conditions for determining $\mathscr{S}$-decomposability; that is, whether M is isomorphic to a direct sum of some elements of $\mathscr{S}$ possibly with multiplicity. In the case that $\mathscr{S}$ is a finite set, Theorem 18 translates into an implementable criterion.

In Section 4, we focus on the equioriented commutative 2D grid. This is the setting for 2-parameter persistence restricted to a finite number of parameter values in each parameter, with “equioriented” meaning that the maps are all in the same directions along each parameter, and with “commutative” meaning commutativity relations are imposed on the maps. See Section 2 for formal definitions. It is clear that over a 1D grid (i.e. the quiver $\vec{A}_n$, Section 2), being a thin indecomposable is equivalent to being isomorphic to an interval representation, since each indecomposable is isomorphic to an interval [3], and conversely, interval representations are automatically thin and indecomposable in general. In subsection 4.1, we show that this relationship holds also in the equioriented commutative 2D grid: any thin indecomposable is isomorphic to an interval representation. In subsection 4.2, we give examples for a non-equioriented commutative 2D grid and for an equioriented commutative 3D grid showing that this relationship does not hold in general. Finally, we provide a count of the total number of intervals in an equioriented commutative 2D grid in Theorem 31 by relating intervals in this setting to the so-called parallelogram polyominos.

In Section 5, we provide a detailed algorithm (Algorithm 1) for determining interval-decomposability, based on Theorem 18, and give its computational complexity. In particular, we give detailed descriptions of the computation of almost split sequences ending at interval representations and of dimensions of homomorphism spaces, which are used to compute multiplicities of interval summands. Furthermore, we propose several heuristics to reduce the number of interval representations to be checked.

Related to the question of determining interval-decomposability, we note the following results. Previous works [9,10] show that a pointwise finite-dimensional persistence module satisfies a certain local property called exactness if and only if it is rectangle-decomposable. Rectangles are intervals, and thus this result gives a criterion for a restricted class of interval-decomposables. Unfortunately, no such local criterion exists for interval-decomposability [11]. We note that our criterion in Theorem 18 is not local, as it relies on the computation of dimensions of certain homomorphism spaces.

In their paper [12], Dey and Xin gave an algorithm to decompose a restricted class of nD persistence modules M. Their algorithm proceeds on the assumption that the module is “distinctly graded”. One formulation of this condition is that there exists a projective presentation $P_1 \to P_0 \to M \to 0$ of M with both P_0 and P_1 square-free[1] modules. Furthermore, in version 5 of their arXiv preprint, they claim that their “algorithm can be applied to determine whether a persistence module is interval decomposable” [12]. When the module is not distinctly graded, one can arbitrarily fix an order on the grades. The number of possible orders is finite, and they claim that at least one of those orders provide a full decomposition of the module, and therefore it is enough to test all possible orders. However we argue that this claim is erroneous. We provide a counter-example by giving an interval-decomposable module M that is not distinctly graded and such that there exists no order on the grades that leads to a full decomposition by applying their algorithm.

2. Background

2.1. Quivers and their representations

We use the language of the representation theory of bound quivers. For more details, we refer the reader to the book [13], for example. Let us recall some basic definitions.

[1] A square-free module is a direct sum of non-isomorphic indecomposables.

A *quiver* is a quadruple $Q = (Q_0, Q_1, s, t)$ of sets Q_0, Q_1 and maps $s, t\colon Q_1 \to Q_0$. If we draw each $a \in Q_1$ as an arrow $a\colon s(a) \to t(a)$, then Q can be presented as a directed graph. Then we call elements of Q_0 (resp. elements of Q_1, $s(a)$ and $t(a)$) vertices of Q (resp. arrows of Q, the source of a and the target of a for each $a \in Q_1$). Let n be a positive integer. We denote by $\overrightarrow{A}_n$ the quiver presented as the directed graph

$$1 \longrightarrow 2 \longrightarrow \cdots \longrightarrow n \ .$$

The quiver $\overrightarrow{A}_n$ plays a central role in persistence theory.

A *subquiver* Q' of a quiver Q is a quiver $Q' = (Q'_0, Q'_1, s', t')$ such that $Q'_0 \subseteq Q_0$, $Q'_1 \subseteq Q_1$, and $s'(a) = s(a)$, $t'(a) = t(a)$ for all $a \in Q'_1$. A subquiver Q' is said to be *full* if it contains all arrows of Q between all pairs of vertices in Q'.

A *quiver morphism* from a quiver Q to a quiver Q' is a pair (f_0, f_1) of maps $f_0\colon Q_0 \to Q'_0$ and $f_1\colon Q_1 \to Q'_1$ such that $f_0 s = s' f_1$, $f_0 t = t' f_1$.

A *path* from a vertex x to a vertex y of length n (≥ 1) in Q is a sequence $\alpha_n \cdots \alpha_2 \alpha_1$ of arrows $\alpha_1, \alpha_2 \ldots, \alpha_n$ of Q such that $s(\alpha_1) = x$, $t(\alpha_n) = y$, and $s(\alpha_{i+1}) = t(\alpha_i)$ for all $1 \leq i \leq n-1$. Here we call x and y the *source* and the *target* of this path, respectively. For each vertex x, We also consider a symbol e_x as a path of length 0 from x to x. Note that each path of length $n \geq 0$ from x to y can be viewed as a quiver morphism $f : \overrightarrow{A}_{n+1} \to Q$ with $f(1) = x$ and $f(n+1) = y$.

Next, we give some definitions concerning convexity and connectedness in quivers.

Definition 1 *([13, p. 303], Convex subquiver).* Let Q be a quiver. A full subquiver Q' of Q is said to be *convex* in Q if and only if for all vertices x, y in Q'_0, and for all paths p from x to y in Q, all vertices of p are in Q'_0 (and thus p is a path in Q').

Definition 2 *(Connected).* A quiver Q is said to be *connected* if it is connected as an undirected graph, namely, if for each pair x, y of vertices of Q there exists a quiver W with underlying graph of the form $1 — 2 — \cdots — n$ for some n (≥ 1) and a quiver morphism $f\colon W \to Q$ such that $f(1) = x$, $f(n) = y$.

In this work, we consider only connected convex subquivers, which are also called interval subquivers.

Definition 3 *(Interval subquiver).* Let Q be a quiver. An *interval* of Q is a convex and connected subquiver of Q.

This definition is a generalization of the one [8,6] for commutative grids used in persistence theory. This in turn generalizes intervals of $\overrightarrow{A}_n$ in the usual sense: It is clear that the interval subquivers of $\overrightarrow{A}_n$ are precisely the full subquivers containing all vertices i for $b \leq i \leq d$, for some $b \leq d \in \mathbb{N}$. The interest in intervals comes mainly from the intuition about $\overrightarrow{A}_n$ in persistence theory: they form the building blocks of representations of $\overrightarrow{A}_n$, are simple to describe (parameters b and d only), and have a useful interpretation as the births and deaths of topological features.

Throughout this work, we let K be a field, and Q a quiver. Recall that a *K-representation* of Q is a family $V = (V(x), V(\alpha))_{x \in Q_0, \alpha \in Q_1}$, where $V(x)$ is a K-vector space for each vertex x, and $V(\alpha) : V(x) \to V(y)$ is a K-linear map for each arrow $\alpha : x \to y$. For example, the zero representation is $0 = (0, 0)_{x \in Q_0, \alpha \in Q_1}$.

A morphism $f : V \to W$ is a family $(f_x)_{x \in Q_0}$ of K-linear maps $f_x : V(x) \to W(x)$ such that $W(\alpha) f_x = f_y V(\alpha)$ for each arrow $\alpha : x \to y$ in Q_1. A *subrepresentation* W of V is a representation $W = (W(x), W(\alpha))_{x \in Q_0, \alpha \in Q_1}$ such that $W(x)$ is a vector subspace of $V(x)$ for each $x \in Q_0$ and the collection of inclusions $W(x) \to V(x)$ forms a morphism $W \to V$. The direct sum $V \oplus W$ of representations $V = (V(x), V(\alpha))_{x \in Q_0, \alpha \in Q_1}$ and $W = (W(x), W(\alpha))_{x \in Q_0, \alpha \in Q_1}$ is the representation $(V(x) \oplus W(x), V(\alpha) \oplus W(\alpha))_{x \in Q_0, \alpha \in Q_1}$. A representation V is *indecomposable* if $V \cong V_1 \oplus V_2$ implies $V_1 = 0$ or $V_2 = 0$. The *dimension* of a representation V is defined to be $\dim V = \sum_{x \in Q_0} \dim V(x)$. We call a representation V *finite-dimensional* if $\dim V < \infty$.

As an example, given $1 \leq b \leq d \leq n$, the interval representation $\mathbb{I}[b, d]$ of $\overrightarrow{A}_n$ is the representation

$$\mathbb{I}[b, d]\colon 0 \longrightarrow \cdots \longrightarrow 0 \longrightarrow \overset{b\text{-th}}{K} \longrightarrow K \longrightarrow \cdots \longrightarrow \overset{d\text{-th}}{K} \longrightarrow 0 \longrightarrow \cdots \longrightarrow 0,$$

which has the vector space $\mathbb{I}[b, d](i) = K$ at the vertices i with $b \leq i \leq d$, and 0 elsewhere, and where the maps between the neighboring vector spaces K are identity maps and zero elsewhere. It is known that $\left\{\mathbb{I}[b, d]\right\}_{1 \leq b \leq d \leq n}$ gives a complete list of indecomposable representations of $\overrightarrow{A}_n$, up to isomorphisms (see [3]).

In what follows, we need the concept of bound quivers. For that, we introduce the path algebra of a quiver Q. Set KQ to be the vector space having the set of paths in Q as its basis. If λ is a path from x to y, and μ a path from y' to z, then we define their composite $\mu\lambda$ to be zero in KQ if $y \neq y'$, otherwise, to be the path obtained by concatenating μ after λ. This defines a multiplication on KQ, and makes it a K-algebra, which is called the *path algebra* of Q. If Q_0 is finite, then KQ has the identity $1 = \sum_{x \in Q_0} e_x$.

Let J be the ideal of KQ generated by all arrows in Q. Then an ideal I of KQ is said to be *admissible* if $J^h \leq I \leq J^2$ for some $h \geq 2$. This is equivalent to the following formulation. First, paths in Q are said to be *parallel* if they have the same source and the same target. A *relation* of Q is a K-linear combination of parallel paths of length at least 2. Then I being admissible is equivalent to saying that (i) I contains all paths of length at least h (expressed by $J^h \leq I$) for some $h \geq 2$, and

(ii) I is generated by a set R of relations: $I = \langle R \rangle$ (expressed by $I \leq J^2$). Note that if Q is a finite acyclic quiver, then there is some $h \geq 2$ such that the length of any path in Q is less than h, which shows that $J^h = 0$. Therefore, in this case, an ideal I is admissible if and only if $I \leq J^2$.

A *bound quiver* is a pair (Q, I) of a quiver Q and an admissible ideal I of the path algebra KQ, which defines the algebra $K(Q, I) := KQ/I$.

Remark 4. In the setting above, we usually present the bound quiver (Q, I) just by (Q, R). Namely, (Q, R) is an abbreviation of $(Q, \langle R \rangle)$, and also set $K(Q, R) := KQ/\langle R \rangle$. Therefore, if sets of relations R and R' generate the same admissible ideal of KQ, then $(Q, R) = (Q, R')$.

A representation V of Q is said to be a *representation of the bound quiver* (Q, R) if V satisfies the relations given by the set of relations R (i.e., if $\sum_{i=1}^{n} t_i V(\mu_i) = 0$ for all $\sum_{i=1}^{n} t_i \mu_i \in R$ with μ_i paths and $t_i \in K$, where $V(\mu) = V(\alpha_m) \cdots V(\alpha_1)$ for a path $\mu = \alpha_m \cdots \alpha_1$). The category of finite-dimensional K-representations of (Q, R) will be denoted by $\mathrm{rep}_K(Q, R)$. In this work, we consider only finite-dimensional representations. It is well-known that the category $\mathrm{rep}_K(Q, R)$ is equivalent to the category of finite-dimensional left $K(Q, R)$-modules ([13, Theorem 1.6 of Chapter III]). We sometimes identify these two categories.

We delay giving examples of representations of bound quivers until after we introduce the bound quiver of interest in this paper, which is the equioriented commutative grid. The 2D persistence modules that is the main concern of this paper are exactly the representations of the equioriented commutative grid. Below, we define the equioriented grid by taking a product of $\vec{A}_n$.

First, we give the general definition of products of quivers.

Definition 5 *(Products of quivers).* Let $Q = (Q_0, Q_1, s, t)$ and $Q' = (Q'_0, Q'_1, s', t')$ be quivers.

- The Cartesian product $Q \times Q'$ is the quiver with the set of vertices $Q_0 \times Q'_0$ and the set of arrows $\{(x, a'), (a, x') \mid x \in Q_0, x' \in Q'_0, a \in Q_1, a' \in Q'_1\}$, where the sources and targets are determined by
$$(a, x') \colon (x, x') \to (y, x') \text{ if } a \colon x \to y,$$
$$(x, a') \colon (x, x') \to (x, y') \text{ if } a' \colon x' \to y'.$$
- The tensor product $Q \otimes Q'$ is the bound quiver $Q \times Q'$ with the ideal generated by the commutativity relations $(a, y')(x, a') - (y, a')(a, x')$ for all arrows $a \colon x \to y$ in Q and $a' \colon x' \to y'$ in Q'.

We also define the following special set of relations.

Definition 6 *(Full commutativity relations).* Let Q be a quiver. The set of *full commutativity relations* of Q is
$$R = \{p_1 - p_2 \mid p_1, p_2 \text{ are parallel paths of length} \geq 2 \text{ in } Q\}.$$

Definition 7 *(Equioriented commutative grid).* Let m, n be positive integers. The bound quiver $\vec{G}_{m,n} = \vec{A}_m \otimes \vec{A}_n$, which is the 2D grid of size $m \times n$ with all arrows in the same direction and with full commutativity relations (by Remark 4), is called the *equioriented commutative grid* of size $m \times n$.

In this work, we use the convention of displaying $\vec{G}_{m,n}$ as a 2D grid with m columns and n rows, with arrows pointing right or up. For example, $\vec{G}_{4,3} = \vec{A}_4 \otimes \vec{A}_3$ is the quiver

$$\begin{array}{ccccccc}
\bullet & \rightarrow & \bullet & \rightarrow & \bullet & \rightarrow & \bullet \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
\bullet & \rightarrow & \bullet & \rightarrow & \bullet & \rightarrow & \bullet \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
\bullet & \rightarrow & \bullet & \rightarrow & \bullet & \rightarrow & \bullet
\end{array}$$

with full commutativity relations.

There is also the following viewpoint: $\vec{G}_{m,n} = \vec{A}_m \otimes \vec{A}_n = (\vec{A}_m \times \vec{A}_n, R)$, where R is the full commutativity relations, can also be understood using the tensor product of path algebras. That is, we have $K\vec{G}_{m,n} = K(\vec{A}_m \otimes \vec{A}_n) \cong K\vec{A}_m \otimes_K K\vec{A}_n$ as algebras, where the notation $K(\vec{A}_m \otimes \vec{A}_n)$ is the quotient $K(\vec{A}_m \times \vec{A}_n)/R$ of the path algebra by the two-sided ideal generated by R (see [14, Proposition 3]).

While not a focus of this paper, we define the equioriented commutative nD grid of size $m_1 \times m_2 \times \ldots \times m_n$ as $\vec{G}_{m_1,\ldots,m_n} := \vec{A}_{m_1} \otimes \ldots \otimes \vec{A}_{m_n}$. Similarly, non-equioriented versions of the commutative nD grid can be defined by taking the tensor product of A_{m_i}-type quivers, where for at least one i, the arrows in ith factor are not pointing in the same direction.

Finally, let us give some small examples. While

$$\begin{tikzcd} K \arrow[r, "\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]"] & K^2 \arrow[r, "{[1\ 1]}"] & K \\ 0 \arrow[u] \arrow[r] & K \arrow[u, "\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]"] \arrow[r, "1"] & K, \arrow[u, "1"] \end{tikzcd}$$

is a representation of the quiver $\vec{A}_3 \times \vec{A}_2 : \begin{tikzcd}[sep=small] \bullet \arrow[r] & \bullet \arrow[r, "\alpha_4"] & \bullet \\ \bullet \arrow[u] \arrow[r] & \bullet \arrow[u, "\alpha_3"] \arrow[r, "\alpha_1"'] & \bullet \arrow[u, "\alpha_2"'] \end{tikzcd}$, it is *not* a representation of $\vec{G}_{3,2} = \vec{A}_3 \otimes \vec{A}_2$ because the commutativity relation $\alpha_4\alpha_3 - \alpha_2\alpha_1$ is not satisfied:

$$[\,1\ 1\,]\begin{bmatrix}1\\1\end{bmatrix} - 1 \cdot 1 = 2 - 1 = 1 \neq 0.$$

On the other hand, the following is an example of a representation of the bound quiver $\vec{G}_{3,2}$

$$\begin{tikzcd} K \arrow[r, "\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]"] & K^2 \arrow[r, "{[0\ 1]}"] & K \\ 0 \arrow[u] \arrow[r] & K \arrow[u, "\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]"] \arrow[r, "1"] & K. \arrow[u, "1"] \end{tikzcd}$$

2.2. Representations of interest

Throughout this section, we let (Q, R) be a bound quiver. We first start with the following straightforward definitions.

Definition 8 *(Thin representations).* A representation $V \in \mathrm{rep}_K(Q, R)$ is *thin* if $\dim_K V(x) \leq 1$ for each vertex x of Q.

Note that we do not require indecomposability for thin representations. If V is thin and indecomposable, we say that V is a thin indecomposable.

Definition 9 *([13, p. 93], Support).* Let $V \in \mathrm{rep}_K(Q, R)$. The *support* of V, denoted by $\mathrm{supp}(V)$, is the full subquiver of Q consisting of the vertices x with $V(x) \neq 0$.

For example, the support $\mathrm{supp}\,\mathbb{I}[b, d]$ of the interval representation $\mathbb{I}[b, d]$ of $\vec{A}_n$ is clearly the full subquiver with vertices i for $b \leq i \leq d$, which is an interval subquiver of $\vec{A}_n$ in the sense of Definition 3. It is also clear that $\mathbb{I}[b, d]$ is thin.

Next, we provide our definition of (pre-)interval representations of a general (bound) quiver, which generalizes the above example.

Definition 10 *(Interval and pre-interval representations).*

(1) A representation $V \in \mathrm{rep}_K(Q, R)$ is an *interval representation* if and only if

- (Thinness) it is thin, and
- (Interval support) its support $\mathrm{supp}(V)$ is an interval of Q, and
- (Identity over support) for all arrows $\alpha \in \mathrm{supp}(V)$, $V(\alpha)$ is an identity map.

Note that this definition is not stable under isomorphism (see Remark 11). Thus, in this work, by interval representation we also mean "isomorphic to an interval representation" if there is no risk of confusion.

(2) If, instead of the third condition (Identity), V satisfies the condition

- (Nonzero over support) for all arrows $\alpha \in \mathrm{supp}(V)$, $V(\alpha)$ is nonzero,

then V is said to be a *pre-interval representation*.

Recall that the support of a representation V is the full subquiver of vertices x with $V(x) \neq 0$. Thus, the "identity/nonzero over support" conditions means that if $V(x)$ and $V(y)$ are nonzero, then *all* arrows $\alpha : x \to y$ have $V(\alpha)$ identity or nonzero, respectively.

Remark 11.

(1) The condition "identity over support" implies that $V(x)$ and $V(y)$ are equal as (one-dimensional) vector spaces. This condition is not stable under isomorphisms. For example, consider $\overrightarrow{A}_2$ and its $\mathbb{R}$-representations

$$V: \ \mathbb{R} \xrightarrow{1} \mathbb{R} \quad \text{and } V': \ \mathbb{R} \xrightarrow{f} \mathbb{R}a$$

where f is the linear map determined by taking $f(1) = a$. Then, $V \cong V'$ and are both pre-interval. Clearly, V is interval, but V' is not. The one-dimensional vector spaces $\mathbb{R}$ and $\mathbb{R}a$ are not equal, only isomorphic.

(2) We introduced the pre-intervals as a more natural algebraic intermediary between the classes of intervals and thin indecomposables. From the above example, one might guess that the pre-intervals are simply the closure of the intervals under taking isomorphisms. In fact, this is true for the $2D$ equioriented commutative grid (Lemma 23). However, this is not true in general. In Section 4.2 we give examples where the three classes (thin indecomposable, pre-interval, isomorphic to an interval) are not equal. Thus, the pre-intervals form a class of its own that may be of interest in the general case.

(3) Under thinness and interval support conditions, a representation V is isomorphic to an interval representation if and only if there exist bases $v_x \in V(x)$ for all $x \in \mathrm{supp}(V)_0$ such that the following holds:

$$V(\alpha)(v_x) = v_y \text{ for each arrow } \alpha : x \to y \text{ in } \mathrm{supp}(V). \tag{2.1}$$

This condition will be used to show that a representation is isomorphic to an interval representation.

(4) If the coefficient field is $K = \mathbb{F}_2$, then every pre-interval representation is isomorphic to an interval representation. We note that in topological data analysis it is indeed common to choose the base field $\mathbb{F}_2$. Thus, it may seem that there is no need to consider the pre-interval representations. However, we note the following two reasons for considering fields other than $\mathbb{F}_2$.

First, homology over $\mathbb{F}_2$ does not capture topological torsion. Therefore, working with other fields provides more information. Second, decomposition of representations over $\mathbb{F}_2$ presents some deep algebraic complications, in the representation-infinite setting. An intuition into these complications can be obtained by contrasting the following two canonical forms arising in matrix decompositions. The Jordan canonical form, available over algebraically closed fields, is relatively simple compared to the rational canonical form, which involves irreducible polynomials in general. In this setting, decomposition over an infinite field (the algebraic closure) involves simpler summands. While we do not purse this line of inquiry further in this work, we note that the above comment concerning underlying fields and canonical forms generalizes to representations having different decompositions when considered over different underlying fields.

We note that our definition generalizes the usual definition of intervals and interval representations in the literature. For example, [7] and [8] defines intervals and interval representations over posets in general, and [6] over the poset $\bar{\mathbb{R}}^n := (\mathbb{R} \cup \{\infty\})^n$. It is clear that, given a poset P, we can construct an acyclic quiver with full commutativity relations (Q, R), and vice-versa, such that $\mathrm{rep}_K(Q, R)$ is equivalent to the category of pointwise finite-dimensional K-linear representations of P.

Then, it can be checked that an interval $\emptyset \neq I \subset P$ in the sense of [7,8] corresponds to a nonempty interval I in the sense of our Definition 3. In this setting, convexity corresponds to the condition that $a, c \in I$ and $a \leq b \leq c$ implies $b \in I$. On the other hand, connectedness corresponds to the condition that for any $a, c \in I$, there is a sequence $a = x_0, x_1, \ldots, x_\ell = c$ in I with x_i and x_{i+1} comparable for all $0 \leq i \leq \ell - 1$. Similarly, given an interval J, the interval module I^J as defined in Definition 2.1 of [8] is precisely an interval representation in the sense of our Definition 10 with support J.

Lemma 12 *(See also [8, Prop. 2.2]). Let V be a nonzero representation of (Q, R). If V is an interval or pre-interval representation, then V is indecomposable.*

Proof. The proof is similar to the proof of Prop. 2.2 in [8]. If V is an interval or pre-interval representation, then it is also thin, so without loss of generality, we assume that the vector spaces of V are K or 0. Then, endomorphisms of V act at each vertex by multiplication by some scalar. By commutativity requirements on endomorphisms together with the "nonzero over support" condition, each pair of these scalars over vertices in the same connected component are equal. Thus, by connectedness, $\mathrm{End}(V) \cong K$, and hence V is indecomposable. $\square$

In general, we have the following hierarchy of these classes of indecomposable representations:

$$\{V \mid V \cong \text{ an interval}\} \subset \{V \mid V \text{ pre-interval}\} \subset \{V \mid V \text{ thin indecomposable}\}. \tag{2.2}$$

Later, we shall show that for the equioriented commutative 2D grid, these three collections are equal. We shall also provide examples of where the inclusions are strict in the general case.

Finally, we provide the following definitions concerning these special classes of indecomposables.

Definition 13. Let (Q, R) be a bound quiver.

(1) A representation $V \in \operatorname{rep}_K(Q, R)$ is said to be *interval-decomposable* (resp. *pre-interval-decomposable*, *thin-decomposable*) if and only if each direct summand in some indecomposable decomposition of V is an interval representation (resp. pre-interval representation, thin representation).
(2) The bound quiver (Q, R) itself is said to be *interval-finite* (resp. *pre-interval-finite*, *thin-finite*) if and only if the number of isomorphism classes of its interval representations (resp. pre-interval representations, thin indecomposables) is finite.

In the rest of this work, we consider only bound quivers (Q, R) such that $KQ/\langle R\rangle$ is a finite-dimensional K-algebra. This holds, for example, if $\langle R\rangle$ is an admissible ideal, or if Q is a finite acyclic quiver. With this assumption, we can use the Auslander-Reiten theory needed for the next section. Furthermore, we fix a complete set $\mathscr{L}$ of representatives of isomorphism classes of (finite-dimensional) indecomposable representations of (Q, R). That is, for each $X \in \operatorname{rep}_K(Q, R)$ let $[X] = \{Y \in \operatorname{rep}_K(Q, R) \mid Y \cong X\}$ be its isomorphism class, and let $\widetilde{\mathscr{L}} = \{[X] \mid X \in \operatorname{rep}_K(Q, R) \text{ indecomposable}\}$ be the set of all isomorphism classes of indecomposable representations of (Q, R). Then, $\mathscr{L}$ is a set formed by choosing one representative $L \in [X]$ for each isomorphism class $[X]$ in $\widetilde{\mathscr{L}}$. Note that $\mathscr{L}$ can be identified with the set of vertices of the Auslander-Reiten quiver of (Q, R). For more details on the Auslander-Reiten theory, we refer the reader to the books [13,15].

2.3. Decomposition theory

We consider only bound quivers (Q, R) such that $A := KQ/\langle R\rangle$ is a finite-dimensional K-algebra. First recall the Krull-Schmidt Theorem, which can be stated as follows.

Theorem 14 *(Krull-Schmidt). For each representation M of (Q, R) there exists a unique function $d_M : \mathscr{L} \to \mathbb{Z}_{\geq 0}$ such that $M \cong \bigoplus_{L\in\mathscr{L}} L^{d_M(L)}$. Therefore for each pair M, N of representations of (Q, R) we have $M \cong N$ if and only if $d_M = d_N$.*

In this subsection, let us review decomposition theory [16,17] which gives an algorithm to compute the multiplicity $d_M(L)$ for all $L \in \mathscr{L}$ by using Auslander-Reiten theory. For the details of Auslander-Reiten theory, we refer the reader to [13, Chapter IV] or [15, Chapter V].

Here, we briefly provide the definitions required for Theorem 17 and its dual.

For a representation M recall that the sum of all simple[2] submodules of M is called the *socle* of M, denoted by $\operatorname{soc} M$, and that the intersection of the kernels of all homomorphisms from M to simple modules is called the *radical* of M, denoted by $\operatorname{rad} M$. We set $\operatorname{top} M := M/\operatorname{rad} M$, and call it the *top* of M. Note that $\operatorname{top} P$ of an indecomposable projective representation P and $\operatorname{soc} I$ of an indecomposable injective representation I are simple.

Definition 15. Let $f : X \to Y$ be a morphism of representations of (Q, R).

(1) f is said to be *left minimal* (resp. *right minimal*) if for any morphism $h \in \operatorname{End}(Y)$ (resp. $h \in \operatorname{End}(X)$) $hf = f$ (resp. $fh = f$) implies that h is an automorphism.
(2) A non-section (resp. non-retraction)[3] f is said to be *left almost split* (resp. *right almost split*) if for every non-section $u : X \to M$ (resp. non-retraction $u : M \to Y$) there is a morphism $v : Y \to M$ such that $vf = u$ (resp. a morphism $v : M \to X$ such that $fv = u$).
(3) f is called a *source map* (resp. *sink map*) if f is both left minimal and left almost split (resp. right minimal and right almost split).

A short exact sequence $0 \to X \xrightarrow{f} Y \xrightarrow{g} Z \to 0$ is an *almost split sequence* if f is a source map and g is a sink map. Source maps and sink maps from every indecomposable representation are given as follows, which is a fundamental theorem in Auslander-Reiten theory:

Theorem 16 *([13, Chapter IV Thm. 3.1 and Prop. 3.5]). Let L be an indecomposable representation of (Q, R), $f : L \to U$ a source map and $g : V \to L$ a sink map. Then*

(1) *If L is injective, then f is given by the composite of the canonical epimorphism $L \to L/\operatorname{soc} L$ followed by an isomorphism $L/\operatorname{soc} L \to U$. In particular, $U \cong L/\operatorname{soc} L =: {}_L E$.*

[2] A module S is said to be *simple* if S has exactly two submodules S and 0.
[3] A homomorphism $f : X \to Y$ is called a *non-section* (resp. *non-retraction*) if $f'f \neq 1_X$ (resp. $ff' \neq 1_Y$) for all $f' : Y \to X$.

(2) *If L is not injective, then f is given by the composite of α followed by an isomorphism ${}_LE \to U$, where*

$$0 \to L \xrightarrow{\alpha} {}_LE \xrightarrow{\beta} \tau^{-1}L \to 0$$

is an almost split sequence. In particular, $U \cong {}_LE$.

(3) *If L is projective, then g is given by the composite of an isomorphism $V \to \operatorname{rad} L$ followed by the inclusion map $\operatorname{rad} L \to L$. In particular, $V \cong \operatorname{rad} L =: E_L$.*

(4) *If L is not projective, then g is given by the composite of an isomorphism $V \to E_L$ followed by β, where*

$$0 \to \tau L \xrightarrow{\alpha} E_L \xrightarrow{\beta} L \to 0$$

is an almost split sequence. In particular, $V \cong E_L$.

Here, $\tau := D\,\mathrm{Tr}$, $\tau^{-1} := \mathrm{Tr}\,D$, where $D = \mathrm{Hom}_K(-, K)$ is the standard K-vector space dual, and where Tr *denotes the transpose (see [13, Chapter IV.2]).*

To define the transpose, we first recall that for a representation X, its *minimal projective presentation* is an exact sequence $P_1 \xrightarrow{p_1} P_0 \xrightarrow{p_0} X \longrightarrow 0$ formed from the projective covers $P_0 \xrightarrow{p_0} X$ and $P_1 \xrightarrow{p_1} \mathrm{Ker}(p_0)$ (for the definition of projective covers, see [13, Chapter I, Definition 5.5]). Applying the A-dual $(-)^t = \mathrm{Hom}_A(-, A)$, where $A := KQ/\langle R\rangle$, we obtain the exact sequence $0 \longrightarrow X^t \xrightarrow{p_0^t} P_0^t \xrightarrow{p_1^t} P_1^t$. The transpose of X is simply $\mathrm{Tr}\,X := \mathrm{Coker}\,p_1^t$.

Returning to Theorem 16, for each indecomposable representation L of (Q, R) we can decompose ${}_LE$ and E_L as

$${}_LE = \bigoplus_{X \in J_L} X^{(a_L(X))}, \quad E_L = \bigoplus_{X \in K_L} X^{(b_L(X))}$$

for a unique subset J_L (resp. K_L) of $\mathscr{L}$ and a unique function $a_L\colon J_L \to \mathbb{Z}_{>0}$ (resp. $b_L\colon K_L \to \mathbb{Z}_{>0}$). Recall that L is injective if and only if $\tau^{-1}L = 0$ (dual version of [15, Prop. IV.1.10(b)]), and that L is projective if and only if $\tau L = 0$.

Theorem 17 *([16, Thm. 3], [17, Cor. 2.3]). Let $M \in \mathrm{rep}_K(Q, R)$ and $L \in \mathscr{L}$. Then $d_M(L)$ is computed by the following four formulae:*

$$d_M(L) = \dim \mathrm{Hom}(L, M) - \dim \mathrm{Hom}({}_LE, M) + \dim \mathrm{Hom}(\tau^{-1}L, M), \tag{2.3}$$

$$d_M(L) = \dim \mathrm{Hom}(L, M) - \sum_{X \in J_L} a_L(X) \dim \mathrm{Hom}(X, M) + \dim \mathrm{Hom}(\tau^{-1}L, M), \tag{2.4}$$

$$d_M(L) = \dim \mathrm{Hom}(M, \tau L) - \dim \mathrm{Hom}(M, E_L) + \dim \mathrm{Hom}(M, L), \tag{2.5}$$

$$d_M(L) = \dim \mathrm{Hom}(M, \tau L) - \sum_{X \in K_L} b_L(X) \dim \mathrm{Hom}(M, X) + \dim \mathrm{Hom}(M, L). \tag{2.6}$$

Here, the number $\dim \mathrm{Hom}(M, N)$ is the dimension of the K-vector space of homomorphisms from M to N. For an indecomposable representation X of (Q, R) the function

$$s_X := \dim \mathrm{Hom}(X, \text{-})\colon\ \mathrm{rep}(Q, R) \to \mathbb{Z}_{\geq 0}, \quad M \mapsto \dim \mathrm{Hom}(X, M)$$

is called the *starting function* from X. Dually,

$$t_X := \dim \mathrm{Hom}(\text{-}, X)\colon\ \mathrm{rep}(Q, R) \to \mathbb{Z}_{\geq 0}, \quad M \mapsto \dim \mathrm{Hom}(M, X)$$

is called the *stopping function* to X. Using these, the formulae (2.4) and (2.6) have the following forms:

$$d_M(L) = s_L(M) - \sum_{X \in J_L} a_L(X) s_X(M) + s_{\tau^{-1}L}(M). \tag{2.7}$$

$$d_M(L) = t_{\tau L}(M) - \sum_{X \in K_L} b_L(X) t_X(M) + t_L(M). \tag{2.8}$$

Note that the value of $s_X(M)$ (or $t_X(M)$) can be computed as the rank of some matrix defined by M for each X (see [16] for details).

For completeness we added the dual versions (2.5) and (2.6), which were not presented in [16, Thm. 3]. Later we will use formula (2.5) to examine the computational complexity of our algorithm for determining interval-decomposability.

3. Determining $\mathscr{S}$-decomposability

To state our theorem, we first generalize the idea of interval-decomposability and thin-decomposability in the following way. Let $\mathscr{S}$ be a subset of the chosen complete set $\mathscr{L}$ of representatives of isomorphism classes of indecomposable representations. Then, $M \in \operatorname{rep}(Q, R)$ is said to be *$\mathscr{S}$-decomposable* if and only if $M \cong \bigoplus_{L\in\mathscr{S}} L^{d_M(L)}$. In this section, we use the decomposition theory to determine whether or not a given persistence module is $\mathscr{S}$-decomposable, provided $\mathscr{S}$ is finite.

Theorem 18. *Let $\mathscr{S}$ be a subset of $\mathscr{L}$, and $M \in \operatorname{rep}(Q, R)$. Then the following are equivalent.*

(1) *M is $\mathscr{S}$-decomposable;*
(2) $\dim M = \sum\limits_{L\in\mathscr{S}} d_M(L) \dim L$;
(3) $\dim M + \sum\limits_{L\in\mathscr{S}} \sum\limits_{X\in J_L} a_L(X) s_X(M) \dim L = \sum\limits_{L\in\mathscr{S}} (s_L(M) + s_{\tau^{-1}L}(M)) \dim L$;
(4) $\dim M + \sum\limits_{L\in\mathscr{S}} \sum\limits_{X\in K_L} b_L(X) t_X(M) \dim L = \sum\limits_{L\in\mathscr{S}} (t_{\tau L}(M) + t_L(M)) \dim L$.

Proof. The isomorphism $M \cong \left(\bigoplus\limits_{L\in\mathscr{S}} L^{d_M(L)}\right) \oplus \left(\bigoplus\limits_{L\in\mathscr{L}\setminus\mathscr{S}} L^{d_M(L)}\right)$ shows that

$$\dim M = \sum_{L\in\mathscr{S}} d_M(L) \dim L + \sum_{L\in\mathscr{L}\setminus\mathscr{S}} d_M(L) \dim L$$

Then we have equivalences (1) $\Leftrightarrow$ $\bigoplus_{L\in\mathscr{L}\setminus\mathscr{S}} L^{d_M(L)} = 0$ $\Leftrightarrow$ $\sum_{L\in\mathscr{L}\setminus\mathscr{S}} d_M(L) \dim L = 0$ $\Leftrightarrow$ (2). The equivalences (2) $\Leftrightarrow$ (3) $\Leftrightarrow$ (4) follow from Equations (2.7) and (2.8). $\square$

As a generalization of Definition 13(2), in the case that $\mathscr{S}$ is finite for a given quiver Q, we say that Q is $\mathscr{S}$-finite. We remark that Theorem 18 gives us a criterion to determine the $\mathscr{S}$-decomposability of a given $M \in \operatorname{rep}(Q, R)$. In particular, we only need to consider a finite number of values $d_M(X)$ for $X \in \mathscr{S}$ and then compare $\dim M$ with $\sum_{X\in\mathscr{S}} d_M(X) \dim X$. If these values are equal, then the given $M \in \operatorname{rep}(Q, R)$ is $\mathscr{S}$-decomposable by the implication (2) $\Rightarrow$ (1). The formula (3) gives a criterion for M to be $\mathscr{S}$-decomposable by using the function dim and the values $s_X(M)$ of starting functions from indecomposable representations $X \in \mathscr{S} \cup (\bigcup_{L\in\mathscr{S}} J_L)$, on which the computation of $d_M(L)$ depends.

Thus, it is important to determine whether or not a particular bound quiver is $\mathscr{S}$-finite. In particular, we are concerned with thin-finiteness or (pre-)interval-finiteness. In Section 4, we study the equioriented commutative 2D grid. Here, we give the following trivial observations of some settings where the criterion given by Theorem 18 can immediately be applied.

Lemma 19. *Let Q be a finite (bound) quiver, and K a finite field. Then Q is thin-finite.*

Proof. Consider the number of possible thin representations of Q. Since Q has a finite number of arrows, and over each arrow, a thin representation V can only have (up to isomorphism) $f : K \to K$ or $K \to 0$ or $0 \to K$, where there are only a finite number of possibilities for $f \in \operatorname{Hom}(K, K) \cong K^{\mathrm{op}}$. Thus the number of possible thin representations (up to isomorphism) of Q is finite. $\square$

Note that because of the hierarchy in Ineq. (2.2), thin-finiteness implies pre-interval-finiteness. For finite quivers, interval-finiteness is automatic and does not require the finiteness of field K, as the next lemma shows.

Lemma 20. *Let Q be a finite (bound) quiver. Then Q is interval-finite.*

Proof. This follows by a similar counting argument for interval representations V as in the previous lemma, but this time the only possibility for $f : K \to K$ is the identity since V is an interval representation. $\square$

4. Equioriented commutative 2D grid

In this section, we focus our attention on the equioriented commutative 2D grid $\vec{G}_{m,n}$. We show that each thin indecomposable of $\vec{G}_{m,n}$ is isomorphic to an interval representation and enumerate all interval representations of $\vec{G}_{m,n}$.

4.1. 2D thin indecomposables are interval representations

First, let us show that interval subquivers of $\vec{G}_{m,n}$ can only have a "staircase" shape. To make this more precise, we define the following.

Let m and n be fixed positive integers, and let $\mathbb{I}_{m,n}$ be the set of all nonempty interval subquivers of $\vec{G}_{m,n}$. For $1 \leq j \leq n$, a *slice* at row j is a pair of numbers $1 \leq b_j \leq d_j \leq m$, denoted $[b_j, d_j]_j$. For $1 \leq s \leq t \leq n$, a *staircase* from s to t is a set of slices $[b_j, d_j]_j$ for $s \leq j \leq t$ such that $b_{j+1} \leq b_j \leq d_{j+1} \leq d_j$ for any $j \in \{s, \ldots, t-1\}$. To make explicit the constants m and n, we say that such a set of slices is a staircase of $\vec{G}_{m,n}$.

Proposition 21. *Let $\mathbb{I}'_{m,n}$ be the set of all staircases of $\vec{G}_{m,n}$. There exists a bijection between $\mathbb{I}_{m,n}$ and $\mathbb{I}'_{m,n}$.*

Proof. We construct a set bijection $f : \mathbb{I}_{m,n} \longrightarrow \mathbb{I}'_{m,n}$ together with its inverse f^{-1}.

For each interval subquiver $I \in \mathbb{I}_{m,n}$, we define $f(I)$ to be the set of slices $f(I) := \{[b_j, d_j]_j \mid s \leq j \leq t\}$ from s to t, where

$$s = \min\{j \mid (i, j) \in I \text{ for some } i\} \text{ and } t = \max\{j \mid (i, j) \in I \text{ for some } i\}$$

and for $s \leq j \leq t$,

$$b_j = \min\{i \mid (i, j) \in I\} \text{ and } d_j = \max\{i \mid (i, j) \in I\}.$$

Note that since I is nonempty, $1 \leq s \leq t \leq n$. Then, for each j with $s \leq j \leq t$, the set $\{i \mid (i, j) \in I\}$ is nonempty by the connectedness condition, and thus $1 \leq b_j \leq d_j \leq m$. Similarly, $b_j \leq d_{j+1}$ follows from the connectedness of I.

The correctness of conditions $b_{j+1} \leq b_j$ and $d_{j+1} \leq d_j$ follows from the convexity of I. To see this, suppose to the contrary that $b_{j+1} > b_j$. Then, we have a path (b_j, j) to $(b_j, j+1)$ to $(b_{j+1}, j+1)$ with both endpoints in I, but $(b_j, j+1)$ is not in I since $b_j < b_{j+1} = \min\{i \mid (i, j+1) \in I\}$. This contradicts convexity. A similar argument shows that $d_{j+1} \leq d_j$. The above arguments show that $f(I)$ is indeed a staircase.

In the opposite direction, given a staircase $I' := \{[b_j, d_j]_j \mid s \leq j \leq t\}$ from s to t, we define $f^{-1}(I')$ to be the full subquiver with vertices

$$\{(i, j) \mid s \leq j \leq t, b_j \leq i \leq d_j\}.$$

It is clear that f and f^{-1} are inverses of each other. $\square$

In general, for a representation $V \in \operatorname{rep}_K(Q, R)$ with $\#Q_0 = n$, the *dimension vector* of V is defined to be

$$\underline{\dim} V := (\dim_K V(x))_{x \in Q_0} \in \mathbb{Z}^n.$$

When we display dimension vectors, we position the numbers $\dim_K V(x)$ corresponding to the position where each vertex $x \in Q_0$ is graphically displayed (see Example 22). By definition, each interval representation M of $\vec{G}_{m,n}$ can be uniquely expressed by its dimension vector, since it is uniquely determined by its support.

By Proposition 21, we identify interval subquivers of $\vec{G}_{m,n}$ with staircases of $\vec{G}_{m,n}$. Thus, we shall also denote an interval by writing it as a set of slices $\{[b_j, d_j]_j \mid s \leq j \leq t\}$, as a staircase from s to t. We can visualize the correspondence $f : \mathbb{I}_{m,n} \longrightarrow \mathbb{I}'_{m,n}$ in the proof of Proposition 21 using the dimension vector notation and staircase notation. Below, we illustrate some examples under this correspondence for $\vec{G}_{6,4}$.

Example 22. The following are examples of intervals in $\vec{G}_{6,4}$.

$$\begin{pmatrix} 011100 \\ 001100 \\ 001110 \\ 000011 \end{pmatrix} \longleftrightarrow \{[5,6]_1, [3,5]_2, [3,4]_3, [2,4]_4\}, \begin{pmatrix} 000000 \\ 011100 \\ 001110 \\ 000000 \end{pmatrix} \longleftrightarrow \{[3,5]_2, [2,4]_3\}.$$

Using this staircase shape, we are able to prove the following

Lemma 23. *Let m, n be positive integers. Any pre-interval representation of $\vec{G}_{m,n}$ is isomorphic to an interval representation.*

Proof. Let V be a pre-interval representation of $\vec{G}_{m,n} = \vec{A}_m \otimes \vec{A}_n$. Then $\operatorname{supp}(V)$ is an interval by definition, and thus a staircase by Proposition 21. Set B to be the quiver $\operatorname{supp}(V)$ with full commutativity relations in it. Then V is regarded as a representation of B.

Let B' be the bound quiver obtained from $\operatorname{supp}(V)$ by flipping all of its vertical arrows, together with the full commutativity relations. Thus the quiver of B' is a subquiver of $\vec{A}_m \otimes (\vec{A}_n)^{\mathrm{op}}$. Then by replacing all maps of V associated to the vertical arrows in $\operatorname{supp}(V)$ by their inverses, we obtain a representation V' of the bound quiver B'. Note that on $\operatorname{supp}(V)$, the maps associated by V are nonzero maps between one-dimensional vector spaces by the definition of a pre-interval representation, and thus have well-defined and unique inverses. To see that the commutative relations in B' are satisfied by V', we note that the left square below is a commutative diagram of nonzero linear maps if and only if the right one is:

$$
\begin{tikzcd}
K \arrow[r, "a_1"] & K \\
K \arrow[u, "b_1"] \arrow[r, "a_2"'] & K \arrow[u, "b_2"']
\end{tikzcd}
\qquad
\begin{tikzcd}
K \arrow[r, "a_1"] \arrow[d, "b_1^{-1}"'] & K \arrow[d, "b_2^{-1}"] \\
K \arrow[r, "a_2"'] & K,
\end{tikzcd}
$$

because $a_1 b_1 = b_2 a_2$ is equivalent to $b_2^{-1} a_1 = a_2 b_1^{-1}$.

We illustrate the construction with the following example, showing the quiver of B and B', respectively:

$$
\begin{tikzcd}
\circ \arrow[r] & \circ \arrow[r] & \circ \arrow[r] & \circ & \\
\circ \arrow[u] \arrow[r] & \circ \arrow[u] \arrow[r] & \circ \arrow[u] \arrow[r] & \circ \arrow[u] \arrow[r] & \circ \\
 & & \circ \arrow[u] \arrow[r] & \circ \arrow[u] \arrow[r] & \circ \arrow[u]
\end{tikzcd}
\qquad
\begin{tikzcd}
x \arrow[r] \arrow[d] & \circ \arrow[r] \arrow[d] & \circ \arrow[r] \arrow[d] & \circ \arrow[d] & \\
\circ \arrow[r] & \circ \arrow[r] & \circ \arrow[r] \arrow[d] & \circ \arrow[r] \arrow[d] & \circ \arrow[d] \\
 & & \circ \arrow[r] & \circ \arrow[r] & \circ.
\end{tikzcd}
\tag{4.1}
$$

We view V' as a representation of B' and not of $\vec{A}_m \otimes (\vec{A}_n)^{\mathrm{op}}$. So for example, there is no problem with the upper right portion of the quiver of B' in Diagram (4.1) not satisfying a zero relation.

In general, let x be the upper left corner of the quiver of B', and take a nonzero element v_x of $K = V'(x)$. For each vertex y of B' there exists a path μ from x to y in the quiver of B', because $\mathrm{supp}(V)$ has a staircase shape. Take $v_y := V'(\mu) v_x$ as the basis of $V'(y)$. Since B' is defined by the full commutativity relations, v_y does not depend on the choice of μ. In this way we can find bases v_y of $V'(y)$ for all vertices y in $\mathrm{supp}(V')$ that satisfy Condition (2.1) in Remark 11. Now v_y are also bases of $V(y) = V'(y)$ for all $y \in \mathrm{supp}(V)_0$ and satisfy Condition (2.1) for V. Thus V is isomorphic to an interval representation. $\square$

Finally, we prove the main result of this subsection.

Theorem 24. *Let m, n be positive integers. Let M be a thin indecomposable representation of the commutative grid $\vec{G}_{m,n} = (Q, R)$. Then M is isomorphic to an interval representation.*

Proof. The proof will be done by contradiction and in two steps. First we show that any thin indecomposable representation that is not a pre-interval should have two non-zeros vector spaces with a path containing a zero map between them. Then we will show that this implies that the representation is decomposable. Lemma 23 will then allow us to conclude.

Assume for contradiction that M is a thin indecomposable that is not a pre-interval representation. As M is an indecomposable representation, its support is connected. Therefore, either the convexity condition on the support of M fails, or the nonzero maps on support condition fails. In the first case, there exist vertices $x, y, z \in Q_0$ such that there is a path from x to y to z, and $M(x) \neq 0$, $M(y) = 0$ and $M(z) \neq 0$. In the second case, there exists an arrow $\alpha : x \to z$ in Q_1 with $M(x) \neq 0$, $M(z) \neq 0$ and $M(\alpha) = 0$.

In either case, we have a path p from x to z with $M(x) \neq 0$ and $M(z) \neq 0$ such that p contains an arrow α with $M(\alpha) = 0$.

Let us consider the representation M on a square with one corner at $(i, j) \in Q_0$ in the grid:

$$
\begin{tikzcd}
M(i, j+1) \arrow[r, "t"] & M(i+1, j+1) \\
M(i, j) \arrow[u, "l"] \arrow[r, "b"'] & M(i+1, j) \arrow[u, "r"']
\end{tikzcd}
\tag{4.2}
$$

where the maps are the values of the representation M on the arrows (for example, $r := M(\beta)$ where β is the arrow $\beta : (i+1, j) \to (i+1, j+1)$). By full commutativity, the two paths (compositions of maps) from $M(i, j)$ to $M(i+1, j+1)$ are equal: $rb = tl$. Since the vector spaces have dimension at most 1 as M is thin, we can conclude the following. If at least one map is zero on one of these paths, then there is a zero map on the other path.

We use the above observation to build a line L intersecting only zero maps in M across the grid that separates M. We start with the arrow α with $M(\alpha) = 0$ found previously and inductively build this line using the following observation. At each square of the grid, at least one of the following patterns is possible:

$$
\begin{tikzcd}
M(c) \arrow[r] & M(d) \\
M(a) \arrow[u] \arrow[r] & M(b) \arrow[u]
\end{tikzcd}
\quad
\begin{tikzcd}
M(c) \arrow[r] & M(d) \\
M(a) \arrow[u] \arrow[r] & M(b) \arrow[u]
\end{tikzcd}
\quad
\begin{tikzcd}
M(c) \arrow[r] & M(d) \\
M(a) \arrow[u] \arrow[r] & M(b) \arrow[u]
\end{tikzcd}
\quad
\begin{tikzcd}
M(c) \arrow[r] & M(d) \\
M(a) \arrow[u] \arrow[r] & M(b) \arrow[u]
\end{tikzcd}
\tag{4.3}
$$

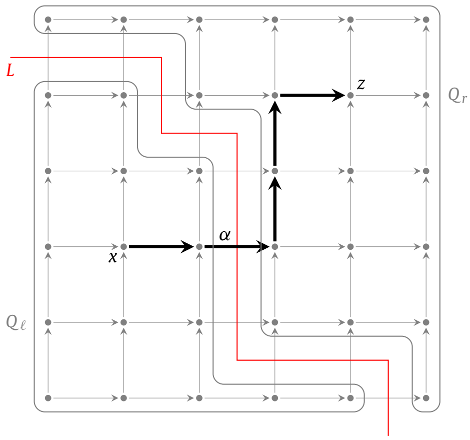


Fig. 1. Starting with the detected path p (thick line) from x to z with $M(x)$ and $M(z)$ nonzero, we find an arrow α with $M(\alpha) = 0$. This region of zeros propagates to the red line L, which divides M into two

where in each pattern, the line (colored red) intersects a pair of arrows β_1, β_2 where $M(\beta_1) = 0$ and $M(\beta_2) = 0$. Note that if more maps are zero, we simply ignore them and choose to extend our line using only one of the four given patterns.

As we are working over a finite 2D grid, this line cannot create a circle. Therefore it goes from one boundary of the grid to another, and divides the grid into two regions with vertices we denote by V_ℓ and V_r, for "left/bottom" and "right/top", respectively. Furthermore, both regions are non-trivial: by construction, $x \in V_\ell$ and $z \in V_r$ with $M(x) \neq 0$ and $M(z) \neq 0$ since the arrow α was found as part of a path from vertex x to z with those properties.

Let $Q_\ell = (V_\ell, E(V_\ell))$ and $Q_r = (V_r, E(V_r))$ be the full subquivers generated by V_ℓ and V_r respectively, and let $E(L)$ be the set of the arrows intersecting the line L constructed above. Then, the grid is partitioned as $\overrightarrow{G}_{m,n} = (Q_0, Q_1) = (V_\ell \sqcup V_r, E(V_\ell) \sqcup E(L) \sqcup E(V_r))$. To see this, we note that by construction $E(V_\ell)$ and $E(V_r)$ are disjoint. Furthermore, $E(L)$ is by definition the arrows going from a vertex of V_ℓ to V_r, and is disjoint from $E(V_\ell)$ and $E(V_r)$. Finally, each arrow on the grid is in one of these three sets. In Fig. 1, we illustrate this partitioning.

Consider representations M_ℓ and M_r obtained by setting M to be zero outside of Q_ℓ and Q_r respectively. The support of M_ℓ is included in Q_ℓ. Note that by construction the arrows exiting Q_ℓ are exactly the arrows $E(L)$, which all support a zero map in M. Hence M_ℓ is a subrepresentation of M. Clearly, M_r is a subrepresentation of M since there are no arrows exiting Q_r. Furthermore, as M restricted to $E(L)$ is 0, we conclude that $M = M_\ell \oplus M_r$.

By the fact that $M_l(x) \neq 0$ and $M_r(z) \neq 0$, it follows that the decomposition above is nontrivial, and thus M is decomposable, a contradiction. Therefore M is a pre-interval representation, and Lemma 23 implies that M is isomorphic to an interval representation. □

4.2. Interesting examples

In this subsection, we give some interesting examples of where a thin indecomposable may not be isomorphic to an interval representation.

Over the equioriented commutative 3D grid, we provide the following example. Let λ be any element of K, and define

$$M(\lambda): \begin{tikzcd} & K \arrow[rr] & & 0 \\ K \arrow[ur, "1"] \arrow[rr, "1"] & & K \arrow[ur] & \\ & K \arrow[uu, "1"] \arrow[rr, "1"] & & K \arrow[uu] \\ 0 \arrow[uu] \arrow[ur] \arrow[rr] & & K \arrow[uu, "1"] \arrow[ur, "\lambda"] & \end{tikzcd}$$

This, and higher-dimensional versions of this indecomposable were studied in the paper [18], where topological realizations were also given for $\lambda = 0$. It is easy to see that $M(\lambda)$ is indecomposable, and for any $\lambda \neq \mu \in K$, $M(\lambda) \ncong M(\mu)$. Furthermore, $M(\lambda)$ is thin. However, if $\lambda \neq 1$, $M(\lambda)$ is not an interval representation, and is not isomorphic to one. Moreover $M(0)$ is not a pre-interval representation and is not isomorphic to one but is still a thin indecomposable.

We note that current applications of topological data analysis do not go beyond the 2D case. However, we are the laying the groundwork for further explorations into the nD case, where this distinction between thin and interval representations becomes relevant. In particular, in the above example, λ can be interpreted as a kind of torsion that cannot be seen using intervals.

Next, if the arrows are not oriented in the same direction, some thin indecomposables may not be interval representations. An example is the representation

$$\begin{tikzcd} K & K \arrow[l, "1"'] \arrow[r, "1"] & K \\ K \arrow[u, "1"] \arrow[d, "1"'] & 0 \arrow[l] \arrow[r] \arrow[u] \arrow[d] & K \arrow[u, "1"'] \arrow[d, "1"] \\ K & K \arrow[l, "1"] \arrow[r, "\lambda"'] & K \end{tikzcd} \tag{4.4}$$

of a non-equioriented commutative 2D grid, where λ is not 1. If λ is not 0 and not 1, this also gives an example of a pre-interval representation that is not an interval representation (and not isomorphic to one).

The above are variations on the same theme: we have an example of a thin indecomposable that is not pre-interval representation (when $\lambda = 0$), and an example of a pre-interval representation that is not isomorphic to an interval representation (when λ is not 0 nor 1). Hence we have strict inclusions in the hierarchy of Ineq. (2.2).

Next, let us provide an example of a bound quiver (Q, R) where pre-interval representations are always isomorphic to an interval representation, but thin indecomposables are not always pre-interval representations. Consider the quiver

$$\begin{tikzcd} \bullet \arrow[r, bend right, "x"'] & \bullet \arrow[l, bend right, "y"'] \end{tikzcd}$$

with relations $R = \{xy - xyxy, yx - yxyx\}$. Then,

$$\begin{tikzcd} K \arrow[r, bend right, "0"'] & K \arrow[l, bend right, "1"'] \end{tikzcd}$$

is an example of a thin indecomposable representation of (Q, R) that is not a pre-interval representation.

Now suppose that $V \in \mathrm{rep}_K(Q, R)$ is a pre-interval representation. In the case that V is a simple representation, it is automatically an interval representation. Otherwise, V is isomorphic to some

$$\begin{tikzcd} K \arrow[r, bend right, "f"'] & K \arrow[l, bend right, "g"'] \end{tikzcd}$$

Then, the relations R imply that $fg - fgfg = 0$ and $gf - gfgf = 0$. Together with the fact that f and g are nonzero because V is a pre-interval representation, we see that f and g are mutually-inverse isomorphisms. Thus, V is isomorphic to

$$\begin{tikzcd} K \arrow[r, bend right, "1"'] & K \arrow[l, bend right, "1"'] \end{tikzcd}$$

which is an interval representation.

4.3. Listing all 2D intervals

By definition, an interval representation can be uniquely identified with its support, an interval subquiver. Recall that $\mathbb{I}_{m,n}$ is the set of all nonempty interval subquivers of the equioriented commutative 2D grid $\vec{G}_{m,n}$. In this subsection, we count the elements of $\mathbb{I}_{m,n}$. Recall that by Proposition 21, we identify interval subquivers of $\vec{G}_{m,n}$ with staircases of $\vec{G}_{m,n}$, and denote an interval by writing it as a set of slices $\{[b_j, d_j]_j \mid s \le j \le t\}$, as a staircase from s to t.

Definition 25 *(Size of interval).* For an interval $I = \{[b_j, d_j]_j \mid s \le j \le t\}$ (a staircase from s to t), we define the size of I as follows.

$$\mathrm{Size}(I) := (d_t - b_s + 1, t - s + 1) \in \mathbb{Z}^2$$

Moreover, for each $(w, h) \in \{1, \ldots, m\} \times \{1, \ldots, n\}$, we set

$$\begin{aligned} F_{m,n}(w,h) &:= \{I \in \mathbb{I}_{m,n} \mid \mathrm{Size}(I) = (w,h)\} \text{ and} \\ R(w,h) &:= \{I \in \mathbb{I}_{w,h} \mid \mathrm{Size}(I) = (w,h)\} = F_{w,h}(w,h) \end{aligned}$$

Note that the size here is simply the size of the bounding box of I. While both sets $F_{m,n}(w,h)$ and $R(w,h)$ contain staircases of the same size (w,h), the underlying 2D commutative grid is different. The set $F_{m,n}(w,h)$ considers staircases in $\vec{G}_{m,n}$, but $R(w,h)$ considers only staircases from $\vec{G}_{w,h}$ that are of maximum size. We introduce these sets to break down our computation of $\#\mathbb{I}_{m,n}$ (see Equation (4.5)).

Example 26. If $(m,n) = (3,3)$, then

$$F_{3,3}(2,2) = \left\{ \begin{pmatrix}110\\110\\000\end{pmatrix}, \begin{pmatrix}011\\011\\000\end{pmatrix}, \begin{pmatrix}000\\110\\110\end{pmatrix}, \begin{pmatrix}000\\011\\011\end{pmatrix}, \begin{pmatrix}100\\110\\000\end{pmatrix}, \begin{pmatrix}010\\011\\000\end{pmatrix}, \begin{pmatrix}000\\100\\110\end{pmatrix}, \begin{pmatrix}000\\010\\011\end{pmatrix}, \right.$$

$$\left. \begin{pmatrix}110\\010\\000\end{pmatrix}, \begin{pmatrix}011\\001\\000\end{pmatrix}, \begin{pmatrix}000\\110\\010\end{pmatrix}, \begin{pmatrix}000\\011\\001\end{pmatrix} \right\} \text{ and } R(2,2) = \{ (\begin{smallmatrix}11\\11\end{smallmatrix}), (\begin{smallmatrix}10\\11\end{smallmatrix}), (\begin{smallmatrix}11\\01\end{smallmatrix}) \}.$$

It is clear that the elements of $F_{m,n}(w,h)$ can be listed by shifting each element of $R(w,h)$, and thus

$$\#F_{m,n}(w,h) = (m-w+1)(n-h+1)\#R(w,h).$$

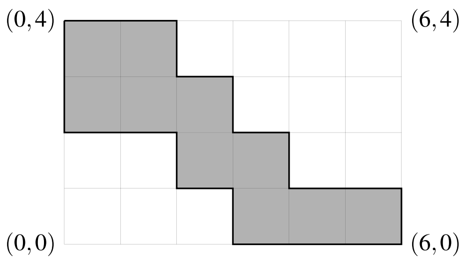
Moreover, the cardinality of the set of all intervals $\mathbb{I}_{m,n}$ can thus be obtained by counting intervals of all possible sizes:

$$\#\mathbb{I}_{m,n} = \sum_{w=1}^{m}\sum_{h=1}^{n} \#F_{m,n}(w,h) = \sum_{w=1}^{m}\sum_{h=1}^{n}(m-w+1)(n-h+1)\#R(w,h). \tag{4.5}$$

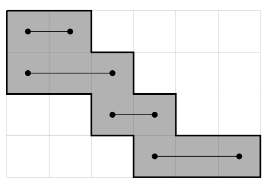
Thus, to calculate $\#\mathbb{I}_{m,n}$, it is enough to calculate the numbers $\#R(w,h)$. Next we give an explicit form for the value of $\#R(w,h)$, by relating it to a well-known concept in combinatorics.

Definition 27 *(Parallelogram polyomino).* A parallelogram polyomino having a $w \times h$ bounding box is a polyomino contained in a rectangle consisting of wh cells and formed by cutting out, from this rectangle, two (possibly empty) non-touching Young diagrams with corners at $(0,0)$ and (w,h).

Equivalently, a parallelogram polyomino with a $w \times h$ bounding box is a pair of lattice non-increasing paths P, Q from $(0,h)$ to $(w,0)$ so that P lies entirely above Q, and P and Q intersect only at $(0,h)$ and $(w,0)$. In the definition above, h is taken to be the height and w the width. An example of a parallelogram polyomino having a 6×4 bounding box is given below.



By interpreting staircases $I \in \mathbb{I}_{w,h}$ as the filled-in boxes on the lattice (not the grid lines!), it is clear that staircases in $R(w,h)$ are in one to one correspondence with parallelogram polyominoes with a $w \times h$ bounding box. The example above is identified to a staircase in the following way.



$\longleftrightarrow$ $\{[1,2]_4,\ [1,3]_3,\ [3,4]_2,\ [4,6]_1\}$

a parallelogram polyomino — a staircase

Lemma 28. *There is a bijection between $R(w,h)$ and the set of parallelogram polyominoes with $w \times h$ bounding box.*

Definition 29 *(Narayana number).* For each pair of integers $1 \le b \le a$, the Narayana number $N(a,b)$ is defined by using binomial coefficients as follows.

$$N(a,b) := \frac{1}{a}\binom{a}{b}\binom{a}{b-1}$$

Narayana numbers are closely related with counting problems of parallelogram polyominoes. Especially, the following fact is well known. For example, see [19].

Proposition 30. *The number of parallelogram polyominoes having an $w \times h$ bounding box is exactly $N(h+w-1,h)$.*

Hence we obtain the following formulas by a straightforward application of the above standard fact.

Theorem 31. *Let m, n be positive integers. Then:*

$$\#R(w,h) = N(h+w-1,h) = \frac{1}{h+w-1}\binom{h+w-1}{h-1}\binom{h+w-1}{w-1}$$

and

$$\#\mathbb{I}_{m,n} = \sum_{w=1}^{m}\sum_{h=1}^{n} \frac{(m-w+1)(n-h+1)}{(h+w-1)}\binom{h+w-1}{h-1}\binom{h+w-1}{w-1}.$$

Proof. We use Lemma 28 and Proposition 30 and note that

$$\binom{h+w-1}{h} = \binom{h+w-1}{w-1}$$

to obtain the first formula, and second formula follows from Equation (4.5). □

In particular, for an equioriented commutative 2D grid of size $m \times 2$ (an equioriented commutative ladder [20]), we obtain the following formula.

Corollary 32. *For each $m \in \mathbb{N}$, we have*

$$\#I_{m,2} = \frac{1}{24}m(m+1)(m^2+5m+30).$$

Remark 33. We can apply Theorem 18 to a given representation M of $\vec{G}_{m,n}$ in order to determine whether or not it is interval-decomposable. In Section 5 we discuss this procedure of determining interval-decomposability. Then, Theorem 31 gives the cardinality of the set $\mathscr{S}$ of intervals over which we need to compute multiplicities. This cardinality is a large number. To mitigate this, we may replace the original quiver $\vec{G}_{m,n}$ by the smallest equioriented commutative 2D-grid containing the support of M. Moreover, in Subsection 5.3, we provide heuristics to further reduce the number of intervals for which we need to compute multiplicities.

5. Algorithms and computational complexity

In this section, we provide a detailed algorithm for determining interval-decomposability, based on Theorem 18. In the final subsection, we also give a remark concerning the use of the decomposition algorithm given in [12], for computing interval-decomposability. Here, we let $\omega < 2.373$ be the matrix multiplication exponent [21,22].

Given a 2D persistence module M in $Q = \vec{G}_{m,n}$, the following procedure can be used to determine whether or not M is interval-decomposable.

Let us first give an overview of Algorithm 1. We initialize dimVecRemaining, which holds the dimensions of vector spaces yet unprocessed by the algorithm. In particular, we let $\text{dimVecRemaining}_{x,y}$ hold the dimension at (x,y) i.e. column x and row y counting from the bottom. For example, below is the underlying quiver of $\vec{G}_{4,3}$ which has 4 columns and 3 rows. For clarity, the (x,y) coordinates of the corner points are labelled.

$$\begin{array}{ccccccc}
(1,3)\bullet & \rightarrow & \bullet & \rightarrow & \bullet & \rightarrow & \bullet(4,3) \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
\bullet & \rightarrow & \bullet & \rightarrow & \bullet & \rightarrow & \bullet \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
(1,1)\bullet & \rightarrow & \bullet & \rightarrow & \bullet & \rightarrow & \bullet(4,1)
\end{array}$$

Algorithm 1 Algorithm for determining interval-decomposability of M.

```
1: function isIntervalDecomposable(M)
2:     dimVecRemaining ← dim M
3:     for x = m, m − 1, ..., 1 do
4:         for y = 1, 2, ..., n do
5:             if dimVecRemaining_{x,y} = 0 then
6:                 continue
7:             end if
8:             for L ∈ getCandidates(x, y) do
9:                 d_M(L) ← multiplicity(M, L)
10:                dimVecRemaining ← dimVecRemaining - d_M(L) dim L
11:                if dimVecRemaining_{x,y} = 0 then
12:                    break
13:                end if
14:            end for
15:            if dimVecRemaining_{x,y} > 0 then
16:                return False
17:            end if
18:        end for
19:    end for
20:    return True
21: end function
```

The main action happens in Line 10, where we decrement dimVecRemaining by the dimension vector of some interval L multiplied by its multiplicity $d_M(L)$ in M. Ignoring for a moment all the places where the algorithm can terminate early, if we simply iterate through all intervals L of the grid $\overrightarrow{G}_{m,n}$, then by Theorem 18 M is interval-decomposable if and only if $\text{dimVecRemaining}_{x,y}$ is 0 for all (x, y), at the end of the algorithm.

Algorithm 1 orders the processing of the intervals L so that there is a possibility of stopping early. In particular, we order the intervals by their lower-right corners (x, y), in order of decreasing x and increasing y (the two outer for-loops in Algorithm 1). The procedure $\textsc{getCandidates}(x, y)$ (in Algorithm 2), generates the intervals with lower-right corner given by (x, y). If, after processing all such candidates for some fixed lower-right corner (x, y), $\text{dimVecRemaining}_{x,y}$ is nonzero, then we know that M cannot be interval-decomposable. Indeed, the way we iterate over all possible lower-right corners ensures that once we finish processing (x, y), the value of $\text{dimVecRemaining}_{x,y}$ can no longer change.

Algorithm 2 Intervals in Q with lower-right corner $d_s = x$ and $s = y$.

```
function getCandidates(x, y)
    Set an empty list 𝓛.
    for b = 1, ..., x do
        Add the interval {[b, x]_y} of height 1 to the end of list 𝓛.
    end for
    for k = 0, ..., length(𝓛) − 1 do
        Read the interval 𝓛[k]:
            (assume that it is the interval from s to t given by {[b_j, d_j]_j | s ≤ j ≤ t})
        if t is equal to the height of Q then
            continue
        end if
        for b_{t+1} = 1, ..., b_t and d_{t+1} = b_t, ..., d_t do
            Add the interval {[b_j, d_j]_j | s ≤ j ≤ t + 1} to the end of list 𝓛.
        end for
    end for
    return 𝓛
end function
```

Next, let us explain the details of $\textsc{getCandidates}(x, y)$ as presented in Algorithm 2. Recall that we use the following notation for a staircase (an interval, by Proposition 21). For $1 \le j \le n$, a *slice* at row j is a pair of numbers $1 \le b_j \le d_j \le m$, denoted $[b_j, d_j]_j$. For $1 \le s \le t \le n$, a *staircase* from s to t is a set of slices $[b_j, d_j]_j$ for $s \le j \le t$ such that $b_{j+1} \le b_j \le d_{j+1} \le d_j$ for any $j \in \{s, \dots, t-1\}$. In Algorithm 2, we enumerate the candidate intervals with the coordinate of the "lower-right" corner fixed as $d_s = x$ and $s = y$. Starting with the lower-right corner, we progressively build up taller and taller intervals.

Next, we also write down Algorithm 3 for computing the multiplicity, to be used in Line 10 of Algorithm 1. The correctness of Algorithm 3 follows from formula (2.5) of Theorem 17. We note that Theorem 17 holds much more generally (not just for intervals). But since in Algorithm 1 we only call Algorithm 3 with an interval representation as input for L, we make this restriction to simplify the discussion. The major components of Algorithm 3 are the computation of the terms of almost split sequences ending at nonprojective intervals and the computation of $\dim \operatorname{Hom}(M, -)$.

Algorithm 3 Computation of the multiplicity $d_M(L)$ for L an interval.

Require: L interval
1: **function** MULTIPLICITY(M, L)
2: **if** L is projective **then**
3: $\tau L,\ E_L \leftarrow 0,\ \operatorname{rad} L$
4: **else**
5: $\tau L,\ E_L \leftarrow$ ALMOSTSPLITSEQUENCETERMS(L)
6: **end if**
7: $d_M(L) \leftarrow \dim \operatorname{Hom}(M, \tau L) - \dim \operatorname{Hom}(M, E_L) + \dim \operatorname{Hom}(M, L)$
8: **return** $d_M(L)$
9: **end function**

Section 3.2 of [23] provides a theoretical procedure for computing the almost split sequence ending at a nonprojective indecomposable Z. We rewrite it in algorithm form as the function ALMOSTSPLITSEQUENCETERMS (Algorithm 4). Here, we restrict our attention to Z with $\operatorname{End} Z \cong K$ since all interval representations Z satisfy this condition, and this simplifies the choice of S in Line 5 (in general another condition needs to be imposed on S).

Algorithm 4 Almost split sequence ending at Z nonprojective with $\operatorname{End}_A(Z) \cong K$.

Require: Z nonprojective indecomposable with $\operatorname{End}_A(Z) \cong K$
1: **function** ALMOSTSPLITSEQUENCETERMS(Z)
2: Compute a minimal projective presentation: $P_1 \xrightarrow{f_1} P_0 \xrightarrow{\varepsilon} Z \to 0$
3: Apply the Nakayama functor $\nu := D \circ \operatorname{Hom}_A(\text{-}, A)$ to obtain $\nu P_1 \xrightarrow{\nu f_1} \nu P_0$
4: Compute $\tau Z := \operatorname{Ker}(\nu f_1)$ ($\cong D \operatorname{Tr} Z$)
5: Compute $\theta_Z : Z \xrightarrow{\text{can}} \operatorname{top} Z \xrightarrow{\pi} S \hookrightarrow \operatorname{top} Z \xrightarrow{\sim} \operatorname{top} P_0 \xrightarrow{\sim} \operatorname{soc} \nu P_0$,
where S is a simple direct summand of $\operatorname{top} Z$
6: Compute the middle term E_Z via pullback

$$\begin{tikzcd}
0 \arrow[r] & X \arrow[r, "f"] \arrow[d, equal] & E_Z \arrow[r, "g"] \arrow[d, "h"] & Z \arrow[d, "\theta_Z"] \\
0 \arrow[r] & \tau Z \arrow[r, hook] & \nu P_1 \arrow[r, "\nu f_1"'] & \nu P_0.
\end{tikzcd}$$

7: **return** $\tau Z,\ E_Z$
8: **end function**

Proposition 34 *(Section 3.2 of [23]). Let A be a finite-dimensional algebra. Then given a non-projective indecomposable A-module Z with $\operatorname{End}_A(Z) \cong K$, Algorithm 4 computes an almost split sequence $0 \to X \xrightarrow{f} E_Z \xrightarrow{g} Z \to 0$ in line 6.*

5.1. Some background definitions for the analysis of the algorithm

In order to provide a complexity analysis (especially for Algorithm 4), we need to describe the objects being constructed in more elementary terms (vector spaces and linear maps). Thus before providing our detailed analysis of the algorithms, let us recall some definitions that we use.

First, let us review some basic concepts. Recall that $\vec{G}_{m,n}$ can be regarded as a subposet of $\mathbb{Z} \times \mathbb{Z}$ by the order

$$(x_1, y_1) \le (x_2, y_2) \iff x_1 \le x_2 \text{ and } y_1 \le y_2$$

for any vertices $(x_1, y_1), (x_2, y_2)$ of $\vec{G}_{m,n}$.

For any pair of two vertices $a = (x_1, y_1)$ and $b = (x_2, y_2)$ of $\vec{G}_{m,n}$, we denote their *join* (resp. *meet*) by $a \vee b$ (resp. $a \wedge b$). These always exist in $\vec{G}_{m,n}$ and are given by

$$a \vee b = (\max(x_1, x_2), \max(y_1, y_2)) \text{ and } a \wedge b = (\min(x_1, x_2), \min(y_1, y_2)).$$

For each interval L, we fix the representation V_L, isomorphic to an interval representation associated to L, by the following. We define

$$V_L(a) = \begin{cases} Ka & \text{if } a \in L_0, \\ 0 & \text{otherwise.} \end{cases}$$

That is, for each $a \in L_0$, we set $V_L(a)$ to be the K-vector space of multiples of the vertex a itself (with a as fixed basis). For each arrow $\alpha : a \to b$ in L, the map $V_L(\alpha) : V_L(a) \to V_L(b)$ is defined by $\lambda a \mapsto \lambda b$ for all $\lambda \in K$.

We review some fundamental facts on representations of quivers and modules over algebras by specializing to our case. First, recall that $\vec{G}_{m,n} = \vec{A}_m \otimes \vec{A}_n = (\vec{A}_m \times \vec{A}_n, R)$, where R is the ideal of full commutativity relations. Set $A := K\vec{G}_{m,n}$, which is the path algebra of the bound quiver $\vec{G}_{m,n}$. That is, $A = K(\vec{A}_m \times \vec{A}_n)/R$. For each $a, b \in (\vec{G}_{m,n})_0$ we denote by $p_{a,b}$

the element of the algebra A represented by a path from b to a, and we set $e_a := p_{a,a}$. Note that $p_{a,b}$ is uniquely determined by a and b because $\vec{G}_{m,n}$ has the full commutativity relations. That is, the coset of any path from b to a in the quotient $A = K(\vec{A}_m \times \vec{A}_n)/R$ contains any other path from b to a. This coset is denoted $p_{a,b}$.

There exists a well-known equivalence ([13, Theorem 1.6 of Chapter III]) between the category of representations of $\vec{G}_{m,n}$ and the category of (left) A-modules, sending each representation V to $\bigoplus_{u\in(\vec{G}_{m,n})_0} V(u)$ with the A-action defined by $V(\alpha)$ for all $\alpha \in (\vec{G}_{m,n})_1$, a quasi-inverse of which sends each A-module M to the representation $(e_a M, \lambda_\alpha)_{a\in(\vec{G}_{m,n})_0, \alpha\in(\vec{G}_{m,n})_1}$, where λ_α denotes the left multiplication by α. By this equivalence we often identify representations of $\vec{G}_{m,n}$ with their corresponding (left) A-modules.

Since it is also used in Algorithm 4, we recall that D is the K-dual given by $D(-) := \mathrm{Hom}_K(-, K) : \mathrm{mod}\, A \to \mathrm{mod}\, A^{\mathrm{op}}$, and that $\nu := D \circ \mathrm{Hom}_A(\text{-}, A) : \mathrm{mod}\, A \to \mathrm{mod}\, A$ is the *Nakayama functor*. See [13, Chap. III.2] for more details. As indicated in Algorithm 4, we use the Nakayama functor to compute the translation τZ (defined as $D\,\mathrm{Tr}\, Z$) by using the fact that $\tau Z \cong \mathrm{Ker}(\nu f_1)$ ([13, Chapter IV Proposition 2.4]). Note that in Algorithm 4 we identified these two and simply wrote $\tau Z = \mathrm{Ker}(\nu f_1)$.

Next, let us describe the indecomposable projective, simple, and indecomposable injective representations of $\vec{G}_{m,n}$. First, let J be the ideal of A generated by all arrows in $\vec{G}_{m,n}$. It is well-known that J is the *Jacobson radical* of A, the intersection of all maximal right ideals in A. The sets

- $\{P(a) := Ae_a \mid a \in (\vec{G}_{m,n})_0\}$
- $\{S(a) := Ae_a/Je_a \mid a \in (\vec{G}_{m,n})_0\}$
- $\{I(a) := \nu P(a) = D(e_a A) \mid a \in (\vec{G}_{m,n})_0\}$

respectively form complete sets of representatives of indecomposable projective, simple, and indecomposable injective A-modules (see [13, Chapter III.2]). Note that $\{e_a + Je_a\}$, $\{p_{x,a} \mid a \le x\}$, $\{p_{a,x} \mid a \ge x\}$ form bases of $S(a)$, $P(a)$, and $e_a A$. Set $\{p^\vee_{a,x} \mid a \ge x\}$ to be the basis of $I(a)$ dual to $\{p_{a,x} \mid a \ge x\}$.

The previous paragraph gives $P(a)$, $S(a)$, $I(a)$ as A-modules. We give their forms as representations of $\vec{G}_{m,n}$. It is clear that $(\mathrm{supp}\, S(a))_0 = \{a\}$, $(\mathrm{supp}\, P(a))_0 = \{x \in (\vec{G}_{m,n})_0 \mid a \le x\}$, and $(\mathrm{supp}\, I(a))_0 = \{x \in (\vec{G}_{m,n})_0 \mid a \ge x\}$. Then as is easily verified, the bijections between bases

$$e_a + Je_a \mapsto a, \quad p_{x,a} \mapsto x, \quad p^\vee_{a,x} \mapsto x$$

define isomorphisms

$$S(a) \to V_{\{a\}}, \quad P(a) \to V_{\{x\in(\vec{G}_{m,n})_0 \mid a\le x\}}, \quad I(a) \to V_{\{x\in(\vec{G}_{m,n})_0 \mid a\ge x\}}, \tag{5.1}$$

respectively by which we identify them. Thus all of $S(a)$, $P(a)$, $I(a)$ for $a \in (\vec{G}_{m,n})_0$ are interval representations, and we fix bases for their vector spaces by these identifications.

Next, let us consider the homomorphisms between certain interval representations and the action of the Nakayama functor. First, recall that for each left A-module M, right A-module N and each $a \in (\vec{G}_{m,n})_0$ we have isomorphisms

$$\begin{aligned} e_a M &\to \mathrm{Hom}_A(P(a), M), \quad m \mapsto \rho_m, \\ Ne_a &\to \mathrm{Hom}_A(e_a A, N), \quad n \mapsto \lambda_n, \end{aligned} \tag{5.2}$$

where ρ_m is the right multiplication by m, and λ_n is the left multiplication by n, that is, $\rho_m(x) := xm$ $(x \in P(a))$, and $\lambda_n(x) := nx$ $(x \in e_a A)$. Note that

$$\rho_{p_{a,b}} : P(a) = Ae_a \to P(b) = Ae_b,$$

which is right multiplication by $p_{a,b} \in A$ represented by any path from b to a, is sent by $\mathrm{Hom}_A(-, A)$ to

$$\lambda_{p_{a,b}} : e_b A \to e_a A,$$

which is left multiplication by $p_{a,b}$, after identification using the first isomorphism in Eq. (5.2) with $M = A$. Applying the K-dual D, we obtain

$$\nu(\rho_{p_{a,b}}) = D(\lambda_{p_{a,b}}) : I(a) \to I(b) \tag{5.3}$$

where $\nu(-) := D\,\mathrm{Hom}(-, A)$ is the Nakayama functor.

We also have

$$\mathrm{Hom}_A(M, I(a)) \cong D(e_a M) \tag{5.4}$$

for all $a \in (\vec{G}_{m,n})_0$. Indeed,

$$\begin{aligned} D(e_a M) &\cong D(M)e_a \cong \mathrm{Hom}_A(e_a A, D(M)) \\ &\cong \mathrm{Hom}_A(DD(M), D(e_a A)) \cong \mathrm{Hom}_A(M, I(a)), \end{aligned}$$

where the second isomorphism follows by the second isomorphism in Eq. (5.2) with $N = D(M)$, and the third isomorphism follows since D is a duality. Alternatively, this follows immediately by tensor-hom adjunction:

$$\begin{aligned} D(e_a M) &\cong \mathrm{Hom}_K(e_a A \otimes_A M, K) \\ &\cong \mathrm{Hom}_A(M, \mathrm{Hom}_K(e_a A, K)) \cong \mathrm{Hom}_A(M, I(a)). \end{aligned}$$

In particular, substituting $M = P(b)$ in Eq. (5.2) and $M = I(b)$ in Eq. (5.4) for $b \in (\vec{G}_{m,n})_0$, we have the following explicit forms for the homomorphisms between indecomposable projectives, and between indecomposable injectives:

$$\mathrm{Hom}_A(P(a), P(b)) = K\rho_{p_{a,b}}, \text{ and } \mathrm{Hom}_A(I(b), I(a)) = KD(\lambda_{p_{a,b}}) = K\nu(\rho_{p_{a,b}}).$$

Moreover, for $M = V_L$ with L an interval subquiver of $\vec{G}_{m,n}$ we have

$$\text{for } a \in L, \;\; \mathrm{Hom}_A(P(a), V_L) = K\rho_a \text{ and } \mathrm{Hom}_A(V_L, I(a)) = KD(\lambda_a)$$

because $V_L(a) = Ka$ for $a \in L$. The following is easy to verify.

Lemma 35. *Let $a \in (\vec{G}_{m,n})_0$ and L be an interval subquiver of $\vec{G}_{m,n}$. Then the explicit form for ρ_a and $D(\lambda_a)$ above as morphisms of representations under the identifications (5.1) are given as morphisms $\varepsilon_{a,V_L} : P(a) \to V_L$ and $\varepsilon'_{V_L,a} : V_L \to I(a)$ defined by*

$$\varepsilon_{a,V_L}(c) = \begin{cases} 1_{Kc} & \text{if } a \le c \text{ and } c \in L_0, \\ 0 & \text{otherwise,} \end{cases}$$

and by

$$\varepsilon'_{V_L,a}(c) = \begin{cases} 1_{Kc} & \text{if } a \ge c \text{ and } c \in L_0, \\ 0 & \text{otherwise,} \end{cases}$$

respectively.

Definition 36. For each $a, b \in (\vec{G}_{m,n})_0$ we set

$$\begin{cases} \varepsilon_{a,b} := \varepsilon_{a,P(b)} : \; P(a) \to P(b), \text{ and} \\ \varepsilon'_{a,b} := \varepsilon'_{I(a),b} : \; I(a) \to I(b). \end{cases}$$

Thus these give the explicit forms of the morphisms $\rho_{p_{a,b}}$ and $D(\lambda_{p_{b,a}}) = \nu\rho_{p_{a,b}}$ (see Eq. (5.3)), respectively. In particular, we have $\nu(\varepsilon_{q,p}) = \varepsilon'_{q,p}$.

5.2. Analysis of the algorithms

We proceed backwards, analyzing the supporting Algorithms 4, 3, and then the overall Algorithm 1, in that order.

First, we look at Algorithm 4. Since we want to analyze its computational complexity when applied to a non-projective interval representation $Z = V_L$ of $\vec{G}_{m,n}$ (Proposition 42), we provide the fine details involved. We devote the next few pages (up to Proposition 42 in page 26) to this discussion and analysis of Algorithm 4. We note that this restriction to intervals simplifies some of the computation, as we can provide explicit forms for the minimal projective presentation and the map νf_1.

(Line 2 of Algorithm 4.) Let V_L be the interval representation associated with the interval subquiver $L = \{[b_j, d_j]_j \mid s \le j \le t\}$. Let $\mathrm{Source}(L)$ be the set of source vertices in L, which is given by

$$\mathrm{Source}(L) = \{(b_j, j) \mid j = \min\{l \mid b_l = b_k\} \text{ for some } k \text{ with } s \le k \le t\}.$$

Below, we explain the computation of a minimal projective presentation $P_1 \xrightarrow{f_1} P_0 \xrightarrow{\varepsilon} V_L \to 0$. For this, we need the concept of upset, a special type of interval. The overall strategy is to first compute minimal projective presentations of V_U for upsets U (Proposition 39). Then, as V_L has the form $V_L \cong V_U / V'_U$ for some upsets U, U' (see Lemma 40), we piece together the minimal projective presentations of V_U, V'_U to have that of V_L.

Definition 37 *(upsets and upset representations).*

(1) A subset U of $(\vec{G}_{m,n})_0$ is called an *upset* if the conditions $x \le y$ in $(\vec{G}_{m,n})_0$ and $x \in U$ imply $y \in U$.

(2) Obviously the intersection of any upsets is again an upset. Therefore for each subset S of $(\vec{G}_{m,n})_0$ there exists the minimum upset U of $(\vec{G}_{m,n})_0$ such that $S \subseteq U$, which we denote by $U(S)$. When $S = \{a\}$ for some $a \in (\vec{G}_{m,n})_0$, we simply write $U(a) := U(\{a\})$.
(3) Since for any upset U, the full subquiver $\mathrm{full}(U)$ of $\vec{G}_{m,n}$ with $\mathrm{full}(U)_0 = U$ turns out to be an interval (see the lemma below), the interval representation $V_U := V_{\mathrm{full}(U)}$ is defined, and is called an *upset representation*.

The following is obvious.

Lemma 38.

(1) *For all $a \in (\vec{G}_{m,n})_0$, $U(a) = (\mathrm{supp}\, P(a))_0$, and $V_{U(a)} = P(a)$.*
(2) *For S a subset of $(\vec{G}_{m,n})_0$, $U(S) = \bigcup_{a \in S} U(a)$ and $V_{U(S)} = \sum_{a \in S} P(a)$.*
(3) *For an interval L,*

$$U(L_0) = U(\mathrm{Source}(L)) = \bigcup_{a \in \mathrm{Source}(L)} U(a) \text{ and } V_{U(L_0)} = \sum_{a \in \mathrm{Source}(L)} P(a)$$

(4) *Every upset is the vertex set of an interval.*
(5) *For an interval $L := \{[b_j, d_j]_j \mid s \le j \le t\}$, L_0 is an upset if and only if $t = n$ and $d_i = m$ for all $s \le i \le n$.*

We now give a minimal projective presentation of an upset representation.

Proposition 39. *Let U be an upset of $(\vec{G}_{m,n})_0$, and set $\{p_1, \dots p_l\} := \mathrm{Source}(U)$, and $q_d := p_d \vee p_{d+1}$ for all $d = 1, \dots, l-1$, where $p_c = (b_{j_c}, j_c)$, $(c = 1, \dots, l)$, with $n \ge j_1 > j_2 > \cdots > j_l \ge 1$. Then a minimal projective presentation of V_U is given by*

$$0 \to \bigoplus_{d=1}^{l-1} P(q_d) \xrightarrow{f_U} \bigoplus_{c=1}^{l} P(p_c) \xrightarrow{\pi_U} V_U \to 0, \tag{5.5}$$

where $\pi_U = (\varepsilon_{p_c, V_U})_{c=1}^{l}$ and

$$f_U = \begin{pmatrix} \varepsilon_{q_1,p_1} & & & \\ -\varepsilon_{q_1,p_2} & \varepsilon_{q_2,p_2} & & \\ & -\varepsilon_{q_2,p_3} & \ddots & \\ & & \ddots & \varepsilon_{q_{l-1},p_{l-1}} \\ & & & -\varepsilon_{q_{l-1},p_l} \end{pmatrix} \text{ (each blank entry is zero).}$$

The homomorphisms $\varepsilon_{p,V}$, $\varepsilon_{q,p}$ are given in Definition 36 and the preceding discussion. We also note that we use a particular ordering for the elements of $\mathrm{Source}(U)$, given by $\{p_1, \dots p_l\} := \mathrm{Source}(U)$ with $p_c = (b_{j_c}, j_c)$, $(c = 1, \dots, l)$ and $n \ge j_1 > j_2 > \cdots > j_l \ge 1$.

Proof. We first show that the equality

$$\sum_{c=1}^{l} \dim P(p_c) - \sum_{d=1}^{l-1} \dim P(q_d) = \dim V_U \tag{5.6}$$

holds. It is enough to show the equality

$$\sum_{c=1}^{l} \dim P(p_c)(p) - \sum_{d=1}^{l-1} \dim P(q_d)(p) = \dim V_U(p) \tag{5.7}$$

for all vertices $p \in (\vec{G}_{m,n})_0$. If $p \notin U$, then $\dim V_U(p) = 0$ by definition, and we have $p \not\ge p_c$ for all $c = 1, \dots, l$, and hence $p \not\ge q_d$ for all $d = 1, \dots, l-1$. Thus the left hand side is also zero, and Eq. (5.7) holds. Assume $p \in U$. We set $\{p_{j_1}, \dots, p_{j_t}\} := \{p_c \mid p_c \le p\}$ with $j_1 > \cdots > j_t$. Note that the indices j_i are contiguous integers. Then $\sum_{c=1}^{l} \dim P(p_c)(p) = \#\{p_c \mid p_c \le p\} = t$, and $\sum_{d=1}^{l-1} \dim P(q_d)(p) = \#\{q_d \mid q_d \le p\} = \#\{q_{j_1}, \dots, q_{j_{t-1}}\} = t - 1$. Since $\dim V_U(p) = 1 = t - (t-1)$, Eq. (5.7) holds also in this case, and hence the equality (5.6) is verified.

Now since all ε_{q_c,p_c} are monomorphisms, f_U is also a monomorphism. On the other hand, π_U is an epimorphism because $\mathrm{Im}\, \pi_U = \sum_{c=1}^{l} \mathrm{Im}\, \varepsilon_{p_c,V_U} = \sum_{c=1}^{l} P(p_c) = V_U$ by Lemma 38(3). Furthermore, since $\varepsilon_{p_c,V_U} \varepsilon_{q_c,p_c} = \varepsilon_{p_{c+1},V_U} \varepsilon_{q_c,p_{c+1}}$, we have $\pi_U f_U = 0$. These facts, together with the equality (5.6) show that the sequence (5.5) above is exact.

Obviously π_U induces an isomorphism between the tops,[4] and hence it is a projective cover of V_U. The exactness of the sequence (5.5) shows that f_U is a projective cover of $\operatorname{Ker}\pi_U$. Therefore the sequence (5.5) is a minimal projective presentation of V_U. $\square$

Lemma 40. *Let $L = \{[b_j, d_j]_j \mid s \le j \le t\} \in \mathbb{I}_{m,n}$ be an interval. Define*

$$U := U(L) = \{[b_j, m]_j \mid s \le j \le n\}$$

where $b_j := b_t$ for $j = t+1, \dots n$, and

$$U' := \{[d_j + 1, m]_j \mid s \le j \le n\}$$

where $d_j + 1 := b_t$ for $j = t+1, \dots n$.

Then U and U' are upsets satisfying

$$V_L \cong V_U / V_{U'}.$$

Proof. Both U and U' are upsets by Lemma 38(5). The statement follows from the following calculations:

$$\begin{aligned} L_0 &= \bigcup_{j=s}^{t} [b_j, d_j]_j = \bigcup_{j=s}^{t} ([b_j, m]_j \setminus [d_j + 1, m]_j) \\ &= \left(\bigcup_{j=s}^{t} [b_j, m]_j\right) \setminus \left(\bigcup_{j=s}^{t} [d_j + 1, m]_j\right) = \left(\bigcup_{j=s}^{n} [b_j, m]_j\right) \setminus \left(\bigcup_{j=s}^{n} [d_j + 1, m]_j\right) \\ &= U \setminus U'. \quad \square \end{aligned}$$

Using the above, we next give a minimal projective presentation of an interval representation V_L.

Proposition 41. *Let L be an interval of $\vec{G}_{m,n}$ and U, U' the upsets defined in Lemma 40. Set $\{p_1, \dots, p_l\} := \mathrm{Source}(L) = \mathrm{Source}(U)$ and $q_d := p_d \vee p_{d+1}$ for all $d = 1, \dots, l-1$, where $p_c = (b_{j_c}, j_c)$, $(c = 1, \dots, l)$, with $n \ge j_1 > \cdots > j_l \ge 1$. Set also $c(r) := \min\{c \mid p_c \le r\}$ for all $r \in \mathrm{Source}(U')$.*

Then a minimal projective presentation of V_L is given by

$$\bigoplus_{r \in \mathrm{Source}(U')} P(r) \oplus \bigoplus_{d=1}^{l-1} P(q_d) \xrightarrow{(f'_L, f_U)} \bigoplus_{c=1}^{l} P(p_c) \xrightarrow{\pi_L} V_L \to 0, \tag{5.8}$$

where $\pi_L := (\varepsilon_{p_c, V_L})_{c=1}^{l}$, and $f'_L := (\delta_{c, c(r)} \varepsilon_{r, p_{c(r)}})_{c,r}$. Here, δ_{ij} is the Kronecker delta.

We recall once again that the homomorphisms $\varepsilon_{p,V}$, $\varepsilon_{q,p}$ are as given in Definition 36 and the preceding discussion.

Proof. For simplicity we put $P_0 := \bigoplus_{c=1}^{l} P(p_c)$ and $P_1 := \bigoplus_{d=1}^{l-1} P(q_d)$. Then we have an exact sequence

$$0 \to P_1 \xrightarrow{f_U} P_0 \xrightarrow{\pi_U} V_U \to 0,$$

which is a minimal projective presentation of V_U by Proposition 39. By the same way we construct a minimal projective presentation of $V_{U'}$ of the form

$$0 \to P'_1 \xrightarrow{f_{U'}} P'_0 \xrightarrow{\pi_{U'}} V_{U'} \to 0,$$

where we note that $P'_0 := \bigoplus_{r \in \mathrm{Source}(U')} P(r)$.

Let $v \colon V_U \to V_L$ be the epimorphism defined by

$$v(x) = \begin{cases} 1_K x, & x \in L_0 \\ 0, & \text{otherwise.} \end{cases}$$

[4] Indeed, by definition of $\varepsilon_{a,b}$ $(a, b \in (\vec{G}_{m,n})_0)$, π_U induces the isomorphism from $\mathrm{top}(\bigoplus_{c=1}^{l} P(p_c)) = \bigoplus_{c=1}^{l} K[p_c]$ to $\mathrm{top}\, V_U = \bigoplus_{c=1}^{l} K[p_c]'$, $[p_c] \mapsto [p_c]'$ for all $c = 1, \dots, l$, where $[p_c]$ and $[p_c]'$ denote the cosets of p_c in $\mathrm{top}(\bigoplus_{c=1}^{l} P(p_c))$ and in $\mathrm{top}\, V_U$, respectively.

Then this induces an isomorphism between the tops, and hence $\pi_L := \nu\pi_U \colon P_0 \to V_L$ is a projective cover of V_L. Set $\Omega V_L := \operatorname{Ker}\pi_L$ and let $\mu\colon \Omega V_L \to P_0$ and $u\colon V_{U'} \to V_U$ be the inclusions. Then there exist unique morphisms g, g' that make the following diagram commute, with exact rows and columns:

$$
\begin{tikzcd}
 & 0 \arrow[d] & 0 \arrow[d] & & \\
 & P_1 \arrow[r, equal] \arrow[d, "g"'] & P_1 \arrow[d, "f_U"] & & \\
0 \arrow[r] & \Omega V_L \arrow[r, hook, "\mu"] \arrow[d, "g'"'] & P_0 \arrow[r, "\pi_L"] \arrow[d, "\pi_U"] & V_L \arrow[r] \arrow[d, equal] & 0 . \\
0 \arrow[r] & V_{U'} \arrow[r, hook, "u"'] \arrow[d] & V_U \arrow[r, "v"'] \arrow[d] & V_L \arrow[r] & 0 \\
 & 0 & 0 & &
\end{tikzcd}
\tag{5.9}
$$

Consider the following diagram of solid arrows with exact rows:

$$
\begin{tikzcd}
0 \arrow[r] & P'_1 \arrow[r, "f_{U'}"] \arrow[d, dashed, "h'"'] & P'_0 \arrow[r, "\pi_{U'}"] \arrow[d, dashed, "h"] & V_{U'} \arrow[r] \arrow[d, equal] & 0 \\
0 \arrow[r] & P_1 \arrow[r, "g"'] & \Omega V_L \arrow[r, "g'"'] & V_{U'} \arrow[r] & 0
\end{tikzcd}
\tag{5.10}
$$

By the projectivity of P'_0 this is completed to the commutative diagram with h, h'. We may take h in such a way that $\mu h\colon P'_0 \to P_0$ is given by the matrix $f'_L := (\delta_{c,c(r)}\varepsilon_{r,p_{c(r)}})_{c,r}$. Indeed, since u is a monomorphism, the equality $ug'h = \pi_U\mu h = \pi_U(\delta_{c,c(r)}\varepsilon_{r,p_{c(r)}})_{c,r} \overset{*}{=} u\pi_{U'}$ shows that $g'h = \pi_{U'}$, where the last equality ($\overset{*}{=}$) holds because the restrictions of both sides to $P(r)$ coincide for all $r \in \operatorname{Source}(U')$.

Since the left square of the diagram (5.10) is a pushout and pullback diagram, we have the following exact sequence:

$$
0 \to P'_1 \xrightarrow{\begin{pmatrix} f_{U'} \\ -h' \end{pmatrix}} P'_0 \oplus P_1 \xrightarrow{(h,g)} \Omega V_L \to 0.
$$

Here (h, g) is a projective cover of ΩV_L. Indeed, since $\pi_{U'}$ is a projective cover of $V_{U'}$, we have $\operatorname{Im} f_{U'} \subseteq \operatorname{rad} P'_0$, and by the form of $\mu h = f'_L$ we have $\operatorname{Im} h' \subseteq \operatorname{rad} P_1$. Therefore $\operatorname{Ker}(h, g) = \operatorname{Im}\begin{pmatrix} f_{U'} \\ -h' \end{pmatrix} \subseteq \operatorname{rad}(P'_0 \oplus P_1)$, as required. By connecting this sequence and the upper horizontal short exact sequence in the diagram (5.9) we obtain a minimal projective presentation

$$
P'_0 \oplus P_1 \xrightarrow{\mu(h,g)} P_0 \xrightarrow{\pi_L} V_L \to 0
$$

of V_L. Here note that $\mu(h, g) = (f'_L, f_U)$. □

Complexity Analysis for Line 2 of Algorithm 4. We let $l = |\operatorname{Source}(L)| = |\operatorname{Source}(U)|$ and $l' := |\operatorname{Source}(U')|$. Furthermore, we set $z := \min\{m, n\}$ and assume that $n = z = \min\{m, n\}$. Note that $l, l' \leq z$.

We give the cost of calculating (symbolically) the minimal projective presentation of V_L as given by Proposition 41. For this, we need to compute U, U' (Lemma 40) and their source vertices $\operatorname{Source}(U')$ and $\operatorname{Source}(U) = \{p_1, \ldots, p_l\}$ where $p_c = (b_{j_c}, j_c)$ for $c = 1, \ldots, l$, with $n \geq j_1 > \cdots > j_l \geq 1$. Then we need to compute $q_d := p_d \vee p_{d+1}$ for all $d = 1, \ldots, l-1$, and $c(r) = \min\{c \mid p_c \leq r\}$ for each $r \in \operatorname{Source}(U')$.

- First, the computation of U and U' from L follows using Lemma 40. This costs $O(z)$ by an obvious iteration over rows.
- Next, let us give the cost of calculating $\operatorname{Source}(U)$ for an upset U. Let $U := \{[b_j, m]_j \mid s \leq j \leq n\}$ be an upset. Then we iterate over the rows starting from the bottom row and going up. First, we record (b_s, s) as a source. Then we iterate $j = s+1, s+2, \ldots, n$, and whenever $b_j < b_{j-1}$, we record (b_j, j) as a source. This costs $O(z)$.
- For each $d = 1, \ldots, l-1$, the calculation of $q_d = p_d \vee p_{d+1}$ costs $O(1)$. This adds up to $O(l)$.
- For each $r \in \operatorname{Source}(U')$, the computation of $c(r) = \min\{c \mid p_c \leq r\}$ can be performed via binary search, costing $O(\log(l))$. Thus, overall the computation of $c(r)$ for all $r \in \operatorname{Source}(U')$ costs $O(l' \log(l))$.

Thus, overall we have a cost of $O(z + l + l' \log(l)) \leq O(z \log(z))$.

This ends our discussion and analysis of Line 2 of Algorithm 4 for the computation of a minimal projective presentation ending at an interval representation V_L. Let us move on to the next line.

(Line 3 of Algorithm 4.) Next, we compute $\nu f_1 : \nu P_1 \to \nu P_0$.

By Proposition 41 the morphism f_1 in the minimal projective presentation of V_L has the form

$$\bigoplus_{r\in\mathrm{Source}(U')} P(r) \oplus \bigoplus_{d=1}^{l-1} P(q_d) \xrightarrow{f_1=(f'_L, f_U)} \bigoplus_{c=1}^{l} P(p_c),$$

where

$$f'_L := (\delta_{c,c(r)}\varepsilon_{r,p_{c(r)}})_{c,r}, \quad \text{and} \quad f_U = \begin{pmatrix} \varepsilon_{q_1,p_1} & & & \\ -\varepsilon_{q_1,p_2} & \varepsilon_{q_2,p_2} & & \\ & -\varepsilon_{q_2,p_3} & \ddots & \\ & & \ddots & \varepsilon_{q_{l-1},p_{l-1}} \\ & & & -\varepsilon_{q_{l-1},p_l} \end{pmatrix}.$$

By ν this is sent to

$$\bigoplus_{r\in\mathrm{Source}(U')} I(r) \oplus \bigoplus_{d=1}^{l-1} I(q_d) \xrightarrow{(\nu f'_L, \nu f_U)} \bigoplus_{c=1}^{l} I(p_c),$$

where

$$\nu f'_L := (\delta_{c,c(r)}\varepsilon'_{r,p_{c(r)}})_{c,r}, \quad \text{and} \quad \nu f_U = \begin{pmatrix} \varepsilon'_{q_1,p_1} & & & \\ -\varepsilon'_{q_1,p_2} & \varepsilon'_{q_2,p_2} & & \\ & -\varepsilon'_{q_2,p_3} & \ddots & \\ & & \ddots & \varepsilon'_{q_{l-1},p_{l-1}} \\ & & & -\varepsilon'_{q_{l-1},p_l} \end{pmatrix} \tag{5.11}$$

by using the remark in Definition 36. Note that there is no need for new computations here. In the previous step we have symbolically calculated the minimal projective presentation by calculating $\mathrm{Source}(U) = \{p_1, \dots, p_l\}$, $\mathrm{Source}(U')$, and $q_d = p_d \vee p_{d+1}$ for $d = 1, \dots, l-1$, and $c(r) = \min\{c \mid p_c \le r\}$ for each $r \in \mathrm{Source}(U')$, at total cost of $O(z\log(z))$. In this step we essentially only turned $\varepsilon_{q,p}$ to $\varepsilon'_{q,p}$.

(Line 4 of Algorithm 4.) In this part, we need to compute

$$\tau L := \mathrm{Ker}(\nu f_1 : \nu P_1 \to \nu P_0)$$

First, we need to express $\nu f_1 = (\nu f'_L, \nu f_U) : \nu P_1 \to \nu P_0$, so far computed only symbolically, in terms of vector spaces and linear maps (as a representation).

The entries of $\nu f_1 = (\nu f'_L, \nu f_U)$ involve the morphisms of the form $\varepsilon'_{q,p}$ (see Equation (5.11)). Fix one such $\varepsilon'_{q,p}$, and recall its definition in Definition 36. For each vertex $v \in (\vec{G}_{m,n})_0$, we compare v with p and q. If $v \le p$ and $v \le q$, then we put the scalar 1_K in the appropriate entry. Over all vertices, this operation costs $O(mn)$ in total. Then, since νf_U contains $2(l-1)$ entries and $\nu f'_L$ contains $l' = |\mathrm{Source}(U')|$ entries involving $\varepsilon'_{q,p}$, expressing νf_1 as a collection of matrices costs $O(mn((l+l')) \le O(mnz)$.

For the computation of the kernel, we also need the internal maps $(\nu P_1)(\alpha)$ for all $\alpha \in (\vec{G}_{m,n})_1$. We let $S_1 := \mathrm{Source}(U') \cup \{q_d \mid 1 \le d \le q-1\}$. Then

$$\nu P_1 = \bigoplus_{r\in\mathrm{Source}(U')} I(r) \oplus \bigoplus_{d=1}^{l-1} I(q_d) = \bigoplus_{r\in S_1} I(r).$$

For fixed arrow α in $\vec{G}_{m,n}$, let $a = \#\{r \in S_1 \mid s(\alpha) \le r\}$ and $b = \#\{r \in S_1 \mid t(\alpha) \le r\}$. We have $(\nu P_1)(\alpha) = \bigoplus_{r\in S_1} I(r)(\alpha) \colon K^a \to K^b$, and for each $r \in S_1$,

$$I(r)(\alpha) : \begin{cases} (Ks(\alpha) \to Kt(\alpha)) : [\lambda s(\alpha) \mapsto \lambda t(\alpha)], \lambda \in K, & s(\alpha), t(\alpha) \le r, \\ 0 \to Kt(\alpha), & s(\alpha) \not\le r, t(\alpha) \le r \\ Ks(\alpha) \to 0, & s(\alpha) \le r, t(\alpha) \not\le r \\ 0 \to 0, & \text{otherwise}. \end{cases} \tag{5.12}$$

Then for each $r \in S_1$, we determine the row and column in the $b \times a$ matrix corresponding to r. Note that only in the case of $s(\alpha) \le r$ and $t(\alpha) \le r$ will there be a corresponding entry. In that case, we put a 1 in the matrix. The rest of the entries of the matrix are 0. Since $\#S_1 = l' + l - 1$, this costs $O(l' + l)$ for each α. Then, since there are $O(mn)$ arrows, we get a total cost of $O(mn(l' + l)) \le O(mnz)$.

Having expressed $\nu f_1 : \nu P_1 \to \nu P_0$ in terms of vector spaces and linear maps, next we discuss the computation of $\operatorname{Ker} \nu f_1$. In general, for a linear map $\phi\colon K^p \to K^q$, we can get an injection $\sigma_\phi\colon \operatorname{Ker}\phi \to K^p$ by performing column operations on the augmented matrix:

$$\left(\begin{array}{c}\phi\\ \hline I_p\end{array}\right) \overset{\text{col ops}}{\rightsquigarrow} \left(\begin{array}{c|c}\overbrace{\text{col echelon form}}^{\operatorname{rank}\phi\text{ columns}} & 0\\ \hline & \sigma_\phi\end{array}\right)$$

where I_p denotes the identity matrix of size p. Since σ_ϕ is a section map, there exists the retraction σ'_ϕ such that $\sigma'_\phi\sigma_\phi = I_{\operatorname{rank}\sigma_\phi}$. This σ'_ϕ is also obtained by the following elementary transformations of the matrix:

$$\left(\begin{array}{c|c}\sigma_\phi & I_p\end{array}\right) \overset{\text{row ops}}{\rightsquigarrow} \left(\begin{array}{c|c}I_{\operatorname{rank}\sigma_\phi} & \sigma'_\phi\\ \hline 0 & \end{array}\right).$$

Hence, for a morphism $F\colon M \to N$ in $\operatorname{rep}(Q, R)$, we can compute $\operatorname{Ker} F$. For each vertex $v \in Q_0$, we have $(\operatorname{Ker} F)(v) := \operatorname{Ker}(F_v) = K^{\operatorname{rank}\sigma_{F_v}}$ and for each arrow $\alpha\colon u \to v$ in Q, we have

$$(\operatorname{Ker} F)(\alpha) := \sigma'_{F_v} M(\alpha)\sigma_{F_u}.$$

Namely, $\operatorname{Ker} F$ is constructed to make the following diagram commutative.

$$\begin{tikzcd}
0 \arrow[r] & (\operatorname{Ker} F)(u) \arrow[r, "\sigma_{F_u}"] \arrow[d, dotted, "(\operatorname{Ker} F)(\alpha)"'] & M(u) \arrow[r, "F_u"] \arrow[d, "M(\alpha)"] & N(u) \arrow[d, "N(\alpha)"] \\
0 \arrow[r] & (\operatorname{Ker} F)(v) \arrow[r, "\sigma_{F_v}"'] & M(v) \arrow[r, "F_v"'] & N(v).
\end{tikzcd}$$

Note that in our setting of computing $\operatorname{Ker} \nu f_1$, we have $q = \dim \nu P_0(u) \le z$ and $p = \dim \nu P_1(u) \le 2z$ for all $u \in (\vec{G}_{m,n})_0$. Then the computation of $(\operatorname{Ker} \nu f_1)(v) := \operatorname{Ker}((\nu f_1)_v)$ for all vertices v costs $O(z^\omega mn)$ via column echelon form computations. Furthermore, the computation of the internal maps $(\operatorname{Ker} \nu f_1)(\alpha) := \sigma'_{(\nu f_1)_v}[(\nu P_1)(\alpha)]\sigma_{(\nu f_1)_u}$ for all arrows α costs $O(z^\omega mn)$ total via matrix multiplications.

(Line 5 of Algorithm 4.) In this part of the algorithm, we compute the morphism θ_L below, which will be used in the next line (Line 6 of Algorithm 4) for computing the middle term E_L of the Auslander-Reiten sequence ending at V_L. Referring to Algorithm 4, $\theta_L := \theta_Z$ is given by the composite

$$\theta_Z\colon Z \xrightarrow{\text{can}} \operatorname{top} Z \xrightarrow{\pi} S \hookrightarrow \operatorname{top} Z \xrightarrow{\overline{\varepsilon}^{-1}} \operatorname{top} P_0 \xrightarrow{\sim} \operatorname{soc} \nu P_0,$$

where S is any simple direct summand of $\operatorname{top} Z$, and $\pi\colon \operatorname{top} Z \to S$, $\operatorname{top} P_0 \xrightarrow{\sim} \operatorname{soc}\nu P_0$ are the canonical isomorphisms. Furthermore, the isomorphism $\operatorname{top} Z \xrightarrow{\overline{\varepsilon}^{-1}} \operatorname{top} P_0$ is the one induced by $\varepsilon : P_0 \to Z$.

Now let $L = \{[b_j, d_j]_j \mid s \le j \le t\} \in \mathbb{I}_{m,n}$, and set $\operatorname{Source}(L) = \{p_1, \ldots, p_l\}$ as in Proposition 41. Then we may take $S := S(p_1)$, and θ_L can be defined by

$$\theta_L := {}^t(\delta_{c,1}\varepsilon'_{V_L,p_1})_{c=1}^{l} \colon V_L \to \nu P_0 = \bigoplus_{c=1}^{l} I(p_c).$$

For each vertex $u \in L_0$, we have

$$(\theta_L)_u = \begin{pmatrix}(\varepsilon'_{V_L,p_1})_u\\ 0\end{pmatrix} \colon V_L(u) \to I(p_1)(u) \oplus \bigoplus_{c=2}^{l} I(p_c)(u)$$

where the entry $\varepsilon'_{V_L,p_1}(u)$ is 1 if $p_1 \ge u$ and 0 otherwise. Computing over all vertices, we have a total cost of $O(mn)$.

(Line 6 of Algorithm 4.) In this line, we compute the middle term $E_L := E_Z$. Note that E_L is given by the pullback

$$\begin{tikzcd}
E_Z \arrow[r, dashed] \arrow[d, dashed] & V_L \arrow[d, "\theta_L"] \\
\nu P_1 \arrow[r, "\nu f_1"] & \nu P_0
\end{tikzcd}$$

and can be computed as the kernel of

$$\nu P_1 \oplus V_L \xrightarrow{(-\nu f_1, \theta_L)} \nu P_0$$

(see [13, Appendix A Definition 5.1]). In particular, we have $E_L \subseteq \nu P_1 \oplus V_L$. We can compute this kernel by the method explained above, or instead, we can build it using information we already have.

Obviously we have $E_L \supseteq \operatorname{Ker} \nu f_1 \oplus \operatorname{Ker} \theta_L = \tau V_L \oplus \operatorname{Ker} \theta_L$. Let $S = S(p_1)$ be the simple direct summand of top Z chosen above (in Line 5 of Algorithm 4), and let $w := \binom{p_1}{p_1} \in \nu P_1 \oplus V_L$, where the first entry p_1 is $p_1 \in I(q_1)$ in

$$\nu P_1 = \bigoplus_{r \in \mathrm{Source}(U')} I(r) \oplus \bigoplus_{d=1}^{l-1} I(q_d),$$

and the second entry p_1 is the obvious $p_1 \in V_L$. Then w is in E_L because

$$(\nu f_1)_{p_1}(p_1) = ((\nu f'_L)_{p_1}, (\nu f_U)_{p_1})(p_1) = (\varepsilon'_{q_1,p_1})_{p_1}(p_1) = p_1 = (\theta_L)_{p_1}(p_1).$$

From the exact sequences of the forms

$$0 \to \tau V_L \to E_L \to V_L \to 0 \quad \text{and} \quad 0 \to \operatorname{Ker} \theta_L \to V_L \to S \to 0$$

we have $\dim E_L = \dim \tau V_L + \dim \operatorname{Ker} \theta_L + 1$. Therefore noting that $w \notin \tau V_L \oplus \operatorname{Ker} \theta_L$, we have $E_L = \tau V_L \oplus \operatorname{Ker} \theta_L \oplus K w$ as a vector space. Let L' be the interval $\mathrm{full}(L_0 \setminus \{p_1\})$. Then we have $\operatorname{Ker} \theta_L = V_{L'}$, and hence we finally have

$$E_L = \tau V_L \oplus V_{L'} \oplus K w \ (\subseteq \nu P_1 \oplus V_L).$$

The representation structure of E_L is defined by those of τV_L, $V_{L'}$ and that of $K w$ defined by

$$\begin{cases} E_L(\alpha_{p_1}) w := \begin{pmatrix} \tau V_L(\alpha_{p_1})(p_1) \\ V_{L'}(\alpha_{p_1})(p_1) \end{pmatrix}, \\ E_L(\beta_{p_1}) w := \begin{pmatrix} \tau V_L(\beta_{p_1})(p_1) \\ V_{L'}(\beta_{p_1})(p_1) \end{pmatrix} = \begin{pmatrix} I(q_1)(\beta_{p_1})(p_1) \\ V_{L'}(\beta_{p_1})(p_1) \end{pmatrix} = \begin{pmatrix} 0 \\ V_{L'}(\beta_{p_1})(p_1) \end{pmatrix}, \end{cases} \tag{5.13}$$

where $q_1 := p_1 \vee p_2$ as in Proposition 39, and $\alpha_{p_1}, \beta_{p_1}$ are the horizontal arrow and the vertical arrow of $\vec{G}_{m,n}$ starting from p_1, respectively. Here we note that each entry of right most terms in (5.13) is given as follows:

$$\tau V_L(\alpha_{p_1})(p_1) = \begin{cases} I(q_1)(\alpha_{p_1})(p_1) = p_1 + (1,0) & \text{if } p_1 + (1,0) \in \operatorname{supp} \tau V_L, \\ 0 & \text{otherwise}, \end{cases}$$

$$V_{L'}(\alpha_{p_1})(p_1) = \begin{cases} p_1 + (1,0) & \text{if } p_1 + (1,0) \in L', \\ 0 & \text{otherwise} \end{cases} ; \text{and}$$

$$V_{L'}(\beta_{p_1})(p_1) = \begin{cases} p_1 + (0,1) & \text{if } p_1 + (0,1) \in L', \\ 0 & \text{otherwise}. \end{cases}$$

Therefore, for each arrow $\alpha : s \to t$ in $\vec{G}_{m,n}$, $E_L(\alpha)$ is given as follows:

$$E_L(\alpha) = \begin{pmatrix} \tau V_L(\alpha) & 0 \\ 0 & V_{L'}(\alpha) \end{pmatrix} : \tau V_L(s) \oplus V_{L'}(s) \to \tau V_L(t) \oplus V_{L'}(t)$$

if $s \neq p_1$ and $t \neq p_1$;

$$E_L(\alpha) = \begin{pmatrix} \tau V_L(\alpha) & 0 \\ 0 & V_{L'}(\alpha) \\ 0 & 0 \end{pmatrix} : \tau V_L(s) \oplus V_{L'}(s) \to \tau V_L(p_1) \oplus V_{L'}(p_1) \oplus (K w)(p_1)$$

if $t = p_1$ (this forces to have $s \neq p_1$), and finally if $s = p_1$ (i.e., if $\alpha \in \{\alpha_{p_1}, \beta_{p_1}\}$),

$$E_L(\alpha_{p_1}) = \begin{pmatrix} \tau V_L(\alpha_{p_1}) & 0 & \tau V_L(\alpha_{p_1})(\pi'_1)_{p_1} \\ 0 & V_{L'}(\alpha_{p_1}) & V_L(\alpha_{p_1})(\pi'_2)_{p_1} \end{pmatrix} : \\ \tau V_L(p_1) \oplus V_{L'}(p_1) \oplus (K w)(p_1) \to \tau V_L(t) \oplus V_{L'}(t),$$

and

$$E_L(\beta_{p_1}) = \begin{pmatrix} \tau V_L(\beta_{p_1}) & 0 & 0 \\ 0 & V_{L'}(\beta_{p_1}) & V_L(\beta_{p_1})(\pi_2')_{p_1} \end{pmatrix}:$$
$$\tau V_L(p_1) \oplus V_{L'}(p_1) \oplus (Kw)(p_1) \to \tau V_L(t) \oplus V_{L'}(t),$$

where we set $\pi_1 \colon \nu P_1 \oplus V_L \to \nu P_1$, and $\pi_2 \colon \nu P_1 \oplus V_L \to V_L$ to be the canonical projections, and since $Kw \leq \nu P_1 \oplus V_L$ and $\pi_1(Kw) \leq \tau V_L$, they restrict to the morphisms $\pi_1' := \pi_1|_{Kw} \colon Kw \to \tau V_L$ and $\pi_2' := \pi_2|_{Kw} \colon Kw \to V_L$.

Note that we have already computed the maps $\tau V_L(\alpha)$, and we only need to copy the known information to create E_L. Thus, for the computational complexity, we only estimate the size of E_L, which is given by

$$\sum_{\alpha \in (\vec{G}_{m,n})_1} \dim E_L(s(\alpha)) \dim E_L(t(\alpha)) \leq mnl^2 \leq mnz^2.$$

Overall, the above arguments show the following result concerning Algorithm 4.

Proposition 42. *For $Z = V_L$ a non-projective interval representation of $\vec{G}_{m,n}$, Algorithm 4, which computes the terms of the almost split sequence ending at Z, can be performed in time complexity $O(mnz^{\omega})$ where $z = \min\{m, n\}$.*

This ends our discussion and analysis of Algorithm 4. We recall that Algorithm 4 is the major component of Algorithm 3 for computing the multiplicity. With our analysis of Algorithm 4 finished, we are ready to use its results to analyze Algorithm 3.

Proposition 43. *For L an interval representation of $\vec{G}_{m,n}$, Algorithm 3, which computes the multiplicity of L in a representation M, can be performed in time complexity $O\left(((\dim M)^{\omega} + mn)z^{\omega}\right)$ where $z = \min\{m, n\}$.*

Proof. In case that L is projective, L has support consisting of all vertices that have a directed path from g, for some fixed vertex g. The module $\operatorname{rad} L$ is simply the interval whose support is the support of L with g excluded. We then set $E_L = \operatorname{rad} L$ and $\tau L = 0$. Otherwise, if L is not projective, we use Algorithm 4 to compute τL and E_L.

Next, we need to compute:

$$d_M(L) = \dim \operatorname{Hom}(M, \tau L) - \dim \operatorname{Hom}(M, E_L) + \dim \operatorname{Hom}(M, L).$$

Thus, we need to compute $\dim \operatorname{Hom}(M, Y)$ for Y equal to τL, E_L, and L. Remark 2 of [16] shows that for M and Y representations of a bound quiver (Q, R), $\operatorname{Hom}(M, Y)$ is isomorphic (as K-vector space) to the kernel of some $D_1 \times D_0$ matrix B, where

$$D_1 = \sum_{\alpha: i \to j \text{ in } Q_1} \dim M(i) \dim Y(j)$$

and

$$D_0 = \sum_{i \in Q_0} \dim M(i) \dim Y(i).$$

In particular, we compute $\dim \operatorname{Hom}(M, Y) = D_0 - \operatorname{rank} B$.

Let us analyze the size of B, which depends on Y. Let $\Upsilon = \max_{i \in Q_0} \dim Y(i)$. Then

$$\begin{aligned} D_1 &\leq \sum_{\alpha: i \to j} \dim M(i) \Upsilon \leq 2 \sum_{i \in Q_0} \dim M(i) \Upsilon = 2\Upsilon \dim M \\ D_0 &\leq \sum_{i \in Q_0} \dim M(i) \Upsilon = \Upsilon \dim M \end{aligned}$$

where the factor 2 for D_1 comes from the fact that in $\vec{G}_{m,n}$, for each vertex i there are at most 2 arrows starting from i. We then note that using Gaussian elimination, $\operatorname{rank} B$ can be computed in time $O(2\Upsilon^{\omega}(\dim M)^{\omega})$ [24].

- In the case that $Y = \operatorname{rad} L$ or $Y = L$, $\Upsilon = 1$.
- In the case that $Y = \tau L$, we give an upper bound for Υ as below. We note that

  $$\Upsilon = \max_{i \in Q_0} \dim \tau L(i) \leq \max_{i \in Q_0} \dim \nu P_1(i) = \dim \nu P_1(1, 1) = \dim P_1(m, n)$$

  since $\tau L = \ker(\nu f_1 : \nu P_1 \to \nu P_0)$. Furthermore, the maximum of $\dim \nu P_1$ occurs at the bottom-left corner $(1, 1)$ since νP_1 is injective. The final equality follows from the definition of ν. Then since $P_1 = \bigoplus_{r \in \mathrm{Source}(U')} P(r) \oplus \bigoplus_{d=1}^{l-1} P(q_d)$ by Proposition 41, we see that

$$\dim P_1(m,n) = l' + l - 1$$

where $l' = \#\mathrm{Source}(U')$ and $l = \#\mathrm{Source}(U)$. Thus $\Upsilon \le l' + l - 1$ for $Y = \tau L$.

- Finally, for the case $Y = E_L$, the middle term of the almost split sequence, we have $\Upsilon \le l' + l$. To see this, note that $\dim E_L(i) = \dim \tau L(i) + \dim L(i)$ so that

$$\Upsilon = \max_{i\in Q_0} \dim E_L(i) \le \max_{i\in Q_0} \dim \tau L(i) + \max_{i\in Q_0} \dim L(i) \le (l' + l - 1) + 1$$

using the previous case.

Recall that $l' \le z$ and $l \le z$, where $z = \min\{m, n\}$. Combining the above, we have a time complexity of

$$\begin{aligned} & O\left(2(1 + (l' + l - 1)^\omega + (l' + l)^\omega)(\dim M)^\omega\right) + O(A) \\ = \; & O\left(z^\omega (\dim M)^\omega\right) + O(mnz^\omega) \end{aligned}$$

if L is nonprojective, where $O(A) = O(mnz^\omega)$ is the cost of computing the terms of the almost split sequence as given in Proposition 42, and $O(z^\omega(\dim M)^\omega)$ if L is projective. □

In the worst case, we need to test all $\#\mathbb{I}_{m,n}$ intervals, and we obtain the following:

Theorem 44. *For M a representation of $\vec{G}_{m,n}$, Algorithm 1, which determines whether or not M is interval-decomposable, can be performed in time complexity*

$$O\left(\{z^\omega(\dim M)^\omega + mnz^\omega\}\#\mathbb{I}_{m,n}\right)$$

where $\dim M$ *is the total dimension of M and* $z = \min\{m, n\}$.

5.3. Interval selection heuristic

An important complexity drawback of the above is the number of intervals we need to check. Given a module M, we need, in the worst case, to compute the multiplicities with respect to all intervals, which is $\#\mathbb{I}_{m,n}$ in number. We now present some heuristics that can reduce the number of intervals to be checked. Even with the use of these heuristics, the correctness of the result is still guaranteed. Their use does not improve the worst case complexity, but we can hope that we avoid checking unnecessary intervals.

Contained-support heuristic. We note that if an interval is a summand of a module M then its support is included in the support of M. Thus, the number of intervals to check can be reduced by only considering intervals included in the support of M. For example, the algorithm GETCANDIDATES(x, y) in Algorithm 2 can be improved by including this heuristic, checking inclusion in the support of dimVecRemaining at each step, as dimVecRemaining represents the part of M still unprocessed.

Line-restriction heuristic. We can further reduce the number of intervals to be tested by the following heuristic, which builds up candidate intervals by stacking 1D intervals to form 2D intervals, with the 1D intervals obtained by decomposing the restriction of M to horizontal lines.

Suppose that $M = \bigoplus_{i=1}^{\ell} T_i \oplus X$, where T_i are all interval representations and X has no interval summands. That is, $\bigoplus_{i=1}^{\ell} T_i$ is the interval-decomposable part of M. We wish to create a set of candidate intervals containing $\{T_i \mid i = 1, \ldots, \ell\}$, without knowing the decomposition given above.

The restriction of the above to a horizontal line L on the commutative grid gives $M|_L = \bigoplus_{i=1}^{\ell} T_i|_L \oplus X|_L$, a decomposition of the 1D persistence module $M|_L$. Note that since T_i are 2D interval representations, they restrict to 1D interval representations $T_i|_L$. The indecomposable decomposition of $M|_L$ necessarily contains all of these 1D intervals (together with intervals coming from $X|_L$). Note that since $M|_L$ is a 1D persistence module, it decomposes into a set of 1D intervals which we denote as $C(L)$.

We stack valid combinations of intervals from $C(L)$ over all horizontal lines L, to produce a set of 2D intervals that necessarily contain $\{T_i \mid i = 1, \ldots, \ell\}$. By a valid stacking, we mean that the resulting 2D object should be a valid interval, i.e. one with staircase shape. This procedure is described in Algorithm 5.

Lemma 45. *Algorithm 5 returns a superset of the intervals appearing in the decomposition of M.*

Proof. We successively build interval modules defined up to the ith line of the grid and extend upwards by stacking. At each stage, one copy of the result is stored in $\mathscr{L}'$ and later sent to $\mathscr{L}$ for further stacking (line 10), while another copy is finished (sent to $\mathscr{P}$) by adding zeros to the rest of the grid (line 11).

Algorithm 5 Combinatorial combination of line restrictions.

Require: 2D persistence module M over $Q = \vec{G}_{m,n}$
1: **function** StackingPotentialIntervals(M)
2: **initialize** $\mathscr{P} = \emptyset$ and $\mathscr{L} = \{0\}$, where 0 is the zero module on the grid up to the 0^{th} line.
3: **for** $i = 1, \dots, n$ **do**
4: $L_i \leftarrow$ line at height i.
5: $C(L_i) \leftarrow$ intervals in decomposition of $M|_{L_i}$.
6: $\mathscr{L}' \leftarrow \{0\}$, where 0 is the zero module on the grid up to the i^{th} line.
7: **for** $T \in \mathscr{L}$ **do**
8: **for** $S \in C(L_i)$ **do**
9: **if** S on T is a valid stacking **then**
10: Add the stacking of S on T to $\mathscr{L}'$.
11: Add the stacking of S on T extended by 0 on the whole grid to $\mathscr{P}$.
12: **end if**
13: **end for**
14: **end for**
15: $\mathscr{L} \leftarrow \mathscr{L}'$
16: **end for**
17: **return** $\mathscr{P}$
18: **end function**

The stacking operation consists of taking such a module $T \in \mathscr{L}$ defined up to the $(i-1)$th line and adding the interval S (at the ith line) on top of it. If T is not the 0 module, then by construction the top line of T is an interval $\mathbb{I}[a,b]$. The considered $S \in C(L_i)$ is always some 1D interval $\mathbb{I}[c,d]$. The stacking of S on T forms a valid 2D interval only when $c \leq a \leq d \leq b$ in order to have a staircase support. Then we form the module $T \to S$, where the arrow supports the canonical map between the two intervals. By construction, this is a 2D interval on the grid defined up to the ith line. On the other hand, if T is 0, then the module $T \to S$ is also well defined and is simply the 2D interval that is nonzero only on the ith line.

We need to check that all 2D interval modules that are part of the decomposition of M are in $\mathscr{P}$. This is a direct consequence of the following observation.

If $M = \bigoplus_{j=1}^{\ell} T_j \oplus X$, where $\bigoplus_{j=1}^{\ell} T_j$ is the interval-decomposable part of M, then for each j the restriction T_j to the ith line $T_j|_{L_i}$ is an interval in $C(L_i)$. Moreover, T_j can be rewritten as the stacking $T_j|_{L_1} \to \cdots \to T_j|_{L_n}$. Since T_j is connected, the 1D intervals $T_j|_{L_i}$ that are nonzero correspond to a contiguous sequence of indices i (line heights), say $s \leq i \leq t$ for some s and t. Thus, T_j is formed in Algorithm 5 using the zero module up to the $(s-1)$th line (Line 6 of Algorithm 5 with $i = s-1$), stacking $T_j|_{L_i} \in C(L_i)$ for $s \leq i \leq t$ (always valid stackings), and finishing up by zeros to the rest of the grid. That is $T_j \in \mathscr{P}$ at the end of the algorithm. $\square$

Image-based heuristic. Another approach, given in Algorithm 6, is to use the ranks of maps to choose which intervals to test. We start from L the zero module and iteratively add interval summands of M with their multiplicity to L. At the end, if M is interval-decomposable, then $M \cong L$. We will greedily work towards equalizing the dimension vectors of M and L. If we reach a point where the greedy procedure fails, this means that the module M is not interval-decomposable.

In lines 4 and 5, the algorithm selects the leftmost vertex s on the lowest possible line where the dimensions of the vector spaces of L and M disagrees. We then look for a rectangle B as large as possible that must be contained in the support of at least one indecomposable summand of M that does not yet appear in L. In lines 8 and 9, we achieve this by selecting a maximal element t such that the rank of the map $M(s \to t)$ is greater than the rank of $L(s \to t)$. Then B is a subset of each support of the intervals we want to find.

To reduce the number of candidates, we remark that those intervals must interact with the map $M(s \to t)$ in the following way. We first initialize $F = M_s$, and progressively consider smaller and smaller subspaces of F as we process intervals. At each iteration of the inner while loop, we consider the subspace not accounted for previously, via a complementary basis in F of the kernel of $M(s \to t)$ in line 10. We note that we exclude the kernel because we want the intervals that contain the rectangle B from s to t in its support. Then the supports of the intervals of interest must be contained in the set of vertices reachable by images and pre-images along walks starting at those basis elements. This is encoded in the sets C_i as defined in line 13. We can compute each set C_i independently, and the intervals must be contained in the union $\bigcup_{i=k+1}^{f} C_i$.

Having obtained candidate intervals, in line 15 we then compute their true multiplicities $d_M(I)$ in M, for example via Algorithm 3 as discussed above. If M is interval-decomposable, then

$$\sum_{B \subset I \subset \bigcup C_i} d_M(I) = \operatorname{rank} M(s \to t) - \operatorname{rank} L(s \to t)$$

since the intervals I being considered (which contain the rectangle from s to t), together with the already-processed $L(s \to t)$ should account for all the rank of $M(s \to t)$. Thus, in line 17 we use this condition to determine whether or not to stop early. If we are not sure that M is not interval-decomposable, in line 19 we update L and F and continue with the iteration.

Algorithm 6 Image based decomposition.

Require: A module M
1: **function** IMAGEBASEDDECOMPOSITION(M)
2: $L \leftarrow 0$.
3: **while** $\underline{\dim} M \neq \underline{\dim} L$ **do**
4: $S \leftarrow \{u \in \mathbb{Z}^2 \mid (\underline{\dim} M - \underline{\dim} L)_u \neq 0\}$.
5: $s \leftarrow$ the minimal element of S on lowest row.
6: $F \leftarrow M_s$.
7: **while** $\dim M_s \neq \dim L_s$ **do**
8: $T \leftarrow \{u \in \mathbb{Z}^2 \mid \operatorname{rank} M(s \to u) \neq \operatorname{rank} L(s \to u)\}$.
9: $t \leftarrow$ a maximal element of T.
10: Compute $\{x_1, \ldots, x_k, x_{k+1}, \ldots, x_f\}$ a basis of F so that $\{x_i\}_{i=1}^k$ is a basis of $F \cap \operatorname{Ker} M(s \to t)$.
11: $B \leftarrow$ the rectangle with lower left corner s, upper right corner t.
12: **for all** $i = k+1, \ldots, f$ **do**
13: $C_i \leftarrow \left\{ r \;\middle|\; \begin{array}{l} \exists (r_l)_{l \in [0,j]} \text{ such that } \forall 0 < l < j-3, r_l \text{ and } r_{l+2} \text{ are incomparable}, \\ r_0 = s,\ r_1 = t,\ r_j = r, \text{ and } \forall l,\ r_l \text{ does not lie below the } s\text{th line}, \\ M(r_j \to r_{j-1})^{-1} M(r_{j-1} \to r_{j-2}) \cdots M(r_2 \to r_1)^{-1} M(r_0 \to r_1) x_i \neq \{0\} \end{array} \right\}$.
14: **end for**
15: Compute $d_M(I)$ for each interval I with $B \subset \operatorname{Supp} I \subset \bigcup_{i=k+1}^{f} C_i$.
16: **if** $\sum_{B \subset I \subset \bigcup C_i} d_M(I) \neq \operatorname{rank} M(s \to t) - \operatorname{rank} L(s \to t)$ **then**
17: **return** that M is not interval-decomposable.
18: **end if**
19: $L \leftarrow L \oplus \bigoplus_{B \subset I \subset \bigcup C_i} I^{(d_M(I))}$ and $F \leftarrow F \cap \operatorname{Ker} M(s \to t)$.
20: **end while**
21: **end while**
22: **return** L, the interval decomposition of M.
23: **end function**

Lemma 46. *If M is interval-decomposable then Algorithm 6 returns the interval decomposition of M. Otherwise Algorithm 6 stops and indicates that M is not interval-decomposable.*

Proof. First, if the algorithm returns a module L then it is necessarily the interval decomposition of M. Indeed, by construction, L is a direct sum of intervals, and every such interval appears exactly the same number of times in L as it does in M. So M is isomorphic to the direct sum of L with another module L'. Moreover, $\underline{\dim} M = \underline{\dim} L$ if L is returned, and so $L' = 0$ and $M \cong L$.

Second, we need to show that the algorithm always terminates. If $\underline{\dim} M \neq \underline{\dim} L$ then $S \neq 0$ and s is well defined. For the second while loop, if $\dim M_s \neq \dim L_s$ then T is not empty because $M(s \to s)$ is the identity with rank equal to $\dim M_s \neq \dim L_s$, and so $s \in T$. Thus, a maximal element t of T exists. Therefore, for every round of this loop, either the algorithm returns that M is not interval-decomposable, or the dimension F decreases and T is reduced.

Finally, we must show that if M is interval-decomposable, Algorithm 6 will always return a module L. Assume that M is interval-decomposable. Arriving at line 16, we have the following properties. We have picked two elements s and t, and we already know an interval decomposable module L that appears in the decomposition of M. As M is interval-decomposable, we have at most $\operatorname{rank} M(s \to t)$ distinct interval modules $I_1, \ldots, I_r$ in the decomposition of M such that $\operatorname{rank} I_i(s \to t) = 1$. Note that $\sum_{i=1}^r d_M(I_i) = \operatorname{rank} M(s \to t)$.

We separate the set $\{I_1, \ldots, I_r\}$ in two sets depending on their lowest left corner, i.e. the leftmost vertex on the lowest line of its support.

(1) If $I \in \{I_1, \ldots, I_r\}$ has a lowest left corner $s' \neq s$ then either s' is on the left of s or on a line below s. In both cases, due to the choice of s at line 5, we have $\underline{\dim} M_{s'} = \underline{\dim} L_{s'}$. By construction, L is a summand of M, hence for every J such that $J_{s'} \neq 0$, $d_L(J) = d_M(J)$. In particular, $d_L(I) = d_M(I)$.

(2) If $I \in \{I_1, \ldots, I_r\}$ has lowest left corner s, we consider $t_1, \ldots, t_j$ the set of all t that have been considered since s has been picked, excluding the current t. An interval $I \in \{I_1, \ldots, I_r\}$ can fall into two categories. If there exists $1 \leq k \leq j$ such that $\operatorname{rank} I(s \to t_k) \neq 0$ then I has been considered when considering the first such k and added to L with the correct multiplicity, i.e. $d_L(I) = d_M(I)$.

Otherwise, $I_s \subset \bigcap_{l=1}^{j} \operatorname{Ker} M(s \to t_j) = F$. Let us show that the support of I is included in $\bigcup C_i$. As M is interval decomposable and I is a summand of M, say $M = I \oplus N$, there exists a choice of basis for M_u such that $M_u = I_u \oplus N_u$ for every vertex $u \in (\vec{G}_{m,n})_0$. In particular $M_s = I_s \oplus N_s$. Let $0 \neq x \in I_s$. The support of I is connected, so for every v in the support of I, there exists a walk $(v_1, \ldots, v_p)$ of elements of the support of I such that v_q and v_{q+1} are adjacent, $v_1 = t$ and $v_p = v$. Furthermore, the map $I(v_q \to v_{q'})$ is an isomorphism if $v_q \leq v_{q'}$.

As the support of I is convex and contains no element below the line of s, we can build an alternating walk of paths by extracting a subsequence $(r_l)_{l=1}^{j}$ of $(v_1, \ldots, v_p)$ such that no r_l lies on a line lower than s, r_l and r_{l+2} are incomparable for $l < j-3$, $r_1 = t$ and $r_j = v$. Then fix $r_0 = s$. For ease of notation, we require that j is even. If the length is odd, we simply put $r_{j+1} = r_j$ and use the subsequence until r_{j+1}.

The map $I(r_j \to r_{j-1})^{-1} I(r_{j-1} \to r_{j-2}) \cdots I(r_2 \to r_1)^{-1} I(r_0 \to r_1)$ is an isomorphism. Translated into M, this implies that

$$M(r_j \to r_{j-1})^{-1} M(r_{j-1} \to r_{j-2}) \cdots M(r_2 \to r_1)^{-1} M(r_0 \to r_1)(x) \neq \{0\}.$$

As x can be expressed as a linear combination of $\{x_1, \dots, x_f\}$, there exists at least one $i \in \{1, \dots, f\}$ such that

$$M(r_j \to r_{j-1})^{-1} M(r_{j-1} \to r_{j-2}) \cdots M(r_2 \to r_1)^{-1} M(r_0 \to r_1)(x_i) \neq \{0\}.$$

Since all elements x_j for $j \le k$ are part of $\operatorname{Ker} M(s \to t)$, we have $i \ge k+1$ and $v \in C_i$.

Moreover, $\operatorname{rank} I(s \to t) \neq 0$ and thus the support of I also contains the rectangle B. Note also that there is no double counting. Therefore $\operatorname{rank} M(s \to t) = \operatorname{rank} L(s \to t) + \sum d_M(I)$, and the algorithm does not incorrectly stop early. $\square$

5.4. *Interval decomposability with [12] decomposition algorithm*

Dey and Xin proposed in [12] a generalization of the persistence algorithm to decompose multidimensional persistence modules. Once an indecomposable decomposition is computed, testing for interval-decomposability is a simple matter as one only needs to check that all elements of the decomposition are interval modules. The generalized persistence algorithm however is limited to a specific case: In the matrix encoding the minimal presentation of the module, no pair of columns nor pair of rows can have the same grade. Translated into the language of this paper, this means that the generators of the projective modules appearing in the minimal projective presentation of the module are all distinct.

It was suggested in [12] that for modules not satisfying this property, an easy workaround can be implemented. In this case, the generalized persistence algorithm does not always provide a full decomposition. It nonetheless returns a direct sum decomposition of the module, the only limitation being that some of the summands might not be indecomposable. The suggested workaround is to arbitrarily fix an order on the rows and columns which have same grades, in essence artificially breaking ties in the grades. By exhaustively checking all such tie-breaking orders on the rows and columns, it was claimed that the algorithm will provide a full decomposition at some point.

This workaround is valid if there exists an order that allows for a full decomposition. Unfortunately, this is not always true as we show with the following example. We show that for whatever order chosen for elements with the same grade, no full decomposition can be obtained through the algorithm from [12]. In fact, we show a stronger statement, that for whatever order chosen for elements with the same grade, no algorithm using only "matrix operations in one direction" can obtain a full decomposition.

Example 47. We consider the field $K = \mathbb{F}_{2^2} = \mathbb{F}_2(\lambda) = \{0, 1, \lambda, \lambda^2\}$, where λ satisfies $\lambda^2 + \lambda + 1 = 0$. Let

$$\begin{tikzcd}
& c \arrow[rr] & & \omega \\
z \arrow[rr] \arrow[ur] & & b \arrow[ur] & \\
& y \arrow[rr] \arrow[uu] & & a \arrow[uu] \\
\alpha \arrow[rr] \arrow[ur] \arrow[uu] & & x \arrow[ur] \arrow[uu] &
\end{tikzcd}
\quad \text{and} \quad M:
\begin{tikzcd}
& K^2 \arrow[rr] & & 0 \\
K^2 \arrow[rr] \arrow[ur] & & K^2 \arrow[ur] & \\
& K^2 \arrow[rr] \arrow[uu] & & K^2 \arrow[uu] \\
0 \arrow[rr] \arrow[ur] \arrow[uu] & & K^2 \arrow[ur, "\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}"'] \arrow[uu] &
\end{tikzcd}$$

be the $2 \times 2 \times 2$ commutative cube $\vec{G}_{2,2,2}$ and a K-representation of $\vec{G}_{2,2,2}$, respectively.

Then one can calculate the following minimal projective presentation for M:

$$P(a)^2 \oplus P(b)^2 \oplus P(c)^2 \xrightarrow{p_1} P(x)^2 \oplus P(y)^2 \oplus P(z)^2 \xrightarrow{p_0} M \longrightarrow 0$$

where $P(v)$ is the indecomposable projective representation with source v, and where the morphism p_1 can be given in matrix form as

$$p_1: \begin{array}{c|cc|cc|cc}
 & P(a)^2 & & P(b)^2 & & P(c)^2 & \\
\hline
P(x)^2 & 1 & 1 & 1 & 0 & 0 & 0 \\
 & 1 & 0 & 0 & 1 & 0 & 0 \\
\hline
P(y)^2 & 1 & 0 & 0 & 0 & 1 & 0 \\
 & 0 & 1 & 0 & 0 & 0 & 1 \\
\hline
P(z)^2 & 0 & 0 & 1 & 0 & 1 & 0 \\
 & 0 & 0 & 0 & 1 & 0 & 1
\end{array}. \tag{5.14}$$

First, let us show that M is decomposable. We note that vertices x, y, and z do not have any arrows between them, and similarly for a, b, c. Thus, allowable matrix operations are restricted to within each block row or column in Equation (5.14). For ease of notation, let X be the matrix $X = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$. Then p_1 transforms as

$$p_1 : \left[\begin{array}{c|c|c} X & I & 0 \\ \hline I & 0 & I \\ \hline 0 & I & I \end{array}\right] \cong \left[\begin{array}{c|c|c} PX & P & 0 \\ \hline I & 0 & I \\ \hline 0 & I & I \end{array}\right] \cong \left[\begin{array}{c|c|c} PX & PP^{-1} & 0 \\ \hline I & 0 & I \\ \hline 0 & P^{-1} & I \end{array}\right] \cong \left[\begin{array}{c|c|c} PX & PP^{-1} & 0 \\ \hline I & 0 & I \\ \hline 0 & PP^{-1} & P \end{array}\right]$$

$$\cong \left[\begin{array}{c|c|c} PX & PP^{-1} & 0 \\ \hline I & 0 & P^{-1} \\ \hline 0 & PP^{-1} & PP^{-1} \end{array}\right] \cong \left[\begin{array}{c|c|c} PX & PP^{-1} & 0 \\ \hline P & 0 & PP^{-1} \\ \hline 0 & PP^{-1} & PP^{-1} \end{array}\right]$$

$$\cong \left[\begin{array}{c|c|c} PXP^{-1} & PP^{-1} & 0 \\ \hline PP^{-1} & 0 & PP^{-1} \\ \hline 0 & PP^{-1} & PP^{-1} \end{array}\right] = \left[\begin{array}{c|c|c} PXP^{-1} & I & 0 \\ \hline I & 0 & I \\ \hline 0 & I & I \end{array}\right]$$

for P any invertible 2×2 matrix, by alternating row and column operations with respect to P. That is, the matrix form of p_1 can be transformed by conjugation of X, without affecting the other block entries.

By letting

$$P = \begin{bmatrix} 1 & \lambda \\ \lambda & 1 \end{bmatrix}, \text{ we have } P^{-1} = \begin{bmatrix} \lambda^2 & 1 \\ 1 & \lambda^2 \end{bmatrix}$$

since $\lambda^2 + \lambda = -1 = 1$ and $\lambda^3 + 1 = 0$ in the base field $K = \mathbb{F}_2(\lambda)$. Thus

$$PXP^{-1} = \begin{bmatrix} 1 & \lambda \\ \lambda & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda^2 & 1 \\ 1 & \lambda^2 \end{bmatrix} = \begin{bmatrix} 1+\lambda & 1 \\ 1+\lambda & \lambda \end{bmatrix} \begin{bmatrix} \lambda^2 & 1 \\ 1 & \lambda^2 \end{bmatrix} = \begin{bmatrix} \lambda^2 & 0 \\ 0 & \lambda \end{bmatrix}.$$

The above computations show that

$$p_1 \cong \begin{array}{c} \\ P(x)^2 \\ \\ P(y)^2 \\ \\ P(z)^2 \end{array} \!\!\!\! \begin{array}{c} \begin{array}{ccc} P(a)^2 & P(b)^2 & P(c)^2 \end{array} \\ \left[\begin{array}{cc|cc|cc} \lambda^2 & 0 & 1 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{array}\right] \end{array} = \left[\begin{array}{c|c|c} \lambda^2 & 1 & 0 \\ \hline 1 & 0 & 1 \\ \hline 0 & 1 & 1 \end{array}\right] \oplus \left[\begin{array}{c|c|c} \lambda & 1 & 0 \\ \hline 1 & 0 & 1 \\ \hline 0 & 1 & 1 \end{array}\right], \tag{5.15}$$

with the two summands each inducing a nontrivial summand of $M \cong \operatorname{Coker} p_1 \cong \operatorname{Coker} p_1' \oplus \operatorname{Coker} p_1''$, where p_1' (p_1'', respectively) is the first (second, respectively) direct summand of p_1 in the decomposition above.

Next, let us consider the workaround proposed by Dey and Xin, for when generators of the projectives appearing in the projective presentation of M have equal grades. This is exactly the case we have here, as we have each two copies of $P(v)$ for $v = a, b, c$ and $v = x, y, z$.

In general, without any restrictions, p_1 given in Equation (5.14) can be transformed into

$$\begin{aligned} &\left[\begin{array}{c|c|c} A_1 1_{P(x)} & 0 & 0 \\ \hline 0 & A_2 1_{P(y)} & 0 \\ \hline 0 & 0 & A_3 1_{P(z)} \end{array}\right] \left[\begin{array}{c|c|c} X & I & 0 \\ \hline I & 0 & I \\ \hline 0 & I & I \end{array}\right] \left[\begin{array}{c|c|c} B_1 1_{P(a)} & 0 & 0 \\ \hline 0 & B_2 1_{P(b)} & 0 \\ \hline 0 & 0 & B_3 1_{P(c)} \end{array}\right] \\ &= \left[\begin{array}{c|c|c} A_1 X B_1 & A_1 B_2 & 0 \\ \hline A_2 B_1 & 0 & A_2 B_3 \\ \hline 0 & A_3 B_2 & A_3 B_3 \end{array}\right] \end{aligned} \tag{5.16}$$

where X is as defined above, A_1, A_2, A_3, B_1, B_2, B_3 are invertible 2×2 K-matrices, and $1_{P(v)}$ is the identity morphism of $P(v)$. Note that since there are no nonzero morphisms among $P(x)$, $P(y)$, $P(z)$, and among $P(a)$, $P(b)$, $P(c)$, the off-diagonal blocks are always zero.

The workaround involves arbitrarily fixing an order for the rows and columns which have the same grades, and running their algorithm. Their algorithm only performs row/column operations in "one direction" with respect to the fixed order.

This involves restricting A_1, A_2, A_3, B_1, B_2, B_3 in the transformation matrices to either being upper or lower triangular. We note that we do not a priori require the matrices to be all upper or all lower. We show that it is impossible to compute a decomposition for p_1 with this restriction.

First, let us study the product AB of two upper or lower invertible 2×2 matrices A and B, since p_1 after transformation (Equation (5.16)) contains blocks of that form. Let $A = \begin{bmatrix} a_1 & a_u \\ a_l & a_2 \end{bmatrix}$ and $B = \begin{bmatrix} b_1 & b_u \\ b_l & b_2 \end{bmatrix}$. Since we impose that A be upper or lower triangular and invertible, we have $a_u = 0$ or $a_l = 0$, and $a_1 \neq 0, a_2 \neq 0$, with similar conditions on B.

Furthermore, we know *one particular* decomposition of p_1 as given in Equation (5.15), with each block row and block column decomposing into two. Thus, any other nontrivial decomposition of p_1 must have its blocks of the form AB be diagonal matrices. Note that given the restrictions on A and B, AB cannot be anti-diagonal.

Requiring the diagonality of

$$AB = \begin{bmatrix} a_1 b_1 + a_u b_l & a_1 b_u + b_2 a_u \\ b_1 a_l + a_2 b_l & a_2 b_2 + a_l b_l \end{bmatrix}$$

is equivalent to requiring that $a_1 b_u + b_2 a_u = 0$ and $b_1 a_l + a_2 b_l = 0$. Since a_1, a_2, b_1, b_2 are all nonzero, we conclude that $a_u = 0$ if and only if $b_u = 0$, and $a_l = 0$ if and only if $b_l = 0$. That is, the "shape" (upper or lower) of A is the same as the "shape" of B.

The transformed p_1 in Equation (5.16) has blocks $A_1 B_2$, $A_2 B_1$, $A_2 B_3$, $A_3 B_2$, and $A_3 B_3$. Requiring that they all be diagonal implies that the shapes of A_1, A_2, A_3, B_1, B_2, B_3 are all the same. That is, the transformation blocks need to all be upper triangular, or all lower triangular.

Finally, we consider the final block $A_1 X B_1$ in Equation (5.16), where $X = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ as before. Suppose that all the transformation blocks are upper triangular. In particular, A_1 and B_1 are upper triangular ($a_l = 0$, $b_l = 0$). Then

$$A_1 X B_1 = \begin{bmatrix} a_1 b_1 + b_1 a_u & a_1 b_2 + a_1 b_u + b_2 a_u \\ a_2 b_1 & a_2 b_u \end{bmatrix}.$$

Since $a_2 b_1$ cannot be zero, this cannot be diagonal. Similarly, in the case that all the transformation blocks are lower triangular, $A_1 X B_1$ cannot be diagonal. By the arguments above, there are no other possibilities for obtaining a nontrivial decomposition of p_1. Thus, given the restrictions on the transformations on p_1, one cannot obtain a full decomposition of p_1.

6. Conclusion

In this paper we presented an algorithm for testing $\mathscr{S}$-decomposability for any finite set $\mathscr{S}$ of indecomposables, based on the procedure [16] for computing the multiplicity of a given indecomposable in the decomposition of a module. We specifically studied the case of interval-decomposability by first providing a characterization and an enumeration method for interval modules in the 2D equioriented commutative grid case. To the extent of our knowledge, this is the first algorithm to test interval-decomposability of a module, without the need for computing its full decomposition if the answer is negative.

Interval modules have a very specific structure that made computation easier, especially the fact that their endomorphism rings are isomorphic to the underlying field K. This slightly simplified Algorithm 4, an essential component of the algorithm, compared to the general procedure (see Section 3.2 of [23]). When considering a different class $\mathscr{S}$ of indecomposables, the aforementioned simplification in Algorithm 4 may no longer hold, but the general procedure is still valid.

Another generalization is to consider interval modules of nD commutative grids with $n > 2$. More generally, in any finite bound quiver, we can still define and enumerate interval modules by using a brute-force approach. Then we can apply our interval-decomposability algorithm. However, the brute-force enumeration comes with an additional cost as we do not yet have an easy characterization of intervals in the general case, in contrast to the staircase shape of 2D intervals.

For the case of 2D equioriented commutative grids considered in this paper, we also provided several heuristics to try to speed up the enumeration of interval modules and the testing of interval-decomposability. It would be interesting to implement these various heuristics and conduct an in-depth comparison on practical instances to evaluate their performances.

Funding

This work was partially supported by JST CREST Mathematics (15656429). M.Y. was partially supported by Osaka City University Advanced Mathematical Institute (MEXT Joint Usage/Research Center on Mathematics and Theoretical Physics JPMXP0619217849). M.B. was partially supported by FWF (P33765).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to thank Tamal Dey and Cheng Xin for bringing to our attention the question of determining interval-decomposability.

References

[1] Edelsbrunner, Letscher, Zomorodian, Topological persistence and simplification, Discrete Comput. Geom. 28 (4) (2002) 511–533, https://doi.org/10.1007/s00454-002-2885-2.
[2] G. Carlsson, V. De Silva, Zigzag persistence, Found. Comput. Math. 10 (4) (2010) 367–405.
[3] P. Gabriel, Unzerlegbare darstellungen I, Manuscr. Math. 6 (1) (1972) 71–103.
[4] G. Carlsson, A. Zomorodian, The theory of multidimensional persistence, Discrete Comput. Geom. 42 (1) (2009) 71–93.
[5] W.W. Crawley-Boevey, On tame algebras and bocses, Proc. Lond. Math. Soc. 3 (3) (1988) 451–483.
[6] T.K. Dey, C. Xin, Computing bottleneck distance for 2-d interval decomposable modules, in: 34th International Symposium on Computational Geometry (SoCG 2018), Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2018.
[7] H.B. Bjerkevik, Stability of higher-dimensional interval decomposable persistence modules, arXiv preprint, arXiv:1609.02086, 2016.
[8] M. Botnan, M. Lesnick, Algebraic stability of zigzag persistence modules, Algebraic Geom. Topol. 18 (6) (2018) 3133–3204.
[9] J. Cochoy, S. Oudot, Decomposition of exact pfd persistence bimodules, Discrete Comput. Geom. 63 (2) (2020) 255–293.
[10] M.B. Botnan, V. Lebovici, S. Oudot, On rectangle-decomposable 2-parameter persistence modules, in: 36th International Symposium on Computational Geometry (SoCG 2020), Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
[11] M.B. Botnan, V. Lebovici, S. Oudot, Local characterizations for decomposability of 2-parameter persistence modules, arXiv preprint, arXiv:2008.02345, 2020.
[12] T.K. Dey, C. Xin, Generalized persistence algorithm for decomposing multi-parameter persistence modules, arXiv preprint, arXiv:1904.03766v5, 2020.
[13] I. Assem, A. Skowroński, D. Simson, Elements of the Representation Theory of Associative Algebras: Volume 1: Techniques of Representation Theory, vol. 65, Cambridge University Press, 2006.
[14] M. Herschend, Tensor products on quiver representations, J. Pure Appl. Algebra 212 (2) (2008) 452–469.
[15] M. Auslander, I. Reiten, S.O. Smalø, Representation Theory of Artin Algebras, vol. 36, Cambridge University Press, 1997.
[16] H. Asashiba, K. Nakashima, M. Yoshiwaki, Decomposition theory of modules: the case of Kronecker algebra, Jpn. J. Ind. Appl. Math. 34 (2) (2017) 489–507, https://doi.org/10.1007/s13160-017-0247-y.
[17] P. Dowbor, A. Mróz, The multiplicity problem for indecomposable decompositions of modules over a finite-dimensional algebra. Algorithms and a computer algebra approach, in: Colloquium Mathematicae, vol. 2, 2007, pp. 221–261.
[18] M. Buchet, E.G. Escolar, Realizations of indecomposable persistence modules of arbitrarily large dimension, in: 34th International Symposium on Computational Geometry (SoCG 2018), Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2018.
[19] J.-C. Aval, M. D'Adderio, M. Dukes, A. Hicks, Y. Le Borgne, Statistics on parallelogram polyominoes and a q,t-analogue of the Narayana numbers, J. Comb. Theory, Ser. A 123 (1) (2014) 271–286.
[20] E.G. Escolar, Y. Hiraoka, Persistence modules on commutative ladders of finite type, Discrete Comput. Geom. 55 (1) (2016) 100–157.
[21] C. Don, W. Shmuel, Matrix multiplication via arithmetic progressions, J. Symb. Comput. 9 (3) (1990) 251–280.
[22] V.V. Williams, Multiplying matrices faster than Coppersmith-Winograd, in: Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing, 2012, pp. 887–898.
[23] P. Gabriel, Auslander-Reiten sequences and representation-finite algebras, in: Representation Theory I, Springer, 1980, pp. 1–71.
[24] O.H. Ibarra, S. Moran, R. Hui, A generalization of the fast lup matrix decomposition algorithm and applications, J. Algorithms 3 (1) (1982) 45–56.