Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

Article

Algorithmic Information Distortions in Node-Aligned and Node-Unaligned Multidimensional Networks*

Felipe S. Abrahão [1,†,], Klaus Wehmuth [1], Hector Zenil [2] and Artur Ziviani [1]**

[1] National Laboratory for Scientific Computing (LNCC), 25651-075, Petropolis, RJ, Brazil; fsa@lncc.br, klaus@lncc.br, ziviani@lncc.br

[2] Alan Turing Institute, British Library, 2QR, 96 Euston Rd, London NW1 2DB. Oxford Immune Algorithmics, RG1 3EU, Reading, U.K.. Algorithmic Dynamics Lab, Unit of Computational Medicine, Department of Medicine Solna, Center for Molecular Medicine, Karolinska Institute, SE-171 77, Stockholm, Sweden. Algorithmic Nature Group, Laboratoire de Recherche Scientifique (LABORES) for the Natural and Digital Sciences, 75005, Paris, France; hector.zenil@cs.ox.ac.uk

** Correspondence: felipesabrahao@gmail.com or fsa@lncc.br

† Current address: National Laboratory for Scientific Computing (LNCC), 25651-075, Petropolis, RJ, Brazil

Abstract: In this article, we investigate limitations of importing methods based on algorithmic information theory from monoplex networks into multidimensional networks (such as multilayer networks) that have a large number of extra dimensions (i.e., aspects). In the worst-case scenario, it has been previously shown that node-aligned multidimensional networks with non-uniform multidimensional spaces can display exponentially larger algorithmic information (or lossless compressibility) distortions with respect to their isomorphic monoplex networks, so that these distortions grow at least linearly with the number of extra dimensions. In the present article, we demonstrate that node-unaligned multidimensional networks, either with uniform or non-uniform multidimensional spaces, can also display exponentially larger algorithmic information distortions with respect to their isomorphic monoplex networks. However, unlike the node-aligned non-uniform case studied in previous work, these distortions in the node-unaligned case grow at least exponentially with the number of extra dimensions. On the other hand, for node-aligned multidimensional networks with uniform multidimensional spaces, we demonstrate that any distortion can only grow up to a logarithmic order of the number of extra dimensions. Thus, these results establish that isomorphisms between finite multidimensional networks and finite monoplex networks do not preserve algorithmic information in general and highlight that the algorithmic information of the multidimensional space itself needs to be taken into account in multidimensional network complexity analysis.

Keywords: multidimensional networks; network complexity; lossless compression; information distortion; graph isomorphism; multiaspect graphs; multilayer networks; information content analysis; algorithmic complexity

Citation: Abrahão, F. S.; Wehmuth, K.; Zenil, H.; Ziviani, A. algorithmic information distortions in node-aligned and node-unaligned multidimensional networks. *Preprints* **2021**, *1*, 0. https://dx.doi.org/

Received:
Accepted:
Published:

1. Introduction

For multidimensional spaces that are sufficiently large and *non-uniform*, previous work has shown that there are distortions in the general case when comparing the irreducible information content (or lossless compressibility) of a *node-aligned* multidimensional network with the irreducible information content of its isomorphic monoplex network [1]. Therefore, this basically implies that currently existing methods that are based on algorithmic information theory (AIT) applied to monoplex networks (or graphs) cannot be straightforwardly imported into the multidimensional case without a proper evaluation of the algorithmic information distortions that might be present.

AIT has been playing an important role in the investigation of network complexity [2–4]. For example, it presented contributions to the challenge of causality discovery in network modeling [5], network summarization [6,7], automorphism group size [8], network

*This paper is an extended version of a previous conference paper [1], whose results correspond to the node-aligned non-uniform case presented in Section 3.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

topological properties [8,9], the principle of maximum entropy and network topology reprogrammability [10], and the reducibility problem of multiplex networks [11]. As the study of multidimensional networks, such as multilayer networks and dynamic multilayer networks, become one of the central topics in network science, further exploration of algorithmic information distortions also become relevant. In this sense, this article extends the work in [1] and presents a theoretical analysis of algorithmic information distortions in node-unaligned multidimensional networks that may have either uniform or non-uniform multidimensional spaces. Thus, our mathematical results explore the possible combinations of node alignment and uniformity that can generate algorithmic information distortions and establish worst-case error margins for these distortions in multidimensional network complexity analyses. In addition, it shows the importance of multidimensional network encodings into which the necessary information for determining the multidimensional space itself is also embedded.

By combining both the previous results from [1] and the new ones from Section 5 in this article, we demonstrate in our final Theorem 18 that *node-unaligned* multidimensional networks (with either *uniform* or *non-uniform* multidimensional spaces) can also display worst-case algorithmic information distortions with respect to their respective isomorphic monoplex networks. However, unlike the node-aligned non-uniform case studied in [1] (in which the worst-case distortions were shown to grow at least linearly with the number of extra node dimensions), we demonstrate that the worst-case distortions in the node-unaligned case can grow at least exponentially with the number of extra node dimensions. In addition, we demonstrate that both the node-unaligned cases and the one studied in [1] contrast with the algorithmic information distortions displayed by *node-aligned* multidimensional networks with *uniform* multidimensional spaces. In this latter case, any algorithmic information distortion can only grow up to a logarithmic order of the number of extra node dimensions. Therefore, the node-aligned uniform case is the one in which the algorithmic information content of any multidimensional network and the algorithmic information content of its isomorphic monoplex network are proved to be much less distorted as the number of node dimensions increases.

It is important to remark that the results in the present article hold independently of the choice of the encoding method or the universal programming language. This is because, given any two distinct encoding methods or any two distinct universal programming languages, the algorithmic complexity of an object represented in one way or the other can only differ by a constant whose value only depends on the choice of encoding methods or universal programming languages, but not on the choice of the object [12–15]. That is, algorithmic complexity is pairwise invariant for any two arbitrarily chosen encodings. Thus, although only dealing with pairs of isomorphic objects in addition to such an encoding invariance, some may deem the existence of the distortion phenomena shown in the present work as counter-intuitive at first glance because we will see later on in Corollaries 7 and 17 that algorithmic information distortions can in fact result from only changing the multidimensional spaces into which isomorphic copies of the objects are embedded.

The remainder article is organized as follows. In Section 2, we recall necessary concepts, definitions, and results from the literature. In Section 3, we present the previous results achieved in [1], which correspond to the node-aligned non-uniform case. In Section 4, we study basic properties of encoded node-unaligned multidimensional networks. In Section 5, we demonstrate the mathematical results. Section 6 concludes the paper.

2. Background

2.1. Multidimensional networks and multiaspect graphs

We directly base our notation regarding classical graphs on [16–18] and regarding multiaspect graphs (MAGs) on [19,20]. In order to avoid ambiguities, minor differences in the notation from [19,20] will be introduced here. In particular, the notation of MAG $H = (A, E)$ is replaced with $\mathscr{G} = (\mathscr{A}, \mathscr{E})$, where the list A of aspects is re-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

placed with $\mathscr{A}$ and the composite edge set E is replaced with $\mathscr{E}$. This way, note that $\mathscr{A} = (\mathscr{A}(\mathscr{G})[1], \ldots, \mathscr{A}(\mathscr{G})[i], \ldots, \mathscr{A}(\mathscr{G})[p])$ is a list of sets, where each set in this list is an *aspect* (or node dimension[1] [21]) denoted by $\mathscr{A}(\mathscr{G})[i]$.

The *companion tuple* of a MAG $\mathscr{G}$ becomes then denoted by $\tau(\mathscr{G})$ [20], where

$$\tau(\mathscr{G}) = (|\mathscr{A}(\mathscr{G})[1]|, \ldots, |\mathscr{A}(\mathscr{G})[p]|)$$

and p is called the *order* of the MAG. As established in [20], it is important to note that the companion tuple completely determines the size of the node-aligned set

$$\mathbb{V}(\mathscr{G}) = \bigtimes_{i=1}^{p} \mathscr{A}(\mathscr{G})[i]$$

of all *composite vertices* $\mathbf{v} = (a_1, \ldots, a_p)$ of $\mathscr{G}$, and as a direct consequence also determines the size of the set

$$\mathbb{E}(\mathscr{G}) = \mathbb{V}(\mathscr{G}) \bigtimes \mathbb{V}(\mathscr{G})$$

of all possible *composite edges* $e = ((a_1, \ldots, a_p), (b_1, \ldots, b_p))$ of $\mathscr{G}$. This way, for every node-aligned MAG $\mathscr{G}$, one has $\mathscr{E}(\mathscr{G}) \subseteq \mathbb{E}(\mathscr{G})$. One may adopt the convention of calling the elements of the first aspect of a MAG as *vertices* (i.e, the mathematical representations of the nodes of the network), so that $\mathscr{A}(\mathscr{G})[1] = V(\mathscr{G})$. As we will explore in more depth in Section 4, note that the *node alignment* property is said to hold for a multidimensional network *iff* every node belongs to (or is ascribed to) every possible permutation of node dimensions' elements. In the case of multidimensional networks represented by MAGs $\mathscr{G}$ where the convention of a vertex $v \in \mathscr{A}(\mathscr{G})[1]$ representing a node holds, a node-aligned MAG formally means that, for every particular permutation $\boldsymbol{\alpha} \in \bigtimes_{i=2}^{p} \mathscr{A}(\mathscr{G})[i]$ of elements in each aspect $\mathscr{A}(\mathscr{G})[i]$, where $i \geq 2$, and for every vertex $v \in \mathscr{A}(\mathscr{G})[1]$, one has that v is paired with $\boldsymbol{\alpha}$ (which in turn is equivalent to the above definition of $\mathbb{V}(\mathscr{G}) = \bigtimes_{i=1}^{p} \mathscr{A}(\mathscr{G})[i]$ and $\mathbb{E}(\mathscr{G}) = \mathbb{V}(\mathscr{G}) \bigtimes \mathbb{V}(\mathscr{G})$).

We denote an *undirected* MAG without *self-loops* by $\mathscr{G}_c = (\mathscr{A}, \mathscr{E})$, so that the set $\mathbb{E}_c$ of all possible undirected and non-self-loop composite edges is defined by

$$\mathbb{E}_c(\mathscr{G}_c) := \{\{\mathbf{u}, \mathbf{v}\} \mid \mathbf{u}, \mathbf{v} \in \mathbb{V}(\mathscr{G}_c)\}$$

and $\mathscr{E}(\mathscr{G}_c) \subseteq \mathbb{E}_c(\mathscr{G}_c)$ always holds. In a direct analogy to simple graphs, we refer to these MAGs $\mathscr{G}_c$ as *simple* MAGs.

Regarding graphs, we follow the common notation and nomenclature [16,18,22]: we denote a general (directed or undirected) *graph* by $G = (V, E)$, where V is the finite set of vertices and $E \subseteq V \times V$; if a graph only contains undirected edges and does not contain self-loops, then it is called a *simple* graph. A graph G is (vertex-)labeled when the members of V are distinguished from one another by labels such as $v_1, v_2, \ldots, v_{|V|}$. If a simple graph is labeled this way by natural numbers, i.e., $V = \{1, \ldots, n\}$ with $n \in \mathbb{N}$, then it is called a *classical* graph.

In a direct analogy to classical graphs, if a simple MAG $\mathscr{G}_c$ is (composite-vertex-)labeled with natural numbers (i.e., for every $i \leq p$, $\mathscr{A}(\mathscr{G})[i] = \{1, \ldots, |\mathscr{A}(\mathscr{G})[i]|\} \subset \mathbb{N}$), then we say that $\mathscr{G}_c$ is a *classical* MAG. For the present purposes of this article, all graphs G will be classical graphs and all MAGs will be simple MAGs (whether node-aligned or node-unaligned). Also note that a classical graph G is a *first-order* (i.e., $p = 1$) classical MAG $\mathscr{G}_c$ with $V(G) = \mathbb{V}(\mathscr{G}_c) = \{1, \ldots, |\mathbb{V}(\mathscr{G}_c)|\}$; and, thus, the term 'vertex' shall not be confused with term 'composite vertex', since they refer to same entity only in the case of first-order MAGs.

We refer to the discrete multidimensional space of a MAG as the discrete cartesian product $\bigtimes_{i=2}^{p} \mathscr{A}(\mathscr{G})[i]$ into which the nodes of the network are embedded. Thus, for

[1] In this article one can employ the terms "aspect" or "node dimension" interchangeably.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

classical MAGs, note that the companion tuple completely determines the discrete *multidimensional space* of the MAGs; and in this particular case the set $\bigtimes_{i=2}^{p} \mathscr{A}(\mathscr{G})[i]$ then becomes a finite discrete space of $(p-1)$-tuples of natural numbers. Later on in Section 4, we will explore other extended forms of the companion tuple in order to deal with node-unaligned MAGs. With the purpose of avoiding confusion, *node-unaligned* MAGs will be denoted by $\mathscr{G}_{ua}$ and we stick with notation $\mathscr{G}$ for *node-aligned* MAGs only.

In the particular case $\mathscr{A}(\mathscr{G})[i] = \mathscr{A}(\mathscr{G})[j]$ holds for every $i, j \leq p$, we say the multidimensional space of the MAG is *uniform*.

Also note that the number of node dimensions of a multidimensional network that is mathematically represented by a MAG is given by the value p, i.e., the order of the MAG. Thus, arbitrarily large multidimensional spaces formally refers to arbitrarily large values of p. Unlike for example in [23,24], in which multidimensional networks refers basically to multiplex networks [25–27], we adopt the convention of defining a general *multidimensional network* as a network that has at least one extra node dimension—this node dimension being for example a set of time instants or a set of layers. As a consequence, both multilayer networks [23,25,27], dynamic networks [28–31], and dynamic multilayer networks [19,32,33] become particular cases of multidimensional networks. Also note that the traditional networks which have no extra node dimension are called a monoplex (i.e., single-layer or monolayer) network [25–27] and, equivalently, can be mathematically represented by a graph. In this article, we are focusing on the multidimensional networks that can be mathematically represented by MAGs (and, therefore, each aspect of a MAG is an equivalent mathematical counterpart of a node dimension). We will employ hereafter the term *multidimensional network* to refer to these networks represented by MAGs.

As established in [19], one can define a *MAG-graph isomorphism* analogously to the classical notion of graph isomorphism: a (node-aligned) MAG $\mathscr{G}$ is isomorphic to a graph G *iff* there is a bijective function $f : \mathbb{V}(\mathscr{G}) \to V(G)$ such that

$$e \in \mathscr{E}(\mathscr{G}) \iff (f(\pi_o(e)), f(\pi_d(e))) \in E(G) \,,$$

where π_o is a function that returns the origin composite vertex of a composite edge and π_d is a function that returns the destination composite vertex of a composite edge.

In order to avoid ambiguities in the nomenclature with the classical isomorphism in graphs (which is usually a vertex label transformation) we call:

- such an isomorphism between a MAG and graph from [19] a *MAG-graph isomorphism*;
- the usual isomorphism between graphs [16,18] as *graph isomorphism*;
- and the isomorphism between two MAGs $\mathscr{G}$ and $\mathscr{G}'$ (i.e., $(\mathbf{u}, \mathbf{v}) \in \mathscr{E}(\mathscr{G})$ *iff* $(f(\mathbf{u}), f(\mathbf{v})) \in \mathscr{E}(\mathscr{G}')$) as *MAG isomorphism*.

It is shown in [19] that a (node-aligned) MAG is isomorphically equivalent to a graph:

Theorem 1. *For every MAG $\mathscr{G}$ of order $p > 0$, where all aspects are non-empty sets, there is a unique (up to a* graph isomorphism*) graph $G_{\mathscr{G}} = (V, E)$ that is* MAG-graph-isomorphic *to $\mathscr{G}$, where*

$$|V(G_{\mathscr{G}})| = \prod_{n=1}^{p} |\mathscr{A}(\mathscr{G})[n]| = |\mathbb{V}(\mathscr{G})| \,.$$

Later on in Section 4, we will introduce a variant of this MAG-graph isomorphism in order to deal with the node-unaligned case.

As one of the core results of the present article, we shall show that, although both a MAG and its isomorphic graph can be encoded and both represent the same abstract relational structure, they may diverge in terms of compressibility (or algorithmic information content) in the general case.

2.2. Algorithmic information theory (AIT)

In this section, we recover some basic notations and definitions from the literature regarding algorithmic information theory. For an introduction to these concepts and notation, see [12–15].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

First, regarding some basic notation, let $l(x)$ denote the length of a string $x \in \{0,1\}^*$. Let $(x)_2$ denote the binary representation of the number $x \in \mathbb{N}$. Let $x \upharpoonright_n$ denote the ordered sequence of the first n bits of the fractional part in the binary representation of $x \in \mathbb{R}$. That is, $x \upharpoonright_n = x_1 x_2 \dots x_n$, where $(x)_2 = y.x_1 x_2 \dots x_n x_{n+1} \dots$ with $y \in \{0,1\}^*$ and $x_1, x_2, \dots, x_n \in \{0,1\}$. We denote the result of the computation of an arbitrary Turing machine $\mathcal{M}$ with input $x \in L$ by the partial computable function $\mathcal{M} \colon L \to L$. In the particular case $\mathcal{M}$ is a universal Turing machine we denote it by $\mathbf{U}$. Let $\mathbf{L'_U}$ denote a binary *prefix-free* (or *self-delimiting*) universal programming language for a prefix universal Turing machine $\mathbf{U}$.

As usual, let $\langle \cdot , \cdot \rangle$ denote an arbitrary computable bijective pairing function [14,15], which can be recursively extended in order to encode any finite ordered n-tuple in the form $\langle \cdot , \dots , \cdot \rangle$.

Let w^* denote the lexicographically first $\mathrm{p} \in \mathbf{L'_U}$ such that $l(\mathrm{p})$ is minimum and $\mathbf{U}(\mathrm{p}) = w$. This way, the algorithmic information content of an object w is given by the (unconditional) *prefix algorithmic complexity* (also known as K-complexity, Kolmogorov prefix complexity, self-delimited program-size complexity, or Solomonoff-Kolmogorov-Chaitin complexity for prefix universal Turing machines), denoted by $\mathbf{K}(w)$, which is the length of the shortest program $w^* \in \mathbf{L'_U}$ such that $\mathbf{U}(w^*) = w$. The *conditional* prefix algorithmic complexity of a binary string y given a binary string x, denoted by $K(y \,|x)$, is the length of the shortest program $w \in \mathbf{L'_U}$ such that $\mathbf{U}(\langle x, w \rangle) = y$.

A real number (or infinite binary sequence) $x \in [0,1] \subset \mathbb{R}$ is said to be 1-*random* (*Martin-Löf random*, *K-random*, or *prefix algorithmically random*) [14] if, and only if, it satisfies

$$\mathbf{K}(x \upharpoonright_n) \geq n - \mathbf{O}(1) \, ,$$

where $n \in \mathbb{N}$ is arbitrary.

With respect to weak asymptotic dominance of functions f and g, we employ the usual notations: $f(x) = \mathbf{O}(g(x))$ for the big $\mathbf{O}$ notation when f is asymptotically upper bounded by g; $f(x) = \mathbf{o}(g(x))$ when g strongly dominates f asymptotically; and $f(x) = \mathbf{\Omega}(g(x))$ for the big **Omega** notation when f is asymptotically lower bounded by g;

As a consequence of the Borel normality of 1-random real numbers [13,34], we have that, for every 1-random $x \in [0,1] \subset \mathbb{R}$, there is a large enough $n_0 \in \mathbb{N}$ such that, for every $n \geq n_0$,

$$\#_1(x \upharpoonright_n) = \frac{n}{2} \pm \mathbf{o}(n)$$

holds, where $\#_1(x \upharpoonright_n)$ denotes the number of occurrences of 1's in $x \upharpoonright_n$.

3. The node-aligned non-uniform case

3.1. Basic properties of encoded node-aligned multiaspect graphs

In a general sense, a (node-aligned) MAG $\mathscr{G}_c$ is said to be *encodable* (i.e., recursively labeled, or with a univocal computably ordered data representation) given $\tau(\mathscr{G}_c)$ *iff* there is an algorithm that, given the companion tuple $\tau(\mathscr{G}_c)$ as input, computes a bijective ordering of composite edges $e \in \mathbb{E}_c(\mathscr{G}_c)$ from any pair of composite vertices $\mathbf{v}, \mathbf{u} \in \mathbb{V}(\mathscr{G}_c)$. That is, if the companion tuple $\tau(\mathscr{G}_c)$ of the MAG $\mathscr{G}_c$ is known, then one can computably retrieve the position of any composite edge $e = \{\mathbf{u}, \mathbf{v}\}$ in the chosen data representation of $\mathscr{G}_c$ from both composites vertices $\mathbf{u}$ and $\mathbf{v}$, and vice-versa. This way, following the usual definition of encodings, a MAG is encodable given $\tau(\mathscr{G}_c)$ *iff* there is a algorithm that, given $\tau(\mathscr{G}_c)$ as input, can univocally encode any possible $\mathscr{E}(\mathscr{G}_c)$ that shares the same companion tuple.

As expected, MAGs that have every element of its aspects labeled as a natural number can always be encoded. The proof of Lemma 2 follows directly from the definition of MAG and the recursive bijective nature of the pairing function.[2] In other words, a classical MAG can always be encoded if the information necessary to determine the companion tuple $\tau(\mathscr{G}_c)$ is previously given.

[2] The reader can find a proof of Lemma 2 in [35].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

Lemma 2. *Any arbitrary classical (node-aligned) MAG $\mathscr{G}_c$ is encodable given $\tau(\mathscr{G}_c)$.*

The encodability given $\tau(\mathscr{G}_c)$ also implies the existence of an algorithm that, given a string $x \in \{0,1\}^*$ of length $|\mathbb{E}_c(\mathscr{G}_c)|$ as input, computes a composite edge set $\mathscr{E}(\mathscr{G}_c)$ and there is another algorithm that, given the encoded composite edge set $\mathscr{E}(\mathscr{G}_c)$ as input, returns a string x. Such strings uniquely represent (up to a MAG isomorphism and/or up to a reordering of composite edges) the characteristic function (or indicator function) of pertinence in the set $\mathscr{E}(\mathscr{G}_c) \subseteq \mathbb{E}_c(\mathscr{G}_c)$, and thus we call them as *characteristic strings* of the MAG:

Definition 1. *Let $\left(e_1, \dots, e_{|\mathbb{E}_c(\mathscr{G}_c)|}\right)$ be any arbitrary ordering of all possible composite edges of a simple MAG $\mathscr{G}_c$. We say that a string $x \in \{0,1\}^*$ with $l(x) = |\mathbb{E}_c(\mathscr{G}_c)|$ is a* (node-aligned) characteristic string *of a simple MAG $\mathscr{G}_c$ iff, for every $e_j \in \mathbb{E}_c(\mathscr{G}_c)$,*

$$e_j \in \mathscr{E}(\mathscr{G}_c) \iff \text{the } j\text{-th digit in } x \text{ is } 1 \,,$$

where $1 \le j \le l(x)$.

In order to ensure uniqueness of representations (now only up to an identity automorphism, if the sequence $\left(e_1, \dots, e_{|\mathbb{E}_c(\mathscr{G}_c)|}\right)$ is previously fixed) from which the algorithmic complexity are calculated, one may also choose to encode a MAG into a string-based representation using the pairing function $\langle \cdot, \cdot \rangle$ and a fixed ordering/indexing of the composite edges:

Definition 2. *Let $\left(e_1, \dots, e_{|\mathbb{E}_c(\mathscr{G}_c)|}\right)$ be any arbitrary ordering of all possible composite edges of a simple MAG $\mathscr{G}_c$. Then, $\langle \mathscr{E}(\mathscr{G}_c) \rangle$ denotes the* (node-aligned) composite edge set string *$\langle \langle e_1, z_1 \rangle, \dots, \langle e_n, z_n \rangle \rangle$ such that*

$$z_i = 1 \iff e_i \in \mathscr{E}(\mathscr{G}_c) \,,$$

where $z_i \in \{0,1\}$ with $1 \le i \le n = |\mathbb{E}_c(\mathscr{G}_c)|$.

Note that a composite edge set string is an encoding of a composite edge list, which in turn is a generalization of edge lists [36] so as to deal with MAGs instead of graphs. Thus, the reader may also interchangeably call the composite edge set string by *composite edge list encoding*.

Also note that composite edge set strings are strictly tied to the way one chooses to encode MAGs given the companion tuple. In other words, once the chosen encoding inherently establishes an ordering $\left(e_1, \dots, e_j, \dots, e_{|\mathbb{E}_c(\mathscr{G}_c)|}\right)$ of composites edges, the only entities that vary in the composite edge set strings are the values of $z_1, z_2, \dots,$ or $z_{|\mathbb{E}_c(\mathscr{G}_c)|}$. For example, as shown in [35], the composite edge sets strings can be constructed from encoding a MAG $\mathscr{G}_c$ (given the companion tuple $\tau(\mathscr{G}_c)$) using the fixed programs $\mathrm{p}_1, \mathrm{p}_2 \in \{0,1\}^*$ and the chosen computable bijective pairing function $\langle \cdot \,, \cdot \rangle$ in such a way that:

(I) if $(a_1, \dots, a_p), (b_1, \dots, b_p) \in \mathbb{V}(\mathscr{G}_c)$, then

$$\mathbf{U}(\langle \langle a_1, \dots, a_p \rangle, \langle b_1, \dots, b_p \rangle, \langle \langle \tau(\mathscr{G}_c) \rangle, \mathrm{p}_1 \rangle \rangle) = (j)_2 \,;$$

(II) if $(a_1, \dots, a_p)$ or $(b_1, \dots, b_p)$ does not belong to $\mathbb{V}(\mathscr{G}_c)$, then

$$\mathbf{U}(\langle \langle a_1, \dots, a_p \rangle, \langle b_1, \dots, b_p \rangle, \langle \langle \tau(\mathscr{G}_c) \rangle, \mathrm{p}_1 \rangle \rangle) = 0 \,;$$

(III) if

$$1 \le j \le |\mathbb{E}_c(\mathscr{G}_c)| = \frac{|\mathbb{V}(\mathscr{G}_c)|^2 - |\mathbb{V}(\mathscr{G}_c)|}{2} \,,$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

then

$$\mathbf{U}(\langle j, \langle \langle \tau(\mathscr{G}_c) \rangle, \mathrm{p}_2 \rangle \rangle) = \langle \langle a_1, \ldots, a_p \rangle, \langle b_1, \ldots, b_p \rangle \rangle ;$$

(IV) if

$$1 \leq j \leq |\mathbb{E}_c(\mathscr{G}_c)| = \frac{|\mathbb{V}(\mathscr{G}_c)|^2 - |\mathbb{V}(\mathscr{G}_c)|}{2}$$

does not hold, then

$$\mathbf{U}(\langle j, \langle \langle \tau(\mathscr{G}_c) \rangle, \mathrm{p}_2 \rangle \rangle) = \langle 0 \rangle .$$

In the case of graphs (or monoplex networks), we remember that, as for example shown in [35], there is always a unified and decidable way to encode a sequence of all possible undirected edges given any unordered pair $\{x, y\}$ of natural numbers $x, y \in \mathbb{N}$, for example by encoding characteristic strings or adjacency matrices of arbitrary finite size. Thus, encoding classical graphs with characteristic strings or with composite edge set strings becomes Turing equivalent and, therefore, both are also equivalent in terms of algorithmic information (see Lemma 4). This is indeed an underlying basic property previously explored, e.g., in [3,6,9]. However, unlike classical graphs, we shall see later on in Corollaries 6 and 16 that the relationship between characteristic strings and composite edge set strings in the case of simple MAGs (whether node-aligned or node-unaligned) does not always behave so well. Additionally, it was shown in [37] that graph isomorphisms do not always preserve algorithmic randomness of *infinite* graphs. Unlike the infinite case, note that the present article only deals with finite MAGs and finite graphs and with infinite families of finite MAGs and finite graphs. Nevertheless, we shall prove in Corollaries 7 and 17 that MAG-graph-isomorphisms between (finite) simple MAGs and (finite) classical graphs do not always preserve algorithmic information.

Before starting the investigation of examples that display worst-case algorithmic information distortion generated by MAG-graph-isomorphisms, we can promptly establish in Lemmas 3 and 4 the upper bound for the algorithmic information distortion between any simple MAG and its MAG-graph-isomorphic classical graph. If the ordering assumed in Definition 1 matches the same ordering in Definition 2, we have in Lemma 3 below that both the MAG and its respective characteristic string are indeed "equivalent" in terms of algorithmic information, but except for the minimum information necessary to encode the multidimensional space (e.g., the algorithmic information of the encoded companion tuple in the form $\langle \tau(\mathscr{G}_c) \rangle = \langle |\mathscr{A}(\mathscr{G})[1]|, \ldots, |\mathscr{A}(\mathscr{G})[p]| \rangle$). As expected, the proof follows easily from the fact that an ordering of composite edges is always embedded into the notion of encodability by composite edge set strings. (Complete proofs of Lemmas 3 and 4 can be found in [35, Lemma 3.2]).

Lemma 3. *Let $x \in \{0,1\}^*$. Let $\mathscr{G}_c$ be an encodable (node-aligned) simple MAG given $\tau(\mathscr{G}_c)$ such that x is the respective characteristic string. Then,*

$$\mathbf{K}(\langle \mathscr{E}(\mathscr{G}_c) \rangle \mid x) \leq \mathbf{K}(\langle \tau(\mathscr{G}_c) \rangle) + \mathbf{O}(1) \tag{1}$$

$$\mathbf{K}(x \mid \langle \mathscr{E}(\mathscr{G}_c) \rangle) \leq \mathbf{K}(\langle \tau(\mathscr{G}_c) \rangle) + \mathbf{O}(1) \tag{2}$$

$$\mathbf{K}(x) = \mathbf{K}(\langle \mathscr{E}(\mathscr{G}_c) \rangle) \pm \mathbf{O}\Big(\mathbf{K}(\langle \tau(\mathscr{G}_c) \rangle)\Big). \tag{3}$$

Note that there is always a way to build a computable sequence $\left(e_1, \ldots, e_{|\mathbb{E}_c(G)|}\right)$ for any classical graph. Thus, since a graph is a MAG of order 1 (and in this case characteristic strings and composite edge set strings become Turing equivalent), Lemma 3 can be improved in the case of classical graphs so that:

 doi:10.20944/preprints202103.0056.v2

Lemma 4. *Let $x \in \{0,1\}^*$. Let G be a classical graph, where x is its characteristic string, $\mathbb{E}_c(G) = \{\{u,v\} \mid u,v \in V\}$, and the sequence $\left(e_1, \dots, e_{|\mathbb{E}_c(G)|}\right)$ is computable. Then,*

$$\mathbf{K}(\langle E(G)\rangle \mid x) \leq \mathbf{O}(1) \tag{4}$$

$$\mathbf{K}(x \mid \langle E(G)\rangle) \leq \mathbf{O}(1) \tag{5}$$

$$\mathbf{K}(x) = \mathbf{K}(\langle E(G)\rangle) \pm \mathbf{O}(1). \tag{6}$$

3.2. A worst-case algorithmic information distortion in node-aligned multidimensional networks

Basically, Lemma 3 assures that the information contained in a simple MAG $\mathscr{G}_c$ and in the characteristic string are the same, except for the algorithmic information necessary to computably determine the companion tuple. Unfortunately, one can show in Theorem 5 below that this information deficiency between the data representation of a MAG (in the form e.g. $\langle \mathscr{E}(\mathscr{G}_c)\rangle$) and its characteristic string cannot be much more improved in general. In other words, as we show in Theorem 5, there are *worst-case* scenarios of multidimensional spaces in which the algorithmic information necessary for retrieving the encoded form of the MAG from its characteristic string is close (except for a logarithmic term) to the upper bound given by Equation 1 in Lemma 3. This shows a fundamental difference between encoding MAGs with characteristic strings and encoding MAGs with composite edge set strings. The proof of Theorem 5 can be found in [1].

Theorem 5. *There are arbitrarily large*[3] *encodable simple (node-aligned) MAGs $\mathscr{G}_c$ given $\tau(\mathscr{G}_c)$ such that*

$$\mathbf{K}(\langle \tau(\mathscr{G}_c)\rangle) + \mathbf{O}(1) \geq \mathbf{K}(\langle \mathscr{E}(\mathscr{G}_c)\rangle \mid x) \geq \mathbf{K}(\langle \tau(\mathscr{G}_c)\rangle) - \mathbf{O}\Big(\log_2\left(\mathbf{K}(\langle \tau(\mathscr{G}_c)\rangle)\right)\Big)$$

with $\mathbf{K}(\langle \mathscr{E}(\mathscr{G}_c)\rangle) \geq p - \mathbf{O}(1)$ and $\mathbf{K}(x) = \mathbf{O}(\log_2(p))$, where x is the respective characteristic string and p is the order of the MAG $\mathscr{G}_c$.

As a consequence of Theorem 5, we show in Corollary 6 below a phenomenon that can only occur for families of node-aligned multidimensional networks embedded into *arbitrarily large* and *non-uniform* multidimensional spaces.

Specifically, Corollary 6 (and also Corollary 16 later on) shows that there are two infinite sets of objects (in particular, one of data representations of multiaspect graphs and the other of strings) whose every member of one set is a particular encoding of a member of the other, but these members of the two sets are not always equivalent in terms of algorithmic information, which is a phenomenon that some may deem to be counter-intuitive at first glance:

Corollary 6. *There are an infinite family F of simple (node-aligned) MAGs and an infinite set X of the correspondent characteristic strings such that, for every constant $c \in \mathbb{N}$, there are $\mathscr{G}_c \in F$ and $x \in X$, where x is the characteristic string of $\mathscr{G}_c$ and*

$$\mathbf{O}\Big(\log_2\left(\mathbf{K}(\langle \mathscr{E}(\mathscr{G}_c)\rangle)\right)\Big) > c + \mathbf{K}(x)\,. \tag{7}$$

The proof of Corollary 6 can be found in [1].

We can now combine Corollary 6 with Theorems 1 and 5 in order to show that, although for every MAG there is a graph that is isomorphic to this MAG, they are not always equivalent in terms of algorithmic information, where in fact the distortion may be exponential with respect to the algorithmic information of the graph:

Corollary 7. *There are an infinite family F_1 of simple (node-aligned) MAGs and an infinite family F_2 of classical graphs, where every classical graph in F_2 is MAG-graph-isomorphic to at least*

[3] And, in particular, those with *non-uniform* multidimensional spaces. As we will see later on in Lemma 14, this exponential distortion presented in Theorem 5 can only occur if the multidimensional space is not uniform.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

one MAG in F_1, such that, for every constant $c \in \mathbb{N}$, there are $\mathscr{G}_c \in F_1$ and a $G_{\mathscr{G}_c} \in F_2$ that is MAG-graph-isomorphic to $\mathscr{G}_c$, where

$$\mathbf{O}\Big(\log_2 \left(\mathbf{K}(\langle \mathscr{E}(\mathscr{G}_c) \rangle) \right) \Big) > c + \mathbf{K}(\langle E(G_{\mathscr{G}_c}) \rangle).$$

The proof of Corollary 7 can be found in [1]. Besides showing that node-aligned multidimensional networks with non-uniform multidimensional spaces can display an exponentially larger algorithmic information distortions with respect to its isomorphic monoplex network, Theorem 5 together with Corollary 7 also show that these distorted values of algorithmic information content grow at least linearly with the value of p (i.e., number of extra node dimensions). With this lower bound for the worst-case distortion increasing rate as we established here, future research is needed for investigating the upper bound for distorted values of algorithmic information content with respect to the number of extra node dimensions.

4. The node-unaligned cases

With the purpose of addressing other variations of multidimensional networks in which a node not belonging to a certain $\boldsymbol{\alpha} \in \bigtimes_{i=2}^{p} \mathscr{A}(\mathscr{G})[i]$ has an important physical meaning, the node alignment can be relaxed. This gives rise to multidimensional networks that are not node aligned, such as node-unaligned multilayer networks [25] or node-unaligned multiplex networks [26].

As a multiplex network is usually understood as a particular case of a multilayer network [23,25,27] in which there is only one extra node dimension (i.e., $d = 1$), we may focus only on a mathematical formulation of multilayer networks that allows nodes to be not aligned, which is given by $M = (V_M, E_M, V, \mathbf{L})$ [25], where:

1. V denotes the set of all possible vertices v;
2. $\mathbf{L} = \{L_a\}_{a=1}^{d}$ denotes a collection of $d \in \mathbb{N}$ sets L_a composed of *elementary layers* $\alpha \in L_a$;
3. $V_M \subseteq V \times L_1 \times \cdots \times L_d$ denotes the subset of all possible vertices paired to elements of $L_1 \times \cdots \times L_d$;
4. $E_M \subseteq V_M \times V_M$ denotes the set of *interlayer* and/or *intralayer* edges connecting two *node-layer tuples* $(v, \alpha_1, \ldots, \alpha_d) \in V_M$.

In this regard, a multilayer network M is said to be node-aligned *iff* $V_M = V \times L_1 \times \cdots \times L_d$. In the case each $\boldsymbol{\alpha} \in \bigtimes_{i=2}^{p} \mathscr{A}(\mathscr{G})[i]$ can be interpreted as (or is representing) a layer, it is important to note that there are some immediate equivalences between $\mathscr{G}$ and M [21]:

- V is the usual set of vertices, where $V(\mathscr{G}) \equiv \mathscr{A}(\mathscr{G})[1]$;
- each set L_a is the $(a-1)$-th aspect $\mathscr{A}(\mathscr{G})[a-1]$ of a MAG $\mathscr{G}$;
- V_M is a subset of the set $\mathbb{V}(\mathscr{G})$ of all composite vertices, where every node-layer tuple $(v, \alpha_1, \ldots, \alpha_d) \in V_M$ is a composite vertex $\mathbf{v} \in \mathbb{V}(\mathscr{G})$;
- $E_M \subseteq \mathscr{E}(\mathscr{G})$ is a subset of the set of all composite edges $(\mathbf{u}, \mathbf{v})$ for which $\mathbf{u}, \mathbf{v} \in V_M$.

Thus, if $V_M = V \times L_1 \times \cdots \times L_d$, then one will have that $V_M \equiv \mathbb{V}(\mathscr{G})$ and $E_M \equiv \mathscr{E}(\mathscr{G})$. This way, besides notation distinctions, it directly follows that a node-aligned multilayer network M is totally equivalent to a MAG $\mathscr{G}$; and, therefore, every result in this paper holding for simple node-aligned MAGs $\mathscr{G}_c$ automatically holds for node-aligned multilayer networks M that only have undirected edges and do not contain self-loops.

In addition to the physical interpretation of nodes and layers that could be intrinsic to a node nonalignment of real-world networks and new methodological challenges in multi-dimensional network analysis that arise from the fact that the multidimensional network is not node unaligned, it becomes important to investigate the network complexity increase (in terms of compressibility or irreducible information content) when the multidimensional network is *node-unaligned* (a problem also raised in [21]).

With this purpose of extending our results to the node-unaligned case, we need to introduce a variation of MAGs so as to allow into the mathematical formalization the possibility of some vertices not being paired with some $\boldsymbol{\alpha}$'s, where $\boldsymbol{\alpha} \in \bigtimes_{i=2}^{p} \mathscr{A}(\mathscr{G})[i]$. Moreover, we need that node-aligned MAGs become particular cases of our new formalization such that the algorithmic information between the two formalizations becomes equivalent when the MAG is node-aligned, which we will show in Lemma 11. To this end, we introduce a modification on the definition of MAG so that the major differences are in the set of composite vertices and, consequentially, in the set of composite edges.

Definition 3. *We define a* node-unaligned MAG *as a triple* $\mathscr{G}_{ua} = (\mathscr{A}, \mathbb{V}_{ua}, \mathscr{E}_{ua})$ *in which*

$$\mathscr{A} = (\mathscr{A}(\mathscr{G}_{ua})[1], \dots, \mathscr{A}(\mathscr{G}_{ua})[i], \dots, \mathscr{A}(\mathscr{G}_{ua})[p])$$

is a list of sets (each of each is an aspect of $\mathscr{G}_{ua}$*),*

$$\mathbb{V}_{ua}(\mathscr{G}_{ua}) \subseteq \mathbb{V}(\mathscr{G}_{ua}) = \bigtimes_{i=1}^{p} \mathscr{A}(\mathscr{G}_{ua})[i]$$

is the set of existing composite vertices, and

$$\mathscr{E}_{ua} \subseteq \mathbb{E}_{ua}(\mathscr{G}_{ua}) = \mathbb{V}_{ua}(\mathscr{G}_{ua}) \times \mathbb{V}_{ua}(\mathscr{G}_{ua})$$

is the set of present composite edges $(\mathbf{u}, \mathbf{v})$*.*

The definition of simple node-unaligned MAGs $\mathscr{G}_{uac} = (\mathscr{A}, \mathbb{V}_{ua}, \mathscr{E}_{ua})$ follows analogously to the aligned case by just restricting the set of all possible composite edges, so that

$$\mathbb{E}_{uac}(\mathscr{G}_{uac}) := \{\{\mathbf{u}, \mathbf{v}\} \mid \mathbf{u}, \mathbf{v} \in \mathbb{V}_{ua}(\mathscr{G}_{uac})\}$$

and $\mathscr{E}_{ua}(\mathscr{G}_{uac}) \subseteq \mathbb{E}_{uac}(\mathscr{G}_{uac})$ hold. In addition, all other terminology of *order*, *multidimensional space*, *uniformity*, *(composite-vertex-)labeling*, and *classical MAGs* apply analogously as in Section 2.1.

Also note that the connections in a node-unaligned MAG are composed of two composites vertices that belong to a subset of the $\mathbb{V}(\mathscr{G}_{ua})$. This immediately implies that, if $\mathscr{G}_{ua} = (\mathscr{A}, \mathbb{V}_{ua}, \mathscr{E}_{ua})$ and $\mathscr{G} = (\mathscr{A}, \mathscr{E})$, where $\mathscr{A}$ is fixed and $\mathscr{E}_{ua} \subseteq \mathscr{E}$, then there is a MAG isomorphism between $\mathscr{G}_{ua}$ and a subMAG [35] of $\mathscr{G}$.

As in the node-aligned case, in order to define an unaligned version for the companion tuple, the latter should completely determine the size of the set $\mathbb{E}_{ua}(\mathscr{G}_{ua})$ and, if $\mathscr{G}_{ua}$ is a classical MAG, then the companion tuple should completely determine the multidimenional space of $\mathscr{G}_{ua}$. In this sense, a node-unaligned version of the companion tuple also needs to carry the necessary and sufficient information for computably retrieving the set $\mathbb{V}_{ua}(\mathscr{G}_{ua})$ from $\mathscr{A}(\mathscr{G}_{ua})$.

Definition 4. *The* node-unaligned companion tuple τ_{ua} *of a MAG* $\mathscr{G}_{ua}$ *is defined by the pair of tuples*

$$\tau_{ua}(\mathscr{G}_{ua}) := \left((|\mathscr{A}(\mathscr{G}_{ua})[1]|, \dots, |\mathscr{A}(\mathscr{G}_{ua})[p]|), \left(m_1, \dots, m_{|\mathbb{V}(\mathscr{G}_{ua})|} \right) \right) ,$$

such that, for every $\mathbf{v}_i \in \mathbb{V}(\mathscr{G}_{ua})$ *in a previously chosen arbitrary computable enumeration of* $\mathbb{V}(\mathscr{G}_{ua})$*,*

$$\mathbf{v}_i \in \mathbb{V}_{ua}(\mathscr{G}_{ua}) \iff m_i = 1 \, ,$$

where $m_i \in \{0, 1\}$ *and* $1 \le i \le |\mathbb{V}(\mathscr{G}_{ua})|$*.*

As for encoding τ_{ua}, one can also employ the recursive pairing function $\langle \cdot, \cdot \rangle$ as usual:

$$\langle \tau_{ua}(\mathscr{G}_{ua}) \rangle := \left\langle \langle |\mathscr{A}(\mathscr{G}_{ua})[1]|, \dots, |\mathscr{A}(\mathscr{G}_{ua})[p]| \rangle, \left\langle m_1, \dots, m_{|\mathbb{V}(\mathscr{G}_{ua})|} \right\rangle \right\rangle$$

 doi:10.20944/preprints202103.0056.v2

Note that there are other alternative forms of computably defining companion tuples, for example as a two-dimensional array in which the first dimension stores the indexes of the composite vertices and the second dimension stores the entries[4] '*belongs to* $\mathbb{V}_{ua}$' or '*does not belong to* $\mathbb{V}_{ua}$'. As our purpose is to achieve analogous results to those obtained in Section 3.2, we employ hereafter the encoding in the form $\langle \tau_{ua}(\mathscr{G}_{ua}) \rangle$, but nevertheless the following results hold for any other Turing equivalent way of encoding the companion tuple as in the form $\langle \tau_{ua}(\mathscr{G}_{ua}) \rangle$ from Definition 4.

It is also interesting to note that the bit string

$$m_1 \cdots m_{|\mathbb{V}(\mathscr{G}_{ua})|}$$

is a form of characteristic string, but for the set $\mathbb{V}_{ua}$ instead of a characteristic string for the entire MAG as in Definition 1.

The MAG-graph isomorphism also suffers a slight modification:

Definition 5. *$\mathscr{G}_{ua}$ is* unaligning MAG-graph-isomorphic *to a graph G iff there is a bijective function $f : \mathbb{V}_{ua}(\mathscr{G}_{ua}) \to V(G)$ such that*

$$e \in \mathscr{E}_{ua}(\mathscr{G}_{ua}) \subseteq \mathbb{E}_{ua}(\mathscr{G}_{ua}) \iff (f(\pi_o(e)), f(\pi_d(e))) \in E(G) \, ,$$

where π_o is a function that returns the origin composite vertex of a composite edge and π_d is a function that returns the destination composite vertex of a composite edge.

This way, we can obtain the following theorem analogously[5] to the proof of Theorem 1:

Theorem 8. *For every MAG $\mathscr{G}_{ua}$ of order $p > 0$, where all aspects are non-empty sets, there is a unique (up to a* graph isomorphism*) graph $G^{ua}_{\mathscr{G}_{ua}} = (V, E)$ that is* unaligning MAG-graph-isomorphic *to $\mathscr{G}_{ua}$, where*

$$|V(G^{ua}_{\mathscr{G}_{ua}})| = |\mathbb{V}_{ua}(\mathscr{G}_{ua})| \, .$$

Short proof. Both the proof of existence and uniqueness follow analogously to the one of [19, Theorem 1, p. 54]. Note that [19, Theorem 1, p. 54] is re-written as Theorem 1 in Section 2.1 with minor notation changes. For the proof of existence it suffices to: construct the set $V(G^{ua}_{\mathscr{G}_{ua}})$ of vertices from $|\mathbb{V}_{ua}(\mathscr{G}_{ua})|$ arbitrarily labeled vertices, instead of $|\mathbb{V}(\mathscr{G}_{ua})|$ arbitrarily labeled vertices; and replace the bijective function $f : \mathbb{V}(\mathscr{G}_{ua}) \to V(G^{ua}_{\mathscr{G}_{ua}})$ with $f' : \mathbb{V}_{ua}(\mathscr{G}_{ua}) \to V(G^{ua}_{\mathscr{G}_{ua}})$. For the proof of uniqueness, besides using function f' instead of f, just replace the bijective function $j : \mathbb{V}(\mathscr{G}_{ua}) \to V(J_{\mathscr{G}_{ua}})$ with $j' : \mathbb{V}_{ua}(\mathscr{G}_{ua}) \to V(J_{\mathscr{G}_{ua}})$. All of the other arguments in [19, Theorem 1, p. 54] then directly applies, except for the necessary respective notation changes. □

It is important to note our choice of notation distinction between the node-aligned and the node-unaligned case when a graph is MAG-graph-isomorphic to a MAG. The graph $G_{\mathscr{G}_{ua}}$ is said to be *aligning MAG-graph-isomorphic* to the MAG when the set of possible composite vertices is complete, that is, when it is taken from $\mathbb{V}(\mathscr{G}_{ua})$. On the other hand, the graph $G^{ua}_{\mathscr{G}_{ua}}$ is said to be *unaligning MAG-graph-isomorphic* to the MAG (which is the case of Definition 5, Theorem 8, Corollary 17, and Theorem 18.(I)(b)) when the set of possible composite vertices are taken from $\mathbb{V}_{ua}(\mathscr{G}_{ua})$ instead of $\mathbb{V}(\mathscr{G}_{ua})$.

4.1. Encoding node-unaligned multiaspect graphs

Encodability of node-unaligned MAGs given the companion tuple works in the same way as the node-aligned case in Section 3.1. That is, a node-unaligned simple MAG $\mathscr{G}_{uac}$ is encodable given $\tau_{ua}(\mathscr{G}_{uac})$ *iff* there is an algorithm that, given $\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle$ as input, can

[4] Or a Boolean variable instead of these alphanumerical strings.
[5] And thus we choose to present only a short proof indicating where the modifications in the proof of [19, Theorem 1, p. 54] should take place.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

univocally encode any possible $\mathscr{E}_{ua}(\mathscr{G}_{uac})$ that shares the same companion tuple. However, there is some important nuances in the composite edge set string that need to be taken into account in order to ensure that $\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle$ is in any event retrievable from $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$.

First, as in the node-aligned case, note that the encodability of *classical* node-unaligned MAGs given the companion tuple can be promptly proved to hold:

Lemma 9. *Any arbitrary node-unaligned classical MAG $\mathscr{G}_{uac}$ is encodable given $\tau_{ua}(\mathscr{G}_{uac})$.*

Proof. For classical node-unaligned MAGs, we have that, for every $i \leq p$,

$$\mathscr{A}(\mathscr{G}_{uac})[i] = \{1, \dots, |\mathscr{A}(\mathscr{G}_{uac})[i]|\} \subset \mathbb{N}$$

and

$$\mathbb{E}_{uac}(\mathscr{G}_{uac}) = \{\{\mathbf{u}, \mathbf{v}\} \mid \mathbf{u}, \mathbf{v} \in \mathbb{V}_{ua}(\mathscr{G}_{uac})\}$$

hold by definition. Note that the recursive bijective pairing function $\langle \cdot, \cdot \rangle$ can be arbitrarily chosen and, therefore, the encoding of the node-unaligned companion tuple in the form

$$\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle := \left\langle \langle |\mathscr{A}(\mathscr{G}_{uac})[1]|, \dots, |\mathscr{A}(\mathscr{G}_{uac})[p]| \rangle, \left\langle m_1, \dots, m_{|\mathbb{V}(\mathscr{G}_{uac})|} \right\rangle \right\rangle$$

univocally determines the value of p, the maximum integer value for each aspect, and the set $\mathbb{V}_{ua}(\mathscr{G}_{ua})$. Thus, it suffices to demonstrate the existence of programs $\mathrm{p'}_1, \mathrm{p'}_2 \in \{0,1\}^*$ such that:

(I) if $(a_1, \dots, a_p), (b_1, \dots, b_p) \in \mathbb{V}_{ua}(\mathscr{G}_{uac})$, then

$$\mathbf{U}(\langle \langle a_1, \dots, a_p \rangle, \langle b_1, \dots, b_p \rangle, \langle \langle \tau_{ua}(\mathscr{G}_{uac}) \rangle, \mathrm{p'}_1 \rangle \rangle) = (j)_2 \, ;$$

(II) if $(a_1, \dots, a_p)$ or $(b_1, \dots, b_p)$ does not belong to $\mathbb{V}_{ua}(\mathscr{G}_{uac})$, then

$$\mathbf{U}(\langle \langle a_1, \dots, a_p \rangle, \langle b_1, \dots, b_p \rangle, \langle \langle \tau_{ua}(\mathscr{G}_{uac}) \rangle, \mathrm{p'}_1 \rangle \rangle) = 0 \, ;$$

(III) if

$$1 \leq j \leq |\mathbb{E}_{uac}(\mathscr{G}_{uac})| = \frac{|\mathbb{V}_{ua}(\mathscr{G}_{uac})|^2 - |\mathbb{V}_{ua}(\mathscr{G}_{uac})|}{2} \, ,$$

then

$$\mathbf{U}(\langle j, \langle \langle \tau_{ua}(\mathscr{G}_{uac}) \rangle, \mathrm{p'}_2 \rangle \rangle) = \langle \langle a_1, \dots, a_p \rangle, \langle b_1, \dots, b_p \rangle \rangle \, ;$$

(IV) if

$$1 \leq j \leq |\mathbb{E}_{uac}(\mathscr{G}_{uac})| = \frac{|\mathbb{V}_{ua}(\mathscr{G}_{uac})|^2 - |\mathbb{V}_{ua}(\mathscr{G}_{uac})|}{2}$$

does not hold, then

$$\mathbf{U}(\langle j, \langle \langle \tau_{ua}(\mathscr{G}_{uac}) \rangle, \mathrm{p'}_2 \rangle \rangle) = \langle 0 \rangle \, .$$

To construct a program that satisfies (I) and (II), let $\mathrm{p'}_1$ be a fixed string that represents on a universal Turing machine the algorithm that, given $\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle$, $\langle a_1, \dots, a_p \rangle$, and $\langle b_1, \dots, b_p \rangle$ as inputs:

(i) builds a sequence from a lexicographical order for the set $\mathbb{V}(\mathscr{G}_{uac}) = \bigtimes_{i=1}^{p} \mathscr{A}(\mathscr{G}_{uac})[i]$, which is always possible because the values $|\mathscr{A}(\mathscr{G}_{uac})[1]|, \dots, |\mathscr{A}(\mathscr{G}_{uac})[p]|$ are given and the labels are natural numbers;

(ii) eliminates (while preserving the previous lexicographical order) in this sequence built on Step (i) those composite vertices $\mathbf{v}_i \in \mathbb{V}(\mathscr{G}_{uac})$ for which $m_i = 0$, according to the given sequence $\left(m_1, \dots, m_{|\mathbb{V}(\mathscr{G}_{uac})|} \right)$ in $\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle$;

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

(iii) upon this lexicographical order for the set $\mathbb{V}_{ua}(\mathscr{G}_{uac})$ resulting from Step (ii), builds a sequence of composite edges $(\mathbf{v}, \mathbf{u}) \in \mathbb{E}_{ua}(\mathscr{G}_{uac})$, where $\mathbf{v}, \mathbf{u} \in \mathbb{V}_{ua}(\mathscr{G}_{uac})$, also following a lexicographical order;

(iv) eliminates in this sequence of composite edges resulting from Step (iii) those composite edges $\mathbf{e}_i \in \mathbb{E}_{ua}(\mathscr{G}_{uac})$ that are self-loops (i.e., when $\mathbf{v} = \mathbf{u}$ holds) or that are symmetric pairs (i.e., when $(\mathbf{u}, \mathbf{v})$ occurs given that $(\mathbf{v}, \mathbf{u})$ has already occurred in the sequence), while preserving the previous lexicographical order of Step (iii);

(v) searches for $((a_1, \dots, a_p), (b_1, \dots, b_p))$ or $((b_1, \dots, b_p), (a_1, \dots, a_p))$ in the previous sequence of composite edges for the set $\mathbb{E}_{uac}(\mathscr{G}_{uac})$ that was built in Step (iv);

  (a) if $((a_1, \dots, a_p), (b_1, \dots, b_p))$ or $((b_1, \dots, b_p), (a_1, \dots, a_p))$ is found, then returns the place $j \in \mathbb{N}$ for which $((a_1, \dots, a_p), (b_1, \dots, b_p))$ or $((b_1, \dots, b_p), (a_1, \dots, a_p))$ occurred in the previous sequence of composite edges built in Step (iv);

  (b) if $((a_1, \dots, a_p), (b_1, \dots, b_p))$ or $((b_1, \dots, b_p), (a_1, \dots, a_p))$ is not found, then returns 0.

To construct a program that satisfies (III) and (IV), a similar algorithm can define $\mathrm{p'}_2$: it works exactly in the same way until Step (iv) of $\mathrm{p'}_1$, but in the new 'Step (v)' program $\mathrm{p'}_2$ searches for the j-th element in the sequence generated by Step (iv) and returns the respective pair of tuples (or returns $\langle 0 \rangle$, if $1 \leq j \leq |\mathbb{E}_{uac}(\mathscr{G}_{uac})|$ does not hold). Finally, note that the existence of such programs $\mathrm{p'}_1$ and $\mathrm{p'}_2$ is (Turing) equivalent to say that the MAG $\mathscr{G}_{uac}$ is encodable given $\tau_{ua}(\mathscr{G}_{uac})$.[6] □

Secondly, note that the characteristic string of a node-unaligned MAG is defined in a similar way as in Section 3.1:

Definition 6. *Let* $\left(e_1, \dots, e_{|\mathbb{E}_{uac}(\mathscr{G}_{uac})|}\right)$ *be any arbitrary ordering of all possible composite edges between existing composite vertices of a* node-unaligned simple *MAG* $\mathscr{G}_{uac}$. *We say that a string* $x' \in \{0,1\}^*$ *with* $l(x') = |\mathbb{E}_{uac}(\mathscr{G}_{uac})|$ *is a* node-unaligned characteristic string *of* $\mathscr{G}_{uac}$ *iff, for every* $e_j \in \mathbb{E}_{uac}(\mathscr{G}_{uac})$,

$$e_j \in \mathscr{E}_{ua}(\mathscr{G}_{uac}) \iff \text{the } j\text{-th digit in } x' \text{ is } 1 \ ,$$

where $1 \leq j \leq l(x')$.

Now, for the node-unaligned composite edge set string, the definition may seem not so straightforwardly translated from Definition 2. As one can see below, it is based on a sequence of the $|\mathbb{E}_c(\mathscr{G}_{uac})|$ composite edges, and not on the sequence of $|\mathbb{E}_{uac}(\mathscr{G}_{uac})|$ composite edges. This is because one needs to embed into node-unaligned composite edge set strings not only the characteristic function of the set $\mathscr{E}_{ua}(\mathscr{G}_{uac})$, as in the node-aligned case, but also the characteristic function of the set $\mathbb{V}_{ua}(\mathscr{G}_{uac})$ (which becomes in turn determined by the k_i's and h_i's in the following definition):

Definition 7. *Let* $\left(e_1, \dots, e_{|\mathbb{E}_c(\mathscr{G}_{uac})|}\right)$ *be any arbitrary ordering of all possible composite edges of a node-unaligned simple MAG* $\mathscr{G}_{uac}$. *Then,* $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$ *denotes the* node-unaligned composite edge set string $\langle \langle e_1, z'_1, k_1, h_1 \rangle, \dots, \langle e_n, z'_n, k_n, h_n \rangle \rangle$ *such that:*

$$z'_i = 1 \iff e_i \in \mathscr{E}_{ua}(\mathscr{G}_{uac}) \ ,$$

[6] The reader is invited to note that, if Step (ii) is eliminated and $\langle \tau(\mathscr{G}_c) \rangle$ is given as input instead of $\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle$, then analogous versions of these two algorithms can be employed to prove the previous node-aligned case in Lemma 2.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

$$k_i = 1 \iff (e_i = (\mathbf{v}, \mathbf{u}) \wedge \mathbf{v} \in \mathbb{V}_{ua}(\mathscr{G}_{uac}))$$

and

$$h_i = 1 \iff (e_i = (\mathbf{v}, \mathbf{u}) \wedge \mathbf{u} \in \mathbb{V}_{ua}(\mathscr{G}_{uac})) ,$$

where $z'_i, k_i, h_i \in \{0,1\}$ *with* $1 \le i \le n = |\mathbb{E}_c(\mathscr{G}_{uac})|$.

This way, we guarantee not only that the characteristic string x can always be computably extracted from $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$, but also that the set $\mathbb{V}_{ua}(\mathscr{G}_{uac})$ can be computed if $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$ is known *a priori* as input. This will be important in the proof of Theorem 13 later on. Moreover, once the ordering of $\mathbb{E}_c(\mathscr{G}_{uac})$ assumed in Definition 7 is preserved by the subsequence that exactly corresponds to the ordering of $\mathbb{E}_{uac}(\mathscr{G}_{uac})$ assumed in Definition 6, we have in Lemma 10 below that both the node-unaligned simple MAG and its respective node-unaligned characteristic string become "equivalent" in terms of algorithmic information, but again (as occurred for the node-aligned case) except for the minimum information necessary to encode the node-unaligned companion tuple:

Lemma 10. *Let* $x' \in \{0,1\}^*$. *Let* $\mathscr{G}_{uac}$ *be an encodable node-unaligned simple MAG given* $\tau_{ua}(\mathscr{G}_{uac})$ *such that* x' *is the respective node-unaligned characteristic string. Then,*

$$\mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle \mid x') \le \mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle) + \mathbf{O}(1) \tag{8}$$

$$\mathbf{K}(x' \mid \langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle) \le \mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle) + \mathbf{O}(1) \tag{9}$$

$$\mathbf{K}(x') = \mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle) \pm \mathbf{O}\Big(\mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle)\Big) . \tag{10}$$

Proof. From the assumption of encodability of $\mathscr{G}_{uac}$ given $\tau_{ua}(\mathscr{G}_{uac})$ in Lemma 10, first note that there must be programs $\mathrm{p'}_1, \mathrm{p'}_2$ as defined in the proof of Lemma 9. Moreover, note that the sequence $\Big(e_1, \dots, e_{|\mathbb{E}_c(\mathscr{G}_{uac})|}\Big)$ can be computably built from $(|\mathscr{A}(\mathscr{G}_{ua})[1]|, \dots, |\mathscr{A}(\mathscr{G}_{ua})[p]|)$ in $\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle$. Remember that, if $(a_1, \dots, a_p)$ or $(b_1, \dots, b_p)$ does not belong to $\mathbb{V}_{ua}(\mathscr{G}_{uac})$, then

$$\mathbf{U}(\langle \langle a_1, \dots, a_p \rangle, \langle b_1, \dots, b_p \rangle, \langle \langle \tau_{ua}(\mathscr{G}_{uac}) \rangle, \mathrm{p'}_1 \rangle \rangle) = 0 .$$

Thus, the sequence $\Big(e_1, \dots, e_{|\mathbb{E}_{uac}(\mathscr{G}_{uac})|}\Big)$, which is a subsequence of $\Big(e_1, \dots, e_{|\mathbb{E}_c(\mathscr{G}_{uac})|}\Big)$, can be computably built from $\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle$. Now, let x' be the correspondent characteristic string (as in Definition 6) of $\mathscr{G}_{uac}$. Therefore, one can construct other programs p and q with $\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle$ already given as first input so that, respectively: p returns x' given $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$ (or given $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle^*$); and q returns $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$ given x' (or given x'^*). In other words, string $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$ is Turing equivalent to string x', if $\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle$ is given. This always holds because the values of $x'_1, \dots, x'_{|\mathbb{E}_{uac}(\mathscr{G}_{uac})|}$ that define the node-unaligned caracteristic string x', where $x' := x'_1 \cdots x'_{|\mathbb{E}_{uac}(\mathscr{G}_{uac})|}$, are computably embedded into the composite edge set string $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$ as a subsequence, since, for $1 \le i \le |\mathbb{E}_c(\mathscr{G}_{uac})|$, wherever $k_i = 0$ or $h_i = 0$ occurs one has that $z'_i = 0$ also occurs. This way, program q just needs to assign $z'_i = 0$ wherever $k_i = 0$ or $h_i = 0$ (which are determined by the values of $m_1, \dots, m_{|\mathbb{V}(\mathscr{G}_{uac})|}$ in $\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle$) and read the values of $x'_1, \dots, x'_{|\mathbb{E}_{uac}(\mathscr{G}_{uac})|}$ in order to insert each of these in the right order wherever $k_i = 1$ and $h_i = 1$. Conversely, program p can computably extract x' from $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$ by ignoring those z'_i wherever $k_i = 0$ or $h_i = 0$.[7] Then, from the minimality of $\mathbf{K}(\cdot)$, we will have that Equations 8 and 9 hold because of the existence of programs q and p respectively. Finally, from Equations 8 and 9, one can then achieve Equation 10 using basic inequalities in AIT regarding unconditional and conditional prefix algorithmic complexity. □

[7] Thus, Equation 9 in particular can be improved to $\mathbf{K}(x' \mid \langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle) \le \mathbf{O}(1)$. The same improvement can also be analogously achieved for Equation 2.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

Note that Lemma 10 holds for any other form of encoding a node-unaligned simple MAG that happen to be equivalent to node-unaligned composite edge set strings $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$. For example, one can equivalently encode $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$ as an three-dimensional array composed of positive integers, Boolean variables, and lists: the first dimension stores the index value $\langle a_1, \dots, a_p \rangle$ of each composite vertex $\mathbf{v} = (a_1, \dots, a_p) \in \mathbb{V}(\mathscr{G}_{uac})$; the second dimension stores a Boolean value determining whether the composite vertex in the first dimension exists or not; and the third dimension stores a list containing the index values of each composite vertices with which the composite vertex in the first dimension forms a composite edge.

As we saw in the node-aligned case, we shall see in the next section that node-unaligned characteristic strings alone are not in general equivalent to composite edge set strings $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$. However, in any event, Lemmas 3 and 10 together with Lemma 4 directly establish that the algorithmic information distortions are always upper bounded by the algorithmic information carried by the companion tuple, whether node-aligned or node-unaligned. Thus, even in the worst-case scenario, the value of the algorithmic complexity of the companion tuple times a independent constant can be always applied as an error margin for the algorithmic information distortions between simple MAGs and their isomorphic classical graphs.

5. Worst-case algorithmic information distortions

In this section, we investigate worst-case algorithmic information distortions for node-unaligned MAGs when the multidimensional space is arbitrarily large. In particular, we study large multidimensional spaces that are non-uniform or uniform. Like we saw in Section 3.2, we show in the following theorems that there are cases in which the algorithmic information necessary for retrieving the encoded form of the node-unaligned simple MAG from its characteristic string is close (except for a logarithmic term) to the upper bound given by Equation 8 in Lemma 10.

Before heading toward the theorems themselves, it is important to show the two cases in Lemmas 11 or 12 for which the set $\mathbb{V}_{ua}(\mathscr{G}_{uac})$ trivializes the problem either by reducing it back to the node-aligned case as in Section 3.2 or by reducing it to a problem of just inserting empty nodes.

The first trivializing case guarantees the consistency of our definitions of node-aligned and node-unaligned MAGs:

Lemma 11. *Let $\mathscr{G}_{uac}$ be a node-unaligned simple MAG with $\mathbb{V}_{ua}(\mathscr{G}_{uac}) = \mathbb{V}(\mathscr{G}_{uac})$, where x is its node-aligned characteristic string (as in Definition 1) and x' is its node-unaligned characteristic string (as in Definition 6). Then,*

$$\mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle) = \mathbf{K}(\langle \mathscr{E}(\mathscr{G}_{uac}) \rangle) \pm \mathbf{O}(1) \,, \tag{11}$$

$$\mathbf{K}(x) = \mathbf{K}(x') \pm \mathbf{O}(1) \,, \tag{12}$$

and

$$\mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle) = \mathbf{K}(\langle \tau(\mathscr{G}_{uac}) \rangle) \pm \mathbf{O}(1) \tag{13}$$

hold.

Proof. In order to show that $\mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle) = \mathbf{K}(\langle \mathscr{E}(\mathscr{G}_{uac}) \rangle) \pm \mathbf{O}(1)$, just note that, since $\mathbb{V}_{ua}(\mathscr{G}_{uac}) = \mathbb{V}(\mathscr{G}_{uac})$, then all the k_i's and h_i's in $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$ assume the value 1. Thus, either eliminating or inserting them into the companion tuples can only add up to $\mathbf{O}(1)$ bits of algorithmic information. In order to show that $\mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle) = \mathbf{K}(\langle \tau(\mathscr{G}_{uac}) \rangle) \pm \mathbf{O}(1)$, just note that $\left\langle m_1, \dots, m_{|\mathbb{V}(\mathscr{G}_{uac})|} \right\rangle$, which appears in $\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle$, is a tuple of 1's and the value of $|\mathbb{V}(\mathscr{G}_{ua})|$ can be computed directly from $\bigtimes_{i=1}^{p} |\mathscr{A}(\mathscr{G}_{uac})[i]|$. Thus, since $\bigtimes_{i=1}^{p} |\mathscr{A}(\mathscr{G}_{uac})[i]|$ in turn can be directly computed from the string $\langle |\mathscr{A}(\mathscr{G}_{uac})[1]|, \dots, |\mathscr{A}(\mathscr{G}_{uac})[p]| \rangle$, which is given in both companion tuples, then either elim-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

inating or inserting $m_1, \dots, m_{|\mathbb{V}(\mathscr{G}_{uac})|}$ into the companion tuples can only add up to $\mathbf{O}(1)$ bits of algorithmic information. Finally, since $\mathbb{V}_{ua}(\mathscr{G}_{uac}) = \mathbb{V}(\mathscr{G}_{uac})$ and, consequentially,

$$\left(e_1, \dots, e_{|\mathbb{E}_{uac}(\mathscr{G}_{uac})|}\right) = \left(e_1, \dots, e_{|\mathbb{E}_c(\mathscr{G}_{uac})|}\right),$$

then one has that $\mathbf{K}(x) = \mathbf{K}(x') \pm \mathbf{O}(1)$ trivially holds because $x = x'$. □

In fact, if $\mathbb{V}_{ua}(\mathscr{G}_{uac}) = \mathbb{V}(\mathscr{G}_{uac})$, the same proof as Lemma 11 can be employed to show that the strings in the left side of the equations in Lemma 11 are in fact respectively Turing equivalent to their counterparts in the right side. Therefore, for any $\mathscr{G}_{uac}$ satisfying Lemma 11, any algorithmic information distortion occurs in the same manner as in the node-aligned case.

The second one guarantees the consistency between network connectedness and empty nodes. An empty node [25] is a totally unconnected node that is added to the network in order to recover the node alignment of an former node-unaligned network. Thus, as expected, if all the composite vertices in $\mathbb{V}_{ua}(\mathscr{G}_{uac})$ are connected to at least another composite vertex in $\mathbb{V}_{ua}(\mathscr{G}_{uac})$, then all the possible unconnected composite vertices are those that redundantly are empty nodes:

Lemma 12. *Let $\mathscr{G}_{uac}$ be a node-unaligned simple MAG in which every composite vertex in $\mathbb{V}_{ua}(\mathscr{G}_{uac})$ is connected to at least another composite vertex in $\mathbb{V}_{ua}(\mathscr{G}_{uac})$. Then,*

$$\mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle) = \mathbf{K}(\langle \mathscr{E}(\mathscr{G}_{uac}) \rangle) \pm \mathbf{O}(1) \tag{14}$$

holds and, additionally, $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$ is in fact Turing equivalent to $\langle \mathscr{E}(\mathscr{G}_{uac}) \rangle$.

Proof. Constructing an algorithm for returning $\langle \mathscr{E}(\mathscr{G}_{uac}) \rangle$ given $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$ as input is straightforward, since one just needs to eliminate all the k_i's and h_i's in $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$. This way, by the minimality of $\mathbf{K}(\cdot)$, we will have that $\mathbf{K}(\langle \mathscr{E}(\mathscr{G}_{uac}) \rangle) \leq \mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle) + \mathbf{O}(1)$ holds. In order to show the inverse case, note that there is always an algorithm that, given $\langle \mathscr{E}(\mathscr{G}_{uac}) \rangle$ as input, identifies those composite vertices $\mathbf{v}$ for which there is an absent composite edge (i.e., a pair in the form $\langle e_i, 0 \rangle$ in $\langle \mathscr{E}(\mathscr{G}_{uac}) \rangle$, where $e_i = (\mathbf{v}, \mathbf{u})$ or $e_i = (\mathbf{u}, \mathbf{v})$) such that no other composite edge composed of $\mathbf{v}$ is present in $\mathscr{E}(\mathscr{G}_{uac})$. Hence, from a list of these composite vertices, it replaces each $\langle e_i, 0 \rangle$ with $\langle e_i, 0, k_i, h_i \rangle$ accordingly, where either k_i or h_i is 0 wherever it corresponds to the place of $\mathbf{v}$ in e_i. This way, $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle$ is computably built from $\langle \mathscr{E}(\mathscr{G}_{uac}) \rangle$ and, by the minimality of $\mathbf{K}(\cdot)$, we will have that $\mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle) \leq \mathbf{K}(\langle \mathscr{E}(\mathscr{G}_{uac}) \rangle) \pm \mathbf{O}(1)$. □

Note that in Lemma 12 one immediately has that $\left\langle E\left(G^{ua}_{\mathscr{G}_{uac}}\right)\right\rangle$ can be computed from $\langle E(G_{\mathscr{G}_{uac}}) \rangle$ with a simple algorithm that identifies totally unconnected vertices. Furthermore, one has that $\langle E(G_{\mathscr{G}_{uac}}) \rangle$ can be computed from $\left\langle E\left(G^{ua}_{\mathscr{G}_{uac}}\right)\right\rangle$, if the value of $|\mathbb{V}(\mathscr{G}_{uac}) \setminus \mathbb{V}_{ua}(\mathscr{G}_{uac})|$ is also given as input. Therefore, for any MAG satisfying Lemma 12, the algorithmic information distortion between the MAG $\mathscr{G}_{uac}$ and the *unaligning* MAG-graph-isomorphic classical graph $G^{ua}_{\mathscr{G}_{uac}}$ can only differ from the algorithmic information distortion between the MAG $\mathscr{G}_{uac}$ and the *aligning* MAG-graph-isomorphic classical graph $G_{\mathscr{G}_{uac}}$ by $\mathbf{O}(\log_2(|\mathbb{V}(\mathscr{G}_{uac}) \setminus \mathbb{V}_{ua}(\mathscr{G}_{uac})|))$ bits of algorithmic information.

Thus, as the reader might notice, the case in which the node nonalignment introduces more irreducible information into the composite edge set string is when $\mathbb{V}_{ua}(\mathscr{G}_{uac}) \neq \mathbb{V}(\mathscr{G}_{uac})$ and not every unconnected composite vertex is empty. Under these conditions that demand a more careful theoretical analysis than the trivializing cases in Lemmas 11 or 12, we study now that there may also be exponential algorithmic information distortions in the node-unaligned case.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

Theorem 13. *There are encodable* node-unaligned simple *MAGs $\mathscr{G}_{uac}$ given $\tau_{ua}(\mathscr{G}_{uac})$ with arbitrarily large* non-uniform *multidimensional spaces such that*

$$\mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle) + \mathbf{O}(1) \geq \mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle \mid x') \geq \mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle) - \mathbf{O}\Big(\log_2 \left(\mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle) \right) \Big)$$

with

$$\mathbf{K}(x') = \mathbf{O}(p) = \mathbf{O}\Big(\log_2 \left(\mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle) \right) \Big) ,$$

where x' is the respective node-unaligned characteristic string and p is the order of $\mathscr{G}_{uac}$.

Proof. This proof shares the same main underlying idea with the proof of Theorem 5 in [1]: we construct MAGs such that, although the companion tuples are incompressible, the characteristic strings and the respective exact number of composite vertices can be computed with much less algorithmic information than the companion tuple demands to be computed. However, now the node-unaligned case demands a more careful theoretical analysis on the compressibility of the node-unaligned companion tuple, the size of the set $\mathbb{V}_{ua}(\mathscr{G}_{uac})$, and the algorithmic information necessary to compute the node-unaligned characteristic string. First, let $\mathscr{G}_{uac}$ be any node-unaligned simple MAG such that, for every $1 \leq i \leq p$ and $1 \leq j \leq |\mathbb{V}(\mathscr{G}_{uac})|$, one has that

$$\begin{aligned} \mathscr{A}(\mathscr{G}_{uac})[i] = \{1,2\} &\iff \text{the } i\text{-th digit of } w_1 \text{ is } 1 \\ \mathscr{A}(\mathscr{G}_{uac})[i] = \{1\} &\iff \text{the } i\text{-th digit of } w_1 \text{ is } 0 \\ m_j = 1 &\iff \text{the } i\text{-th digit of } w_2 \text{ is } 1 \\ m_j = 0 &\iff \text{the } i\text{-th digit of } w_2 \text{ is } 0 \end{aligned}$$

hold, where $\mathscr{E}_{ua}(\mathscr{G}_{uac}) \subseteq \mathbb{E}_{uac}(\mathscr{G}_{uac})$ and $p \in \mathbb{N}$ and $w_1, w_2, x' \in \{0,1\}^*$ are arbitrary. Since w_1 and w_2 are arbitrary, let w_1 and w_2 be bit strings that, respectively, are arbitrarily long finite initial segments $y \upharpoonright_{l(w_1)}$ and $y \upharpoonright_{l(w_2)}$ of a 1-random real number y. From Section 2.2, remember that, if y is a 1-*random real number*, then $\mathbf{K}(y \upharpoonright_n) \geq n - \mathbf{O}(1)$, where $n \in \mathbb{N}$ is arbitrary. From Lemma 9, we have that $\mathscr{G}_{uac}$ is encodable given $\tau_{ua}(\mathscr{G}_{uac})$. Therefore, there is a program q′ that represents an algorithm running on a prefix universal Turing machine **U** that proceeds as follows:

(i) receive $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle^*$ as input;
(ii) calculate the value of $\mathbf{U}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle^*)$ and build a sequence $(e_1, \dots, e_n)$ of the composite edges $e_i \in \mathbb{E}_c(\mathscr{G}_{uac})$, where $n = |\mathbb{E}_c(\mathscr{G}_{uac})|$, in the exact same order that they appear in $\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle = \mathbf{U}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle^*)$;
(iii) build a finite ordered set

$$\mathbb{V}' := \left\{ \mathbf{v} \middle| e' \in (e_1, \dots, e_n) \text{ , where } \left(e' = \langle \mathbf{v}, \mathbf{u} \rangle \lor e' = \langle \mathbf{u}, \mathbf{v} \rangle \right) \right\} ;$$

(iv) build a finite list $\left[A_1, \dots, A_p \right]$ of finite ordered sets

$$A_i := \left\{ a_i \middle| a_i \text{ is the } i\text{-th element of } \mathbf{v} = \left(a_1, \dots, a_p \right) \in \mathbb{V}' \right\} ,$$

where p is finite;
(v) for every i with $1 \leq i \leq p$, make

$$z_i := |A_i| ;$$

(vi) build a finite ordered set

$$\mathbb{V}'_{ua} := \left\{ \mathbf{v} \middle| \begin{array}{l} e_i \in (e_1, \dots, e_n) \text{ , where} \\ \quad (e_i = (\mathbf{v}, \mathbf{u}) \land k_i = 1) \\ \text{or} \\ \quad (e_i = (\mathbf{u}, \mathbf{v}) \land h_i = 1) \end{array} \right\} ;$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

(vii) build a list

$$\left[m_1, m_2, \dots, m_{|\mathbb{V}'|}\right]$$

from

$$\mathbb{V}'[j] \in \mathbb{V}'_{ua} \iff m_j := 1$$
$$\mathbb{V}'[j] \notin \mathbb{V}'_{ua} \iff m_j := 0\,;$$

(viii) return the bit string

$$\left\langle \langle z_1, z_2, \dots, z_p \rangle, \left\langle m_1, m_2, \dots, m_{|\mathbb{V}'|} \right\rangle \right\rangle .$$

Therefore, from our construction of $\mathscr{G}_{uac}$ and Definition 4, we will have that

$$\mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle) \leq l(\langle \langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle^*, \mathrm{q}' \rangle) \leq \mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle) + \mathbf{O}(1) \tag{15}$$

holds by the minimality of $\mathbf{K}(\cdot)$ and by our construction of q'. Note that

$$l(w_2) = |\mathbb{V}(\mathscr{G}_{uac})|\,,$$

where

$$\bigtimes_{i=1}^{p} |\mathscr{A}(\mathscr{G}_{uac})[i]| = |\mathbb{V}(\mathscr{G}_{uac})|\,, \tag{16}$$

and any $|\mathscr{A}(\mathscr{G}_{uac})[i]|$ can only assume values in $\{1, 2\}$. One can trivially construct: an algorithm that returns $y \upharpoonright_{l(w_2)}$ given $\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle$ as input; and another algorithm that returns $\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle$ given $y \upharpoonright_{l(w_2)}$ and $y \upharpoonright_{l(w_1)}$ as inputs. Also note that $l(w_1) = p$ and $l(w_2) = |\mathbb{V}(\mathscr{G}_{uac})|$. This way, we will have that[8]

$$\mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle) \leq \mathbf{K}(y \upharpoonright_{l(w_1)}) + \mathbf{K}(y \upharpoonright_{l(w_2)}) + \mathbf{O}(1) \leq p + |\mathbb{V}(\mathscr{G}_{uac})| + \mathbf{O}(\log_2(p)) \tag{17}$$

and, since y is 1-random,

$$|\mathbb{V}(\mathscr{G}_{uac})| - \mathbf{O}(1) \leq \mathbf{K}(y \upharpoonright_{l(w_2)}) \leq \mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle) + \mathbf{O}(1)\,. \tag{18}$$

Other property that follows from the fact that y is 1-random and w_1 and w_2 being arbitrarily long is that both w_1 and w_2 are Borel normal [13,34], as in Section 2.2. Therefore, we will have that

$$|\mathbb{E}_{uac}(\mathscr{G}_{uac})| = \frac{\left(\frac{2^{\left(\frac{p}{2} \pm \mathbf{o}(p)\right)}}{2} \pm \mathbf{o}\left(2^{\left(\frac{p}{2} \pm \mathbf{o}(p)\right)}\right)\right)^2 - \left(\frac{2^{\left(\frac{p}{2} \pm \mathbf{o}(p)\right)}}{2} \pm \mathbf{o}\left(2^{\left(\frac{p}{2} \pm \mathbf{o}(p)\right)}\right)\right)}{2}\,, \tag{19}$$

where

$$|\mathbb{V}(\mathscr{G}_{uac})| = 2^{\left(\frac{p}{2} \pm \mathbf{o}(p)\right)}$$

and

$$|\mathbb{V}_{ua}(\mathscr{G}_{uac})| = \frac{|\mathbb{V}(\mathscr{G}_{uac})|}{2} \pm \mathbf{o}(|\mathbb{V}(\mathscr{G}_{uac})|)\,.$$

This way, since the exact number of 1's appearing in w_1 is given by $\frac{p}{2} \pm \mathbf{o}(p)$, there is a program of size

$$\mathbf{O}\left(\log_2(\frac{p}{2})\right) + \mathbf{O}(\log_2(\mathbf{o}(p))) + \mathbf{O}(1) \leq \mathbf{O}(\log_2(p))$$

[8] In particular, from the Borel normality of w_2 and Equation 22 later on, we can further improve Equation 17 to

$$\mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle) \leq |\mathbb{V}(\mathscr{G}_{uac})| + \mathbf{O}(\log_2(\log_2(|\mathbb{V}(\mathscr{G}_{uac})|)))\,.$$

that returns the integer value $|\mathbb{V}(\mathscr{G}_{uac})|$ as output. Hence, since the exact number of 1's appearing in w_2 is given by $\frac{|\mathbb{V}(\mathscr{G}_{uac})|}{2} \pm \mathbf{o}(|\mathbb{V}(\mathscr{G}_{uac})|)$, there is a program of size

$$\mathbf{O}\left(\log_2\left(\frac{2^{\left(\frac{p}{2}\pm\mathbf{o}(p)\right)}}{2}\right)\right) + \mathbf{O}\left(\log_2\left(\mathbf{o}\left(2^{\left(\frac{p}{2}\pm\mathbf{o}(p)\right)}\right)\right)\right) + \mathbf{O}(\log_2(p)) \leq \mathbf{O}(p) + \mathbf{o}(p) + \mathbf{O}(\log_2(p)) = \mathbf{O}(p) \tag{20}$$

that returns the integer value $|\mathbb{E}_{uac}(\mathscr{G}_{uac})|$ as output. Consequentially, we will have that

$$\mathbf{K}(l(x')) = \mathbf{O}(p) \tag{21}$$

holds by the fact that $l(x') = |\mathbb{E}_{uac}(\mathscr{G}_{uac})|$. Moreover, since p is arbitrarily large and $|\mathbb{V}(\mathscr{G}_{uac})| = 2^{\left(\frac{p}{2}\pm\mathbf{o}(p)\right)}$, the Borel normality of w_1 also guarantees that

$$\mathbf{\Omega}(p) = \log_2(|\mathbb{V}(\mathscr{G}_{uac})|) \ . \tag{22}$$

Now, since $\mathscr{E}_{ua}(\mathscr{G}_{uac})$ and p were arbitrary, we can choose any node-unaligned characteristic string x' such that

$$\mathbf{K}\left(x' \,\middle|\, l(x')\right) = \mathbf{O}(1) \tag{23}$$

holds and that there are some composite vertices in $\mathbb{V}_{ua}(\mathscr{G}_{uac})$ that are not connected to any other composite vertex in $\mathbb{V}_{ua}(\mathscr{G}_{uac})$.[9] Thus, from Equations 15, 18, 21, 22 and since $\mathbf{K}\left(x' \,\middle|\, l(x')\right) = \mathbf{O}(1)$ and p is arbitrarily large, we will have that

$$\begin{aligned} \mathbf{K}(x') &\leq \mathbf{K}(l(x')) + \mathbf{O}(1) \leq \\ &\leq \mathbf{O}(p) \leq \\ &\leq \mathbf{O}(\log_2(|\mathbb{V}(\mathscr{G}_{uac})|)) \leq \\ &\leq \mathbf{O}\Big(\log_2\left(\mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac})\rangle)\right)\Big) \leq \\ &\leq \mathbf{O}\Big(\log_2\left(\mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac})\rangle)\right)\Big) \end{aligned} \tag{24}$$

holds by the minimality of $\mathbf{K}(\cdot)$ and by basic inequalities in AIT regarding unconditional and conditional prefix algorithmic complexity. Therefore, from Equations 15 and 24 together with basic inequalities in AIT, we will have that

$$\begin{aligned} \mathbf{K}\Big(\langle \tau_{ua}(\mathscr{G}_{uac})\rangle\Big) &\leq \mathbf{K}\Big(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac})\rangle\Big) + \mathbf{O}(1) \leq \\ &\leq \mathbf{K}(x') + \mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac})\rangle \,|\, x') + \mathbf{O}(1) \leq \\ &\leq \mathbf{O}\Big(\log_2\left(\mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac})\rangle)\right)\Big) + \mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac})\rangle \,|\, x') \ . \end{aligned}$$

Finally, the proof of $\mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac})\rangle) + \mathbf{O}(1) \geq \mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac})\rangle \,|\, x')$ follows directly from Lemma 10. □

As for Theorem 5, the reader is then invited to note that the proof of Theorem 13 also works for many other forms of companion tuples $\tau_{ua}(\mathscr{G}_{uac})$. For example, keep w_1 and w_2

[9] A trivial example of x' satisfying these requirements is x' being a binary sequence starting with 1 and repeating 0's until the length matches $|\mathbb{E}_{uac}(\mathscr{G}_{uac})|$, but any other example of x' in which the node-unaligned MAG is denser also works, as long as it satisfies those two requirements.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

being long enough finite initial segments of a 1-random real number y and then define $\tau_{ua}(\mathscr{G}_{uac})$ such that

$$\begin{aligned} \mathscr{A}(\mathscr{G}_{uac})[i] = \{1, \dots, (f_1(p) + f_2(p))\} &\iff \text{the } i\text{-th digit of } w \text{ is } 1 \\ \mathscr{A}(\mathscr{G}_{uac})[i] = \{1, \dots, f_1(p)\} &\iff \text{the } i\text{-th digit of } w \text{ is } 0 \\ m_i = 1 &\iff \text{the } i\text{-th digit of } w_2 \text{ is } 1 \\ m_i = 0 &\iff \text{the } i\text{-th digit of } w_2 \text{ is } 0 \,, \end{aligned}$$

where $f_1 : \mathbb{N} \to \mathbb{N} \setminus \{0\}$ and $f_2 : \mathbb{N} \to \mathbb{Z} \setminus \{0\}$ are arbitrary total computable functions.

For the purpose of comparison, the next immediate question arises from whether there might be such a worst-case distortions between composite edge set strings and characteristic strings when the multidimensional space is *uniform* and the network is *node-aligned*. As the reader might expect, we show in Lemma 14 below that node-aligned MAGs with uniform multidimensional spaces are more tightly associated to their characteristic strings in terms of the algorithmic information and, thus, it cannot display the same distortions as in Theorems 5 and 13, which grow exponentially with p. In particular, the distortions in the node-aligned uniform case can only grow up to a logarithmic term of the order p; and this algorithmic information necessary to compute the value of p can only grow up to a double logarithmic term of length of the node-aligned characteristic string:

Lemma 14. *Let $\mathscr{G}_c$ be an arbitrary* node-aligned classical *MAG with arbitrarily large* uniform *multidimensional spaces, where $|\mathbb{V}(\mathscr{G}_c)| \geq 3$ and $|\mathscr{A}(\mathscr{G}_c)[i]| \geq 2$ for every $i \leq p$. Then,*

$$\mathbf{K}(x) \leq \mathbf{K}(\langle \mathscr{E}(\mathscr{G}_c) \rangle) + \mathbf{O}(1) \leq \mathbf{K}(x) + \mathbf{O}(\log_2(p)) \leq \mathbf{K}(x) + \mathbf{O}(\log_2(\log_2(l(x)))) \,,$$

where x is the respective node-aligned characteristic string *and p is the order of $\mathscr{G}_c$.*

Proof. Since the multidimensional space is uniform (i.e., when $\mathscr{A}(\mathscr{G}_c)[i] = \mathscr{A}(\mathscr{G}_c)[j]$ holds for every $i, j \leq p$), there is a simple algorithm that always compute the integer value $|\mathscr{A}(\mathscr{G}_c)[i]|$, for any $i \leq p$, when $\bigtimes_{i=1}^{p} |\mathscr{A}(\mathscr{G}_c)[i]|$ and p are given as inputs. In addition, in this case, $\langle \tau(\mathscr{G}_c) \rangle$ can be computably built if $|\mathscr{A}(\mathscr{G}_c)[i]|$, for any $i \leq p$, and the value of p are given as inputs. Moreover, by solving a simple quadratic equation with just one possible positive integer solution, there also is a simple algorithm that always returns the integer value $\bigtimes_{i=1}^{p} |\mathscr{A}(\mathscr{G}_c)[i]|$ when $l(x) = |\mathbb{E}_c(\mathscr{G}_c)|$ is given as input. Note here that, since $|\mathbb{V}(\mathscr{G}_c)| \geq 3 > 2$, then

$$l(x) = |\mathbb{E}_c(\mathscr{G}_c)| = \frac{|\mathbb{V}(\mathscr{G}_c)|^2 - |\mathbb{V}(\mathscr{G}_c)|}{2} \geq |\mathbb{V}(\mathscr{G}_c)| \geq 1 \,. \tag{25}$$

From the self-delimiting (or prefix-free) property for $\mathbf{K}(\cdot)$, we clearly have that $l(x)$ is always computable given x^* as input. We have from Lemmas 2 and 3 that $\langle \mathscr{E}(\mathscr{G}_c) \rangle$ can always be computed if x^* and $\langle \tau(\mathscr{G}_c) \rangle$ are given as inputs. Thus, by combining all of these algorithms, we will have that $\langle \mathscr{E}(\mathscr{G}_c) \rangle$ can be computed given x^* and p as inputs. Hence, from the minimality of $\mathbf{K}(\cdot)$, we will have that

$$\mathbf{K}(\langle \mathscr{E}(\mathscr{G}_c) \rangle) \leq \mathbf{K}(x) + \mathbf{O}(\log_2(p)) \tag{26}$$

holds. Since

$$|\mathbb{V}(\mathscr{G}_c)| = |\mathscr{A}(\mathscr{G}_c)[1]|^p \,,$$

and $|\mathscr{A}(\mathscr{G}_c)[1]| \geq 2$, then from Equation 25 we also have that

$$p \leq \log_2(|\mathbb{V}(\mathscr{G}_c)|) \leq \log_2(l(x)) \,. \tag{27}$$

Therefore, from Equation 26,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

$$\mathbf{K}(\langle \mathscr{E}(\mathscr{G}_c) \rangle) \leq \mathbf{K}(x) + \mathbf{O}(\log_2(p)) \leq \mathbf{K}(x) + \mathbf{O}(\log_2(\log_2(l(x))))$$

holds. Finally, to prove that $\mathbf{K}(x) \leq \mathbf{K}(\langle \mathscr{E}(\mathscr{G}_c) \rangle) + \mathbf{O}(1)$, just note that one can always computably extract x from $\langle \mathscr{E}(\mathscr{G}_c) \rangle$. □

An interesting future research is to investigate whether one can construct an worst-case example of node-aligned multidimensional network with uniform multidimensional space that actually displays a distortion of the tight order of $\log_2(p)$. In any event, Lemma 14 already demonstrates an upper bound for the worst-case distortion increasing rate with respect to the value of p (i.e., with the number of extra node dimensions). In particular, as mentioned before, this upper bound is given by only a logarithmic term of p. On the other hand, although we saw in Lemma 14 that uniform multidimensional spaces can only display very small distortions in the node-aligned case, we show below that worst-case distortions that grow exponentially with p are still possible in the node-unaligned case:

Theorem 15. *There are encodable* node-unaligned simple *MAGs $\mathscr{G}_{uac}$ given $\tau_{ua}(\mathscr{G}_{uac})$ with $|\mathscr{A}(\mathscr{G}_{uac})[i]| \geq 2$ for every $i \leq p$ and with arbitrarily large* uniform *multidimensional spaces such that*

$$\mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle) + \mathbf{O}(1) \geq \mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle \mid x') \geq \mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle) - \mathbf{O}\Big(\log_2 \left(\mathbf{K}(\langle \tau_{ua}(\mathscr{G}_{uac}) \rangle) \right) \Big)$$

with

$$\log_2 \Big(\mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle) \Big) = \mathbf{\Omega}(p)$$

and

$$\mathbf{K}(x') = \mathbf{O}\Big(\log_2 \left(\mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle) \right) \Big) ,$$

where x' is the respective node-unaligned characteristic string and p is the order of $\mathscr{G}_{uac}$.

Proof. The underlying idea of this proof is similar to the proof of Theorem 13, but with the fundamental distinction in the set of all composite vertices $\bigtimes_{i=1}^{p} |\mathscr{A}(\mathscr{G}_{uac})[i]| = |\mathbb{V}(\mathscr{G}_{uac})|$, so that now $|\mathbb{V}(\mathscr{G}_{uac})| = |\mathscr{A}(\mathscr{G}_{uac})[1]|^p$ holds instead of $|\mathbb{V}(\mathscr{G}_{uac})| = 2^{\left(\frac{p}{2} \pm \mathbf{o}(p)\right)}$. Let $\mathscr{G}_{uac}$ be any node-unaligned simple MAG such that, for every $1 \leq i \leq p$, one has that $\mathscr{A}(\mathscr{G}_{uac})[i] = X = \{1, \dots, |X|\}$ and, for every $1 \leq j \leq |\mathbb{V}(\mathscr{G}_{uac})|$, one has that

$$\begin{aligned} m_j = 1 &\iff \text{the } j\text{-th digit of } w_2 \text{ is } 1 \\ m_j = 0 &\iff \text{the } j\text{-th digit of } w_2 \text{ is } 0 \, , \end{aligned}$$

where $\mathscr{E}_{ua}(\mathscr{G}_{uac}) \subseteq \mathbb{E}_{uac}(\mathscr{G}_{uac})$, $X \subset \mathbb{N}$, $p \in \mathbb{N}$ and $w_2, x' \in \{0,1\}^*$ are arbitrary. As in the proof of Theorem 13, let w_2 be an arbitrarily long finite initial segment $y \upharpoonright_{l(w_2)}$ of a 1-random real number y. From Lemma 9, we have that $\mathscr{G}_{uac}$ is encodable given $\tau_{ua}(\mathscr{G}_{uac})$. This way, Equations 15, 18 hold in the same way as in Theorem 13. However, now Equation 22 assumes a distinct and more precise form: since

$$|\mathbb{V}(\mathscr{G}_{uac})| = |X|^p = |\mathscr{A}(\mathscr{G}_{uac})[1]|^p ,$$

and $|X| = |\mathscr{A}(\mathscr{G}_{uac})[i]| \geq 2$, then we will have that

$$p \leq p \log_2(|X|) = \log_2(|\mathbb{V}(\mathscr{G}_{uac})|) \, . \tag{28}$$

Note that $p < |\mathbb{V}(\mathscr{G}_{uac})|$ and $|X| \leq |\mathbb{V}(\mathscr{G}_{uac})|$. Additionally, from the Borel normality of w_2, we know that

$$|\mathbb{V}_{ua}(\mathscr{G}_{uac})| = \frac{|\mathbb{V}(\mathscr{G}_{uac})|}{2} \pm \mathbf{o}(|\mathbb{V}(\mathscr{G}_{uac})|) \, .$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

Thus, there will be a program of size

$$\begin{aligned} \mathbf{O}(\log_2(|X|)) + \mathbf{O}(\log_2(p)) + \mathbf{O}(\log_2(\mathbf{o}(|X|^p))) + \mathbf{O}(1) \leq \\ \mathbf{O}(\log_2(|X|)) + \mathbf{O}(\log_2(p)) + \mathbf{O}(p \, \log_2(|X|)) \leq \\ \mathbf{O}(p \, \log_2(|X|)) + \mathbf{O}(\log_2(p)) \end{aligned} \tag{29}$$

that returns the integer value $|\mathbb{E}_{uac}(\mathscr{G}_{uac})|$ as output, where $|\mathbb{E}_{uac}(\mathscr{G}_{uac})| = l(x')$. This way, one can choose a node-unaligned characteristic string x' such that

$$\mathbf{K}\left(x' \,\middle|\, l(x')\right) = \mathbf{O}(1) \tag{30}$$

holds and that there are some composite vertices in $|\mathbb{V}_{ua}(\mathscr{G}_{uac})|$ that are not connected to any other composite vertex in $|\mathbb{V}_{ua}(\mathscr{G}_{uac})|$, where $l(x') = |\mathbb{E}_{uac}(\mathscr{G}_{uac})|$. Hereafter, the rest of the proof follows analogously to the proof of Theorem 13. □

The proofs of Corollaries 16 and 17 follows from Theorems 8, 13 and 15 and Lemma 4 in a totally analogous manner as Corollaries 6 and 7 follows from Theorem 1 and 5 and Lemma 4. Thus, we choose to leave the following proofs up to the reader.

Corollary 16. *There are an infinite family F' of* node-unaligned simple *MAGs, which may have either* uniform *or* non-uniform *multidimensional spaces, and an infinite set X' of the correspondent node-unaligned characteristic strings such that, for every constant $c \in \mathbb{N}$, there are $\mathscr{G}_{uac} \in F'$ and $x' \in X'$, where x' is the node-unaligned characteristic string of $\mathscr{G}_{uac}$ and*

$$\mathbf{O}\Big(\log_2 \left(\mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle)\right)\Big) > c + \mathbf{K}(x') \, . \tag{31}$$

Corollary 17. *There are an infinite family F'_1 of* node-unaligned simple *MAGs, which may have either* uniform *or* non-uniform *multidimensional spaces, and an infinite family F'_2 of* classical graphs, *where every classical graph in F'_2 is unaligning MAG-graph-isomorphic to at least one MAG in F'_1, such that, for every constant $c \in \mathbb{N}$, there are $\mathscr{G}_{uac} \in F'_1$ and a $G^{ua}_{\mathscr{G}_{uac}} \in F'_2$ that is unaligning MAG-graph-isomorphic to $\mathscr{G}_{uac}$, where*

$$\mathbf{O}\Big(\log_2 \left(\mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle)\right)\Big) > c + \mathbf{K}\Big(\Big\langle E\Big(G^{ua}_{\mathscr{G}_{uac}}\Big)\Big\rangle\Big).$$

Besides showing that node-unaligned multidimensional networks can display an exponentially larger algorithmic information distortions with respect to its isomorphic monoplex network, Theorems 13 and 15 together with Corollary 17 show that these distorted values of algorithmic information content grows *at least* exponentially with the order p (i.e., with number of extra node dimensions). In the same way as mentioned at the end of Section 3.2, future research is also needed for establishing the upper bound for distorted values of algorithmic information content with respect to p.

Finally, we can combine our results in order to achieve our last theorem, which summarizes the present article:

Theorem 18.

(I) *There are an infinite family F''_1 of* simple *MAGs and an infinite family F''_2 of* classical graphs, *where every classical graph in F''_2 is MAG-graph-isomorphic to at least one MAG in F''_1, such that:*

(a) *if the simple MAGs in F''_1 are* node-aligned *and have a* non-uniform *multidimensional space, then for every constant $c \in \mathbb{N}$,* there are *$\mathscr{G}_c \in F''_1$ and a $G_{\mathscr{G}_c} \in F''_2$ that is (aligning) MAG-graph-isomorphic to $\mathscr{G}_c$, where*

$$\mathbf{O}\Big(\log_2 \left(\mathbf{K}(\langle \mathscr{E}(\mathscr{G}_c) \rangle)\right)\Big) > c + \mathbf{K}(\langle E(G_{\mathscr{G}_c}) \rangle) \, ,$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2021 doi:10.20944/preprints202103.0056.v2

and this exponential distortion grows at least linearly *with the order p of the MAG $\mathscr{G}_c$, i.e.,*

$$\mathbf{K}(\langle \mathscr{E}(\mathscr{G}_c) \rangle) \geq p - \mathbf{O}(1) \,.$$

(b) *if the simple MAGs in F_1'' are* node-unaligned *and have either* non-uniform *or* uniform *multidimensional space, then for every constant $c \in \mathbb{N}$,* there are *$\mathscr{G}_{uac} \in F_1''$ and a $G^{ua}_{\mathscr{G}_{uac}} \in F_2''$ that is (unaligning) MAG-graph-isomorphic to $\mathscr{G}_{uac}$, where*

$$\mathbf{O}\Big(\log_2 \left(\mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle) \right) \Big) > c + \mathbf{K}\Big(\Big\langle E\Big(G^{ua}_{\mathscr{G}_{uac}} \Big) \Big\rangle \Big) \,,$$

and this exponential distortion grows at least exponentially *with the order p of the MAG $\mathscr{G}_{uac}$, i.e.,*

$$\log_2 \Big(\mathbf{K}(\langle \mathscr{E}_{ua}(\mathscr{G}_{uac}) \rangle) \Big) = \mathbf{\Omega}(p) \,.$$

(II) *Let F_1'' be an arbitrary infinite family of* node-aligned classical *MAGs with* uniform *multidimensional spaces. Let F_2'' be an arbitrary infinite family of* classical graphs *such that every classical graph in F_2'' is MAG-graph-isomorphic to at least one MAG in F_1'' and that both these graph and MAG share the same characteristic string. Then,* for every *$\mathscr{G}_c \in F_1''$ and $G_{\mathscr{G}_c} \in F_2''$ that is (aligning) MAG-graph-isomorphic to $\mathscr{G}_c$, one has that*

$$\mathbf{K}(\langle E(G_{\mathscr{G}_c}) \rangle) \leq \mathbf{K}(\langle \mathscr{E}(\mathscr{G}_c) \rangle) + \mathbf{O}(1) \leq \mathbf{K}(\langle E(G_{\mathscr{G}_c}) \rangle) + \mathbf{O}(\log_2(p))$$

and, therefore, any distortion can only grow up to *a* logarithmic *order of p.*

Proof. The proof of Theorem 18.(I)(a) follows directly from Theorem 5 and Corollary 7, which were previously presented in [1]. The proof of Theorem 18.(I)(b) follows directly from Theorems 13 and 15 together with Corollary 17. The proof of Theorem 18.(II) follows directly from Lemma 14 and Lemma 4 by fixing a computable ordering for both the sets $\mathbb{E}_c(\mathscr{G}_c)$ and $\mathbb{E}_c(G_{\mathscr{G}_c})$. □

6. Conclusions

This article presented mathematical results on network complexity, irreducible information content, and lossless compressibility analysis of node-aligned or node-unaligned multidimensional networks. We studied the limitations for algorithmic information theory (AIT) applied to monoplex networks or graphs to be imported into multidimensional networks, in particular, in the case the number of extra node dimensions (i.e., aspects) in these networks is sufficiently large. Our results demonstrate the existence of worst-case algorithmic information distortions when the algorithmic information content of a multidimensional network is compared with the algorithmic information content of its isomorphic monoplex network. More specifically, our proofs show that these distortions exist when a logarithmically compressible network topology of a monoplex network is embedded into a high-algorithmic-complexity multidimensional space.

Previous results in [1] have shown that *node-aligned* multidimensional networks with *non-uniform* multidimensional spaces can display an exponentially larger algorithmic complexity than the algorithmic complexity of its isomorphic monoplex network. In addition, [1] shows that these distorted values of algorithmic information content grow at least linearly with number of extra node dimensions.

When dealing with either *uniform* or *non-uniform* multidimensional spaces, we show in this article that *node-unaligned* multidimensional networks can also display exponential algorithmic information distortions with respect to the algorithmic information content of their respective isomorphic monoplex networks. Unlike the case studied in [1], these worst-case distortions in the node-unaligned case are shown to grow at least exponentially with the number of extra node dimensions. Thus, the node-unaligned case is more impactful than the previous node-aligned one precisely because exponential distortions may take place even with uniform multidimensional spaces. Furthermore, the distortions may grow much faster as the number of extra node dimensions increases.

On the other hand, we demonstrated that *node-aligned* multidimensional networks with *uniform* multidimensional spaces are limited to only display algorithmic information distortions that grow up to a logarithmic order of the number of extra node dimensions. As one might expect, the node alignment in conjunction with the uniformity of the multidimensional space guarantee that, in any event, the algorithmic information content of the multidimensional network and the algorithmic information content of its isomorphic monoplex network are tightly associated, except maybe for a logarithmic factor of the number of extra node dimensions.

The results in this article show that evaluations of the algorithmic information content of networks may be extremely sensitive to whether or not one is taking into account not only the total number of node dimensions, but also the respective sizes of each node dimension, and the ordering that they appear in the mathematical representation. Due to the need for additional irreducible information in order to compute the shape of the high-algorithmic-complexity multidimensional space, the present article shows that isomorphisms between finite multidimensional networks and finite monoplex networks do not preserve algorithmic information in general, so that the irreducible information content of a multidimensional network may be highly dependent on the choice of its encoded isomorphic copy. In order to avoid distortions in the general case when studying network complexity or lossless compression of multidimensional networks, it also highlights the importance of embedding the information necessary to determine the multidimensional space itself into the encoding of the multidimensional network. To such an end, network representation methods that take into account the algorithmic complexity of the data structure itself (unlike adjacency matrices, tensors, or characteristic strings) are required for importing algorithmic-information-based methods into the multidimensional case. In this way, given the relevance of algorithmic information theory in the challenge of causality discovery in network modeling, network summarization, network entropy, and compressibility analysis of networks, we believe this paper makes a fundamental contribution to the study of the complexity of multidimensional networks that have a large number of node dimensions, which in turn also imposes a need for being accompanied by more sophisticated algorithmic complexity approximating methods than those for monoplex networks or graphs.

Funding: Authors acknowledge the partial support from CNPq: F. S. Abrahão (301.322/2020-1), K. Wehmuth (303.193/2020-4), and A. Ziviani (310.201/2019-5). Authors acknowledge the INCT in Data Science – INCT-CiD (CNPq 465.560/2014-8) and FAPERJ (E-26/203.046/2017).

Acknowledgments: We thank the anonymous reviewers of [1] for remarks and suggested improvements that led to the elaboration of the present article. We also thank Cristian Calude, Mikhail Prokopenko, and Gregory Chaitin for suggestions and directions on related topics investigated in this article.

References

1. Abrahão, F.S.; Wehmuth, K.; Zenil, H.; Ziviani, A. An Algorithmic Information Distortion in Multidimensional Networks. Complex Networks & Their Applications IX; Benito, R.M.; Cherifi, C.; Cherifi, H.; Moro, E.; Rocha, L.M.; Sales-Pardo, M., Eds.; Springer International Publishing: Cham, 2021; Vol. 944, *Studies in Computational Intelligence*, pp. 520–531, http://link.springer.com/10.1007/978-3-030-65351-4_42.
2. Mowshowitz, A.; Dehmer, M. Entropy and the complexity of graphs revisited. *Entropy* **2012**, *14*, 559–570. doi:10.3390/e14030559.
3. Zenil, H.; Kiani, N.; Tegnér, J. A Review of Graph and Network Complexity from an Algorithmic Information Perspective. *Entropy* **2018**, *20*, 551. doi:10.3390/e20080551.
4. Morzy, M.; Kajdanowicz, T.; Kazienko, P. On Measuring the Complexity of Networks: Kolmogorov Complexity versus Entropy. *Complexity* **2017**, *2017*, 1–12. doi:10.1155/2017/3250301.

5. Zenil, H.; Kiani, N.A.; Zea, A.A.; Tegnér, J. Causal deconvolution by algorithmic generative models. *Nature Machine Intelligence* **2019**, *1*, 58–66. doi:10.1038/s42256-018-0005-0.
6. Zenil, H.; Kiani, N.A.; Abrahão, F.S.; Rueda-Toicen, A.; Zea, A.A.; Tegnér, J. Minimal Algorithmic Information Loss Methods for Dimension Reduction, Feature Selection and Network Sparsification. *arXiv Preprints* **2020**. https://arxiv.org/abs/1802.05843v8.
7. Zenil, H.; Kiani, N.A.; Tegnér, J. Quantifying loss of information in network-based dimensionality reduction techniques. *Journal of Complex Networks* **2016**, *4*, 342–362. doi:10.1093/comnet/cnv025.
8. Zenil, H.; Soler-Toscano, F.; Dingle, K.; Louis, A.A. Correlation of automorphism group size and topological properties with program-size complexity evaluations of graphs and complex networks. *Physica A: Statistical Mechanics and its Applications* **2014**, *404*, 341–358. doi:10.1016/j.physa.2014.02.060.
9. Buhrman, H.; Li, M.; Tromp, J.; Vitányi, P. Kolmogorov Random Graphs and the Incompressibility Method. *SIAM Journal on Computing* **1999**, *29*, 590–599. doi:10.1137/S0097539797327805.
10. Zenil, H.; Kiani, N.A.; Tegnér, J. The Thermodynamics of Network Coding, and an Algorithmic Refinement of the Principle of Maximum Entropy. *Entropy* **2019**, *21*, 560. doi:10.3390/e21060560.
11. Santoro, A.; Nicosia, V. Algorithmic Complexity of Multiplex Networks. *Physical Review X* **2020**, *10*, 021069. doi:10.1103/PhysRevX.10.02106
12. Chaitin, G. *Algorithmic Information Theory*, 3 ed.; Cambridge University Press, 2004.
13. Calude, C.S. *Information and Randomness: An algorithmic perspective*, 2 ed.; Springer-Verlag, 2002.
14. Downey, R.G.; Hirschfeldt, D.R. *Algorithmic Randomness and Complexity*; Theory and Applications of Computability, Springer New York: New York, NY, 2010; http://doi.org/10.1007/978-0-387-68441-3.
15. Li, M.; Vitányi, P. *An Introduction to Kolmogorov Complexity and Its Applications*, 4 ed.; Texts in Computer Science, Springer: Cham, 2019; http://doi.org/10.1007/978-3-030-11298-1.
16. Bollobás, B. *Modern graph theory*; Graduate texts in mathematics, Springer Science & Business Media, 1998; p. 394.
17. Brandes, U.; Erlebach, T. Fundamentals. In *Network Analysis*; Brandes, U.; Erlebach, T., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2005; Vol. 3418, *Lecture Notes in Computer Science*, pp. 7–15. doi:10.1007/978-3-540-31955-9_2.
18. Diestel, R. *Graph Theory*, 5 ed.; Vol. 173, *Graduate Texts in Mathematics*, Springer-Verlag Berlin Heidelberg, 2017; p. 428, https://www.springer.com/br/book/9783662536216.
19. Wehmuth, K.; Fleury, É.; Ziviani, A. On MultiAspect graphs. *Theoretical Computer Science* **2016**, *651*, 50–61. doi:10.1016/j.tcs.2016.08.017.
20. Wehmuth, K.; Fleury, É.; Ziviani, A. MultiAspect Graphs: Algebraic Representation and Algorithms. *Algorithms* **2017**, *10*, 1–36. doi:10.3390/a10010001.
21. Abrahão, F.S.; Wehmuth, K.; Zenil, H.; Ziviani, A. On incompressible multidimensional networks. *arXiv Preprints* **2018**. http://arxiv.org/abs/1812.01170.
22. Harary, F. *Graph Theory*; Addison Wesley series in mathematics, CRC Press, 2018; https://www.taylorfrancis.com/books/9780429493768.
23. Boccaletti, S.; Bianconi, G.; Criado, R.; del Genio, C.; Gómez-Gardeñes, J.; Romance, M.; Sendiña-Nadal, I.; Wang, Z.; Zanin, M. The structure and dynamics of multilayer networks. *Physics Reports* **2014**, *544*, 1–122. doi:10.1016/j.physrep.2014.07.001.
24. Berlingerio, M.; Coscia, M.; Giannotti, F.; Monreale, A.; Pedreschi, D. Foundations of Multidimensional Network Analysis. 2011 International Conference on Advances in Social Networks Analysis and Mining. IEEE, 2011, pp. 485–489, http://ieeexplore.ieee.org/document/5992618/.
25. Kivela, M.; Arenas, A.; Barthelemy, M.; Gleeson, J.P.; Moreno, Y.; Porter, M.A. Multilayer networks. *Journal of Complex Networks* **2014**, *2*, 203–271. doi:10.1093/comnet/cnu016.
26. Cozzo, E.; de Arruda, G.F.; Rodrigues, F.A.; Moreno, Y. *Multiplex Networks*; SpringerBriefs in Complexity, Springer International Publishing: Cham, 2018; https://link.springer.com/book/10.1007/978-3-319-92255-3.
27. De Domenico, M.; Solé-Ribalta, A.; Cozzo, E.; Kivelä, M.; Moreno, Y.; Porter, M.A.; Gómez, S.; Arenas, A. Mathematical Formulation of Multilayer Networks. *Physical Review X* **2013**, *3*, 041022. doi:10.1103/PhysRevX.3.041022.
28. Rossetti, G.; Cazabet, R. Community Discovery in Dynamic Networks. *ACM Computing Surveys* **2018**, *51*, 1–37. doi:10.1145/3172867.
29. Michail, O. An Introduction to Temporal Graphs: An Algorithmic Perspective. In *Algorithms, Probability, Networks, and Games*; Zaroliagis, C.; Pantziou, G.; Kontogiannis, S., Eds.; Springer International Publishing: Cham, 2015; Vol. 9295, *Lecture Notes in Computer Science*, pp. 308–343. doi:10.1007/978-3-319-24024-4_18.
30. Pan, R.K.; Saramäki, J. Path lengths, correlations, and centrality in temporal networks. *Physical Review E* **2011**, *84*, 016105. doi:10.1103/PhysRevE.84.016105.
31. Costa, E.C.; Vieira, A.B.; Wehmuth, K.; Ziviani, A.; da Silva, A.P.C. Time Centrality in Dynamic Complex Networks. *Advances in Complex Systems* **2015**, *18*. doi:10.1142/S021952591550023X.
32. Wehmuth, K.; Costa, B.; Bechara, J.V.; Ziviani, A. A Multilayer and Time-Varying Structural Analysis of the Brazilian Air Transportation Network. Latin America Data Science Workshop (LADAS) co-located with 44th International Conference on Very Large Data Bases (VLDB 2018), 2018, pp. 57–64, http://ceur-ws.org/Vol-2170/paper8.pdf.
33. Wehmuth, K.; Ziviani, A. Avoiding Spurious Paths in Centralities Based on Shortest Paths in High Order Networks. 2018 Eighth Latin-American Symposium on Dependable Computing (LADC). IEEE, 2018, pp. 19–26, https://ieeexplore.ieee.org/document/8671613/.
34. Calude, C.S. Borel Normality and Algorithmic Randomness. In *Developments in Language Theory*; World Scientific Publishing, 1994; pp. 113–129. doi:10.1142/9789814534758.

35. Abrahão, F.S.; Wehmuth, K.; Zenil, H.; Ziviani, A. Algorithmic information and incompressibility of families of multidimensional networks. *arXiv Preprints* **2020**. Research report no. 8/2018, National Laboratory for Scientific Computing (LNCC), Petrópolis, Brazil. https://arxiv.org/abs/1810.11719v8.
36. Newman, M. *Networks: an introduction*; Oxford University Press, 2010; http://doi.org/10.1093/acprof:oso/9780199206650.001.0001.
37. Khoussainov, B. A quest for algorithmically random infinite structures. Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) - CSL-LICS; ACM Press: New York, New York, USA, 2014; pp. 1–9, https://doi.org/10.1145/2603088.2603114.