AN ALGORITHMICALLY RANDOM FAMILY OF MULTIASPECT GRAPHS AND ITS TOPOLOGICAL PROPERTIES

FELIPE S. ABRAHÃO, KLAUS WEHMUTH, ARTUR ZIVIANI

National Laboratory for Scientific Computing (LNCC)
25651-075 – Petropolis, RJ – Brazil

ABSTRACT. This article presents a theoretical investigation of incompressibility and randomness in generalized representations of graphs along with its implications on network topological properties. We extend previous studies on plain algorithmically random classical graphs to plain and prefix algorithmically random MultiAspect Graphs (MAGs). First, we show that there is an infinite recursively labeled infinite family of nested MAGs (or, as a particular case, of nested classical graphs) that behaves like (and is determined by) an algorithmically random real number. Then, we study some of their important topological properties, in particular, vertex degree, connectivity, diameter, and rigidity.

1

1. Introduction

In this article, we study algorithmic randomness or incompressibility of generalized representations of graphs. With this purpose, we apply theoretical tools from algorithmic information theory [10, 14, 20, 23, 25] to generalized graphs which represent dyadic (or 2-place) relations between two arbitrary n-ary tuples [31, 32]. In the context of measuring deterministic information content and its complexity, the general problem that we tackle is to establish equivalences between the algorithmic information of generalized graphs and the algorithmic information of strings. Thereafter, we aim at investigating network topological properties of algorithmically random generalized graphs.

Measuring the information content of graphs or networks by statistic-informational[1] tools, e.g., Shannon entropy related measures, is one of the current subjects of increasing importance in network modeling and network analysis [5, 17, 18, 24, 26, 28, 30]. Furthermore, the study of topological properties of graphs (or networks) defined on stochastic random processes (e.g., Erdős–Rényi random graph) has been of central importance to graph theory [2, 4], complex networks theory [1, 3, 22], or in the broad field of network science [7, 24]. As already pointed in [9], many of the topological properties we study here are indeed statistically expected to hold for some stochastic-randomly generated graphs. On the other hand, headed by algorithmic complexity and algorithmic randomness [10, 13, 20, 25], we have the study of randomness of fixed (finite or infinite) objects and of information content measures for deterministic (i.e., computable) generation processes [11, 23, 35]. Therefore, in the context of graphs or networks, algorithmic information theory (also known as 'Kolmogorov complexity' theory or 'Solomonoff-Kolmogorov-Chaitin complexity' theory) has presented theoretical and empirical tools in order to investigate relationships between algorithmic complexity and properties of graphs or complex networks [8, 9, 15, 25, 28, 36–38].

Here, we follow this second line of research by a theoretical investigation of algorithmic complexity and algorithmic randomness in generalized graph representations. In this work, we present definitions, lemmas, theorems, and corollaries. Our results are based on a formalization of generalized graphs, called *MultiAspect Graphs* (MAGs), as presented in [31, 32]. These MAGs are formal representations of dyadic (or 2-place) relationships between two arbitrary n-ary tuples. In this way, it has been shown that MAGs enable one to formally represent and computationally analyze high order networks, e.g., dynamic networks [16, 34] or dynamic multilayer networks [33]. In addition, this article is mainly based on previous applications of the incompressibility method to classical labeled graphs [9, 25]. Thus, the general scope of this work is not only algorithmic randomness of MAGs, but also the implications of incompressibility on high order networks' topological properties.

In this article, our main goals are to: study recursive labeling in MAGs; show that the algorithmic information content carried by MAG is well-defined; show that there are recursively labeled infinite families of MAGs (and, consequentially, also of classical graphs) that behave like algorithmically random real numbers; present some topological properties of such MAGs and families of MAGs, in particular, vertex degree, connectivity, diameter, and rigidity.

---

[1] Or probabilistic-informational.

To this end, we base our definitions and notations on previous work in algorithmic information theory[2] [20, 25], graph theory[3] [2, 6, 19], MAGs[4] [31, 32], and algorithmically random classical graphs[5] [9, 25]. In this sense, the main idea behind our results derives from a standard application of known inequalities in algorithmic information theory to graph and MAG representation formalisms. Thus, the present article is as a proper extension of the results on algorithmically random classical graphs in [9, 25] to MAGs.

In Section 2, we define *recursively labeled* MAGs and show how such mathematical objects are determined by the algorithmic information of arbitrarily chosen binary strings.[6] In fact, unlike classical graphs, the algorithmic information of a MAG and the string that determines its (composite) edge set may be not so tightly associated regarding (plain or prefix) algorithmic complexity and mutual algorithmic information.[7] However, once we define *recursively labeled (finite or infinite) families* of MAGs in Section 2.1, we see that, in this case, both become algorithmically equivalent.[8] This recovers the property of a binary string determining the presence or absence of a edge, as we previously had for classical graphs [9, 25].

In Section 3, we introduce[9] prefix algorithmic randomness (i.e., K-randomness) for MAGs and show[10] that there are infinite families of MAGs (or classical graphs) in which every member is incompressible (i.e., weakly K-random) regarding prefix algorithmic complexity (i.e., K-complexity). In addition, we show in Section 3.1 that there are recursively labeled infinite families[11] of MAGs in which a member is a *MultiAspect subGraph* (subMAG) of the other. That is, such families are defined by an infinite sequence of MAGs such that the former is always a subMAG of the latter. Therefore, one can obtain[12] a *recursively labeled infinite nested family* of MAGs that is as prefix algorithmically random (i.e., K-random or 1-random) as a prefix algorithmically random real number[13], like the halting probability.

In Section 4, we relate these results on prefix algorithmic randomness with plain algorithmic randomness of MAGs in a manner directly analogous to plain algorithmic randomness of classical graphs in [9, 25].[14] Thus, as we show[15] in Section 5, this enables one to extend previous results on network topological properties in [9, 25] to plain algorithmically random MAGs or prefix algorithmically random nested families of MAGs.[16]

---

[2] See Sections 1.1.2 and 1.2.2 .

[3] See Section 1.1.1.

[4] See Sections 1.1.1 and 1.2.1.

[5] See Sections 1.1.3 and 1.2.3.

[6] See Definition 2.1 and Lemma 2.2.

[7] See Lemma 2.2.

[8] See Definition 2.2 and Corollary 2.3.1.

[9] See Definition 3.1.

[10] See Lemma 3.1.

[11] See Definition 3.3.

[12] See Theorem 3.1.

[13] Which is univocally represented by an infinite binary string

[14] See Definition 4.1 and Theorem 4.1.

[15] See Theorem 5.1.

[16] See Corollaries 5.1.1 and 5.1.3.

1.1. **Preliminary definitions and notations.**

1.1.1. *Graphs and MultiAspect Graphs.* We base our notation regarding classical graphs and MultiAspect Graphs directly on $[9, 25, 31, 32]$. In order to avoid ambiguities, minor differences in the notation from $[31, 32]$ will be introduced in this section.

Notation 1.1.1. Let $(\,.\,,.\,)$ denote an *ordered pair* which is defined by the cartesian product $\times$ of two sets with cardinality 1 each. Thus, the union of all these ordered pairs is the cartesian product of two sets X and Y where

$$x \in X \land y \in Y \iff (x, y) \in X \times Y$$

Notation 1.1.2. Let $\{\,.\,,.\,\}$ denote a *unordered pair* which is set with cardinality 2.[17]

Definition 1.1.1. A *labeled* (directed or undirected) *graph* $G = (V, E)$ is defined by an ordered pair (V, E), where $V = \{1, \ldots, n\}$ is the finite set of labeled vertices with $n \in \mathbb{N}$ and E is the edge set such that

$$E \subseteq V \times V$$

Note 1.1.1.1. If a labeled graph G does not contain *self-loops*[18], i.e., for every $x \in V$,

$$(x, x) \notin E \;,$$

then we say G is a *traditional directed graph.*

Definition 1.1.1.1. A labeled *undirected* graph $G = (V, E)$ without *self-loops* is a labeled graph with a restriction in the edge set E such that each edge is an *unordered pair* with

$$E \subseteq \{\{x, y\} \mid x, y \in V\} = \mathbb{E}_c\,(G)\ \text{``}\subsetneq\text{''}\ V \times V$$

where[19] there is $Y \subseteq V \times V$ such that

$$\{x, y\} \in E \subseteq \mathbb{E}_c\,(G) \iff (x, y) \in Y \land (y, x) \in Y$$

We also refer to these graphs as *classical graphs.*

Note 1.1.1.2. For the present purposes of this article, and as classically found in the literature, all graphs G will be classical graphs.

Notation 1.1.1.1. Let $V(G)$ denote the set of vertices of G.

Notation 1.1.1.2. Let $|V(G)|$ denote the cardinality of the set of vertices in G.

Notation 1.1.1.3. Let $E(G)$ denote the edge set of G.

Notation 1.1.1.4. Let $|E(G)|$ denote the cardinality of the edge set in G.

Definition 1.1.2. As in $[2, 6, 19]$, we say a graph G' is a *subgraph* of a graph G, denoted as $G' \subseteq G$, *iff*

$$V\,(G') \subseteq V\,(G) \,\land\, E\,(G') \subseteq E\,(G)$$

---

[17] That is, it is a unordered pair is a multiset with cardinality 2 where the multiplicity of each element is 1.

[18] That is, there is no edge or arrow linking the same vertex to itself.

[19] That is, the adjacency matrix of this graph is symmetric.

Definition 1.1.2.1. We say a graph G' is a *vertex-induced subgraph* of G *iff*

$$V(G') \subseteq V(G)$$

and, for every $u, v \in V(G')$,

$$(u,v) \in E(G) \implies (u,v) \in E(G')$$

In addition, we denote this G' as $G[V(G')]$.

As defined in [31,32], we may generalize the notion of graph in order to represent dyadic (or 2-place) relations between n-ary tuples:

Definition 1.1.3. Let $\mathscr{G} = (\mathscr{A}, \mathscr{E})$ be a MultiAspect Graph (MAG), where $\mathscr{E}$ is the set of existing composite edges of the MAG and $\mathscr{A}$ is a class of sets, each of which is an *aspect*. Each aspect $\sigma \in \mathscr{A}$ is a finite set and the number of aspects $p = |\mathscr{A}|$ is called the *order* of $\mathscr{G}$. By an immediate convention, we call a MAG with only one aspect as a *first order* MAG, a MAG with two aspects as a *second order* MAG and so on. Each composite edge (or arrow) $e \in \mathscr{E}$ may be denoted by an ordered $2p$-tuple $(a_1, \ldots, a_p, b_1, \ldots, b_p)$, where a_i, b_i are elements of the i-th aspect with $1 \le i \le p = |\mathscr{A}|$.

Note 1.1.3.1. Thus, the aspects in $\mathscr{A}$ determine which variant a graph $\mathscr{G}$ will be (and how the set $\mathscr{E}$ will be defined). For example, a time-varying graph as in [16,34] or a multilayered graph as in [33].

Notation 1.1.3.1. $\mathscr{A}(\mathscr{G})$ denotes the set of aspects of $\mathscr{G}$ and $\mathscr{E}(\mathscr{G})$ denotes the *composite edge set* of $\mathscr{G}$.

Notation 1.1.3.2. We denote the i-th aspect of $\mathscr{G}$ as $\mathscr{A}(\mathscr{G})[i]$. So, $|\mathscr{A}(\mathscr{G})[i]|$ denotes the number of elements in $\mathscr{A}(\mathscr{G})[i]$. In order to match the classical graph case, we adopt the convention of calling the elements of the first aspect of a MAG as *vertices*. Therefore, we denote the set $\mathscr{A}(\mathscr{G})[1]$ of elements of the first aspect of a MAG $\mathscr{G}$ as $V(\mathscr{G})$. Thus, a vertex should not be confused with a composite vertex (see Notation 1.1.3.3).

Notation 1.1.3.3. The set of all *composite vertices* $\mathbf{v}$ of $\mathscr{G}$ is denoted by

$$\mathbb{V}(\mathscr{G}) = \underset{i=1}{\overset{p}{\times}} \mathscr{A}(\mathscr{G})[i]$$

and the set of all *composite edges* e of $\mathscr{G}$ is denoted by

$$\mathbb{E}(\mathscr{G}) = \underset{n=1}{\overset{2p}{\times}} \mathscr{A}(G)[(n-1) \pmod{p} + 1)] \,,$$

so that, for every ordered pair $(\mathbf{u}, \mathbf{v})$ with $\mathbf{u}, \mathbf{v} \in \mathbb{V}(\mathscr{G})$, we have $(\mathbf{u}, \mathbf{v}) = e \in \mathbb{E}(\mathscr{G})$. Also, for every $e \in \mathbb{E}(\mathscr{G})$ we have $(\mathbf{u}, \mathbf{v}) = e$ such that $\mathbf{u}, \mathbf{v} \in \mathbb{V}(\mathscr{G})$. Thus,

$$\mathscr{E}(\mathscr{G}) \subseteq \mathbb{E}(\mathscr{G})$$

Note 1.1.3.2. The terms *vertex* and *node* may be employed interchangeably in this article. However, we choose to use the term *node* preferentially within the context of networks, where nodes may realize operations, computations or would have some kind of agency, like in real networks. Thus, we choose to use the term *vertex* preferentially in the mathematical context of graph theory.

Definition 1.1.4. We denote the *companion tuple* of a MAG $\mathscr{G}$ as defined in [31,32] by $\tau(\mathscr{G})$ where

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

Notation 1.1.4.1. Let $\langle \tau(\mathscr{G}) \rangle$ denote the string $\langle |\mathscr{A}(\mathscr{G})[1]|, \ldots, |\mathscr{A}(\mathscr{G})[p]| \rangle$. See also Notation 1.1.10.

Definition 1.1.4.1. We define the *composite diameter* of $\mathscr{G}$ in an analogous way to diameter in classical graphs, which is defined as the maximum shortest path length. Thus, we define the composite diameter $\mathrm{D}_{\mathscr{E}}(\mathscr{G})$ as the maximum value in the set of the minimum number of steps (or hops) in $\mathscr{E}(\mathscr{G})$ necessary to reach a composite vertex $\mathbf{v}$ from a composite vertex $\mathbf{u}$, for any $\mathbf{u}, \mathbf{v} \in \mathbb{V}(\mathscr{G})$. See also [32] for paths and distances in MAGs.

Definition 1.1.5. As in [31,32], a sub-determination is a generalization of the aggregation concept applied to time-varying and multilayer graphs, in which all layers can be aggregated, resulting in a traditional graph. We define a sub-determination ζ of a MAG $\mathscr{G}$ as a partition of the set $\mathbb{V}(\mathscr{G})$ into equivalence classes (in respect to $\mathscr{E}(\mathscr{G})$) taking into account only a sublist of the $|\mathscr{A}(\mathscr{G})|$ aspects. Therefore, a sub-determination ζ generates another MAG $\mathscr{G}_\zeta = (\mathscr{A}_\zeta(\mathscr{G}), \mathscr{E}_\zeta(\mathscr{E}(\mathscr{G})))$ such that

$$\mathscr{E}_\zeta: \qquad \mathbb{E}(\mathscr{G}) \qquad \rightarrow \qquad \mathbb{E}_\zeta(\mathscr{G})$$
$$(a_1, \ldots, a_p, b_1, \ldots, b_p) \mapsto (a_{\zeta_1}, \ldots, a_{\zeta_{p_\zeta}}, b_{\zeta_1}, \ldots, b_{\zeta_{p_\zeta}})$$

where

$$\mathbb{E}_\zeta(\mathscr{G}) = \underset{n=1}{\overset{2p_\zeta}{\times}} \mathscr{A}_\zeta(G)[(n-1) \pmod{p_\zeta} + 1)]$$

and $(\zeta_1, \ldots, \zeta_{p_\zeta})$ is a subsequence of $(1, \ldots, p)$ with $|\mathscr{A}_\zeta(\mathscr{G})| = p_\zeta$.

Definition 1.1.6. As in [31], we say a MAG $\mathscr{G}$ without self-loops (i.e., for every $\mathbf{u} \in \mathbb{V}(\mathscr{G})$, $(\mathbf{u}, \mathbf{u}) \notin \mathscr{E}(\mathscr{G})$) is isomorphic to a traditional directed graph G when there is a bijective function $f : \mathbb{V}(\mathscr{G}) \rightarrow V(G)$ such that an edge $e \in \mathscr{E}(\mathscr{G})$ if, and only if, the edge $(f(\pi_o(e)), f(\pi_d(e))) \in E(G)$ where π_o is a function that returns the origin vertex of an edge and the function π_d is a function that returns the destination vertex of an edge.

Definition 1.1.7. We define an *undirected* MAG $\mathscr{G}_c = (\mathscr{A}, \mathscr{E})$ without *self-loops* as a restriction $\mathbb{E}_c$ in the set of all composite edges $\mathbb{E}$ such that

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

where[20] there is $Y \subseteq \mathbb{E}(\mathscr{G}_c)$ such that

$$\{\mathbf{u}, \mathbf{v}\} \in \mathscr{E}(\mathscr{G}_c) \iff (\mathbf{u}, \mathbf{v}) \in Y \wedge (\mathbf{v}, \mathbf{u}) \in Y$$

And we will have directly from this definition that

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

Note 1.1.7.1. We refer to these MAGs $\mathscr{G}_c$ in Definition 1.1.7 as *simplified* MAGs.

Note 1.1.7.2. Note that a classical graph G, as in Definition 1.1.1.1, is a labeled first order $\mathscr{G}_c$ with $\mathbb{V}(\mathscr{G}_c) = \{1, \ldots, |\mathbb{V}(\mathscr{G}_c)|\}$.

---

[20] That is, the adjacency matrix of this graph is symmetric.

Definition 1.1.8. As defined in [16, 34], let[21] $G_t = (V, \mathscr{E}, T)$ be a second order MAG representing *Time-Varying Graph* (TVG), where V is the set of vertices, T is the set of time instants, and $\mathscr{E} \subseteq V \times T \times V \times T$ is the set of edges.

Definition 1.1.8.1. We define the set of time instants of the graph $G_t = (V, \mathscr{E}, T)$ as $T(G_t) = \{t_0, t_1, \ldots, t_{|T(G_t)|-1}\}$.

Note 1.1.8.1. Let $V(G_t)$ denote the set of vertices of G_t.

Note 1.1.8.2. Let $|V(G_t)|$ be the cardinality of the set of vertices in G_t.

Note 1.1.8.3. We adopt the convention that there is a natural ordering for $T(G_t)$ such that

$$\forall i \in \mathbb{N} \ (\ 0 \le i \le |T(G_t)| - 1 \implies t_i = i + 1 \ )$$

.

Definition 1.1.8.2. We define a *trans-temporal* edge as an edge $e = (u, t_i, v, t_j) \in \mathscr{E}(G_t)$ with $j \neq i + 1$ and $j \neq i$.

1.1.2. *Turing machines and algorithmic information theory.* In this section, we present notations and definitions regarding algorithmic information theory and its formalization on Turing machines. For a complete introduction to these notations and definitions, see [9, 20, 25].

Notation 1.1.3. Let $\lg(x)$ denote the binary logarithm $\log_2(x)$.

Notation 1.1.4. Let $\{0, 1\}^*$ be the set of all finite binary strings.

Notation 1.1.5. Let $l(x)$ denote the length of a finite string $x \in \{0, 1\}^*$.[22] In addition, let $|X|$ denote the number of elements (i.e., the cardinality) in a set, if X is a finite set.

Notation 1.1.6. Let $(x)_2$ denote the string which is a binary represenation of the number x. In addition, let $(x)_L$ denote the representation of the number $x \in \mathbb{N}$ in language L.

Notation 1.1.7. 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, $x \upharpoonright_n = x_1 x_2 \ldots x_n \equiv (x_1, x_2, \ldots, x_n)$ where $(x)_2 = y.x_1 x_2 \ldots x_n x_{n+1} \ldots$, and $x_1 x_2 \ldots x_n, y \in \{0, 1\}^*$.

Notation 1.1. Let $\mathbf{U}(x)$ denote the output of a universal Turing machine $\mathbf{U}$ when x is given as input in its tape. Thus, $\mathbf{U}(x)$ denotes a *partial recursive* function

$$\varphi_{\mathbf{U}} : L \to L \ , \\ x \mapsto y = \varphi_{\mathbf{U}}(x)$$

where L is a language. In particular, $\varphi_{\mathbf{U}}(x)$ is a *universal* partial recursive function [25, 29]. Note that, if x is a non-halting program on $\mathbf{U}$, then this function $\mathbf{U}(x)$ is undefined for x.

---

[21] It can be equivalently denoted as $G_t = (V, T, \mathscr{E})$.

[22] In [20], $l(x)$ is denoted by $|x|$.

Notation 1.1.1. Wherever $n \in \mathbb{N}$ or $n \in \{0,1\}^*$ appears in the domain or in the codomain of a partial (or total) recursive function

$$\varphi_{\mathcal{U}}: L \to \quad L \quad ,$$
$$x \mapsto y = \varphi_{\mathcal{U}}(x)$$

where $\mathcal{U}$ is a Turing machine, running on language L, it actually denotes

$$(n)_L$$

Notation 1.1.8. Let $\mathbf{L_U}$ denote a binary universal programming language for a universal Turing machine $\mathbf{U}$.

Notation 1.1.9. Let $\mathbf{L'_U}$ denote a binary *prefix-free* (or *self-delimiting*) universal programming language for a prefix universal Turing machine $\mathbf{U}$.[23]

Notation 1.1.10. As in [20, 25], let $\langle \cdot, \cdot \rangle$ denote an arbitrary recursive bijective pairing function. This notation can be recursively extended to $\langle \cdot, \langle \cdot, \cdot \rangle \rangle$ and, then, to an ordered tuple $\langle \cdot, \cdot, \cdot \rangle$. This iteration can be recursively applied with the purpose of defining finite ordered tuples $\langle \cdot, \dots, \cdot \rangle$.

Definition 1.1.9. As in [20, 25], the (unconditional) *plain algorithmic complexity* (also known as C-complexity, plain Kolmogorov complexity, plain program-size complexity or plain Solomonoff-Komogorov-Chaitin complexity) of a finite binary string w, denoted by $C(w)$, is the length of the shortest program $w^* \in \mathbf{L_U}$ such that $\mathbf{U}(w^*) = w$.[24] The *conditional* plain algorithmic complexity of a binary finite string y given a binary finite string x, denoted by $C(y|x)$, is the length of the shortest program $w^* \in \mathbf{L_U}$ such that $\mathbf{U}(\langle x, w^* \rangle) = w$. Note that $C(y) = C(y|\epsilon)$, where ϵ is the empty string. We also have the *joint* plain algorithmic complexity of strings x and y denoted by $C(x,y) := C(\langle x, y \rangle)$ and the *C-complexity of information* in x about y denoted by $I_C(x : y) := C(y) - C(y|x)$.

Note 1.1.9.1. For an edge set $\mathscr{E}(\mathscr{G})$, let $C(\mathscr{E}(\mathscr{G})) := C(\langle \mathscr{E}(\mathscr{G}) \rangle)$ denote

$$C(\langle \langle e_1, z_1 \rangle, \dots, \langle e_n, z_n \rangle \rangle)$$

such that

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

where $z_i \in \{0,1\}$ with $1 \le i \le n = |\mathbb{E}(\mathscr{G})|$. The same applies analogously to $C(E(G))$ and to the conditional, joint, and C-complexity of information case.

Definition 1.1.10. As in [20, 25], the (unconditional) *prefix algorithmic complexity* (also known as K-complexity, prefix Kolmogorov complexity, prefix program-size complexity or prefix Solomonoff-Komogorov-Chaitin complexity) of a finite binary string w, denoted by $K(w)$, is the length of the shortest program $w^* \in \mathbf{L'_U}$ such that $\mathbf{U}(w^*) = w$.[25] The *conditional* prefix algorithmic complexity of a binary finite string y given a binary finite string x, denoted by $K(y|x)$, is the length of the shortest

---

[23] Note that, although the same letter $\mathbf{U}$ is used in Notation 1.1.8, the two universal Turing machines may be different, since, in $\mathbf{L_U}$, the Turing machine does not need to be prefix-free. Thus, every time the domain of function $\mathbf{U}(x)$ is in $\mathbf{L_U}$, $\mathbf{U}$ denotes an arbitrary universal Turing machine. Analogously, every time the domain of function $\mathbf{U}(x)$ is in $\mathbf{L'_U}$, $\mathbf{U}$ denotes a *prefix* universal Turing machine. If $\mathbf{L'_U}$ or $\mathbf{L_U}$ are not being specified, then assume an arbitrary universal Turing machine.

[24] w^* denotes the lexicographically first $\mathrm{p} \in \mathbf{L_U}$ such that $l(\mathrm{p})$ is minimum and $\mathbf{U}(p) = w$.

[25] w^* denotes the lexicographically first $\mathrm{p} \in \mathbf{L'_U}$ such that $l(\mathrm{p})$ is minimum and $\mathbf{U}(p) = w$.

program $w^* \in \mathbf{L'_U}$ such that $\mathbf{U}(\langle x, w^* \rangle) = w$. Note that $K(y) = K(y|\epsilon)$, where ϵ is the empty string. Similarly, we have the *joint* prefix algorithmic complexity of strings x and y denoted by $K(x, y) := K(\langle x, y \rangle)$, the *K-complexity of information* in x about y denoted by $I_K(x : y) := K(y) - K(y|x)$, and the *mutual algorithmic information* of two string x and y denoted by $I_A(x ; y) := K(y) - K(y|x^*)$.

Note 1.1.10.1. For an edge set $\mathscr{E}(\mathscr{G})$, let $K(\mathscr{E}(\mathscr{G})) := K(\langle \mathscr{E}(\mathscr{G}) \rangle)$ denote

$$K(\langle \langle e_1, z_1 \rangle, \ldots, \langle e_n, z_n \rangle \rangle)$$

such that

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

where $z_i \in \{0, 1\}$ with $1 \le i \le n = |\mathbb{E}(\mathscr{G})|$. The same applies analogously to $K(E(G))$ and to the conditional, joint, K-complexity of information, and mutual case.

Definition 1.1.11. Let $\Omega \in [0, 1] \subset \mathbb{R}$ denote the *halting probability* (also known as Chaitin's constant or Omega number). The halting probability is defined by

$$\Omega = \sum_{\substack{\exists y (\mathbf{U}(\mathrm{p})=y) \\ \text{where } \mathrm{p} \in \mathbf{L'_U}}} \frac{1}{2^{l(\mathrm{p})}}$$

Definition 1.1.12. We say a string $x \in \{0, 1\}^*$ is *weakly K-random* (*K-incompressible up to a constant, c-K-incompressible, prefix algorithmically random up to a constant* or *prefix algorithmically incompressible up to a constant*) if, and only if, for a fixed constant $d \in \mathbb{N}$,

$$K(x) \ge l(x) - d$$

Definition 1.1.13. We say a real number $x \in [0, 1] \subset \mathbb{R}$ is *1-random* (*K-incompressible up to a constant, K-random* or *prefix algorithmically random*) if, and only if, it satisfies

$$K(x \restriction_n) \ge n - \mathbf{O}(1)$$

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

Notation 1.1.13.1. In order to avoid ambiguities between plain and prefix algorithmic complexity and ambiguities in relation to randomness deficiencies, we choose to say that an algorithmically random real number in respect to prefix algorithmic complexity in Definition 1.1.13 is $\mathbf{O}(1)$-*K-random*.

Note 1.1.13.1. That is, a real number $x \in [0, 1] \subset \mathbb{R}$ is $\mathbf{O}(1)$-K-random *iff* it is weakly K-random for every initial segment $x \restriction_n$. See [20].

1.1.3. *Algorithmically random graphs.* Here, we restate the definition of a labeled graph that has a randomness deficiency at most $\delta(n)$ from [9, 25]:

Definition 1.1.14. A classical graph G with $|V(G)| = n$ is $\delta(n)$-random if, and only if, it satisfies

$$C(E(G)|n) \ge \binom{n}{2} - \delta(n)$$

where

$$\delta : \mathbb{N} \to \mathbb{N}$$
$$n \mapsto \delta(n)$$

Notation 1.1.14.1. In order to avoid ambiguities between plain and prefix algorithmic complexity, we choose to say that a $\delta(n)$-random graph G in Definition 1.1.14 is $\delta(n)$-*C-random*.

Definition 1.1.15. We say a classical graph is *rigid* if, and only if, its only automorphism is the identity automorphism.

1.2. **Background results.**

1.2.1. *MultiAspect Graphs.* This section restates some previous results in [31, 32]. First, it has been shown that a MAG is basically equivalent to a traditional directed graph [32].

Theorem 1.2.1. *For every MAG $\mathscr{G}$ without self-loops of order $p > 0$, where all aspects are non-empty sets, there is a unique (up to a graph isomorphism) traditional directed graph $G_{\mathscr{G}} = (V, E)$ with $|V(G)| = \prod_{n=1}^{p} |\mathscr{A}(\mathscr{G})[n]|$ that is isomorphic (as in Definition 1.1.6) to $\mathscr{G}$.*

As an immediate corollary of Theorem 1.2.1, we have that the same holds for the undirected case. To achieve a simple proof of that in Corollary 1.2.1.1, just note that any undirected MAG (or graph) can be equivalently represented by a directed MAG (or graph, respectively) in which, for every oriented edge (i.e., arrow), there must be an oriented edge in the exact opposite direction.[26] In other words, the adjacency matrix must be symmetric.[27]

Corollary 1.2.1.1. *For every MAG $\mathscr{G}_c$ (as in Definition 1.1.7) of order $p > 0$, where all aspects are non-empty sets, there is a unique (up to a graph isomorphism) classical graph $G_{\mathscr{G}_c} = (V, E)$ with $|V(G)| = \prod_{n=1}^{p} |\mathscr{A}(\mathscr{G}_c)[n]|$ that is isomorphic to $\mathscr{G}_c$.*

From these results, we also have that the concepts of *walk*, *trail*, and *path* become well-defined for MAGs analogously to within the context of graphs. See section 3.5 in [32].

1.2.2. *Algorithmic information theory.* We now restate some important relations in algorithmic information theory[28] [11, 14, 20, 23, 25]. The following results can be found in [12–14, 20, 21, 25, 27].

---

[26] Remember Notation 1.1.2.

[27] See also the proof of Lemma 2.2.

[28] Also known as Kolmogorov complexity or Solomonoff-Kolmogorov-Chaitin complexity.

Lemma 1.2.1. *For every $x, y \in \{0,1\}^*$ and $n \in \mathbb{N}$,*

(1) $$C(x) \leq l(x) + \mathbf{O}(1)$$

(2) $$K(x) \leq l(x) + \mathbf{O}(\lg(l(x)))$$

(3) $$C(y|x) \leq C(y) + \mathbf{O}(1)$$

(4) $$K(y|x) \leq K(y) + \mathbf{O}(1)$$

(5) $$C(y|x) \leq K(y|x) + \mathbf{O}(1) \leq C(y|x) + \mathbf{O}(\lg(C(y|x)))$$

(6) $$C(x) \leq C(x,y) + \mathbf{O}(1) \leq C(y) + C(x|y) + \mathbf{O}(\lg(C(x,y)))$$

(7) $$K(x) \leq K(x,y) + \mathbf{O}(1) \leq K(y) + K(x|y) + \mathbf{O}(1)$$

(8) $$C(x) \leq K(x) + \mathbf{O}(1)$$

(9) $$K(n) = \mathbf{O}(\lg(n))$$

(10) $$K(x) \leq C(x) + K(C(x)) + \mathbf{O}(1)$$

(11) $$I_A(x;y) = I_A(y;x) \pm \mathbf{O}(1)$$

Note 1.2.1.1. Note that the inverse relation $K(x,y) + \mathbf{O}(1) \geq K(y) + K(x|y) + \mathbf{O}(1)$ does not hold in general in Equation (7). In fact, one can show that $K(x,y) = K(y) + K(x \mid \langle y, K(y) \rangle) \pm \mathbf{O}(1)$, which is the key step to prove Equation (11).

Lemma 1.2.2. *Let $f_c \colon \mathbb{N} \to \mathbb{N}$ be a computable function, then*
$$n \mapsto f_c(n)$$

$$K(f_c(n)) \leq K(n) + \mathbf{O}(1)$$

One of the most important results in algorithmic information theory is the investigation and proper formalization of a mathematical theory for randomness [10, 20, 25]. This is what has allowed the previous results that we are extending, as restated in Section 1.2.3. In this article, we also choose to employ one of these important mathematical objects: the *halting probability* (see Definition 1.1.11). This is a widely known example of infinite binary sequence, or real number, that is algorithmically random in respect to prefix algorithmic complexity.

Theorem 1.2.2. *Let $n \in \mathbb{N}$. Then,*

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

That is, Ω is $\mathbf{O}(1)$-K-random.

Theorem 1.2.3. *Let $x \in [0,1] \subset \mathbb{R}$ be a real number. Then, the following are equivalent:*

(12) $$x \text{ is } \mathbf{O}(1)\text{-}K\text{-random}$$

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

(14) $$C(x \upharpoonright_n |n) \geq n - K(n) - \mathbf{O}(1)$$

1.2.3. *Algorithmically random graphs.* As pointed in the Introduction 1, our first goal in this article is to extended the results from [9, 25]. By defining algorithmically random graphs, the application of the incompressibility method to graph

theory generated fruitful lemmas and theorems with the purpose of studying diameter, connectivity, degree, statistics of subgraphs, unlabeled graphs counting, and automorphisms. In this section, we present some of these results.

Lemma 1.2.3. *A fraction of at least* $1 - \frac{1}{2^{\delta(n)}}$ *of all classical graphs G with $|V(G)| = n$ is $\delta(n)$-C-random.*

Lemma 1.2.4. *The degree $\mathbf{d}(v)$ of a vertex $v \in V(G)$ in a $\delta(n)$-C-random classical graph G with $|V(G)| = n$ satisfies*

$$\left| \mathbf{d}(v) - \left(\frac{n-1}{2} \right) \right| = \mathbf{O}\left(\sqrt{n\ (\delta(n) + \lg(n))} \right)$$

Lemma 1.2.5. *All $\mathbf{o}(n)$-C-random classical graphs G with $|V(G)| = n$ have $\frac{n}{4} + \mathbf{o}(n)$ disjoint paths of length 2 between each pair of vertices $u, v \in V(G)$. In particular, all $\mathbf{o}(n)$-C-random classical graphs G with $|V(G)| = n$ have diameter 2.*

Lemma 1.2.6. *Let $c \in \mathbb{N}$ be a fixed constant. Let G be a $(c \lg(n))$-C-random classical graph with $|V(G)| = n$. Let $X_{f(n)}(v)$ denote the set of the least $f(n)$ neighbors of a vertex $v \in V(G)$, where*

$$f : \mathbb{N} \to \ \mathbb{N}$$
$$n \mapsto f(n)$$

Then, for every vertices $u, v \in V(G)$,

$$\{u, v\} \in E(G)$$

or

$$\exists i \in V(G)(i \in X_{f(n)}(v) \wedge \{u, i\} \in E(G) \wedge \{i, v\} \in E(G))$$

with $f(n) \geq (c + 3) \lg(n)$.

Lemma 1.2.7. *If*

$$\delta(n) \leq n + \lg(n) + 2$$

then all $\delta(n)$-C-random classical graphs are rigid.

2. Recursively labeled MultiAspect Graphs

In this section, we will introduce a model of MultiAspect Graph (MAG) representation. First, we need to generalize the concept of a labeled graph in order to grasp the set of composite vertices. As with labeled graphs [9, 25, 28, 36, 38], where there is an enumeration of its vertices assigning a natural number to each one of them, we want that the edge set $\mathscr{E}$ continues to be uniquely (up to an automorphism) represented by a finite binary string. In fact, we will assume a more general condition than a fixed lexicographical ordering of the $|\mathbb{E}_c(\mathscr{G}_c)|$ edges. Thus, we will introduce MAGs that are *recursively labeled*.

We say that a MAG $\mathscr{G}_c$ from Definition 1.1.7 is recursively labeled if, and only if, there is an algorithm that, given the companion tuple $\tau(\mathscr{G}_c)$ (see Definition 1.1.4) as input, returns a recursive bijective ordering of composite edges $e \in \mathbb{E}_c(\mathscr{G}_c)$. More formally,

Definition 2.1. A MAG $\mathscr{G}_c$ (as in Definition 1.1.7) is *recursively labeled* given $\tau(\mathscr{G}_c)$ *iff* there are programs $\mathrm{p}_1, \mathrm{p}_2 \in \{0,1\}^*$ such that, for every $a_i, b_i \in \mathscr{A}(\mathscr{G}_c)[i]$ with $1 \le i \le p = |\mathscr{A}(\mathscr{G}_c)|$,

$$\text{(15)} \qquad \mathbf{U}\left(\langle\langle a_1, \ldots, a_p\rangle, \langle b_1, \ldots, b_p\rangle, \langle\langle \tau(\mathscr{G}_c)\rangle, \mathrm{p}_1\rangle\rangle\right) = (j)_2$$

$$\text{(16)} \qquad \mathbf{U}\left(\langle j, \langle\langle \tau(\mathscr{G}_c)\rangle, \mathrm{p}_2\rangle\rangle\right) = \langle\langle a_1, \ldots, a_p\rangle, \langle b_1, \ldots, b_p\rangle\rangle = (e_j)_2$$

where

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

Note that this Definition 2.1 can be easily extended to arbitrary MAGs as defined in 1.1.3, and not only undirected MAGs without self-loops. Also note that in the case of a first order $\mathscr{G}_c$, the usual notion of a labeled classical graph satisfies[29] Definition 2.1.

We can show that Definition 2.1 is satisfiable:

Lemma 2.1. *Any MAG $\mathscr{G}_c$ with $\mathscr{A}(\mathscr{G}_c)[i] \subset \mathbb{N}$, where $1 \le i \le p \in \mathbb{N}$, is recursively labeled given $\tau(\mathscr{G}_c)$ (i.e., it satisfies Definition 2.1).*

Proof. Let $\mathscr{G}_c$ be a MAG with $\mathscr{A}(\mathscr{G}_c)[i] \subset \mathbb{N}$, where $1 \le i \le p \in \mathbb{N}$. In this case, since $\langle \cdot, \cdot \rangle$ represents a recursive bijective pairing function, the companion tuple $\langle \tau(\mathscr{G}_c)\rangle = \langle |\mathscr{A}(\mathscr{G}_c)[1]|, \ldots, |\mathscr{A}(\mathscr{G}_c)[p]|\rangle$ univocally determines the value of p and the maximum value for each aspect.[30] Hence, one can always define a recursive lexicographical ordering $<_{\mathbb{V}}$ of the set $\{\langle x_1, \ldots, x_p\rangle \mid (x_1, \ldots, x_p) \in \mathbb{V}(\mathscr{G}_c)\}$ by starting at $\langle 1, \ldots, 1\rangle$ and, from a recursive iteration of this procedure from the right character to the left character, ordering all possible arrangements of the rightmost characters while one maintains the leftmost characters fixed, but respecting the limitations $|\mathscr{A}(\mathscr{G}_c)[i]|$ with $1 \le i \le p \in \mathbb{N}$. That is, from choosing an arbitrary well-known lexicographical ordering of ordered pairs, one can iterate this for a lexicographical ordering of n-tuples by $(x_1, (x_2, x_3)) = (x_1, x_2, x_3)$, $(x_1, (x_2, (x_3, x_4))) = (x_1, x_2, x_3, x_4)$, and so on. Alternatively, one may contruct the order relation $<_{\mathbb{V}}$ from functions $D(\mathbf{u}, \tau)$ and $N(d, i, \tau)$ defined in [31, Section 3.2: Ordering of Composite Vertices and Aspects, p. 9]. Therefore, from this recursive bijective ordering of composite vertices, we can now construct a sequence defined by a recursive bijective ordering $<_{\mathbb{E}_c}$ of the composite edges of a MAG $\mathscr{G}_c$. First, one build a sequence by applying a classical lexicographical ordering $<_{\mathbb{E}}$ to the set of pairs

$$\left\{ \langle\langle x_1, \ldots, x_p\rangle, \langle y_1, \ldots, y_p\rangle\rangle \mid (x_1, \ldots, x_p), (y_1, \ldots, y_p) \in \mathbb{V}(\mathscr{G}_c)\right\}$$

Then, one excludes the occurrence of self-loops and the second occurrence of symmetric pairs $(\langle y_1, \ldots, y_p\rangle, \langle x_1, \ldots, x_p\rangle)$, generating a subsequence of the previous sequence. Note that the procedure for determining whether the two composites vertices in an composite edge are equal or not is always decidable, so that self-loops on composite vertices will not return index values under order relation $<_{\mathbb{E}_c}$. Additionally, note that, since the sequence of composites edges was formerly arranged in lexicographical order relation $<_{\mathbb{E}}$, then, for every $a, b \in \mathbb{N}$ under order relation $<_{\mathbb{V}}$,

$$a <_{\mathbb{V}} b \implies (a, b) <_{\mathbb{E}} (b, a)$$

---

[29] From a standard application of a lexicographical ordering of the edges.

[30] See also [31] for more properties of the companion tuple regarding generalized graph representations.

This way, since subsequences preserve order, if $i_{(a,b)_{<_{\mathbb{E}}}}$ is the index value of the pair (a,b) in the sequence built under order relation $<_{\mathbb{E}}$ and $a <_{\mathbb{V}} b$, then

$$i_{(b,a)_{<_{\mathbb{E}_c}}} := i_{(a,b)_{<_{\mathbb{E}}}} \text{ and } i_{(a,b)_{<_{\mathbb{E}_c}}} := i_{(a,b)_{<_{\mathbb{E}}}}$$

Thus, p_1 is a fixed string that represents on a universal Turing machine the algorithm that, given $\tau(\mathscr{G}_c)$, $\langle a_1, \ldots, a_p \rangle$, and $\langle b_1, \ldots, b_p \rangle$ as inputs,

(i) builds the sequence of composite edges by the order relation $<_{\mathbb{E}_c}$ described before such that, for each step of this construction,

(a) search for $((a_1, \ldots, a_p), (b_1, \ldots, b_p))$ or $((b_1, \ldots, b_p), (a_1, \ldots, a_p))$ in this sequence;

(b) if one of these pairs is found, it returns the index value of the first one of these pairs found in this sequence;

(c) else, it continues building the sequence.

Note that one of these pairs must be always eventually found, since $\mathscr{A}(\mathscr{G}_c)[i] \subset \mathbb{N}$ with $1 \leq i \leq p = |\mathscr{A}(\mathscr{G}_c)| \in \mathbb{N}$. An analogous algorithm defines p_2, but searching for the j-th element in the sequence and returning the respective pair of tuples instead. $\qquad \square$

Furthermore, with this pair of programs p_1, p_2 and with $\langle \tau(\mathscr{G}_c) \rangle$, one can always build an algorithmic that, given a bit string $x \in \{0,1\}^*$ of length $|\mathbb{E}_c(\mathscr{G}_c)|$ as input, computes the composite edge set $\mathscr{E}(\mathscr{G}_c)$ and build another algorithm that, given the composite edge set $\mathscr{E}(\mathscr{G}_c)$ as input, returns the string x. Thus, this string x determines (up to an automorphism) the recursively labeled MAG $\mathscr{G}_c$. That is, it is a *representative string* of the MAG. This gives rise to the following Lemma:

Lemma 2.2. *Let* $x \in \{0,1\}^*$. *Let* $\mathscr{G}_c$ *be a recursively labeled MAG given* $\tau(\mathscr{G}_c)$ *(as in Definition 2.1) with* $l(x) = |\mathbb{E}_c(\mathscr{G}_c)|$ *such that, for every* $e \in \mathbb{E}_c(\mathscr{G}_c)$,

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

where $1 \leq j \leq l(x)$. *Then,*

(17) $\qquad C(\mathscr{E}(\mathscr{G}_c) \,|\, x) \leq K(\mathscr{E}(\mathscr{G}_c) \,|\, x) + \mathbf{O}(1) = K(\langle \tau(\mathscr{G}_c) \rangle) + \mathbf{O}(1)$

(18) $\qquad C(x \,|\, \mathscr{E}(\mathscr{G}_c)) \leq K(x \,|\, \mathscr{E}(\mathscr{G}_c)) + \mathbf{O}(1) = K(\langle \tau(\mathscr{G}_c) \rangle) + \mathbf{O}(1)$

(19) $\qquad\qquad\qquad\qquad\qquad K(x) = K(\mathscr{E}(\mathscr{G}_c)) \pm \mathbf{O}\left(K(\langle \tau(\mathscr{G}_c) \rangle)\right)$

(20) $\qquad I_A(x; \mathscr{E}(\mathscr{G}_c)) = I_A(\mathscr{E}(\mathscr{G}_c); x) \pm \mathbf{O}(1) = K(x) - \mathbf{O}\left(K(\langle \tau(\mathscr{G}_c) \rangle)\right)$

Proofs.

(proof of 17) First, remember notation of $\mathscr{E}(\mathscr{G})$ in Definitions 1.1.10 and 1.1.9 from which we have that

$$K(\langle \mathscr{E}(\mathscr{G}) \rangle) = K(\langle \langle e_1, z_1 \rangle, \ldots, \langle e_n, z_n \rangle \rangle)$$

such that

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

where $z_i \in \{0,1\}$ with $1 \leq i \leq n = |\mathbb{E}(\mathscr{G})|$. Thus, for MAGs $\mathscr{G}_c$ defined in Definition 1.1.7, we will have that[31]

$$K(\langle \mathscr{E}(\mathscr{G}_c) \rangle) = K(\langle \langle e_1, z_1 \rangle, \ldots, \langle e_n, z_n \rangle \rangle)$$

such that

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

where $z_i \in \{0,1\}$ with

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

We also have that, since $\mathscr{G}_c$ is a recursively labeled MAG, there is p_2 such that Equation (16) holds independently of the chosen companion tuple $\tau(\mathscr{G}_c)$. Let $\langle \tau(\mathscr{G}_c) \rangle$ be a self-delimiting string that encodes the companion tuple $\tau(\mathscr{G}_c)$. Let p be a binary string that represents on a universal Turing machine the algorithm that reads the companion tuple $\langle \tau(\mathscr{G}_c) \rangle$ as his first input and reads the string x as its second input[32]. Then, it reads each j-th bit of x, runs $\langle j, \langle \langle \tau(\mathscr{G}_c) \rangle, p_2 \rangle \rangle$ and, from the outputs of $\langle j, \langle \langle \tau(\mathscr{G}_c) \rangle, p_2 \rangle \rangle$, returns the string $\langle \langle e_1, z_1 \rangle, \ldots, \langle e_n, z_n \rangle \rangle$ where $z_j = 1$, if the j-th bit of x is 1, and $z_j = 0$, if the j-th bit of x is 0. Therefore, we will have that there is a binary string $\langle \langle \tau(\mathscr{G}_c) \rangle, p \rangle \in \mathbf{L}'_{\mathbf{U}}$ that represents an algorithm running on a prefix (or self-delimiting) universal Turing machine $\mathbf{U}$ that, given x as input, runs p taking also $\langle \tau(\mathscr{G}_c) \rangle$ as the first input. Since p_2 is fixed, we have that there is a self-delimiting binary encoding of $(\langle \tau(\mathscr{G}_c) \rangle, p)$ with

$$l\left(\langle \langle \tau(\mathscr{G}_c) \rangle, p \rangle\right) \leq K(\langle \tau(\mathscr{G}_c) \rangle) + \mathbf{O}(1)$$

Then, by the minimality of $K(\cdot)$, we will have that

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

The inequality $C(\mathscr{E}(\mathscr{G}_c) \,|\, x) \leq K(\mathscr{E}(\mathscr{G}_c) \,|\, x) + \mathbf{O}(1)$ follows directly from Lemma 1.2.1.

(proof of 18) This proof follows analogously to the proof of Equation (17), but using program p_1 instead of p_2 in order to build the string x from $\langle \mathscr{E}(\mathscr{G}_c) \rangle$.

(proof of 19) This proof follows analogously to the proof of Equation 7 in Lemma 1.2.1. Let p be a shortest self-delimiting description of $\langle \mathscr{E}(\mathscr{G}_c) \rangle$. From Equation 18, we know there is q, independent of the choice of p, such that it is a shortest self-delimiting description of x given $\langle \mathscr{E}(\mathscr{G}_c) \rangle$, where

$$K(x \,|\, \mathscr{E}(\mathscr{G}_c)) = K(\langle \tau(\mathscr{G}_c) \rangle) + \mathbf{O}(1)$$

Thus, there is string s, independent of the choice of p and q, that represents the algorithm running on a universal Turing machine that, given p and q

---

[31] In fact, the reader is invited to note that, since $\mathscr{G}_c$ is recursively labeled, this is equivalent to the case in which the adjacency matrix of $\mathscr{G}_c$ is symmetric. That is, let $\langle \mathscr{E}(\mathscr{G}) \rangle_{sym}$ denote $\langle \langle e_1, z_1 \rangle, \ldots, \langle e_{2n}, z_{2n} \rangle \rangle$ where the sequence $(e_1, \ldots, e_{2n})$ represents the adjacency matrix, which by assumption we know it is symmetric. Then, $K(\langle \mathscr{E}(\mathscr{G}) \rangle) = K(\langle \mathscr{E}(\mathscr{G}) \rangle_{sym}) \pm \mathbf{O}(K(\langle \tau(\mathscr{G}_c) \rangle))$. In the case of a recursively labeled family in Definition 2.2, this relation can be improved to $K(\langle \mathscr{E}(\mathscr{G}) \rangle) = K(\langle \mathscr{E}(\mathscr{G}) \rangle_{sym}) \pm \mathbf{O}(1)$.

[32] If, as in the proof of Lemma 20, the string x^* (i.e., a shortest self-delimiting description of x) is being given as input instead, then p reads the output of x^*.

as its inputs, calculates the output of p, runs q with the output of p as its input, and returns this last output. We will have that there is a prefix universal machine $\mathbf{U}$ in which $\langle p, q, s \rangle \in \mathbf{L}'_{\mathbf{U}}$ and, from Equation 18,

$$| \langle p, q, s \rangle | \leq K(\mathscr{E}(\mathscr{G}_c)) + K(x \,|\, \mathscr{E}(\mathscr{G}_c)) + \mathbf{O}(1) \leq$$
$$\leq K(\mathscr{E}(\mathscr{G}_c)) + K(\langle \tau(\mathscr{G}_c) \rangle) + \mathbf{O}(1)$$

Then, by the minimality of $K(\cdot)$, we will have that

$$K(x) \leq | \langle p, q, s \rangle | \leq K(\mathscr{E}(\mathscr{G}_c)) + K(\langle \tau(\mathscr{G}_c) \rangle) + \mathbf{O}(1)$$

Therefore,

$$K(x) \leq K(\mathscr{E}(\mathscr{G}_c)) + \mathbf{O}\left(K(\langle \tau(\mathscr{G}_c) \rangle)\right)$$

The proof of $K(\mathscr{E}(\mathscr{G}_c)) \leq K(x) + K(\langle \tau(\mathscr{G}_c) \rangle) + \mathbf{O}(1)$ follows in the same manner, but using Equation (17) instead.

(proof of 20) We have from Definition 1.1.10 that

(21) $$I_A(x \,; \mathscr{E}(\mathscr{G}_c)) = K(\mathscr{E}(\mathscr{G}_c)) - K(\mathscr{E}(\mathscr{G}_c) \,|\, x^*)$$

Now, we build a program for $\mathscr{E}(\mathscr{G}_c)$ given x^* almost identical to the one in the proof of Equation (17). First, remember that, since $\mathscr{G}_c$ is a recursively labeled MAG, there is p_2 such that Equation (16) holds independently of the chosen companion tuple $\tau(\mathscr{G}_c)$. Let $\langle \tau(\mathscr{G}_c) \rangle$ be a self-delimiting string that encodes the companion tuple $\tau(\mathscr{G}_c)$. Let p be a binary string that represents the algorithm running on a universal Turing machine that reads the companion tuple $\langle \tau(\mathscr{G}_c) \rangle$ as his first input and reads the output of x^* as its second input. Then, it reads each j-th bit of[33] x, runs $\langle j, \langle \langle \tau(\mathscr{G}_c) \rangle, p_2 \rangle \rangle$ and, from the outputs of $\langle j, \langle \langle \tau(\mathscr{G}_c) \rangle, p_2 \rangle \rangle$, returns the string $\langle \langle e_1, z_1 \rangle, \ldots, \langle e_n, z_n \rangle \rangle$ where $z_j = 1$, if the j-th bit of x is 1, and $z_j = 0$, if the j-th bit of x is 0. Threfore, we will have that there is a binary string $\langle \langle \tau(\mathscr{G}_c) \rangle, p \rangle \in \mathbf{L}'_{\mathbf{U}}$ that represents an algorithm running on a prefix (or self-delimiting) universal Turing machine $\mathbf{U}$ that, given x as input, runs p taking also $\langle \tau(\mathscr{G}_c) \rangle$ as the first input. Since p_2 is fixed, we have that there is a self-delimiting binary encoding of $(\langle \tau(\mathscr{G}_c) \rangle, p)$ with

$$l\left(\langle \langle \tau(\mathscr{G}_c) \rangle, p \rangle\right) \leq K(\langle \tau(\mathscr{G}_c) \rangle) + \mathbf{O}(1)$$

Then, by the minimality of $K(\cdot)$, we will have that

$$K(\mathscr{E}(\mathscr{G}_c) \,|\, x^*) \leq l\left(\langle \langle \tau(\mathscr{G}_c) \rangle, p \rangle\right) \leq K(\langle \tau(\mathscr{G}_c) \rangle) + \mathbf{O}(1)$$

Thus, from Step 21, we will have that

$$\mathbf{O}(1) + K(\mathscr{E}(\mathscr{G}_c)) \geq I_A(x \,; \mathscr{E}(\mathscr{G}_c)) \geq$$
$$\geq K(\mathscr{E}(\mathscr{G}_c)) - (K(\langle \tau(\mathscr{G}_c) \rangle) + \mathbf{O}(1))$$

Therefore,

$$I_A(x \,; \mathscr{E}(\mathscr{G}_c)) = K(\mathscr{E}(\mathscr{G}_c)) - \mathbf{O}\left(K(\langle \tau(\mathscr{G}_c) \rangle)\right)$$

For the proof of $I_A(\mathscr{E}(\mathscr{G}_c) \,; x) = K(x) - \mathbf{O}\left(K(\langle \tau(\mathscr{G}_c) \rangle)\right)$, the same follows analogously to the previous proof, but using an almost identical recursive procedure to the one for Equation (18) instead. Finally, the proof of $I_A(x; \mathscr{E}(\mathscr{G}_c)) = I_A(\mathscr{E}(\mathscr{G}_c); x) \pm \mathbf{O}(1)$ follows directly from Lemma 1.2.1.

---

[33] Note that x is the output of x^* on the chosen universal Turing machine.

□

Basically, Lemma 2.2 assures that the information contained in a MAG $\mathscr{G}_c$ and in the bit string representing the characteristic function (or indicator function) of pertinence in the set $\mathscr{E}(\mathscr{G}_c)$ are the same, except for the information necessary to compute the set of composite vertices $\mathbb{V}(\mathscr{G}_c)$. This is an important idea that will be employed in further sections.

As a MAG is a generalization of graphs, we may also want that Lemma 2.2 remains sound regarding classical graphs. Indeed, as we show in the next Corollary 2.2.1, this follows from the immediate fact that a first order $\mathscr{G}_c$ is (up to an notation automorphism) a classical graph. However, we will show in Corollary 2.3.2 that Corollary 2.2.1 can be improved.

Corollary 2.2.1. *Let $x \in \{0,1\}^*$. Let G be a classical graph from Definition 1.1.1.1 with $l(x) = |\mathbb{E}_c(G)|$ such that, for every $e \in \mathbb{E}_c(G)$,*

$$e \in E(G) \iff \quad \text{the } j\text{-th digit in } x \text{ is } 1$$

where $1 \leq j \leq l(x)$ and $\mathbb{E}_c(G) := \{\{x,y\} \mid x, y \in V\}$. Then,

(22) $C(E(G) \mid x) \leq K(E(G) \mid x) + \mathbf{O}(1) = \mathbf{O}(\lg(|V(G)|))$

(23) $C(x \mid E(G)) \leq K(x \mid E(G)) + \mathbf{O}(1) = \mathbf{O}(\lg(|V(G)|))$

(24) $K(x) = K(E(G)) \pm \mathbf{O}(\lg(|V(G)|))$

(25) $I_A(x; E(G)) = I_A(E(G); x) \pm \mathbf{O}(1) = K(x) - \mathbf{O}(\lg(|V(G)|))$

Proof. We have that, by definition, every classical graph G is (up to a notation isomorphism) a first order MAG $\mathscr{G}_c$. The key idea of the proof is that, since $V(G) = \{1, \ldots, n\} = \mathbb{V}(\mathscr{G}_c) \subset \mathbb{N}$, one can always define a recursive bijective ordering, so that we will have that $\mathscr{G}_c$ is recursively labeled given $|V(G)|$. To this end, let G be a first order MAG satisfying Lemma 2.1 with $p = 1$. Then, we will have that $V(G) = \mathbb{V}(\mathscr{G}_c)$ and $\langle \tau(\mathscr{G}_c) \rangle = (|V(G)|)_2$. Note that, from Equation (9) in Lemma 1.2.1, we have that $K(|V(G)|) = \mathbf{O}(\lg((|V(G)|)))$. Therefore, the rest of the proof follows directly from Lemma 2.2.   □

2.1. **Recursively labeled family of MultiAspect Graphs.** Another family of MAGs from Definition 2.1 that may be of interest is the one in which the ordering of edges does not depend on the size of $|\mathbb{V}(\mathscr{G}_c)|$ or, more specifically, on the set of aspects $\mathscr{A}(\mathscr{G}_c)$. The main idea underlying the definition of such family is that the ordering of the composite edges does not change as the companion tuple $\tau(\mathscr{G}_c)$ changes. Note that the companion tuple is highly informative in fully characterizing the respective MAG [31]. First, since $\langle \tau(\mathscr{G}_c) \rangle$ is being given as an input, it needs to be self-delimited. In addition, the recursive bijective pairing $\langle \cdot, \cdot \rangle$ allows one to univocally retrieves the tuple $(|\mathscr{A}(\mathscr{G})[1]|, \ldots, |\mathscr{A}(\mathscr{G})[p]|)$. Thus, the companion tuple also informs the order of the MAG. Secondly, note that, for the same value of $p = |\mathscr{A}(\mathscr{G})|$ and the same value of $|\mathbb{V}(\mathscr{G}_c)|$, one may have different companion tuples. Therefore, these give rise to the need of grasping the strong notion of *recursive labeling* into distinct families of MAGs as follows:

Definition 2.2. A family $F_{\mathscr{G}_c}$ of MAGs $\mathscr{G}_c$ (as in Definition 1.1.7) is *recursively labeled iff* there are programs $\mathrm{p}'_1, \mathrm{p}'_2 \in \{0,1\}^*$ such that, for every $\mathscr{G}_c \in F_{\mathscr{G}_c}$ and for every $a_i, b_i \in \mathscr{A}(\mathscr{G}_c)[i]$ with $1 \le i \le p = |\mathscr{A}(\mathscr{G}_c)|$,

(26) $$\mathbf{U}\left(\langle\langle a_1, \dots, a_p\rangle, \langle b_1, \dots, b_p\rangle, \mathrm{p}'_1\rangle\right) = (j)_2$$

(27) $$\mathbf{U}\left(\langle j, \mathrm{p}'_2\rangle\right) = \langle\langle a_1, \dots, a_p\rangle, \langle b_1, \dots, b_p\rangle\rangle = (e_j)_2$$

where

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

The reader is also invited to note that this Definition 2.2 can be easily extended to arbitrary MAGs $\mathscr{G}$, as in Definition 1.1.3. In this case, we will have $|\mathbb{E}(\mathscr{G})| = |\mathbb{V}(\mathscr{G})|^2$ instead of

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

To show that Definition 2.2 is satisfiable by an *infinite* (recursively enumerable) family of MAGs, we will define an infinite family of MAGs $\mathscr{G}_c$ such that every one of them has the same order and no condition of the presence or absence of a composite edge was taken in to account. The key idea of this proof is to start with an arbitrarily chosen MAG and construct an infinite family from an iteration in which only the number of elements in the aspects increases.

Lemma 2.3. *There is an recursively labeled infinite family $F_{\mathscr{G}_c}$ of MAGs $\mathscr{G}_c$ with arbitrary symmetric adjacency matrix (i.e., with arbitrary composite edge set in $\mathbb{E}_c$). In particular, there is an recursively labeled infinite family $F_{\mathscr{G}_c}$ of MAGs $\mathscr{G}_c$ with arbitrary symmetric adjacency matrix such that every one of them has the same order p.*

Proof. Let $p, n_0 \in \mathbb{N}$ be arbitrary values. Let $\mathscr{G}_{c0}$ be a fixed arbitrary MAG with $\mathscr{A}(\mathscr{G}_{c0})[i] \subset \mathbb{N}$, where $1 \le i \le p = |\mathscr{A}(\mathscr{G}_{c0})| \in \mathbb{N}$, such that, for every $i, j \le p$, we have $|\mathscr{A}(\mathscr{G}_{c0})[i]| = |\mathscr{A}(\mathscr{G}_{c0})[j]| = n_0 \in \mathbb{N}$. Then, we build another arbitrary MAG $\mathscr{G}_{c1}$ with $\mathscr{A}(\mathscr{G}_{c1})[i] \subset \mathbb{N}$, where $1 \le i \le p = |\mathscr{A}(\mathscr{G}_{c1})| = |\mathscr{A}(\mathscr{G}_{c0})| \in \mathbb{N}$, such that, for every $i, j \le p$, we have $|\mathscr{A}(\mathscr{G}_{c1})[i]| = |\mathscr{A}(\mathscr{G}_{c1})[j]| = n_0 + 1 = n_1 \in \mathbb{N}$. From an iteration of this process, we will obtain an infinite family $F_{\mathscr{G}_c} = \{\mathscr{G}_{c0}, \mathscr{G}_{c1}, \dots, \mathscr{G}_{ci}, \dots\}$ where no assumption was taken regarding the presence or absence of composite edges in their respective edge sets $\mathscr{E}$, so that any $\mathscr{G}_{ci} \in F_{\mathscr{G}_c}$ can be defined by any chosen composite edge set $\mathscr{E}(\mathscr{G}_{ci})$. In addition, there is a total order in respect to the set of all composite vertices $\mathbb{V}$ such that $\mathbb{V}(\mathscr{G}_{ci}) \subsetneq \mathbb{V}(\mathscr{G}_{ci+1})$, where $i \ge 0$. The next step is to construct a recursive ordering of composite edges for each one of these MAGs. Like in Lemma 2.1, we will construct a recursively ordered sequence of composite edges, which is independent of $\mathscr{E}$. From this sequence, the algorithms that the strings p'_1 and p'_2 represent will become immediately defined. To achieve this proof, we know that there is an algorithm that applies to $\mathscr{G}_{c0}$ the ordering satisfying the proof of Lemma 2.1. Let $(\mathbb{E}_c(\mathscr{G}_{ci}))$ denote an arbitrary sequence $\left(e_1, e_2, \dots, e_{|\mathbb{E}_c(\mathscr{G}_{ci})|}\right)$ of all possible composite edges of MAG $\mathscr{G}_{ci}$ with $i \ge 0$. Then, one applies the iteration of:

- If $(\mathbb{E}_c(\mathscr{G}_{ck}))$, where $k \ge 0$, is a sequence of composite edges such that, for every $\mathscr{G}_{ci}$ with $0 \le i \le k$, $(\mathbb{E}_c(\mathscr{G}_{ci}))$ is a prefix[34] of $(\mathbb{E}_c(\mathscr{G}_{ck}))$, then:

---

[34] Also note that a sequence is always a prefix of itself.

> (i) apply to $\mathbb{V}(\mathscr{G}_{ci+1})$ the recursive ordering $<_{\mathbb{E}_c}$ satisfying Lemma 2.1 ;
> (ii) Concatenate after the last element of $(\mathbb{E}_c(\mathscr{G}_{ck}))$ the elements of $\mathbb{E}_c(\mathscr{G}_{ci+1})$ that were not already in $(\mathbb{E}_c(\mathscr{G}_{ck}))$, while preserving the order relation $<_{\mathbb{E}_c}$ previously applied to $\mathbb{V}(\mathscr{G}_{ci+1})$.

Note that $p, n_0 \in \mathbb{N}$ are fixed values. Thus, p'_1 is a fixed string that represents on a universal Turing machine the algorithm that, given $\langle a_1, \ldots, a_p \rangle$ and $\langle b_1, \ldots, b_p \rangle$ as inputs,

> (i) builds the sequence of composite edges by the iteration described before such that, for each step of this iteration,
>> (a) search for $((a_1, \ldots, a_p), (b_1, \ldots, b_p))$ or $((b_1, \ldots, b_p), (a_1, \ldots, a_p))$ in this sequence;
>> (b) if one of these pairs is found, it returns the index value of the first one of these pairs found in this sequence.
>> (c) else, continue the iteration;

Note that one of these pairs must be always eventually found, since $\mathscr{A}(\mathscr{G}_{cj})[i] \subset \mathbb{N}$ with $1 \le i \le p = |\mathscr{A}(\mathscr{G}_{cj})| \in \mathbb{N}$ and $j \ge 0$. An analogous algorithm defines p'_1, but searching for the j-th element in the sequence and returning the respective pair of tuples instead.

$\square$

Thus, the sequence of composite edges of MAGs in this family that has a smaller set of composite vertices is always a prefix of the sequence of composite edges of the one that has a larger set of composite vertices. Note that we have kept the order of all MAGs in this family fixed with the purpose of avoiding some prefix ordering asymmetries due to dovetailing natural numbers inside the composite vertices for different values of p. This way, we have shown that Definition 2.2 is satisfiable.

One of the immediate properties of a recursively labeled family of MAGs $\mathscr{G}_c$ is that the information contained in the edge set of such MAGs is even more tightly associated with a binary string in Lemma 2.2. Thus, by replacing $\langle \langle \tau(\mathscr{G}_c) \rangle, \mathrm{p}_2 \rangle$ with p'_2, $\langle \langle \tau(\mathscr{G}_c) \rangle, \mathrm{p}_1 \rangle$ with p'_1, and $\langle \langle \tau(\mathscr{G}_c) \rangle, p \rangle$ with p in the proofs of Lemma 2.2, the following corollary holds:

Corollary 2.3.1. *Let $F_{\mathscr{G}_c}$ be a recursively labeled family (as in Definition 2.2) of MAGs $\mathscr{G}_c$. Then, for every $\mathscr{G}_c \in F_{\mathscr{G}_c}$ and $x \in \{0,1\}^*$, where $l(x) = |\mathbb{E}_c(\mathscr{G}_c)|$ such that, for every $e \in \mathbb{E}_c(\mathscr{G}_c)$,*

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

where $1 \le j \le l(x)$, the following relations hold

$$(28) \qquad C(\mathscr{E}(\mathscr{G}_c)\,|\,x) \le K(\mathscr{E}(\mathscr{G}_c)\,|\,x) + \mathbf{O}(1) = \mathbf{O}(1)$$

$$(29) \qquad C(x\,|\,\mathscr{E}(\mathscr{G}_c)) \le K(x\,|\,\mathscr{E}(\mathscr{G}_c)) + \mathbf{O}(1) = \mathbf{O}(1)$$

$$(30) \qquad\qquad\qquad K(x) = K(\mathscr{E}(\mathscr{G}_c)) \pm \mathbf{O}(1)$$

$$(31) \qquad I_A(x; \mathscr{E}(\mathscr{G}_c)) = I_A(\mathscr{E}(\mathscr{G}_c); x) \pm \mathbf{O}(1) = K(x) - \mathbf{O}(1)$$

Regarding classical graphs, one can assume a constant $p = |\mathscr{A}(\mathscr{G}_c)| = 1$ from the recursive ordering in Lemma 2.3, which satisfies Definition 2.2. Thus, since the composite edge sets $\mathscr{E}$ were arbitrary, there will be a recursively labeled infinite family that is equivalent (up to an edge re-ordering) to the family of all classical

graphs G, as in Definition 1.1.1.1. In other words, a classical graph is always a first order MAG that belongs to a recursively labeled family of MAGs, as previously stated in $[9, 25]$. In this regard, from Corollary 2.2.1 and the proof of Lemma 2.3 with order $p = 1$, we will have that:

Corollary 2.3.2. *Let $x \in \{0, 1\}^*$. Let G be a classical graph from Definition 1.1.1.1 with $l(x) = |\mathbb{E}_c(G)|$ such that, for every $e \in \mathbb{E}_c(G)$,*

$$e \in E(G) \iff \quad \text{the } j\text{-th digit in } x \text{ is } 1$$

where $1 \leq j \leq l(x)$ and $\mathbb{E}_c(G) \coloneqq \{\{x, y\} \mid x, y \in V\}$. Then,

$$(32) \qquad C(E(G) \mid x) \leq K(E(G) \mid x) + \mathbf{O}(1) = \mathbf{O}(1)$$

$$(33) \qquad C(x \mid E(G)) \leq K(x \mid E(G)) + \mathbf{O}(1) = \mathbf{O}(1)$$

$$(34) \qquad\qquad\qquad\qquad\qquad\qquad K(x) = K(E(G)) \pm \mathbf{O}(1)$$

$$(35) \qquad I_A(x; E(G)) = I_A(E(G); x) \pm \mathbf{O}(1) = K(x) - \mathbf{O}(1)$$

Thus, these results ensure that one can apply the incompressibility method to MAGs analogously to classical graphs as in $[9, 25]$. In particular, Corollary 2.3.2 ends up showing that our definitions and constructions of recursive labeling are consistent with the statements in $[9, 25]$. In the next section, we will show the existence of algorithmically random MAGs from a widely known example of K-random real number.

3. A family of K-random MultiAspect Graphs

One of the goals of this article is to show the existence of an infinite family of MAGs that contains a nested sequence of MAGs in which one is a subMAG of the other. Additionally, we want these MAGs to be $\mathbf{O}(1)$-K-random in respect to[35] its subMAGs. For this purpose,[36] we will use an infinite $\mathbf{O}(1)$-K-random binary sequence as the source of information to build the edge set $\mathscr{E}$. This is the main idea of our construction.

We will give a constructive method for building an edge set $\mathscr{E}(\mathscr{G}_c)$ that is algorithmic-informationally equivalent[37] to the n bits of Ω. Therefore, unlike the usage of C-random finite binary sequences like in Lemma 1.2.3 from $[9, 25]$, one can achieve a method for constructing a collection of *prefix algorithmically random* MAGs (or graphs) using an infinite K-random sequence as source.

The key idea is to define a direct bijection between a recursively ordered sequence of composite edges, which in turn defines the composite edge sets $\mathscr{E}(\mathscr{G}_c)$, and the bits of Ω. As an immediate consequence, $\mathscr{E}(\mathscr{G}_c)$ will be $\mathbf{O}(1)$-K-random, that is, algorithmically random in respect to prefix algorithmic complexity (see Sections 1.1.2 and 1.2.2). Further, from previously established relations between $\mathbf{O}(1)$-K-randomness and C-randomness in Section 1.2.2 and from Theorem 1.2.1, we will show in Section 4 that this MAG is isomorphically equivalent to a $\mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|))$-C-random classical graph (see Section 1.1.3). Therefore, promptly enabling a direct application of the results in Section 1.2.3 to this MAG.

---

[35] In fact, in respect to a finite collection of these subMAGS.

[36] See Section 3.1.

[37] See Theorem 3.1 and Corollary 2.3.1.

Definition 3.1. We say a MAG $\mathscr{G}_c$ (as in Definition 1.1.7) is (weakly) $\mathbf{O}(1)$-K-random *iff* it satisfies

$$K(\mathscr{E}(\mathscr{G}_c)) \geq \binom{|\mathbb{V}(\mathscr{G}_c)|}{2} - \mathbf{O}(1)$$

Thus, a $\mathbf{O}(1)$-K-random MAG $\mathscr{G}_c$ is an undirected MAG without self-loops with a topology (which is determined by the edge set $\mathscr{E}$) that can only be compressed up to a constant in a prefix universal Turing machine. That is, to decide the existence or non existence of a composite edge, one roughly needs the same number of bits of algorithmic information as the total number of possible composite edges. This follows the same intuition behind the definition of $\mathbf{O}(1)$-K-random real numbers (or infinite binary sequences). Additionally, it bridges[38] *plain algorithmic randomness* (i.e., C-randomness) in classical graphs from [9, 25] and *prefix algorithmic randomness* (i.e., K-randomness) by, in this case, assuming a constant randomness deficiency in respect to the prefix algorithmic complexity of the whole composite edge set $\mathscr{E}(\mathscr{G}_c)$. This differs from Definition 1.1.14 in [9, 25], which takes into account the conditional plain algorithmic complexity of the edge set given the number of vertices. Nevertheless, the reader may notice that we have from Lemma 1.2.1 that, for every $\mathbf{O}(1)$-K-random MAG $\mathscr{G}_c$,

$$K(\mathscr{E}(\mathscr{G}_c) \mid |\mathbb{V}(\mathscr{G}_c)|) \geq \binom{|\mathbb{V}(\mathscr{G}_c)|}{2} - \mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|))$$

That is, for $\mathbf{O}(1)$-K-random MAGs $\mathscr{G}_c$, informing the quantity of composite vertices to compress the edge set cannot give much more information than the one necessary to compute this very informed quantity. Thus, one may define a MAG $\mathscr{G}_c$ with

$$K(\mathscr{E}(\mathscr{G}_c) \mid |\mathbb{V}(\mathscr{G}_c)|) \geq \binom{|\mathbb{V}(\mathscr{G}_c)|}{2} - \delta(\mathscr{G}_c)$$

as a *strongly* $\delta(\mathscr{G}_c)$-K-random MAG $\mathscr{G}_c$. This way, every *weakly* $\mathbf{O}(1)$-K-random MAG $\mathscr{G}_c$ is *strongly* $\mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|))$-K-random. In addition, it follows directly from Equation (5) in Lemma 1.2.1 that every $\delta(\mathscr{G}_c)$-C-random MAG $\mathscr{G}_c$ (as we will define[39] in Section 4) is *strongly* $(\delta(\mathscr{G}_c) + \mathbf{O}(1))$-K-random. However, the investigation of strongly $\delta(\mathscr{G}_c)$-K-random MAGs is not in the scope of this article and we will only deal with the weak case hereafter. This is the reason we have left the term "weakly" between parenthesis in Definition 3.1, so that we will omit this term in this article.

It is also important to note that, since a MAG is a finite object, the asymptotic "big $\mathbf{O}$" notation in Definition 3.1 is equivalent to giving a fixed constant $c \in \mathbb{N}$. So, for a fixed value of c we say that a MAG $\mathscr{G}_c$ is $\mathbf{O}(1)$-K-random if

$$K(\mathscr{E}(\mathscr{G}_c)) \geq \binom{|\mathbb{V}(\mathscr{G}_c)|}{2} - c$$

This makes a direct parallel to weakly K-random finite binary strings as in Definition 1.1.12. Thus, despite the notation being similar, when we talk about $\mathbf{O}(1)$-K-randomness of MAGs it refers to prefix algorithmic randomness of finite objects (i.e., with a representation in a finite binary sequence) and when we talk about $\mathbf{O}(1)$-K-randomness of real numbers (as in Definition 1.1.13) it refers to prefix algorithmic randomness of an infinite binary sequence. However, as we will see

---

[38] See Section 1.2.2.
[39] See Definition 4.1.

in Section 3.1, there will be a strict relation between initial segments of $\mathbf{O}(1)$-K-random real numbers and nested subMAGs (or subgraphs). The reader may also want to see [10, 20, 25] for more properties and subtleties regarding algorithmically random finite sequences and algorithmically random infinite sequences.

Now, with the purpose of showing the existence of a $\mathbf{O}(1)$-K-random MAG $\mathscr{G}_c$, we will just combine previous results in algorithmic information theory with the ones that we have achieved in Section 2. In fact, we will show that the existence of an infinite family of MAGs satisfying Definition 3.1 holds within a recursively labeled family of MAGs using the K-incompressibility of the halting probability:

Lemma 3.1. *There is a recursively labeled* infinite *family $F_{\mathscr{G}_c}$ (as in Definition 2.2) of MAGs $\mathscr{G}_c$ that are $\mathbf{O}(1)$-K-random.*

Proof. From Lemma 2.3, we know there will be an infinite family $F'_{\mathscr{G}_c}$ that is recursively labeled with arbitrary presence or absence of composite edges in each MAG in this family. From Theorem 1.2.2, we have that

$$K(\Omega \restriction_n) \geq n - \mathbf{O}(1)$$

where $n \in \mathbb{N}$ is arbitrary. Since $F'_{\mathscr{G}_c}$ contains arbitrary arrangements of presence or absence of composite edges, we can now define family $F_{\mathscr{G}_c}$ as a subset of $F'_{\mathscr{G}_c}$ in which, for infinitely many $\mathscr{G}_c \in F_{\mathscr{G}_c} \subset F'_{\mathscr{G}_c}$ with $n = |\mathbb{E}_c(\mathscr{G}_c)|$, we have that

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

where $1 \leq j \leq n \in \mathbb{N}$. Then, for every $\mathscr{G}_c \in F_{\mathscr{G}_c}$, we will have from Corollary 2.3.1 that

$$K(\mathscr{E}(\mathscr{G}_c)) \pm \mathbf{O}(1) = K(\Omega \restriction_n) \geq n - \mathbf{O}(1)$$

Therefore, since

$$\binom{|\mathbb{V}(\mathscr{G}_c)|}{2} = \frac{|\mathbb{V}(\mathscr{G}_c)|^2 - |\mathbb{V}(\mathscr{G}_c)|}{2} = |\mathbb{E}_c(\mathscr{G}_c)| = n$$

we will have that

$$K(\mathscr{E}(\mathscr{G}_c)) \geq \binom{|\mathbb{V}(\mathscr{G}_c)|}{2} - \mathbf{O}(1)$$

$\square$

Additionally, since classical graphs are first order MAGs $\mathscr{G}_c$, the following corollary holds as a consequence of Corollary 2.3.2:

Corollary 3.1.1. *There is an infinite number of classical graphs G (as in Definition 1.1.1.1) that are $\mathbf{O}(1)$-K-random.*

3.1. **An infinite family of nested MultiAspect subGraphs.** We know that a real number is $\mathbf{O}(1)$-K-random if, and only if, it is weakly K-random for every initial segment (i.e., every prefix)—see Definition 1.1.13—of its representation in an infinite binary sequence. Thus, asking the same about K-randomness in MAGs or graphs would be a natural consequence of the previous results we have presented in this article. In fact, we will see that the same idea can be captured by nesting subgraphs of subgraphs and so on. As we are dealing with a generalization of

graphs, in particular MultiAspect Graphs (MAGs), the same must also be done for MAGs. In this section, we will extend the notion of subgraphs[40] to MAGs. Then, we will see in Theorem 3.1 that there is an infinite family of MAGs (and classical graphs in Corollary 3.1.1) which behaves like initial segments of a $\mathbf{O}(1)$-K-random real number.

The following definition is just an extension of the common definition of subgraphs, as in Definition 1.1.2.

Definition 3.2. Let $\mathscr{G}'$ and $\mathscr{G}$ be MAGs as in Definition 1.1.3. We say a MAG $\mathscr{G}'$ is a *MultiAspect subGraph* (subMAG) of a MAG $\mathscr{G}$, denoted as $\mathscr{G}' \subseteq \mathscr{G}$, *iff*

$$\mathbb{V}(\mathscr{G}') \subseteq \mathbb{V}(\mathscr{G}) \ \wedge \ \mathscr{E}(\mathscr{G}') \subseteq \mathscr{E}(\mathscr{G})$$

Definition 3.2.1. We say a MAG $\mathscr{G}'$ is a *vertex-induced subMAG* of MAG $\mathscr{G}$ *iff*

$$\mathbb{V}(\mathscr{G}') \subseteq \mathbb{V}(\mathscr{G})$$

and, for every $\mathbf{u}, \mathbf{v} \in \mathbb{V}(\mathscr{G}')$,

$$(\mathbf{u}, \mathbf{v}) \in \mathscr{E}(\mathscr{G}) \implies (\mathbf{u}, \mathbf{v}) \in \mathscr{E}(\mathscr{G}')$$

In addition, we denote this vertex-induced subMAG $\mathscr{G}'$ as $\mathscr{G}[\mathbb{V}(\mathscr{G}')]$.

Now, we can construct a family of nested subMAGs in a way such that there is a total order for the subgraph operation. In this manner, for every two elements of this family, one of them must be a subMAG of the other. First, we will define a nested family of MAGs in Definition 3.3. Then, we will prove the existence of a nested family that is recursively labeled and infinite in Lemma 3.2.

Definition 3.3. We say a family $F_{\mathscr{G}}^*$ of MAGs $\mathscr{G}$ (as in Definition 1.1.3) is a *nested family* of MAGs $\mathscr{G}$ *iff*, for every $\mathscr{G}, \mathscr{G}', \mathscr{G}'' \in F_{\mathscr{G}}^*$, the following hold

(1)
$$\mathscr{G}' \subseteq \mathscr{G} \ \wedge \ \mathscr{G} \subseteq \mathscr{G}' \implies \mathscr{G} = \mathscr{G}'$$

(2)
$$\mathscr{G}' \subseteq \mathscr{G} \ \wedge \ \mathscr{G} \subseteq \mathscr{G}'' \implies \mathscr{G}' \subseteq \mathscr{G}''$$

(3)
$$\mathscr{G}' \subseteq \mathscr{G} \ \vee \ \mathscr{G} \subseteq \mathscr{G}'$$

Definition 3.3.1. We say a family $F_{\mathscr{G}}^{v*}$ of MAGs $\mathscr{G}$ (as in Definition 1.1.3) is a *vertex-induced nested family* of MAGs $\mathscr{G}$ *iff*, for every $\mathscr{G}, \mathscr{G}', \mathscr{G}'' \in F_{\mathscr{G}}^{v*}$, the following hold

(1)
$$\mathscr{G}' = \mathscr{G}[\mathbb{V}(\mathscr{G}')] \subseteq \mathscr{G} \ \wedge \ \mathscr{G} = \mathscr{G}'[\mathbb{V}(\mathscr{G})] \subseteq \mathscr{G}' \implies \mathscr{G} = \mathscr{G}'$$

(2)
$$\mathscr{G}' = \mathscr{G}[\mathbb{V}(\mathscr{G}')] \subseteq \mathscr{G} \ \wedge \ \mathscr{G} = \mathscr{G}''[\mathbb{V}(\mathscr{G})] \subseteq \mathscr{G}'' \implies \mathscr{G}' = \mathscr{G}''[\mathbb{V}(\mathscr{G}')] \subseteq \mathscr{G}''$$

(3)
$$\mathscr{G}' = \mathscr{G}[\mathbb{V}(\mathscr{G}')] \subseteq \mathscr{G} \ \vee \ \mathscr{G} = \mathscr{G}'[\mathbb{V}(\mathscr{G})] \subseteq \mathscr{G}'$$

---

[40] As in Definition 1.1.2.

It follows directly from these definitions that every vertex-induced nested family in Definition 3.3.1 is a nested family in Definition 3.3. In addition, since simplified MAGs $\mathscr{G}_c$, as in Definition 1.1.7, are just a particular case of the ones in Definition 1.1.3, we can easily extend both Definitions 3.3 and 3.3.1 to families $F^*_{\mathscr{G}_c}$ and $F^{v*}_{\mathscr{G}_c}$ respectively.

As we will see in Lemma 3.2, one can define a vertex-induced nested family of MAGs that is recursively labeled and infinite. Moreover, there is a non-denumerable amount of these families. The key idea is to bring the same recursive ordering of composite edges from Lemma 2.3. Therefore, since it contains arbitrary configurations of composite edge sets, one can define a sequence of subMAGs drawn from an infinite binary sequence (i.e., a real number with an infinite fractional part) like we did in Lemma 3.1.

Lemma 3.2. *There is a non-denumerable amount of recursively labeled (vertex-induced) nested infinite families $F^{v*}_{\mathscr{G}_c}$ of MAGs $\mathscr{G}_c$. In particular, every real number $x \in [0,1] \subset \mathbb{R}$ with an infinite fractional part can univocally determine the presence or absence of a composite edge in every $\mathscr{G}_c \in F^{v*}_{\mathscr{G}_c}$.*

Proof. We will only prove the second part of the theorem, since we know that the cardinality of the set of real numbers $x \in [0,1] \subset \mathbb{R}$ with infinite fractional part in its binary representation is non-denumerable. Therefore, we will prove that an arbitrary real number $x \in [0,1] \subset \mathbb{R}$ with an infinite fractional part can univocally determine the presence or absence of a composite edge for every $\mathscr{G}_c \in F^{v*}_{\mathscr{G}_c}$, where $F^{v*}_{\mathscr{G}_c}$ is an infinite recursively labeled vertex-induced nested family of MAGs. To achieve this, let $x \in [0,1] \subset \mathbb{R}$ be an arbitrary real number with an infinite fractional part. Let $F'_{\mathscr{G}_c}$ be a family of MAGs defined in the proof of Lemma 2.3. Thus, there is $p \in \mathbb{N}$ which, for every $\mathscr{G}_c \in F'_{\mathscr{G}_c}$ and $i,j \leq p$, we have that $\mathscr{A}(\mathscr{G}_c)[i] \subset \mathbb{N}$, $|\mathscr{A}(\mathscr{G}_c)| = p$ and $|\mathscr{A}(\mathscr{G}_c)[i]| = |\mathscr{A}(\mathscr{G}_c)[j]|$. Hence, for every $\mathscr{G}_c, \mathscr{G}'_c, \mathscr{G}''_c \in F'_{\mathscr{G}_c}$, we will have that

(36) $$\mathbb{V}(\mathscr{G}'_c) \subseteq \mathbb{V}(\mathscr{G}_c) \wedge \mathbb{V}(\mathscr{G}_c) \subseteq \mathbb{V}(\mathscr{G}'_c) \implies \mathbb{V}(\mathscr{G}_c) = \mathbb{V}(\mathscr{G}'_c)$$

(37) $$\mathbb{V}(\mathscr{G}'_c) \subseteq \mathbb{V}(\mathscr{G}_c) \wedge \mathbb{V}(\mathscr{G}_c) \subseteq \mathbb{V}(\mathscr{G}''_c) \implies \mathbb{V}(\mathscr{G}'_c) \subseteq \mathbb{V}(\mathscr{G}''_c)$$

(38) $$\mathbb{V}(\mathscr{G}'_c) \subseteq \mathbb{V}(\mathscr{G}_c) \vee \mathbb{V}(\mathscr{G}_c) \subseteq \mathbb{V}(\mathscr{G}'_c)$$

(39) $$\mathbb{V}(\mathscr{G}_c) \subseteq \mathbb{V}(\mathscr{G}'_c) \implies |\mathbb{E}_c(\mathscr{G}_c)| \leq |\mathbb{E}_c(\mathscr{G}'_c)|$$

and

(40) $$\text{for every } e_i \in \mathbb{E}_c(\mathscr{G}_c) \text{ and } e'_j \in \mathbb{E}_c(\mathscr{G}'_c) \text{ with } i \leq |\mathbb{E}_c(\mathscr{G}_c)| \text{ and } j \leq |\mathbb{E}_c(\mathscr{G}'_c)|,$$
$$\mathbb{V}(\mathscr{G}_c) \subseteq \mathbb{V}(\mathscr{G}'_c) \wedge j \leq |\mathbb{E}_c(\mathscr{G}_c)| \implies e_i = e'_j$$

We now define a family $F^{v*}_{\mathscr{G}_c} \subset F'_{\mathscr{G}_c}$ such that, for every $\mathscr{G}_c, \mathscr{G}'_c \in F^{v*}_{\mathscr{G}_c} \subset F'_{\mathscr{G}_c}$, we have that

$$e_i \in \mathscr{E}(\mathscr{G}_c) \iff \text{ the } i\text{-th digit in } x \upharpoonright_n \text{ is } 1$$

(41) $$\text{and}$$

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

where $1 \leq j, i \leq n \in \mathbb{N}$, and

(42) $$n = \begin{cases} |\mathbb{E}_c(\mathscr{G}_c)| & \text{if } \mathscr{G}_c \text{ is a vertex induced subMAG of } \mathscr{G}'_c \\ |\mathbb{E}_c(\mathscr{G}'_c)| & \text{if } \mathscr{G}'_c \text{ is a vertex induced subMAG of } \mathscr{G}_c \end{cases}$$

In fact, from Equations (39) and (40), this family $F^{v*}_{\mathscr{G}_c}$ can be easily constructed as follows:

(a) if $\mathbb{V}(\mathscr{G}_c) \subseteq \mathbb{V}(\mathscr{G}'_c)$, then

$$n := \binom{|\mathbb{V}(\mathscr{G}_c)|}{2} \leq \binom{|\mathbb{V}(\mathscr{G}'_c)|}{2}$$

and

$$e_i \in \mathscr{E}(\mathscr{G}_c) \iff \text{ the } i\text{-th digit in } x \upharpoonright_n \text{ is } 1$$

and

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

(b) if $\mathbb{V}(\mathscr{G}'_c) \subseteq \mathbb{V}(\mathscr{G}_c)$, then

$$n := \binom{|\mathbb{V}(\mathscr{G}'_c)|}{2} \leq \binom{|\mathbb{V}(\mathscr{G}_c)|}{2}$$

and

$$e_i \in \mathscr{E}(\mathscr{G}_c) \iff \text{ the } i\text{-th digit in } x \upharpoonright_n \text{ is } 1$$

and

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

To prove that this construction can always be correctly applied infinitely many often, note that, since $F'_{\mathscr{G}_c}$ is infinite and Equations (36) and (38) hold, we have that

$$\mathbb{V}(\mathscr{G}'_c) \subseteq \mathbb{V}(\mathscr{G}_c) \; \lor \; \mathbb{V}(\mathscr{G}_c) \subseteq \mathbb{V}(\mathscr{G}'_c)$$

holds infinitely many often in $F^{v*}_{\mathscr{G}_c}$. $\qquad\qquad \square$

In this way, a family of graphs that satisfies Lemma 3.2 immediately gives us an infinite sequence of nested subMAGs. The issue we are going to tackle then is whether such chain of subMAGs could behave like initial segments of a K-random an infinite binary sequence or not. To this end, we capture this idea by making an analogous definition to Definition 1.1.13:

Definition 3.4. We say a nested infinite family $F^*_{\mathscr{G}_c}$ (as in Definition 3.3) of MAGs $\mathscr{G}_c$ (as in Definition 1.1.7) is $\mathbf{O}(1)$-K-random *iff*, for every $\mathscr{G}_c \in F^*_{\mathscr{G}_c}$, we have that

$$K(\mathscr{E}(\mathscr{G}_c)) \geq \binom{|\mathbb{V}(\mathscr{G}_c)|}{2} - \mathbf{O}(1)$$

Definition 3.4.1. Let $x \in [0,1] \subset \mathbb{R}$ be an arbitrary real number with an infinite fractional part. We denote as F_x the nested family $F^*_{\mathscr{G}_c}$ (as in Definition 3.3) of MAGs $\mathscr{G}_c$ (as in Definition 1.1.7) in which, for every $\mathscr{G}_c \in F^*_{\mathscr{G}_c}$ with $n = |\mathbb{E}_c(\mathscr{G}_c)|$,

$$K(\mathscr{E}(\mathscr{G}_c)) = K(x \upharpoonright_n) \pm \mathbf{O}(1)$$

In fact, such $\mathbf{O}(1)$-K-random nested family was already constructed for the proof of Lemma 3.1 and this may be seen as particular case of Lemma 3.2. We will use this particularity in the following theorem:

Theorem 3.1. *There is a recursively labeled (vertex-induced) nested infinite family $F^{v*}_{\mathscr{G}_c}$ (as in Definition 3.3.1) of MAGs $\mathscr{G}_c$ (as in Definition 1.1.7) that is $\mathbf{O}(1)$-K-random. In particular, there is a $\mathbf{O}(1)$-K-random recursively labeled (vertex-induced) nested infinite family F_Ω (as in Definition 3.4.1) of MAGs $\mathscr{G}_c$ (as in Definition 1.1.7) such that, for every $\mathscr{G}_c \in F_\Omega$ with $n = |\mathbb{E}_c(\mathscr{G}_c)|$,*

$$K(\mathscr{E}(\mathscr{G}_c)) = K(\Omega \upharpoonright_n) \pm \mathbf{O}(1)$$

Proof. It only suffices to prove the second part of the theorem, since the existence of such family F_Ω directly proves the first part. Thus, let $F_{\mathscr{G}_c}^{v*}$ be a family defined as in the proof of Lemma 3.2. Since the real number $x \in [0, 1] \subset \mathbb{R}$ was arbitrary, we can assume $x = \Omega$, as in Definition 1.1.11. However, we already have from the proofs of Lemmas 3.1 and 3.2 that this family immediately satisfies Lemma 3.1 and, therefore, every $\mathscr{G}_c \in F_{\mathscr{G}_c}^{v*}$ is $\mathbf{O}(1)$-K-random. In addition, since this family is recursively labeled, we will have from Corollary 2.3.1 that

$$K(\mathscr{E}(\mathscr{G}_c)) = K(\Omega \restriction_n) \pm \mathbf{O}(1)$$

Therefore, from Definition 3.4.1, we can denote this family $F_{\mathscr{G}_c}^{v*}$ by F_Ω. □

Additionally, the following corollary can be achieved directly from Corollary 2.3.2, instead of Corollary 2.3.1, and from Corollary 3.1.1, instead of Lemma 3.1, by assuming that the order of every MAG in the family satisfying Theorem 3.1 is $p = 1$:

Corollary 3.1.1. *There is a recursively labeled (vertex-induced) nested infinite family F_G^{v*} (as in Definition 3.3.1) of classical graphs (as in Definition 1.1.1.1) that is $\mathbf{O}(1)$-K-random. In particular, there is a $\mathbf{O}(1)$-K-random recursively labeled (vertex-induced) nested infinite family F_Ω (as in Definition 3.4.1) of classical graphs G (as in Definition 1.1.7) such that, for every $G \in F_\Omega$ with $n = |\mathbb{E}_c(G)|$,*

$$K(E(G)) = K(\Omega \restriction_n) \pm \mathbf{O}(1)$$

4. C-randomness of K-random MultiAspect Graphs

We have shown that randomness, regarding prefix algorithmic complexity, (i.e., prefix algorithmic randomness or K-randomness) in MultiAspect Graphs (MAGs) defines a class of MAGs with a topology that can only be described by the same amount of algorithmic information (except for a constant) as the number of possible connections. In Section 3, these results were achieved by extending the same concept of randomness of classical graphs regarding plain algorithmic complexity (i.e., plain algorithmic randomness or C-randomness) in [9, 25].[41] Therefore, a natural consequence would be studying the relation between (weakly) $\mathbf{O}(1)$-K-random MAGs and $\delta(|\mathbb{V}(\mathscr{G}_c)|)$-C-random MAGs.

One of the important results in algorithmic information theory (see Section 1.2.2) is that one can retrieve a lower bound for plain algorithmic complexity of finite segments of infinite binary sequences that are $\mathbf{O}(1)$-K-random. Thus, in this section, we apply this same property to MAGs. In particular, we study $\delta(n)$-C-randomness in MAGs that are $\mathbf{O}(1)$-K-random.

Definition 4.1. We say a MAG $\mathscr{G}_c$ (as in Definition 1.1.7) is $\delta(|\mathbb{V}(\mathscr{G}_c)|)$-C-random *iff* it satisfies

$$C\left(\mathscr{E}(\mathscr{G}_c) \,\middle|\, |\mathbb{V}(\mathscr{G}_c)|\right) \geq \binom{|\mathbb{V}(\mathscr{G}_c)|}{2} - \delta(|\mathbb{V}(\mathscr{G}_c)|)$$

where

$$\delta: \mathbb{N} \to \mathbb{N}$$
$$n \mapsto \delta(n)$$

---

[41] See also Sections 1.1.3 and 1.2.3.

This definition directly extends Definition 1.1.14 to MAGs, taking into account that Corollary 1.2.1.1 gives us an isomorphic representation of a MAG $\mathscr{G}_c$ as a classical graph. Therefore, it enables a proper interpretation of previous results in [9, 25] into the context of MAGs.

However, before studying the properties of $\delta(|\mathbb{V}(\mathscr{G}_c)|)$-C-random MAGs, we will investigate the relation between $\mathbf{O}(1)$-K-random MAGs, and $\delta(|\mathbb{V}(\mathscr{G}_c)|)$-C-random MAGs. The main idea is constructing MAGs from composite edge sets determined by binary strings that are prefixes of $\mathbf{O}(1)$-K-random real numbers. This will give rise not only to $\mathbf{O}(1)$-K-random MAGs, which are basically weakly K-random finite strings (see Definition 1.1.12), but also to $\delta(|\mathbb{V}(\mathscr{G}_c)|)$-C-random MAGs. Therefore, together with previous studies on algorithmic randomness, as restated in Theorem 1.2.3, we will now be able to obtain the following theorem:

Theorem 4.1. *Let $F_{\mathscr{G}_c}$ be a recursively labeled infinite family of MAGs $\mathscr{G}_c$ (as in Definition 2.2) such that, for every $\mathscr{G}_c \in F_{\mathscr{G}_c}$ and $x \in [0,1] \subset \mathbb{R}$, if $l(x \restriction_n) = |\mathbb{E}_c(\mathscr{G}_c)|$ and*

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

where $1 \le j \le l(x \restriction_n)$, $n \in \mathbb{N}$ and $e \in \mathbb{E}_c(\mathscr{G}_c)$, then $x \in [0,1] \subset \mathbb{R}$ is $\mathbf{O}(1)$-K-random (as in Definition 1.1.13). Thus, every MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}$ is $\mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|))$-C-random and (weakly) $\mathbf{O}(1)$-K-random. In addition, there is such family $F_{\mathscr{G}_c}$ with $\Omega = x \in [0,1] \subset \mathbb{R}$.

Proof. First, we prove that every MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}$ is $\mathbf{O}(1)$-K-random. We have that for every $\mathscr{G}_c$ there is $x \restriction_n \in \{0,1\}^*$ with $n = l(x \restriction_n) = |\mathbb{E}_c(\mathscr{G}_c)|$ and

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

where $1 \le j \le l(x \restriction_n)$, $n \in \mathbb{N}$ and $e \in \mathbb{E}_c(\mathscr{G}_c)$. Hence, by hypothesis, we will have that $x \in [0,1] \subset \mathbb{R}$ is $\mathbf{O}(1)$-K-random. Thus, from Definition 1.1.13 and Corollary 2.3.1, we will have that

$$K(\mathscr{E}(\mathscr{G}_c)) \pm \mathbf{O}(1) = K(x \restriction_n) \ge l(x \restriction_n) - \mathbf{O}(1) = \binom{|\mathbb{V}(\mathscr{G}_c)|}{2} - \mathbf{O}(1)$$

Thus, from Definition 3.1, we will have that every MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}$ is $\mathbf{O}(1)$-K-random. Now, in order to prove that every MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}$ is $\mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|))$-C-random, note that Theorem 1.2.3 implies that, if $x \in [0,1] \subset \mathbb{R}$ is $\mathbf{O}(1)$-K-random, then

$$(43) \qquad\qquad C(x \restriction_n) \ge n - K(n) - \mathbf{O}(1)$$

In addition, we know that the following inequalities hold:

(1) from Equation (7) in Lemma 1.2.1,

$$K(\mathscr{E}(\mathscr{G}_c)) \le K(|\mathbb{V}(\mathscr{G}_c)|) + K(\mathscr{E}(\mathscr{G}_c) \,|\, |\mathbb{V}(\mathscr{G}_c)|) + \mathbf{O}(1)$$

(2) from Equations (8) and (9) in Lemma 1.2.1,

$$C(|\mathbb{V}(\mathscr{G}_c)|) \le K(|\mathbb{V}(\mathscr{G}_c)|) + \mathbf{O}(1) \le \mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|))$$

and

$$K(n) \le \mathbf{O}(\lg(n))$$

(3) from Equation (5) in Lemma 1.2.1,

$$K(\mathscr{E}(\mathscr{G}_c) \,|\, |\mathbb{V}(\mathscr{G}_c)|) \le C(\mathscr{E}(\mathscr{G}_c) \,|\, |\mathbb{V}(\mathscr{G}_c)|) + \mathbf{O}(\lg(C(\mathscr{E}(\mathscr{G}_c) \,|\, |\mathbb{V}(\mathscr{G}_c)|)))$$

(4) from[42] Equations (2), (4) and (5) in Lemma 1.2.1 and Corollary 2.3.1,

$$C(\mathscr{E}(\mathscr{G}_c) \mid |\mathbb{V}(\mathscr{G}_c)|) \leq K(\mathscr{E}(\mathscr{G}_c) \mid |\mathbb{V}(\mathscr{G}_c)|) + \mathbf{O}(1) \leq$$

$$\leq K(\mathscr{E}(\mathscr{G}_c)) + \mathbf{O}(1) = K(x \upharpoonright_n) \pm \mathbf{O}(1) \leq$$

$$\leq n + \mathbf{O}(\lg(n)) \leq \mathbf{O}(n^2)$$

(5) from Equation (8) in Lemma 1.2.1, Equation (43) and Corollary 2.3.1,

$$K(\mathscr{E}(\mathscr{G}_c)) \pm \mathbf{O}(1) = K(x \upharpoonright_n) + \mathbf{O}(1) \geq C(x \upharpoonright_n) \geq n - K(n) - \mathbf{O}(1)$$

Then, since

$$\mathbf{O}(\lg(n + \mathbf{O}(\lg(n)))) = \mathbf{O}(\lg(n)) = \mathbf{O}\left(\lg\left(\frac{|\mathbb{V}(\mathscr{G}_c)|^2 - |\mathbb{V}(\mathscr{G}_c)|}{2}\right)\right) = \mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|)) \ ,$$

we will have that

$$\mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|)) + C(\mathscr{E}(\mathscr{G}_c) \mid |\mathbb{V}(\mathscr{G}_c)|) + \mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|)) \pm \mathbf{O}(1) \geq$$

$$\geq n - \mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|)) - \mathbf{O}(1) = \binom{|\mathbb{V}(\mathscr{G}_c)|}{2} - \mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|))$$

Let $\delta(|\mathbb{V}(\mathscr{G}_c)|) = \mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|))$. Thus, from Definition 4.1, we will have that $\mathscr{G}_c$ is $\mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|))$-C-random. In order to prove that there is such family $F_{\mathscr{G}_c}$ with $\Omega = x \in [0, 1] \subset \mathbb{R}$, just use the one from the proof of Lemma 3.1. $\qquad\square$

With this result, we can study plain algorithmic randomness in nested infinite $\mathbf{O}(1)$-K-random families of MAGs. Thus, by choosing a family of MAGs that satisfies Theorem 3.1 we will have from Corollary 2.3.1 that the conditions of Theorem 4.1 are immediately satisfied. Hence,

Corollary 4.1.1. *Let $F_{\mathscr{G}_c}^{v*}$ be a recursively labeled (vertex-induced) nested infinite $\mathbf{O}(1)$-K-random family (as in Theorem 3.1) of MAGs $\mathscr{G}_c$ (as in Definition 2.2). Then, every MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}^{v*}$ is $\mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|))$-C-random.*

Furthermore, the same case for classical graphs applies as a particular case by employing Corollary 3.1.1 instead of Theorem 3.1:

Corollary 4.1.2. *Let F_G^{v*} be a recursively labeled (vertex-induced) nested infinite $\mathbf{O}(1)$-K-random family (as in Corollary 3.1.1) of classical graphs G (as in Definition 1.1.1.1). Then, every classical graph $G \in F_G^{v*}$ is $\mathbf{O}(\lg(|V(G)|))$-C-random.*

5. Some topological properties of algorithmically random MultiAspect Graphs

In this section, we extend the incompressibility method on classical graphs in [9, 25] to plain algorithmically random MAGs. Hence, we will investigate diameter, connectivity, degree, and automorphisms.

The key idea of the following results derives directly from applying the equivalence of MAGs and graphs, as in Section 1.2.1, along with previous results from [9, 25], restated in Section 1.2.3.

---

[42] Or Equations (10), (9), (4) and (5).

Theorem 5.1. *Let $F_{\mathscr{G}_c} \neq \varnothing$ be an arbitrary recursively labeled family of MAGs $\mathscr{G}_c$ (as in Definition 2.2). Then, for every $\mathscr{G}_c \in F_{\mathscr{G}_c}$*

$$\mathscr{G}_c \text{ is } (\delta(|\mathbb{V}(\mathscr{G}_c)|) + \mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|)))\text{-}C\text{-random iff } G \text{ is}$$
$$(\delta(|V(G)|) + \mathbf{O}(\lg(|V(G)|)))\text{-}C\text{-random}$$

where G is isomorphic (as in Definition 1.1.6) to $\mathscr{G}_c$.

Proof. The existence and uniqueness of G is guaranteed by Corollary 1.2.1.1, which follows from the proof of Theorem 1.2.1 in [31] with a symmetric adjacency matrix. Thus, we will first describe a recursive procedure for constructing this unique isomorphic classical graph G from $\mathscr{G}_c \in F_{\mathscr{G}_c}$ and vice-versa. Then, it will only remain to prove that (except for the information necessary to compute the size of the graph):

$$\mathscr{G}_c \text{ is } \delta(|\mathbb{V}(\mathscr{G}_c)|)\text{-C-random } \textit{iff } G \text{ is } \delta(|V(G)|)\text{-C-random}$$

In order to construct such classical graph G from $\mathscr{G}_c \in F_{\mathscr{G}_c}$, it is important to remember the proof of Theorem 1.2.1 in [31]. We assume here the same procedure described there for the existence of G. Since $\mathscr{G}_c$ belongs to a recursively labeled family, as in Definition 2.2, then we can take the recursive bijective pairing function $\langle \cdot, \cdot, \ldots, \cdot \rangle$ on which this recursive labeling applies to this family. Hence, there is a recursive bijective function

$$
\begin{aligned}
f\colon \quad \mathbb{V}(\mathscr{G}_c) \quad &\to \quad V(G) = \{1, \ldots, n\} \subset \mathbb{N} \\
(a_1, \ldots, a_p) &\mapsto f((a_1, \ldots, a_p)) = \langle a_1, \ldots, a_p \rangle \in \mathbb{N}
\end{aligned}
$$

that performs a bijective relabeling between vertices of G and composite vertices of $\mathscr{G}_c$. Note that $|V(G)| = |\mathbb{V}(\mathscr{G}_c)| \in \mathbb{N}$. Therefore, given $\mathscr{E}(\mathscr{G}_c)$ as input, there is an algorithm that reads this string $\langle \mathscr{E}(\mathscr{G}_c) \rangle$ and replace each composite vertex by its corresponding label in $V(G)$ using function f, and then returns $\langle E(G) \rangle$. On the other hand, given $E(G)$ as input, there is an algorithm that reads this string $\langle E(G) \rangle$ and replace each vertex by its corresponding label in $\mathbb{V}(\mathscr{G}_c)$ using function f^{-1}, and then returns $\langle \mathscr{E}(\mathscr{G}_c) \rangle$. Thus, since $|V(G)| = |\mathbb{V}(\mathscr{G}_c)| \in \mathbb{N}$, we will have that

$$K\left(\mathscr{E}(\mathscr{G}_c) \,\middle|\, |\mathbb{V}(\mathscr{G}_c)|\right) = K\left(E(G) \,\middle|\, |V(G)|\right) \pm \mathbf{O}(1)$$

Now, we split the proof in two cases: first, when $K\left(\mathscr{E}(\mathscr{G}_c) \,\middle|\, |\mathbb{V}(\mathscr{G}_c)|\right) \leq K\left(E(G) \,\middle|\, |V(G)|\right) + \mathbf{O}(1)$; second, when $K\left(\mathscr{E}(\mathscr{G}_c) \,\middle|\, |\mathbb{V}(\mathscr{G}_c)|\right) + \mathbf{O}(1) \geq K\left(E(G) \,\middle|\, |V(G)|\right)$. The second case will follow analogously to the first one. So, for the first case, suppose

$$(44) \qquad K\left(\mathscr{E}(\mathscr{G}_c) \,\middle|\, |\mathbb{V}(\mathscr{G}_c)|\right) \leq K\left(E(G) \,\middle|\, |V(G)|\right) + \mathbf{O}(1)$$

From Equation (5) in Lemma 1.2.1, we have that

$$C\left(\mathscr{E}(\mathscr{G}_c) \,\middle|\, |\mathbb{V}(\mathscr{G}_c)|\right) \leq K\left(\mathscr{E}(\mathscr{G}_c) \,\middle|\, |\mathbb{V}(\mathscr{G}_c)|\right) + \mathbf{O}(1)$$

and

$$K\left(E(G) \,\middle|\, |V(G)|\right) + \mathbf{O}(1) \leq C\left(E(G) \,\middle|\, |V(G)|\right) + \mathbf{O}\left(\lg\left(C\left(E(G) \,\middle|\, |V(G)|\right)\right)\right)$$

Then, from Equations (3) and (1) in Lemma 1.2.1 and

$$\mathbf{O}\left(\lg\left(\frac{|V(G)|^2 - |V(G)|}{2}\right)^2\right) = \mathbf{O}(\lg(|V(G)|))$$

and

$$l\left(\langle E(G)\rangle\right) \le 2\,\mathbf{O}(\lg\left(|V(G)|\right))\left(\frac{|V(G)|^2 - |V(G)|}{2}\right) + \mathbf{O}\left(\lg\left(\frac{|V(G)|^2 - |V(G)|}{2}\right)\right) + \mathbf{O}(1) \le$$

$$\le \mathbf{O}\left(\left(\frac{|V(G)|^2 - |V(G)|}{2}\right)^2\right)$$

we will have by supposition (see Equation (44)) that

$$C\left(\mathscr{E}(\mathscr{G}_c)\,\big|\,|\mathbb{V}(\mathscr{G}_c)|\right) \le K\left(\mathscr{E}(\mathscr{G}_c)\,\big|\,|\mathbb{V}(\mathscr{G}_c)|\right) + \mathbf{O}(1) \le K\left(E(G)\,\big|\,|V(G)|\right) + \mathbf{O}(1) \le$$

$$\le C\left(E(G)\,\big|\,|V(G)|\right) + \mathbf{O}\left(\lg\left(C\left(E(G)\,\big|\,|V(G)|\right)\right)\right) \le$$

$$\le C\left(E(G)\,\big|\,|V(G)|\right) + \mathbf{O}\left(\lg\left(l\left(\langle E(G)\rangle\right) + \mathbf{O}\left(1\right)\right)\right) \le$$

$$\le C\left(E(G)\,\big|\,|V(G)|\right) + \mathbf{O}\left(\lg\left(\frac{|V(G)|^2 - |V(G)|}{2}\right)^2\right) \le$$

$$\le C\left(E(G)\,\big|\,|V(G)|\right) + \mathbf{O}(\lg(|V(G)|))$$

For the second case,

$$K\left(\mathscr{E}(\mathscr{G}_c)\,\big|\,|\mathbb{V}(\mathscr{G}_c)|\right) + \mathbf{O}(1) \ge K\left(E(G)\,\big|\,|V(G)|\right)$$

we will have analogously that

$$C\left(E(G)\,\big|\,|V(G)|\right) \le C\left(\mathscr{E}(\mathscr{G}_c)\,\big|\,|\mathbb{V}(\mathscr{G}_c)|\right) + \mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|))$$

For this, just note that one can use the recursive function f^{-1} to construct the composite vertices in $\mathbb{V}(\mathscr{G}_c)$ from vertices in $V(G)$, so that

$$l\left(\langle \mathscr{E}(\mathscr{G}_c)\rangle\right) \le 2\,\mathbf{O}(\lg\left(|V(G)|\right))\left(\frac{|V(G)|^2 - |V(G)|}{2}\right) + \mathbf{O}\left(\lg\left(\frac{|V(G)|^2 - |V(G)|}{2}\right)\right) + \mathbf{O}(1)$$

Thus, from Definitions 4.1 and 1.1.14, we will have that

$$\mathscr{G}_c \text{ is } (\delta(|\mathbb{V}(\mathscr{G}_c)|) + \mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|)))\text{-C-random } \textit{iff } G \text{ is}$$
$$(\delta(|V(G)|) + \mathbf{O}(\lg(|V(G)|)))\text{-C-random}$$

where $|V(G)| = |\mathbb{V}(\mathscr{G}_c)| \in \mathbb{N}$. $\square$

This way, Theorem 5.1 establishes a way to study common properties between algorithmically random MAGs and algorithmically random graphs. It takes into account algorithmic randomness for plain algorithmic complexity in both cases. In fact, we have shown that the plain algorithmic complexity of simplified MAGs and its isomorphic classical graph is roughly the same, except for the amount of algorithmic information necessary[43] to encode the length of the program that performs this isomorphism on an arbitrary universal Turing machine. As a consequence, it allows us to properly extend some results in [9, 25] on plain algorithmically random classical graphs to simplified MAGs:

Corollary 5.1.1. *Let $F_{\mathscr{G}_c} \ne \varnothing$ be an arbitrary recursively labeled family of MAGs $\mathscr{G}_c$ (as in Definition 2.2). Then, the following hold:*

---

[43] Upper bounded by $\mathbf{O}(\lg(|V(G)|))$.

(1) *If $F_{\mathscr{G}_c}$ is also a family in which every MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}$ has the same number of composite vertices $|\mathbb{V}(\mathscr{G}_c)|$ and this family contains all possible arrangements of presence or absence of composite edges, then a fraction of at least*

$$1 - \frac{1}{2^{\delta(|\mathbb{V}(\mathscr{G}_c)|)}}$$

of all MAGs that belong to this family $F_{\mathscr{G}_c}$ is $\delta(|\mathbb{V}(\mathscr{G}_c)| + \mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|)))$-C-random.

(2) *The degree $\mathbf{d}(\mathbf{v})$ of a composite vertex $\mathbf{v} \in \mathbb{V}(\mathscr{G}_c)$ in a $\delta(|\mathbb{V}(\mathscr{G}_c)|)$-C-random MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}$ satisfies*

$$\left| \mathbf{d}(\mathbf{v}) - \left(\frac{|\mathbb{V}(\mathscr{G}_c)| - 1}{2} \right) \right| = \mathbf{O}\left(\sqrt{|\mathbb{V}(\mathscr{G}_c)| \, (\delta(|\mathbb{V}(\mathscr{G}_c)|) + \mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|)))} \right)$$

(3) *All $\mathbf{o}(|\mathbb{V}(\mathscr{G}_c)|)$-C-random MAGs $\mathscr{G}_c \in F_{\mathscr{G}_c}$ have*

$$\frac{|\mathbb{V}(\mathscr{G}_c)|}{4} + \mathbf{o}(|\mathbb{V}(\mathscr{G}_c)|)$$

disjoint paths of length 2 between each pair of composite vertices $\mathbf{u}, \mathbf{v} \in \mathbb{V}(\mathscr{G}_c)$. In particular, all $\mathbf{o}(|\mathbb{V}(\mathscr{G}_c)|)$-C-random MAGs $\mathscr{G}_c \in F_{\mathscr{G}_c}$ have composite diameter 2.

(4) *Let $c \in \mathbb{N}$ be a fixed constant. Let $\mathscr{G}_c \in F_{\mathscr{G}_c}$ be $(\mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|)))$-C-random. Let $X_{f(|\mathbb{V}(\mathscr{G}_c)|)}(\mathbf{v})$ denote the set of the least $f(|\mathbb{V}(\mathscr{G}_c)|)$ neighbors of a composite vertex $\mathbf{v} \in \mathbb{V}(\mathscr{G}_c)$, where*

$$f\colon \begin{array}{ccc} \mathbb{N} & \to & \mathbb{N} \\ |\mathbb{V}(\mathscr{G}_c)| & \mapsto & f(|\mathbb{V}(\mathscr{G}_c)|) \end{array}$$

Then, for every composite vertices $\mathbf{u}, \mathbf{v} \in \mathbb{V}(\mathscr{G}_c)$,

$$\{\mathbf{u}, \mathbf{v}\} \in \mathscr{E}(\mathscr{G}_c)$$

or

$$\exists \mathbf{i} \in \mathbb{V}(\mathscr{G}_c)\left(\mathbf{i} \in X_{(\lg(|\mathbb{V}(\mathscr{G}_c)|))^2}(\mathbf{v}) \wedge \{\mathbf{u}, \mathbf{i}\} \in \mathscr{E}(\mathscr{G}_c) \wedge \{\mathbf{i}, \mathbf{v}\} \in \mathscr{E}(\mathscr{G}_c) \right)$$

(5) *All $\mathbf{o}(|\mathbb{V}(\mathscr{G}_c)| + \lg(|\mathbb{V}(\mathscr{G}_c)|))$-C-random MAGs $\mathscr{G}_c \in F_{\mathscr{G}_c}$ are rigid under permutations of composite vertices.*

Proof. The proofs of all five statements come directly from Theorem 5.1. Hence, we specifically obtain the desired proofs of Items 1, 2, 3, 4, and 5 from Lemmas 1.2.3, 1.2.4, 1.2.5, 1.2.6, and 1.2.7 respectively. Note that one needs to apply the respective corrections to the randomness deficiencies $\delta(x)$ from Theorem 5.1 regarding asymptotic dominance. Also note that, in Item 4, if a classical graph is $(c \lg(|V(G)|))$-C-random, then $((c+3) \lg(|V(G)|)) \leq \mathbf{o}((\lg(|V(G)|))^2)$, which satisfies Lemma 1.2.6. $\square$

In addition, we can directly[44] combine Corollary 5.1.1 with Theorem 4.1 or Corollary 4.1.1 into the following Corollaries 5.1.2 and 5.1.3, which can be easily extended to classical graphs too. This result ends our present investigation of algorithmic randomness of MultiAspect Graphs (MAGs) by relating graph-topological properties like in [9, 25] with algorithmically random MAGs regarding prefix algorithmic complexity.

---

[44] Hence, we omit the proof.

Corollary 5.1.2. *Let $F_{\mathscr{G}_c}$ be a recursively labeled infinite family of MAGs $\mathscr{G}_c$ (as in Definition 2.2) such that, for every $\mathscr{G}_c \in F_{\mathscr{G}_c}$ and $x \in [0,1] \subset \mathbb{R}$, if $l(x \upharpoonright_n) = |\mathbb{E}_c(\mathscr{G}_c)|$ and*

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

where $1 \le j \le l(x)$, $n \in \mathbb{N}$ and $e \in \mathbb{E}_c(\mathscr{G}_c)$, then $x \in [0,1] \subset \mathbb{R}$ is $\mathbf{O}(1)$-K-random. Then, the following hold:

(1) *The degree $\mathbf{d}(\mathbf{v})$ of a composite vertex $\mathbf{v} \in \mathbb{V}(\mathscr{G}_c)$ in a MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}$ satisfies*

$$\left| \mathbf{d}(\mathbf{v}) - \left(\frac{|\mathbb{V}(\mathscr{G}_c)| - 1}{2} \right) \right| = \mathbf{O}\left(\sqrt{|\mathbb{V}(\mathscr{G}_c)| \left(\mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|)) \right)} \right)$$

(2) *Every MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}$ has*

$$\frac{|\mathbb{V}(\mathscr{G}_c)|}{4} + \mathbf{o}(|\mathbb{V}(\mathscr{G}_c)|)$$

disjoint paths of length 2 between each pair of composite vertices $\mathbf{u}, \mathbf{v} \in \mathbb{V}(\mathscr{G}_c)$.

(3) *Every MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}$ has composite diameter 2.*

(4) *For every composite vertices $\mathbf{u}, \mathbf{v} \in \mathbb{V}(\mathscr{G}_c)$ in a MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}$,*

$$\{\mathbf{u}, \mathbf{v}\} \in \mathscr{E}(\mathscr{G}_c)$$

or

$$\exists \mathbf{i} \in \mathbb{V}(\mathscr{G}_c) \big(\mathbf{i} \in X_{(\lg(|\mathbb{V}(\mathscr{G}_c)|))^2}(\mathbf{v}) \wedge \{\mathbf{u}, \mathbf{i}\} \in \mathscr{E}(\mathscr{G}_c) \wedge \{\mathbf{i}, \mathbf{v}\} \in \mathscr{E}(\mathscr{G}_c) \big)$$

(5) *Every MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}$ is rigid under permutations of composite vertices.*

As we have investigated in Section 3.1, this result also holds for nested families:

Corollary 5.1.3. *Let $F_{\mathscr{G}_c}^{v*}$ be a recursively labeled (vertex-induced) nested infinite $\mathbf{O}(1)$-K-random family (as in Theorem 3.1) of MAGs $\mathscr{G}_c$ (as in Definition 2.2). Then, the following hold:*

(1) *The degree $\mathbf{d}(\mathbf{v})$ of a composite vertex $\mathbf{v} \in \mathbb{V}(\mathscr{G}_c)$ in a MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}^{v*}$ satisfies*

$$\left| \mathbf{d}(\mathbf{v}) - \left(\frac{|\mathbb{V}(\mathscr{G}_c)| - 1}{2} \right) \right| = \mathbf{O}\left(\sqrt{|\mathbb{V}(\mathscr{G}_c)| \left(\mathbf{O}(\lg(|\mathbb{V}(\mathscr{G}_c)|)) \right)} \right)$$

(2) *Every MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}^{v*}$ has*

$$\frac{|\mathbb{V}(\mathscr{G}_c)|}{4} + \mathbf{o}(|\mathbb{V}(\mathscr{G}_c)|)$$

disjoint paths of length 2 between each pair of composite vertices $\mathbf{u}, \mathbf{v} \in \mathbb{V}(\mathscr{G}_c)$.

(3) *Every MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}^{v*}$ has composite diameter 2.*

(4) *For every composite vertices $\mathbf{u}, \mathbf{v} \in \mathbb{V}(\mathscr{G}_c)$ in a MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}^{v*}$,*

$$\{\mathbf{u}, \mathbf{v}\} \in \mathscr{E}(\mathscr{G}_c)$$

or

$$\exists \mathbf{i} \in \mathbb{V}(\mathscr{G}_c) \big(\mathbf{i} \in X_{(\lg(|\mathbb{V}(\mathscr{G}_c)|))^2}(\mathbf{v}) \wedge \{\mathbf{u}, \mathbf{i}\} \in \mathscr{E}(\mathscr{G}_c) \wedge \{\mathbf{i}, \mathbf{v}\} \in \mathscr{E}(\mathscr{G}_c) \big)$$

(5) Every MAG $\mathscr{G}_c \in F_{\mathscr{G}_c}^{v*}$ is rigid under permutations of composite vertices.

6. Conclusions

In this article, we have theoretically investigated algorithmic randomness of generalizations of graphs, in particular, MultiAspect Graphs (MAGs). In addition, we have extended previous results on network topological properties for classical graphs to MAGs.

First, we have defined recursive labeling for MAGs. Unlike classical graphs, the algorithmic information of a MAG and the representative binary string, which determines its composite edge set, may be not equivalent (i.e., up to a constant) regarding (plain or prefix) algorithmic complexity and mutual algorithmic information. In fact, we have shown that the algorithmic information content of a MAG and its representative binary string may differ on the order of the prefix algorithmic complexity of the companion tuple, which determines the set of composite vertices.

Then, we have extended the conception of recursive labeling in order to define recursively labeled families of MAGs. In this case, we have shown that MAGs in a recursively labeled family are tightly associated (analogously to the case for classical labeled graphs) with its respective representative binary string that algorithmically determines the presence or absence of a edge.

We have also introduced prefix algorithmic randomness for MAGs. In this regard, we have shown that there are infinite families of MAGs in which every member is incompressible regarding prefix algorithmic complexity. This shows that the same phenomenon of incompressibility of finite strings in classical algorithmic information theory also holds for high order networks.

In addition, recursively labeled infinite families of nested subMAGs were formally constructed with the purpose of defining an infinite object that could behave like an infinite binary sequence. In fact, we have shown that there is such infinite family which is prefix algorithmically random. This is an exact analogous phenomenon to prefix algorithmic randomness for infinite binary sequences (or real numbers), e.g., the halting probability (i.e., the Omega number or Chaitin's constant).

In the context of MAGs, we have also investigated some relationships between prefix algorithmic randomness and plain algorithmic randomness. This shows, and reinforces, some previous relations and equivalences in algorithmic information theory, but now in respect to MAGs (or high order networks) and classical graphs. Thus, we suggest further investigations of other possible equivalences brought from algorithmic information theory in future research.

Furthermore, we have extended previous results on network topological properties of plain algorithmically random classical graphs to plain algorithmically random MAGs or prefix algorithmically random nested families of MAGs. In particular, vertex degree, connectivity, diameter, and rigidity. In fact, as it was the case for classical graphs, this shows that there are several useful properties that could be embedded or analyzed in high order networks.

This article shows that an incompressible MAG have on average high degree composite vertices, particularly, with degrees on the order of half of the size of the network (i.e., half of the value given by the number of composite vertices minus 1) within an strong-asymptotically dominated standard deviation. In this sense,

an incompressible MAG tends to be an expected "almost regular" graph in the limit when the size of the networks increases indefinitely. For sufficiently large set of composite vertices, incompressible MAGs also cross a phase transition in which the diameter between composite vertices becomes 2. They are $\frac{|\mathbb{V}(\mathscr{G}_c)|}{4} + \mathbf{o}(|\mathbb{V}(\mathscr{G}_c)|)$-connected and there is a $\left(\lg(|\mathbb{V}(\mathscr{G}_c)|)\right)^2$-vertex star with any arbitrary composite vertex as the center that is directly linked to any other composite vertex of the respective incompressible MAG. In addition, incompressible MAGs are rigid (i.e., when only the identity automorphism holds) in respect to permutations of composite vertices.

It is also important to note that, since a classical graph is a first order simplified MAG, we have also shown that all the results in this article hold for classical graphs.

References

[1] A.-L. Barabasi, *Scale-Free Networks: A Decade and Beyond*, Science **325** (2009jul), no. 5939, 412–413, available at `http://www.sciencemag.org/cgi/doi/10.1126/science.1173299`.

[2] Béla Bollobás, *Modern graph theory*, Graduate texts in mathematics, Springer Science & Business Media, 1998.

[3] Béla Bollobás and Oliver Riordan, *The diameter of a scale-free random graph*, Combinatorica **24** (2004jan), no. 1, 5–34.

[4] Béla Bollobás, *Random Graphs*, Cambridge University Press, Cambridge, 2001.

[5] Fabio Boschetti, Mikhail Prokopenko, Ian Macreadie, and Anne-Marie Grisogono, *Defining and Detecting Emergence in Complex Networks*, Knowledge-based intelligent information and engineering systems, 9th international conference proceedings (kes 2005), lncs 3684, 2005, pp. 573–580.

[6] Ulrik Brandes and Thomas Erlebach, *Fundamentals*, Network analysis, 2005, pp. 7–15.

[7] _____ (ed.), *Network Analysis*, Lecture Notes in Computer Science, vol. 3418, Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2005.

[8] Harry Buhrman, Jaap-Henk Hoepman, and Paul Vitányi, *Space-efficient Routing Tables for Almost All Networks and the Incompressibility Method*, SIAM Journal on Computing **28** (1999jan), no. 4, 1414–1432.

[9] Harry Buhrman, Ming Li, John Tromp, and Paul Vitányi, *Kolmogorov Random Graphs and the Incompressibility Method*, SIAM Journal on Computing, available at `http://doi.org/10.1137/S0097539797327805`.

[10] Cristian S. Calude, *Information and Randomness: An algorithmmic perspective*, 2nd ed., Springer-Verlag, 2002.

[11] _____, *Information: The Algorithmic Paradigm*, Formal theories of information, 2009, pp. 79–94.

[12] Cristian S. Calude, Peter H. Hertling, Bakhadyr Khoussainov, and Yongge Wang, *Recursively enumerable reals and Chaitin Ω numbers*, Theoretical Computer Science **255** (2001mar), no. 1-2, 125–149, available at `http://doi.org/10.1016/S0304-3975(99)00159-0`.

[13] Gregory Chaitin, *A Theory of Program Size Formally Identical to Information Theory*, Journal of the ACM **22** (1975), no. 3, 329–340.

[14] _____, *Algorithmic Information Theory*, 3rd ed., Cambridge University Press, 2004.

[15] Qi Cheng and Fang Fang, *Kolmogorov random graphs only have trivial stable colorings*, Information Processing Letters **81** (2002feb), no. 3, 133–136.

[16] Eduardo Chinelate Costa, Alex Borges Vieira, Klaus Wehmuth, Artur Ziviani, and Ana Paula Couto da Silva, *Time Centrality in Dynamic Complex Networks*, Advances in Complex Systems **18** (2015), no. 07n08, available at `http://arxiv.org/abs/1504.00241http://dx.doi.org/10.1142/S021952591550023X`.

[17] Matthias Dehmer, *Information-Theoretic Concepts for the Analysis of Complex Networks*, Applied Artificial Intelligence **22** (2008aug), no. 7-8, 684–706.

[18] Matthias Dehmer, Frank Emmert-Streib, and Alexander Mehler (eds.), *Towards an Information Theory of Complex Networks*, Birkhäuser Boston, Boston, 2011.

[19] Reinhard Diestel, *Graph Theory*, 5th ed., Graduate Texts in Mathematics, vol. 173, Springer-Verlag Berlin Heidelberg, 2017 (English).

[20] Rodney G. Downey and Denis R. Hirschfeldt, *Algorithmic Randomness and Complexity*, Theory and Applications of Computability, Springer New York, New York, NY, 2010.

[21] Peter Gács, *Exact Expressions for Some Randomness Tests*, Zeitschrift für Mathematische Logik und Grundlagen der Mathematik **26** (1980), no. 25-27, 385–394, available at `http://doi.org/10.1002/malq.19800262502`.

[22] Carlos Gershenson and Mikhail Prokopenko, *Complex Networks*, Artificial Life **17** (2011oct), no. 4, 259–261.

[23] Peter D. Grünwald and Paul M. B. Vitányi, *Algorithmic Information Theory*, Philosophy of information, 2008, pp. 281–317.

[24] Ted G. Lewis, *Network Science*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2009.

[25] Ming Li and Paul Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications*, 2nd ed., Springer Science & Business Media, New York, 1997.

[26] Joseph T. Lizier, Mikhail Prokopenko, and Albert Y. Zomaya, *A Framework for the Local Information Dynamics of Distributed Computation in Complex Systems*, Guided self-organization, 2014, pp. 115–158.

[27] Joseph S. Miller and Liang Yu, *On initial segment complexity and degrees of randomness*, Transactions of the American Mathematical Society **360** (2008jan), no. 06, 3193–3211, available at `http://doi.org/10.1090/S0002-9947-08-04395-X`.

[28] Abbe Mowshowitz and Matthias Dehmer, *Entropy and the complexity of graphs revisited*, Entropy **14** (2012mar), no. 3, 559–570, available at `http://doi.org/10.3390/e14030559`.

[29] Hartley Rogers Jr., *Theory of Recursive Functions and Effective Computability*, MIT Press, Cambridge, MA, USA, 1987.

[30] R Solé and Sergi Valverde, *Information theory of complex networks: On evolution and architectural constraints*, Complex networks, 2004aug, pp. 189–207.

[31] Klaus Wehmuth, Éric Fleury, and Artur Ziviani, *MultiAspect Graphs: Algebraic Representation and Algorithms*, Algorithms **10** (2016dec), no. 1, 1, available at `http://doi.org/10.3390/a10010001`.

[32] _____, *On MultiAspect graphs*, Theoretical Computer Science **651** (2016), 50–61, available at `http://doi.org/10.1016/j.tcs.2016.08.017`.

[33] Klaus Wehmuth and Artur Ziviani, *Centralities in High Order Networks*, 2018 (English).

[34] Klaus Wehmuth, Artur Ziviani, and Eric Fleury, *A unifying model for representing time-varying graphs*, 2015 ieee international conference on data science and advanced analytics (dsaa), 2015oct, pp. 1–10.

[35] Hector Zenil, *Philosophy of Information: The Urgent Need to Move away from Entropy towards Algorithmic Information*, Proceedings **1** (2017jun), no. 3, 104.

[36] Hector Zenil, Narsis Kiani, and Jesper Tegnér, *A Review of Graph and Network Complexity from an Algorithmic Information Perspective*, Entropy **20** (2018jul), no. 8, 551, available at `http://doi.org/10.3390/e20080551`.

[37] Hector Zenil, Narsis A. Kiani, and Jesper Tegnér, *Low-algorithmic-complexity entropy-deceiving graphs*, Physical Review E **96** (2017jul), no. 1, 012308, available at `1608.05972`.

[38] Hector Zenil, Fernando Soler-Toscano, Kamaludin Dingle, and Ard A. Louis, *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 **404** (2014jun), 341–358, available at `1306.0322`.