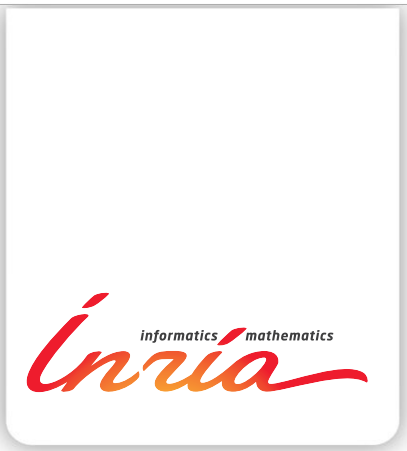


A Unifying Model for Representing Time-Varying Graphs

Klaus Wehmuth, Artur Ziviani, Éric Fleury

RESEARCH REPORT
N° 8466
January 2014
Project-Team DANTE

ISSN 0249-6399 ISRN INRIA/RR--8466--FR+ENG



A Unifying Model for Representing Time-Varying Graphs

Klaus Wehmuth*, Artur Ziviani*, Éric Fleury†

Project-Team DANTE

Research Report n° 8466 — January 2014 — 39 pages

Abstract: We propose a novel model for representing finite discrete Time-Varying Graphs (TVGs). We show how key concepts, such as degree, path, and connectivity, are handled in our model. We also analyze the data structures built following our proposed model and demonstrate that, for most practical cases, the asymptotic memory complexity of our model is restricted to the cardinality of the set of edges. Moreover, we prove that if the TVG nodes can be considered as independent entities at each time instant, the analyzed TVG is isomorphic to a directed static graph. This is an important theoretical result since this allows the use of the isomorphic directed graph as a tool to analyze both the properties of a TVG and the behavior of dynamic processes over a TVG. We also show that our unifying model can represent several previous (classes of) models for dynamic networks found in the recent literature, which in general are unable to represent each other. In contrast to previous models, our proposal is also able to intrinsically model cyclic (i.e. periodic) behavior in dynamic networks. These representation capabilities attest the expressive power of our proposed unifying model for TVGs.

Key-words: dynamic networks, temporal networks, graphs, dynamic graphs, complex networks

* National Laboratory for Scientific Computing (LNCC) – Av. Getúlio Vargas, 333 – 25651-075 - Petrópolis, RJ - Brazi

† LIP – UMR CNRS 5668 – Ecole Normale Supérieure de Lyon (ENS de Lyon) / INRIA – 46, allée d'Italie 69364 – Lyon Cedex 07, France

RESEARCH CENTRE GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Modèle de représentation des graphes temporels

Résumé : Nous proposons un modèle (TVG pour *Time-Varying Graphs*) pour représenter les graphes dynamiques (*i.e.*, des graphes susceptibles d'évoluer au cours du temps). Nous montrons qu elles définitionsclefs comme le degré, la notion de chemin, de connectivité sont prise en compte par ce modèle. Une analyse de la complexité des structures de données nécessaire à la représentation de ce modèle montre que la complexité asymptotique est en $O(m)$ (cardinalité du nombre d'arêtes du graphe dynamique). Si les sommets d'un TVG peuvent être considérés comme des entités indépendantes à chaque instant, alors on démontre que le graphe TVG est isomorphe à un graphe orienté static. Notre modèle permet de représenter et de prendre en compte les différentes propositions existantes qui n'étaient pas en mesure de se représenter les unes les autres.

Mots-clés : réseaux dynamiques, réseaux temporels, graphes, graphes dynamique, réseaux complexes

1 Introduction

Graph theory has been used for the analysis of several networked systems, being at the core of the new field of Network Science. Much of the utility of the graph abstraction resides in the fact that it can represent relations between a set of objects as well as their connectivity properties, which derive from edge transitivity in a straightforward way without the need of further assumptions that are not explicit in the graph abstraction itself. In this context, there is a lot of studies focusing on investigating the behavior of dynamic processes, such as random walks or information diffusion, over complex networks [PSV01, KKT03, BBV08, IFM09].

More recently, there is an increasing interest in investigating not only the process dynamics on networks, but also the dynamics of networks, i.e. when the network structure (nodes and edges) may vary over time [GDM09, FNR+12, HS13]. This, however, brings a difficulty since the original graph abstraction was not originally created considering time relations between nodes. As a consequence, the need to extend the basic graph abstraction in order to include time relations between nodes arose, leading to many models for Time-Varying Graphs (TVGs) [BXFJ03, Fer04, Hol05, Kos09, TMML09, HS12, CFQS12, KA12].

As recent models appear extending the basic graph concept to include time relations (Section 4 discusses related work), they are nonetheless not general enough to satisfy the needs of different networked systems and also in many cases rely on assumptions that are not explicitly part of the model. For instance, in models based on snapshots (i.e., a series of static graphs), such as those in [BXFJ03, Fer04], it is implicitly assumed that a node in a given snapshot is connected to 'itself' in the next snapshot, making it possible to extend the transitivity of edges over time. This assumption, however, is not made explicit in the snapshot model. Therefore, when analysed without this implicit assumption, a snapshot model is a sequence of disconnected graphs and therefore no connectivity is possible between different time instants. The need to handle this assumption, which is not explicitly part of the model, brings difficulties since the structure of the model by itself is no longer sufficient to properly represent its behavior, making the understanding, usage, and analysis of such models more complex.

In this paper, we propose a new unifying model for representing finite discrete TVGs. Our proposed model is sufficiently general to capture the needs of distinct dynamic networks [AHFVZ12, SVZ12, HS12, GaVSZ13], whereas not requiring any further assumption that is not explicitly contained in the model itself. Further, our model aims at preserving the strictly discrete nature of the basic graph abstraction, while also allowing to properly represent time relations between nodes. Moreover, we prove that if the TVG nodes can be considered as independent entities at each time instant, the analyzed TVG is isomorphic to a directed static graph. This is an important theoretical result because this allows the use of the isomorphic directed graph as a tool to analyze both the properties of a TVG and the behavior of dynamic processes over a TVG. We also demonstrate that, for most practical cases, the asymptotic memory complexity of our TVG model is determined by the cardinality of the set of edges. Furthermore, we also show the unifying properties of our proposed model for representing TVGs by describing how it represents several previous (classes of) models for dynamic networks found in the recent literature, which in general are unable to represent each other. In contrast to previous models, our proposal is able to intrinsically model cyclic (i.e. periodic) behavior in dynamic networks. These representation features attest the expressive power of our proposed unifying model for TVGs.

This paper proceeds as follows. Section 2 introduces our proposed unifying model for representing TVGs and its main properties. Section 3 discusses data structures to properly represent TVGs using our model. In Section 4, we show how our unifying model can be used to represent previous models for dynamic networks while these models in general are unable to represent each other. Finally, we conclude in Section 5.

2 Proposed model for representing TVGs and its main properties

Time-varying graphs (TVGs) are graphs in which nodes, or edges may vary in time. In this section, we formally define our proposed model for representing TVGs and present its main properties as well. To that end, we use the same notation adopted by [BJG08] for directed graphs.

2.1 Proposed model for representing TVGs

Our proposed model represents a TVG as an object $H = (V, E, T)$, where V is the set of nodes, T is the finite set of time instants for which the TVG is defined, and $E \subseteq V \times T \times V \times T$ is the set of edges. As a matter of notation, we denote $V(H)$ as the set of all nodes in H, $E(H)$ the set of all edges in H, and $T(H)$ the set of all time instants in H. We also define $n(H) = |V(H)|$ the number of nodes in H, $m(H) = |E(H)|$ the number of edges in H, and $\tau(H) = |T(H)|$ the number of time instants in which H is defined.

A *dynamic edge* e in a TVG H is defined as an ordered quadruple $e = (u, t_a, v, t_b)$, where $u, v \in V(H)$ are the origin and destination nodes (u possibly equal to v) while $t_a, t_b \in T(H)$ are the origin and destination time instants, respectively (t_a possibly equal to t_b). Therefore, the dynamic edge $e = (u, t_a, v, t_b)$ should be understood as a connection from node u at time t_a to node v at time t_b. As hinted by its temporal nature, a dynamic edge is a directed edge. If one needs to represent an undirected edge in the TVG, both (u, t_a, v, t_b) and its reciprocal (v, t_b, u, t_a) should be present in $E(H)$. As a matter of notation, in the remaining of the paper, a TVG will be represented by a upper case letter, usually H or K, a node will be represented by a lower case letter, usually u, v, r or s, a time instant will be represented as t_a, where the index a denotes its position in time, and a dynamic edge will be represented as the ordered quadruple (u, t_a, v, t_b) (or in a shorter form by the letter e).

We define four canonical projections, each projection mapping a dynamic edge into each one of its components:

$$\begin{aligned} \pi_1 : E(H) &\rightarrow V(H) \\ (u, t_a, v, t_b) &\mapsto u, \end{aligned}$$

$$\begin{aligned} \pi_2 : E(H) &\rightarrow T(H) \\ (u, t_a, v, t_b) &\mapsto t_a, \end{aligned}$$

$$\begin{aligned} \pi_3 : E(H) &\rightarrow V(H) \\ (u, t_a, v, t_b) &\mapsto v, \end{aligned}$$

$$\begin{aligned} \pi_4 : E(H) &\rightarrow T(H) \\ (u, t_a, v, t_b) &\mapsto t_b. \end{aligned}$$

One may decide to classify an edge $e = (u, t_a, v, t_b)$ into four classes depending on its temporal characteristic:

1. *Spatial edges* connect two nodes at the same time instant, e is in the form of $e = (u, t_a, v, t_a)$, where $u \neq v$;

2. *Temporal edges* connect the same node at two distinct time instants, e is in the form of $e = (u, t_a, u, t_b)$, where $t_a \neq t_b$;

3. *Mixed edges* connect distinct nodes in distinct time instants, e is in the form of $e = (u, t_a, v, t_b)$, where $u \neq v$ and $t_a \neq t_b$;

4. *Spatial-temporal self-loop edges* connect the same node at the same time instant, e is in the form of, $e = (u, t_a, u, t_a)$.

We define a *temporal node* as an ordered pair (u, t_a), where $u \in V(H)$ is a node and $t_a \in T(H)$ is a time instant. A temporal node is the representation of a given node at a given time instant. The set of temporal nodes in a TVG H is given by $V(H) \times T(H)$, the cartesian product of the set of nodes and the set of time instants. We denote $VT(H) = V(H) \times T(H)$ as the set of all temporal nodes on TVG H. As a matter of notation, a temporal node is represented by the ordered pair that defines it, e.g. (u, t_a), or as u_{t_a} for a short notation. We use the canonical projections $\pi_1(u_t)$ and $\pi_2(u_t)$ to extract the node and time instant that compose the temporal node. We also use the notation $(u, \cdot)$ to denote a node u at any time instant, which is indeed equivalent to stating that node $u \in V$.

The definition of a TVG H is as general as possible and does not impose any order on the time set $T(H)$. One may want to stick to the classical time notion and impose a total order on $T(H)$. Within such a context where $T(H)$ has a linear order, both mixed or temporal dynamic edges $e = (u, t_a, v, t_b)$ can also be classified as progressive or regressive depending on the order of their temporal components. Dynamic edges that are originated at an earlier time instant and destined to a later time instant are progressive ($t_a < t_b$), whereas dynamic edges originated at a later time instant and destined to an earlier time instant are regressive ($t_a > t_b$). Regressive edges are particularly useful for creating cyclic TVGs, which in turn can be applied to model networks with a cyclic periodic behavior. A simple example of this is a wireless DTN (Delay/Disruption-Tolerant Network) [Fal03] with n nodes, out of which $n - 1$ nodes are fixed and mutually disconnected, while the remaining node is a mobile one that behaves on a cyclic pattern connecting to a single fixed node at each time instant. A similar scheme can be used to model Wireless Sensor Networks (WSNs) with a mobile sink [CKN08, VVZF10] that regularly visits the sensor nodes to gather the most recent monitored information.

A sub-TVG J of the TVG H is defined in a straightforward way: $J = (V, E, T), V(J) \subseteq V(H), E(J) \subseteq E(H), T(J) \subseteq T(H)$, such that for all $e \in E(J)$, $\pi_1(e), \pi_3(e) \in V(J)$ and $\pi_2(e), \pi_4(e) \in T(J)$.

2.2 TVG isomorphism

We define the TVG isomorphism as an extension of the concept of the graph isomorphism. Two TVGs H and K are isomorphic if there is a pair of bijective functions f and g, where $f : V(H) \to V(K)$ and $g : T(H) \to T(K)$, such that a dynamic edge $(u, t_a, v, t_b) \in E(H)$ if and only if the dynamic edge $(f(u), g(t_a), f(v), g(t_b)) \in E(K)$.

Since the TVG isomorphism is an equivalence relation, the set of all TVGs isomorphic to a given TVG H form an equivalence class in the set of all TVGs. This equivalence relation partitions the set of all TVGs. Further, since the functions f and g are bijections, it follows that if two TVGs H and K are isomorphic, they necessarily have the same number of nodes and the same number of time instants, i.e. $|V(H)| = |V(K)|$ and $|T(H)| = |T(K)|$. From the requirement that a dynamic edge (u, t_a, v, t_b) exists in H if and only if the dynamic edge $(f(u), g(t_a), f(v), g(t_b))$ exists in K, it can be seen that two isomorphic TVGs also have the same number of dynamic edges, i.e. $|E(H)| = |E(K)|$.

The TVG isomorphism is not a time order preserving isomorphism. A time order preserving isomorphism can be obtained by further requiring the sets $T(H)$ and $T(K)$ to have a linear order and the function g to be an order isomorphism, which is an order preserving bijection whose inverse is also order preserving. Hence, for any $t_a, t_b \in T(H)$ we have that $t_a < t_b$ if and only if $g(t_a) < g(t_b)$. Note that since g is a bijection, it follows that $t_a = t_b$ if and only if $g(t_a) = g(t_b)$.

2.3 Isomorphism between TVGs and directed graphs

In this section, we show that there is an isomorphism between TVGs and directed graphs. We also discuss the circumstances under which this isomorphism holds and the properties it preserves.

Theorem 2.1 *For every TVG H with n nodes and τ time instants, where $n > 1$ and $\tau > 1$, there is a directed graph G with $n \times \tau$ nodes which is isomorphic to H. This directed graph G is unique up to a graph isomorphism.*

Proof 2.1 *We show that for any given TVG H with n nodes and τ time instants, there is a unique (up to a graph isomorphism) directed graph G with $n \times \tau$ nodes for which there is a bijective function $f : VT(H) \to V(G)$, such that any dynamic edge $(u, t_a, v, t_b) \in E(H)$ if and only if the edge $(f((u, t_a)), f((v, t_b))) \in E(G)$.*

- *Existence of G:*

  Given an arbitrary TVG H with n nodes and τ time instants, we construct a directed graph G that satisfies the isomorphism conditions. We start with a graph G with $n \times \tau$ nodes and no edges ($E(G) = \emptyset$). Note that the number of nodes in G equals the number of elements in the set $VT(H) = V(H) \times T(H)$, i.e. $|V(G)| = |VT(H)| = n \times \tau$. We then take an arbitrary bijective function $f : VT(H) \to V(G)$. Since by construction the sets $VT(H)$ and $V(G)$ have the same number of elements, such bijection exists. Finally, for every dynamic edge $(u, t_a, v, t_b) \in E(H)$, we add an edge $(f((u, t_a)), f((v, t_b)))$ to $E(G)$. Since f is injective, it follows that if $(a, t_b) \neq (c, t_d)$, then $f((a, t_b)) \neq f((c, t_d))$. Therefore each distinct dynamic edge (u, t_a, v, t_b) is mapped to a distinct edge $(f((u, t_a)), f((v, t_b)))$. As the only edges in $E(G)$ are the ones mapped from dynamic edges in $E(H)$, it follows that $(u, t_a, v, t_b) \in E(H)$ if and only if $(f((u, t_a)), f((v, t_b))) \in E(G)$, as required. Note that by this property we have that the number of dynamic edges in H is the same as the number of edges in G, i.e. $|E(H)| = |E(G)|$. This gives us a directed graph G and a bijective function f that satisfy the isomorphism requirements. Therefore, we have shown that the required graph G exists.

- *Uniqueness of G:*

  Let's assume that in addition to the TVG H, the directed graph G, and the bijective function f described above, we also have another directed graph J with $n \times \tau$ nodes and a bijective function $j : VT(H) \to V(J)$, such that any dynamic edge $(u, t_a, v, t_b) \in E(H)$ if and only if the dynamic edge $(j((u, t_a)), j((v, t_b))) \in E(J)$. Since both f and j are bijective functions, it follows that the composite function $(j \circ f^{-1}) : V(G) \to V(J)$ is also a bijection. Further, from the definitions of f and j, it follows that the nodes $w, z \in V(G)$ are adjacent in G if and only if the nodes $(j \circ f^{-1})(w), (j \circ f^{-1})(z) \in V(J)$ are adjacent in J. To observe this, note that if the edge $(w, z) \in E(G)$, then the dynamic edge [1] $(f^{-1}(w), f^{-1}(z)) \in E(H)$ and, as a consequence, the edge $(j(f^{-1}(w)), j(f^{-1}(z))) \in E(J)$. The converse follows from the

[1] This is in fact a slight abuse of notation, since $(f^{-1}(w)$ and $(f^{-1}(z)$ are ordered pairs. The formally correct notation is $(\pi_1(f^{-1}(w)), \pi_2(f^{-1}(w)), \pi_1(f^{-1}(z)), \pi_2(f^{-1}(z)))$.

same argument applied to an edge in $E(J)$. Therefore, G and J are isomorphic directed graphs, and thus G is unique up to a graph isomorphism.

Given the existence and uniqueness of the directed graph G, the existence of the function f, and since H is an arbitrary TVG with n nodes and τ time instants, we conclude that the theorem holds.

Note that once a bijective function f is determined, all the permutations of f are also bijective functions from $VT(H)$ to $V(G)$. Of course, as shown in the second part of the proof of Theorem 2.1, all the graphs generated by these permutations are isomorphic to each other as well.

An intuitive interpretation of this isomorphism is that the temporal information contained on the dynamic edges and by extent in the set $T(H)$ can be injected into the node set of a directed graph which can be constructed in a way to preserve the edge structure of the TVG, i.e. the relations between nodes and time instants present in the TVG.

Since this isomorphism preserves the edge structure of the TVG, the isomorphic directed graph can be used as a tool to analyze properties of the TVG. However, it should be noted that this isomorphism can not preserve all properties of the TVG (since it is not an identity) and, therefore, care should be taken on its use, to make sure that the properties preserved are sufficient to justify the results obtained. For example, a walk on the TVG corresponds to a walk on the temporal node representation of the TVG (see Section 2.8).

Corollary 2.1 *Given a TVG H and a directed graph G isomorphic to H, there is a bijective function from $E(H)$ to $E(G)$, built upon the isomorphism characterizing the bijection f, and which takes each dynamic edge in H to its corresponding edge in G.*

Proof 2.2 *Let H be a TVG and G a directed graph isomorphic to H. Since H and G are isomorphic, a dynamic edge (u, t_a, v, t_b) belongs to $E(H)$ if and only if a corresponding edge $(f((u, t_a)), f((v, t_b)))$ belongs to $E(G)$, and the function f is a bijection from $VT(H)$ to $V(J)$. Consider the following function*

$$\begin{aligned} h : E(H) &\rightarrow E(G) \\ (u, t_a, v, t_b) &\mapsto (f(u, t_a), f(v, t_b)). \end{aligned}$$

First, note that h is indeed a function, since it has a properly defined domain $E(H)$, codomain $E(G)$, and association rule $(u, t_a, v, t_b) \mapsto ((u, t_a), (v, t_b))$. Further, note that this association rule is valid since for each $e \in E(H)$ there is an element $e_s \in E(G)$ such that $e_s = h(e)$. This is true because since H and G are isomorphic, it follows that if $e = (u, t_a, v, t_b) \in E(H)$ then $e_s = (f(u, t_a), f(v, t_b)) \in E(G)$. Furthermore, note that h is defined in terms of the bijection f used to construct the isomorphism between H and G and also that h associates each dynamic edge of H with the edge in G corresponding to it in accordance with the isomorphism between H and G.

We now show that the function h is injective. Let $e_1, e_2 \in E(H)$. Without loss of generality, we can consider that $e_1 = (u, t_a, v, t_b)$ and $e_2 = (r, t_c, s, t_d)$. We intend to show that if $h(e_1) = h(e_2)$ then $e_1 = e_2$. If $h(e_1) = h(e_2)$, then $(f(u, t_a), f(v, t_b)) = (f(r, t_c), f(s, t_d))$, so that $f(u, t_a) = f(r, t_c)$ and $f(v, t_b) = f(s, t_d)$. Since f is bijective, it follows that $u = r$, $t_a = t_c$, $v = s$ and $t_b = t_d$. Therefore, $e_1 = e_2$ and so h is injective.

Further, we show that h is surjective. Let $e_s \in G$ be any edge in the directed graph G. Without loss of generality, we can assume that $e_s = (w, z)$, where $w, z \in V(G)$ are nodes of the directed graph G. Since G is isomorphic to H, it follows that there is a dynamic edge

$e = (u, t_a, v, t_b) \in E(H)$, such that $w = f((u, t_a))$ and $z = f((v, t_b))$. Then, by the definition of h, it follows that $e_s = (w, z) = h(e) = (f(u, t_a), f(v, t_b))$. Therefore, for any given edge $e_s \in E(G)$, there is a dynamic edge $e \in E(H)$ such that $e_s = h(e)$, and so h is a surjective function.

Since the function h associates each dynamic edge $e \in E(H)$ with its isomorphic corresponding edge $e_s \in G$, and h is both injective and surjective, the corollary holds.

2.3.1 Order preserving isomorphism

From the TVG definition used in this work, we have that the set $T(H)$ containing the time instants at which the TVG is defined is not required to be an ordered set. As a consequence, the isomorphism defined in Theorem 2.1 does not necessarily preserves time ordering. If, however, a time order preserving isomorphism is required, it can be obtained by imposing additional requirements to the involved sets and functions as explained next.

First, we need the sets $T(H)$ and $VT(H)$ to be totally ordered sets, such that for all $t_a, t_b \in T(H)$ and all $u, v \in V(H)$, $t_a < t_b$ if and only if $(u, t_a) < (v, t_b)$. Since this can not be done directly, we define an equivalence relation in $VT(H)$, partitioning it into equivalence classes and then create an order on these equivalence classes. To achieve this, we define an equivalence relation $\sim$, such that $(u, t_a) \sim (v, t_b)$ if and only if $t_a = t_b$. The equivalence relation $\sim$ partitions the set $VT(H)$ into τ equivalence classes, such that two elements of $VT(H)$ are on the same equivalence class if and only if they have the same temporal component. Let $\mathbb{T}(H)$ be the set of equivalence classes generated by $\sim$ on $VT(H)$ and $\Gamma_{t_a} \in \mathbb{T}(H)$ the equivalence class of ordered pairs (u, t_x), where $t_x = t_a$. By construction, each equivalence class $\Gamma \in \mathbb{T}(H)$ has the same number of elements, $|\Gamma| = |V(H)| = n$, equal to the number of nodes in the TVG H. The set $\mathbb{T}(H)$ can be totally ordered in the way desired.

Second, we define a partition on the set $V(G)$ that has τ equivalence classes and each equivalence class contains $|V(H)|$ elements. This is possible since $|V(G)| = |VT(H)|$. Let $\mathbb{T}(G)$ be the set of equivalence classes in this partition. Note that $|\mathbb{T}(G)| = |\mathbb{T}(H)|$. We then define a total order over $\mathbb{T}(G)$. This establishes an equivalence relation on $V(G)$, which we denote by $\approx$.

Finally, we define a bijection $f : VT(H) \to V(G)$, which not only satisfies the condition for isomorphism (i.e., $(u, t_a, v, t_b) \in E(H)$ if and only if $(f((u, t_a)), f((v, t_b))) \in E(G)$), but also is consistent with the partitions in $VT(H)$ and $V(G)$ as well as preserves the order of these partitions. Formally, the condition of being consistent with the partitions on $VT(H)$ and $V(G)$ can be written as: for all $t_a, t_b \in T(H)$ and all $u, v \in V(H)$, $(u, t_a) \sim (v, t_b)$ if and only if $f((u, t_a)) \approx f((v, t_b))$. The order preservation condition can be written as: for all $t_a, t_b \in T(H)$ and all $u, v \in V(H)$, $(u, t_a) < (v, t_b)$ if and only if $f((u, t_a)) < f((v, t_b))$.

It remains to be shown that such an order preserving isomorphism in fact exists. We do this by example, showing that the natural isomorphism defined next in Section 2.3.2 is in fact an order preserving isomorphism.

2.3.2 Natural isomorphism

A special case can be constructed, which characterizes a natural isomorphism (i.e. a natural choice of isomorphism), by making the directed graph G such that its node set $V(G)$ is equal to the set $VT(H)$ and using the identity function $I : VT(H) \to V(G)$ as the bijective function to characterize the isomorphism.

Note that, in this case, the edges added to the graph G by the process described in Theorem 2.1 are such that for every dynamic edge $(u, t_a, v, t_b) \in E(H)$ an edge $((u, t_a), (v, t_b)) \in E(G)$ exists in the directed graph G.

To see that the natural isomorphism is also an order preserving isomorphism, note that in this case $VT(H) = V(G)$ such that both sets can be partitioned in the same way and both partitions can be ordered in the same way. By doing this, the identity function naturally is consistent with the partitions and preserves order as required in Section 2.3.1.

2.3.3 Generalized isomorphism

At this point, an interesting supposition is the existence of an isomorphism between the set of all TVGs with finite nodes and time instants to the set of all directed graphs with finite nodes. This supposition is equivalent to state that there is a bijective function from the set of all TVGs to the set of all directed graphs, such that each TVG is associated to a unique graph that preserves the edge topology of the TVG and also that, in the same manner, each graph is associated to a unique TVG, always up to a graph and TVG isomorphism.

This supposition, however, is false. It is possible to associate each TVG with a unique graph that preserves its edge structure, but it is not possible to associate each graph with a unique TVG. This means that there is a function like the one proposed by the initial supposition, but this function is rather surjective and not injective. Therefore, no generalized isomorphism is possible.

To note why this is indeed the case, note that a given directed graph with p nodes can be associated to TVGs with different dimensions, depending on the prime decomposition of p. Consider for instance a directed graph with $p = 21$, i.e., a directed graph with 21 nodes. This directed graph could be associated with a TVG with 7 nodes and 3 time instants, but also to a TVG with 3 nodes and 7 time instants. Still, there are the two trivial cases of a TVG with 1 node and 21 time instants or a TVG with 21 nodes and 1 time instant—in this case, the directed graph itself. In particular, if p is a prime number, only these two trivial cases are possible.

2.4 TVG representation by temporal nodes

In this section, we show that it is possible to create a representation of any given TVG using temporal nodes. This is equivalent to the natural isomorphism between TVGs and directed networks, presented in Section 2.3.2.

From the definition of temporal nodes, we have that given a TVG H, a temporal node is defined as $(u, t_a) \in VT(H) = V(H) \times T(H)$. Therefore, if $VT(H)$ is considered as the node set of a directed graph, it follows that an edge on this graph is an element of the set $VT(H) \times VT(H)$, which is an ordered pair of temporal nodes. For instance, an edge between the temporal nodes (u, t_a) and (v, t_b) is represented as the ordered pair $((u, t_a), (v, t_b))$ of temporal nodes.

In this environment, given a TVG H, it is straightforward to create a directed graph $G = (VT(H), ET(H))$, where $VT(H)$ is the temporal nodes set, and $ET(H)$ is obtained from the set $E(H)$ of dynamic edges of the TVG H, such that for all $u, v \in V(H)$ and all $t_a, t_b \in T(H)$, the edge $((u, t_a), (v, t_b)) \in ET(H)$ if and only if the dynamic edge $(u, t_a, v, t_b) \in E(H)$. Another way for obtaining the set $ET(H)$ is by using the bijective function h defined in Corollary 2.1. Note that since in this environment the function f used to define the isomorphism between H and G is the identity function $I : VT(H) \to VT(H)$, we have that the function h related to the isomorphism between $E(H)$ and $ET(H)$ (see Corollary 2.1) is written as

$$\begin{aligned} h : E(H) &\to ET(H) \\ (u, t_a, v, t_b) &\mapsto ((u, t_a), (v, t_b)), \end{aligned}$$

where $ET(H) \subseteq VT(H) \times VT(H)$.

To see that the graph $G = (VT(H), ET(H))$ is indeed the graph obtained by the natural isomorphism discussed in Section 2.3.2, note that since $VT(H) = V(H) \times T(H)$, we have the identity function $I : VT(H) \to VT(H)$, and that by the definition of the temporal node representation, for all $u, v \in V(H)$ and all $t_a, t_b \in T(H)$, the edge $((u, t_a), (v, t_b)) \in ET(H)$ if and only if the dynamic edge $(u, t_a, v, t_b) \in E(H)$.

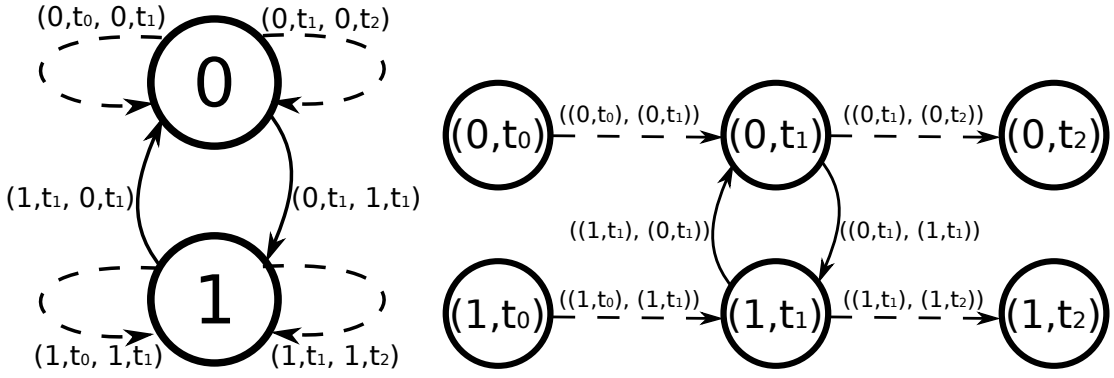
For any given TVG $H = (V, E, T)$, we can now define the function

$$\begin{aligned} g : (V(H), E(H), T(H)) &\to (VT(H), ET(H)) \\ H &\mapsto (I(V(H), T(H)), h(E(H))), \end{aligned}$$

such that $g(H)$ is the temporal node representation of the TVG H.

The main idea behind the temporal node representation is that if the nodes of a TVG are considered as distinct objects at each time instants, then the TVG can be seen as a simple directed graph. This, however, should not be misunderstood as a statement that a TVG and a directed graph are one and the same object. It is important to remember that in order to make this representation possible, the assumption that each node is a distinct object at each time instant has to be made. We show that some of the properties of the TVG are in fact preserved in this graph representation, which can make the analysis of the TVG easier. Therefore, this representation comes with a caveat, indicating that proper care should be taken to apply this representation only in the cases in which the properties under study are preserved under the temporal node representation of the TVG.

Figure 1(a) shows an illustrative TVG on its native form and Figure 1(b) shows the same illustrative TVG in the form of the temporal node representation. Note that the temporal node representation has six (2×3) temporal nodes and same number of edges as the TVG in its native representation. Further, the edges in the TVG and in the graph only differ in their representation. In the native representation of the illustrative TVG, the dynamic edges are represented as an ordered quadruple, whereas in the temporal node representation the corresponding edge is represented as a pair of temporal nodes. Additionally, Figure 1(b) also shows the temporal nodes grouped in a way consistent with the equivalence relation $\sim$ defined in Section 2.3.1, which groups elements with the same temporal coordinate in equivalence classes. From Figure 1(b), it can also be seen how the temporal node representation preserves the temporal order of the original TVG.



(a) Native representation of an illustrative TVG

(b) Temporal node representation of an illustrative TVG

Figure 1: Native and temporal node representations of an illustrative TVG.

2.5 Aggregated directed graph

The aggregated graph associated to a TVG is a directed graph created by projecting all dynamic edges onto a single graph. In order to obtain the aggregated graph of a TVG, we first define the projection

$$\begin{aligned} \pi : V \times T \times V \times T &\rightarrow V \times V \\ (u, t_a, v, t_b) &\mapsto (u, v), \end{aligned}$$

that takes a dynamic edge to an edge by simply dropping the time coordinates of the dynamic edge.

Using this projection, we can now define the function

$$\begin{aligned} Agg : (V, E, T) &\rightarrow (V, V \times V) \\ H &\mapsto (I(V(H)), \pi(E(H)), \end{aligned}$$

where I is the identity function in $V(H)$. It follows from this definition that for a given TVG H, $Agg(H)$ is a directed graph such that if $(u, t_a, v, t_b) \in E(H)$, then $(u, v) \in E(Agg(H))$. In this transformation, spatial and mixed edges are mapped to an edge on the aggregated graph, while temporal edges are contracted into a single node. In this way, temporal edges are ignored, which is consistent with the definition that the aggregated graph is a simple directed graph.

Although the concept of aggregated graph is often found in the TVG literature, it should be noted that for a given TVG H, the nodes of the graph $Agg(H)$ in general do not have the same degree of the nodes on H. It should also be noted that the existence of a path connecting two nodes u and v in $Agg(H)$ does not imply the existence of a path connecting nodes u and v on the TVG H (a path in a TVG is formally defined in Section 2.8.3).

In general, the aggregated graph of a TVG is not isomorphic or in any way equivalent to the temporal node representation of the TVG. To see that this is the case, it suffices to note that for a given TVG H, the temporal node representation of H has $|V(H)| \times |T(H)|$ nodes, while $Agg(H)$ has $|V(H)|$ nodes.

2.6 Degree

Since dynamic edges are naturally directed, we adopt the same notation as in directed graphs of the indegree of a node u denoted as $deg^-(u)$ and the outdegree of a node u denoted as $deg^+(u)$. In addition, we distinguish between node degree and temporal node degree.

In order to properly define the node degree of a TVG, we use the canonical projections defined in Section 2.1. We then define the node outdegree and indegree as

$$\begin{aligned} deg^+(u) &= |\{e \in E(H) | \pi_1(e) = u\}|, \\ deg^-(u) &= |\{e \in E(H) | \pi_3(e) = u\}|, \end{aligned}$$

where $deg^+(u)$ is the number of dynamic edges originating at node u and $deg^-(u)$ is the number of dynamic edges destined to the node u.

The temporal node degree considers the degree of a node at each time instant. This follows directly from the definition of temporal node:

$$\begin{aligned} deg^+((u, t)) &= |\{e \in E(H) | \pi_1(e) = u, \pi_2(e) = t\}|, \\ deg^-((u, t)) &= |\{e \in E(H) | \pi_3(e) = u, \pi_4(e) = t\}|. \end{aligned}$$

That is, $deg^+((u,t))$ is the number of dynamic edges originated at a node u at the time instant t, while $deg^-((u,t))$ is the number of dynamic edges destined to node u at time instant t. It follows from these definitions that for any given TVG H, the outdegree of a node u is given by

$$deg^+(u) = \sum_{t \in T(H)} deg^+((u,t)),$$

and its indegree is given by

$$deg^-(u) = \sum_{t \in T(H)} deg^-((u,t)).$$

Further, we have that

$$|E(H)| = \sum_{u \in V(H)} deg^+(u) = \sum_{u \in V(H)} \sum_{t \in T(H)} deg^+((u,t)),$$

and

$$|E(H)| = \sum_{u \in V(H)} deg^-(u) = \sum_{u \in V(H)} \sum_{t \in T(H)} deg^-((u,t)).$$

2.7 Adjacency

The concept of adjacency in a TVG can be defined in terms of node and edge adjacencies:

- *Node adjacency* establishes a relation between nodes, where two nodes are considered adjacent if and only if they share a common dynamic edge, i.e. a dynamic edge is incident to both nodes. Therefore, node adjacency is equivalent to the existence of a dynamic edge between the nodes. In other words, in a given TVG H, two nodes $u, v \in V(H)$ are adjacent if and only if there is at least one dynamic edge $e \in E(H)$ such that $u = \pi_1(e)$ and $v = \pi_3(e)$. Note that there is no time constraint in the concept of node adjacency. If a dynamic edge is incident to two nodes, they are adjacent nodes regardless of the time instants at which the dynamic edge is incident to each one of them.

- *Temporal node adjacency* establishes a relation between temporal nodes, i.e. a relation between nodes at specific time instants. As with nodes, temporal nodes are considered adjacent if and only if they share a common dynamic edge. However, differently from the node adjacency, when a dynamic edge is incident to a pair of temporal nodes, the time instants of the dynamic edges matches the time instants of the temporal nodes, both at the origin and the destination. In a given TVG H, two temporal nodes $u_t, v_t \in VT(H)$ are adjacent if and only if there is a dynamic edge $e \in E(H)$ such that $u_t = (\pi_1(e), \pi_2(e))$ and $v_t = (\pi_3(e), \pi_4(e))$.

- *Edge adjacency* defines an edge relation where two dynamic edges are considered adjacent if and only if they are incident to the same node at the same time instant. In a given TVG H, two dynamic edges $e_a, e_b \in E(H)$ are adjacent if and only if there is a temporal node $u_t \in VT(H)$, such that $u_t \in \{(\pi_1(e_a), \pi_2(e_a)), (\pi_3(e_a), \pi_4(e_a))\}$ and $u_t \in \{(\pi_1(e_b), \pi_2(e_b)), (\pi_3(e_b), \pi_4(e_b))\}$.

In order to present the relations between the adjacency in a TVG and the adjacency in its isomorphic directed graph G, we introduce the following theorems.

Theorem 2.2 *Given a TVG H and its isomorphic directed graph G, a pair of temporal nodes $u_t, v_t \in VT(H)$ is adjacent in H if and only if their corresponding nodes in G are adjacent.*

Proof 2.3 *Let H be a TVG, G its isomorphic directed graph and $u_t, v_t \in VT(H)$ a pair of adjacent temporal nodes in H.*

Since u_t and v_t are adjacent, it follows that there is a dynamic edge $e \in E(H)$ such that $u_t = (\pi_1(e), \pi_2(e))$ and $v_t = (\pi_3(e), \pi_4(e))$. Since H and G are isomorphic, it follows from Theorem 2.1 that $e \in E(H)$ if and only if the edge $e_s = (f(u_t), f(v_t)) \in E(G)$. Therefore, since $f(u_t), f(v_t) \in V(G)$ are the nodes in G corresponding to u_t and v_t, and the edge e_s is incident to both of them, the theorem holds.

Theorem 2.3 *Given a TVG H and its isomorphic directed graph G, a pair of nodes $u, v \in V(H)$ is adjacent in H if and only if there is at least one pair of adjacent nodes $r, s \in V(G)$ in the graph G, such that r corresponds to a temporal node u_t for which $u = \pi_1(u_t)$ and s corresponds to a temporal node v_t for which $v = \pi_1(v_t)$.*

Proof 2.4 *Let H be a TVG, G a directed graph isomorphic to H, and $u, v \in V(H)$ a pair of nodes in the TVG H. From Theorem 2.2, we have that a pair of nodes $r, s \in V(G)$ is adjacent if and only if there is a pair of adjacent temporal nodes $u_t, v_t \in VT(H)$. From the definition of temporal node adjacency, u_t and v_t are adjacent if and only if there is a dynamic edge e such that $u_t = (\pi_1(e), \pi_2(e))$ and $v_t = (\pi_3(e), \pi_4(e))$. Further, from the definition of node adjacency, u and v are adjacent if and only if there is a dynamic edge e such that $u = \pi_1(e)$ and $v = \pi_3(e)$. Note that $\pi_1(v_t) = \pi_3(e)$.*

Therefore, since by hypothesis $u = \pi_1(u_t)$ and $v = \pi_1(v_t)$, the theorem holds.

Theorem 2.4 *For a given a TVG H and its isomorphic directed graph G, a pair of dynamic edges in H is adjacent if and only if their corresponding edges are adjacent in G.*

Proof 2.5 *Let H be a TVG and G its isomorphic directed graph as per Theorem 2.1.*

- $\Longrightarrow$
  Let $e_1, e_2 \in E(H)$ be two adjacent dynamic edges in H. Since they are adjacent, by our definition both edges are incident to the same node at the same time instant. Therefore, without loss of generality, we can assume that $e_1 = (u, t_a, v, t_b)$ and $e_2 = (v, t_b, w, t_c)$, making both edges incident to node v at time t_b.
  By Theorem 2.1, it follows that there are two edges $e_{1s}, e_{2s} \in G$, such that $e_{1s} = (f((u, t_a)), f((v, t_b)))$ and $e_{2s} = (f((v, t_b)), f((w, t_c)))$. Since both edges e_{1s} and e_{2s} are incident to the node $f((v, t_b))$, they are adjacent edges in G.
- $\Longleftarrow$
  Let $e_3, e_4 \in E(G)$ be two adjacent edges in G. Since they are adjacent, they are incident to the same node and thus, without loss of generality, they can be written as $e_3 = (a, b)$ and $e_4 = (b, c)$, where $a, b, c \in V(G)$ are nodes in G. Hence, by Theorem 2.1, $f^{-1}(a), f^{-1}(b), f^{-1}(c) \in VT(H)$ are temporal nodes in H and $e_{3t} = (f^{-1}(a), f^{-1}(b))$ and $e_{4t} = (f^{-1}(b), f^{-1}(c))$ are dynamic edges in H. Since both edges are incident to the temporal node $f^{-1}(b)$, it follows that they are incident to the same node at the same time instant, being therefore adjacent dynamic edges in H.

Since the temporal node representation is a particular case of the isomorphism shown in Theorem 2.1, it follows that Theorems 2.2, 2.3, and 2.4 are also valid for the temporal node representation. This property makes it possible to analyze concepts derived from edge adjacency in the temporal node representation of the TVG.

It is well known that if a pair of nodes u and v are adjacent in a TVG, then these nodes are also adjacent in the aggregated graph of the TVG. However, the reciprocal is not true.

For the dynamic edge adjacency, we have the following theorem.

Theorem 2.5 *If two spatial or mixed edges in a TVG H are incident to the same node, then their projections on the aggregate graph are adjacent.*

Proof 2.6 *Let $e_1, e_2 \in E(H)$ be a pair of spatial or mixed dynamic edges in TVG H, such that both are incident to a given node v. Since these edges are spatial or mixed dynamic edges, they have distinct origin and destination nodes. Hence, without loss of generality, we can define $e_1 = (u, t_a, v, t_b)$ and $e_2 = (v, t_c, w, t_d)$, where $u \neq v$ and $v \neq w$ are nodes in TVG H and $t_a, t_b, t_c, t_d \in T(H)$ are time instants. Applying the projection π used to get the aggregate graph (see Section 2.5) to the edges, we get $\pi(e_1) = (u, v)$ and $\pi(e_2) = (v, w)$. Since $u \neq v$ and $v \neq w$, (u, v) and (v, w) are two distinct directed edges, and since both are incident to node v, it follows that they are adjacent edges in $Agg(H)$, the aggregated graph of the TVG H.*

Note that the dynamic edges e_1 and e_2 are not necessarily adjacent in the TVG. For the Theorem 2.5 to hold, it is sufficient that these dynamic edges share the same node, regardless of the time instants. This happens because all information concerning time instants in the dynamic edges is dropped by the projection π. Further, note that the edges e_1 and e_2 do not have to necessarily be both spatial or both mixed.

2.8 Walks, trails, paths, and cycles

In this section, we analyze concepts that are derived from the basic concept of adjacency (see Section 2.7). Since in the environment of a TVG the time relations are considered in the same way as relations between nodes, it is natural to use an adjacency definition which takes time relations into account. Therefore, the concepts of temporal node adjacency and edge adjacency are used because they depend not only on the nodes, but also on time instants to determine if adjacency occurs.

We show in this section that walks, trails, paths, and cycles, as they are defined for TVGs, are preserved under the isomorphism between TVGs and directed graphs (see Section 2.2), and in particular by the temporal node representation (i.e. natural isomorphism between a TVG and a directed graph), so that properties related to walks, trails, paths, and cycles can be analyzed on the temporal node representation of the TVG.

2.8.1 Walk

We define a walk on a TVG H as an alternating sequence $W = [u_{t_1}, e_1, u_{t_2}, e_2, u_{t_3}, ..., u_{t_{k-1}}, e_{k-1}, u_{t_k}]$ of temporal nodes $u_{t_i} \in VT(H)$ and dynamic edges $e_j \in E(H)$, such that $u_{t_i} = (\pi_1(e_i), \pi_2(e_i))$ and $u_{t_{i+1}} = (\pi_3(e_i), \pi_4(e_i))$ for $1 \leq i < k$. Note that from this definition we have that all pairs of consecutive temporal nodes u_{t_i} and $u_{t_{i+1}}$ are adjacent and also that all pairs of consecutive dynamic edges e_j and e_{j+1} are adjacent.

A walk is closed if $u_{t_1} = u_{t_k}$ and open otherwise. The set of temporal nodes in the walk W is denoted as $VT(W)$ and the set of dynamic edges in the walk W is denoted as $E(W)$. Since the dynamic edges in W contain time instants, we can denote the set of all time instants in W as $T(W)$. Further, if W is a walk on a TVG H, then $VT(W) \subseteq VT(H)$, $E(W) \subseteq E(H)$ and $T(W) \subseteq T(H)$.

Note that each dynamic edge e_i in a walk W can be determined by the temporal nodes u_{t_i} and $u_{t_{i+1}}$ by writing $e_i = (\pi_1(u_{t_i}), \pi_2(u_{t_i}), \pi_1(u_{t_{i+1}}), \pi_2(u_{t_{i+1}}))$. Therefore, W can be fully described by the sequence of its temporal nodes, $W_V = [u_{t_1}, u_{t_2}, ..., u_{t_k}]$. We may refer to a walk using this notation in cases where the precise determination of the edges is not needed. The sequence of temporal nodes W_V is not necessarily equal to the set $VT(W)$ of temporal nodes in the walk, since in W_V there may be repeated temporal nodes.

Further, each dynamic edge e_j in a walk W also fully determines the temporal nodes u_{t_j} and $u_{t_{j+1}}$, since $u_{t_j} = (\pi_1(e_j), \pi_2(e_j))$ and $u_{t_{j+1}} = (\pi_4(e_j), \pi_4(e_j))$. Hence, W can also be determined by its sequence of dynamic edges $W_E = [e_1, e_2, ..., e_n]$. We may use this notation when the precise identification of the nodes is not needed. The sequence (or list) of edges W_E is not necessarily equal to the set of edges $E(W)$, since there may be repeated edges in W_E. The length of a walk is determined by the number of dynamic edges the walk contains, i.e. $Len(W) = |W_E|$.

As a short notation, in cases where there is no ambiguity, or the identity of the temporal nodes and dynamic edges in the walk is not relevant, we may also identify a walk W only by its start and end nodes as $W = u_1 \rightarrow u_k$.

Theorem 2.6 *An alternating sequence W of temporal nodes and dynamic edges in a TVG H is a walk on H if and only if there is a corresponding walk G_W in the temporal node representation of H.*

Proof 2.7 *As stated in Section 2.4, for the temporal node representation $g(H)$ of the TVG H there is a bijective function $h : E(H) \rightarrow ET(H) = E(g(H))$ such that $h((u, t_a, v, t_b)) = ((u, t_a), (v, t_b))$.*

- $\Longrightarrow$
  Let $W = [u_{t_1}, e_1, u_{t_2}, e_2, u_{t_3}, ..., u_{t_{k-1}}, e_{k-1}, u_{t_k}]$ be a walk on TVG H and $g(H)$ be the temporal node representation of the TVG H. Since the nodes present in W are temporal nodes, they correspond directly to temporal nodes in $g(H)$, because the bijection between $VT(H)$ and $V(g(H))$ is the identity. Thus, it is only necessary to translate the dynamic edges e_i into edges in $E(g(H))$. Each dynamic edge e_i is determined by the temporal nodes u_{t_i} and $u_{t_{i+1}}$ such that $e_i = (\pi_1(u_{t_i}), \pi_2(u_{t_i}), \pi_1(u_{t_{i+1}}), \pi_2(u_{t_{i+1}}))$. Applying the function h to e_i we have that $h((\pi_1(u_{t_i}), \pi_2(u_{t_i}), \pi_1(u_{t_{i+1}}), \pi_2(u_{t_{i+1}}))) = ((\pi_1(u_{t_i}), \pi_2(u_{t_i})), (\pi_1(u_{t_{i+1}}), \pi_2(u_{t_{i+1}}))) = (u_{t_i}, u_{t_{i+1}}) = e_{s_i}$. Therefore, we have that the sequence $G_W = [u_{t_1}, e_{s_1}, u_{t_2}, e_{s_2}, u_{t_3}, ..., u_{t_{k-1}}, e_{s_{k-1}}, u_{t_k}]$ is the walk corresponding to W in $g(H)$, since it is the walk composed of the nodes in $V(g(H))$ corresponding to the temporal nodes in W.

- $\Longleftarrow$
  Let $G_W = [u_{t_1}, e_{s_1}, u_{t_2}, e_{s_2}, u_{t_3}, ..., u_{t_{k-1}}, e_{s_{k-1}}, u_{t_k}]$ be a walk on $g(H)$. Then $u_{t_i} \in V(g(H))$ are nodes in $g(H)$ and $e_{s_i} \in E(g(H))$ are edges in $g(H)$. By construction of $g(H)$, $V(g(H)) = VT(H)$, such that $u_{t_i} \in VT(H)$ are also temporal nodes in H. Since W is a walk on the graph $g(H)$, it follows that $e_{s_i} = (u_{t_i}, u_{t_{i+1}})$. Applying h^{-1} to e_{s_i}, we have $h^{-1}((u_{t_i}, u_{t_{i+1}})) = (\pi_1(u_{t_i}), \pi_2(u_{t_i}), \pi_1(u_{t_{i+1}})) = e_i$, such that the sequence $W = [u_{t_1}, e_1, u_{t_2}, e_2, u_{t_3}, ..., u_{t_{k-1}}, e_{k-1}, u_{t_k}]$ is the corresponding walk on the TVG H.

Corollary 2.2 *The length of a walk W on a TVG H is the same as the length of the corresponding walk G_W on the directed graph $g(H)$.*

Proof 2.8 *It follows from Theorem 2.6 that the sequence of temporal nodes u_{t_i} in the walk H is the same as the sequence of temporal nodes u_{t_i} in the walk G_W. Since each pair of consecutive temporal nodes u_{t_i} and $u_{t_{i+1}}$ determines a dynamic edge in W and also an edge in G_W, it follows that the number of dynamic edges in W is the same as the number of edges in G_W. Therefore, from the definition of length of a walk, we have that $Len(W) = Len(G_W)$ and the corollary holds.*

Theorem 2.7 *Given a TVG H and its aggregated graph $Agg(H)$, the projection of a walk on H onto $Agg(H)$ is a walk on $Agg(H)$.*

Proof 2.9 *Let H be a TVG, $Agg(H)$ its aggregated directed graph and $W = [u_{t_1}, e_1, u_{t_2}, e_2, u_{t_3}, ..., u_{t_{k-1}}, e_{k-1}, u_{t_k}]$ a walk on TVG H. If the walk W only has spatial or mixed dynamic edges, then this*

theorem holds as a direct consequence of Theorem 2.5, since consecutive edges in a walk are adjacent in the TVG and therefore are incident to the same node. If, however, the walk has temporal dynamic edges, they contract to a single node and, therefore, are not reflected in $Agg(H)$. When this happens, note that the nodes before and after each temporal edge are the same. This happens because a temporal edge has the form $e = (u, t_a, u, t_b)$, connecting a node to itself at different time instants. Therefore, every sequence of temporal dynamic edges can be ignored and the theorem holds.

2.8.2 Trail

We define a trail in a TVG H as a walk on H where all edges are distinct. We can identify a trail $W = [u_{t_1}, e_1, u_{t_2}, e_2, u_{t_3}, ..., u_{t_{k-1}}, e_{k-1}, u_{t_k}]$ with the TVG H_W = (V(W), E(W), T(W)). Since $V(W) \subseteq V(H)$, $E(W) \subseteq E(H)$ and $T(W) \subseteq T(H)$, it follows that the trail W is a sub-TVG of H.

Theorem 2.8 *A walk H_W on H is a trail on H if and only if there is a corresponding trail G_W on $g(H)$.*

Proof 2.10 • $\Longrightarrow$

Let H_W be a trail on H and let $g(H)$ be the temporal node representation of the TVG H. Since H_W is a trail, it is also a walk. Hence, by Theorem 2.6, there is a corresponding walk G_W on $g(H)$. Since H_W is a trail on H, we have that all dynamic edges $e_s \in E(H_W) \subseteq E(H)$ are distinct. From Corollary 2.1, we have that there is a bijection $h : E(H) \to E(g(H))$ and, therefore, since the dynamic edges e_s in H_W are distinct, so are the corresponding edges in G_W. Since G_W is a walk on $g(H)$ and its edges are distinct, it follows that G_W is a trail on $g(H)$.

• $\Longleftarrow$

Let G_W be a trail on $g(H)$, which is the temporal node representation of a TVG H. Since G_W is also a walk on $g(H)$, by Theorem 2.6, there is a corresponding walk H_W on H. Since G_W is a trail, the edges $e_{s_i} \in E(G_W) \subseteq E(g(H))$ are all distinct. As that by Corollary 2.1 there is a bijection $h : E(H) \to E(g(H))$, it follows that the corresponding dynamic edges in the walk H_W are also distinct. Therefore, since H_W is a walk and has distinct edges, it follows that H_W is a trail on the TVG H.

A trail is closed when the first and last temporal nodes are the same, i.e. $u_{t_1} = u_{t_k}$, and open otherwise. A closed trail is also called a tour or a circuit.

From Theorem 2.7, we have that the projection of a trail W onto the aggregated graph of H is a walk. To see that this projection is not necessarily a trail, consider the trail $W = [(u, t_1), (u, t_1, v, t_2), (v, t_2), (v, t_2, w, t_3), (w, t_3), (w, t_3, u, t_4), (u, t_4), (u, t_4, v, t_5), (v, t_5)]$. The projection of W onto $Agg(H)$ is the walk $W_{agg} = [u, (u, v), v, (v, w), w, (w, u), u, (u, v), v]$. Note that the edge (u, v) appears twice in W_{agg}. Therefore, the walk W_{agg} is not a trail.

Since a trail is also a walk, the length of a trail is determined in the same way as the length of a walk, by the number of edges. In particular, it follows from Corollary 2.2 that the length of a trail H_W on a TVG H is the same as the length of the corresponding trail G_W on $g(H)$, i.e. $Len(H_W) = Len(G_W)$.

2.8.3 Path

We define a path on a TVG H as a walk on H where all temporal nodes are distinct. We can identify a path $P = [u_{t_1}, e_1, u_{t_2}, e_2, u_{t_3}, ..., u_{t_{k-1}}, e_{k-1}, u_{t_k}]$ with the TVG H_P = (V(P), E(P),

T(P)), where $V(P), E(P)$, and $T(P)$ are the sets of temporal nodes, dynamic edges, and time instants on the path, respectively. Since $V(P) \subseteq V(H)$, $E(P) \subseteq E(H)$, and $T(P) \subseteq T(H)$, it follows that the path P is a sub-TVG of H.

Theorem 2.9 *A walk P on H is a path on H if and only if there is a corresponding path G_P on $g(H)$.*

Proof 2.11 *Since P is also a walk on H, we have from Theorem 2.6 that P is a walk on H if and only if G_P is a walk on $g(H)$. Therefore, the only point left to be shown is that all nodes in P are distinct if and only if all nodes in G_P are distinct. However, it also follows from Theorem 2.6 that the temporal nodes in P and in G_P are the same. Therefore, if the nodes of P are distinct, so are the nodes of G_P, and the theorem holds.*

From Theorem 2.7, we have that the projection of a path P onto the aggregated graph of H is a walk. To show this, we use the same example presented for trails. Let the sequence $P = [(u,t_1),(u,t_1,v,t_2),(v,t_2),\ (v,t_2,w,t_3),(w,t_3),(w,t_3,u,t_4),(u,t_4),(u,t_4,v,t_5),(v,t_5)]$ be a path on TVG H. The projection of P onto $Agg(H)$ is the walk $W_{agg} = [u,(u,v),v,(v,w),w,(w,u),u,(u,v),v]$. Note that the node u appears twice in W_{agg}. The walk W_{agg} is thus not a path.

Since a path is also a walk, the length of the path is also determined by the number of edges on it. Further, it follows from Corollary 2.2 that the length of a path P on a TVG H is the same as the length of the corresponding path G_P on $g(H)$, i.e. $Len(P) = Len(G_P)$.

This notion of path stems from the adjacency definitions used on TVGs, where both nodes and time instants are considered to define adjacency (i.e. two dynamic edges are adjacent if and only if they are both incident to a node at the same given time instant). Therefore, this definition is consistent with the walk and trail definitions stated before, since all of them are based on this same adjacency notion.

However, the path definition we present is not the same that is usually used, in which a path can not pass twice through a same node, regardless of time instants. We present this definition since, as stated above, it is consistent with the development started on the walk definition and also because it leads to interesting and useful results, such as that a path is preserved by the natural isomorphism (i.e. temporal nodes representation). Further, if required, the more traditional path definition can be easily reached, since it coincides with the projection of the path onto the aggregated graph of the TVG. Note that this is consistent with the idea that the aggregated graph and projections over it are obtained by ignoring the time instant coordinates present on dynamic edges.

Once we have the path definition, it can be seen that a mixed edge can be decomposed into a pair of temporal and spatial edges forming a path with two dynamic edges. However, it should be noted that such a decomposition leads to a TVG that is topologically different (and therefore not isomorphic) as compared with the original TVG with the mixed edge. To see that this is indeed the case, note that when this decomposition is performed, it is necessary to create a temporal and a spatial dynamic edge (unless they already exist) and that the original mixed edge is then removed. Since in any case the original TVG and the TVG with the decomposition don't have the same number of edges, they cannot be isomorphic. Similarly, any path containing mixed dynamic edges can be decomposed into a path containing only temporal and spatial dynamic edges.

2.8.4 Cycle

A cycle on a TVG is defined as a closed path, i.e. a path that starts and ends on the same temporal node.

Theorem 2.10 *A path P on a TVG H is a cycle if and only if the corresponding path G_P in $g(H)$ is a cycle.*

Proof 2.12 *From Theorem 2.9, we have that P is a path on H if and only if G_P is a path on $g(H)$. Hence, we only need to show that P starts and ends on the same temporal node if and only if G_P does the same. It follows from Theorem 2.6 that the sequence of temporal nodes in P and in G_P is the same. Therefore, if P starts and ands on a given temporal node u_t, so does G_P and the theorem holds.*

Additionally, since a cycle is a special case of a path, it follows that the length of a cycle is determined in the same way as the length of a path. Further, the length of a cycle P on H is the same as the length of the corresponding cycle G_P on $g(H)$.

We define the girth of a TVG H as the length of the shortest cycle in H. If H has no cycle, we consider that its girth is infinite. It follows directly from Theorem 2.10 and Corollary 2.2 that a TVG H and its temporal node representation $g(H)$ have the same girth.

If a cycle on a TVG is constructed using only spatial dynamic edges, this cycle is equivalent to the concept of cycles on a directed graph. Nevertheless, if a cycle contains any dynamic edge which has a non-zero temporal component (i.e. a temporal or mixed edge), then the cycle has to contain a regressive dynamic edge, since the presence of progressive edges makes the use of a regressive edge necessary to make it possible for a path to return to the temporal node where it started.

2.9 Shortest paths on a TVG

Given two temporal nodes $u_{t_a}, u_{t_b} \in VT(H)$ in a TVG H, we define the shortest path between u_{t_a} and u_{t_b} as the path with the smallest length starting at temporal node u_{t_a} and ending at temporal node u_{t_b}.

We can define four distinct variants of shortest path on a TVG:

1. Shortest path from a temporal node to a temporal node, i.e. $(u, t_a) \to (v, t_b)$;

2. Shortest path from a node to a temporal node. This is the shortest path starting at any temporal node u_t such that $\pi_1(u_t) = u$ and ending at the temporal node (v, t_b); i.e. $(u, \cdot) \to (v, t_b)$;

3. Shortest path from a temporal node to a node. This is the shortest path starting at a temporal node (u, t_a) and ending at any temporal node v_t such that $\pi_1(v_t) = v$; i.e. $(u, t) \to (v, \cdot)$;

4. Absolute shortest path. This is the shortest path from a node to another node. In other words, this is the shortest path starting at any temporal node u_t such that $\pi_1(u_t) = u$ and ending at any temporal node v_t such that $\pi_1(v_t) = v$; i.e. $(u, \cdot) \to (v, \cdot)$.

Note that even though the variants 2, 3, and 4 involve the concept of nodes, a path always starts and ends on a temporal node. The meaning of the nodes in the shortest path variants is the freedom to select a temporal node at any time instant, such that the path is minimized.

Theorem 2.11 *A path P between two temporal nodes u_{t_a} and u_{t_b} on a TVG H is a shortest path between these nodes if and only if the corresponding path G_P in $g(H)$ is a shortest path between u_{t_a} and u_{t_b} on $g(H)$.*

Proof 2.13 • $\Longrightarrow$ *(by contradiction)*
Let $u_{t_a}, u_{t_b} \in VT(H)$ be two temporal nodes on a TVG H and P be a shortest path from u_{t_a} to u_{t_b}. Further, let G_P be the corresponding path from u_{t_a} to u_{t_b} on the graph $g(H)$. From Corollary 2.2, we have that $Len(P) = Len(G_P)$.
Let's suppose that G_P is not a shortest path from u_{t_a} to u_{t_b} on $g(H)$. This means that there is a path G_{P_S} from u_{t_a} to u_{t_b} on $g(H)$, such that $Len(G_{P_S}) < Len(G_P)$. Then, by Theorem 2.9 and Corollary 2.2, there must be a corresponding path P_S from u_{t_a} to u_{t_b} on H, such that $Len(P_S) < Len(P)$. This is a contradiction, since P is a shortest path from u_{t_a} to u_{t_b} on H. Therefore, G_P is a shortest path from u_{t_a} to u_{t_b} on $g(H)$.

• $\Longleftarrow$ *(by contradiction)*
Let $u_{t_a}, u_{t_b} \in V(g(H)$ be two temporal nodes on $g(H)$ and G_P be a shortest path from u_{t_a} to u_{t_b}. Further, let P be the corresponding path from u_{t_a} to u_{t_b} on the TVG H. From Corollary 2.2, we have that $Len(G_P) = Len(P)$.
Let's suppose that P is not a shortest path from u_{t_a} to u_{t_b} on TVG H. Then, there must be a path P_S from u_{t_a} to u_{t_b} in H, such that $Len(P_S) < Len(P)$. Thus, from Theorem 2.9 and Corollary 2.2, there must be a corresponding path G_{P_S} from u_{t_a} to u_{t_b} on $g(H)$, such that $Len(G_{P_S}) < Len(G_P)$. This is a contradiction, since G_P is a shortest path from u_{t_a} to u_{t_b} on $g(H)$. Therefore, P is a shortest path from u_{t_a} to u_{t_b} on the TVG H.

Figure 2 shows a simple example of a TVG for illustrating TVG shortest paths. The shortest path from $(0, t_0)$ to $(1, t_2)$ is the sequence $[(0, t_0), (0, t_1), (1, t_1), (1, t_2)]$ and has length 3, while the shortest path from $(0, t_0)$ to $(1, \cdot)$ is formed by $[(0, t_0), (0, t_1), (1, t_1)]$ and has length 2. In the same way, the shortest path from $(0, \cdot)$ to $(1, \cdot)$ is $[(0, t_1), (1, t_1)]$, while the shortest path from $(0, \cdot)$ to $(1, t_2)$ is $[(0, t_1), (1, t_1), (1, t_2)]$.

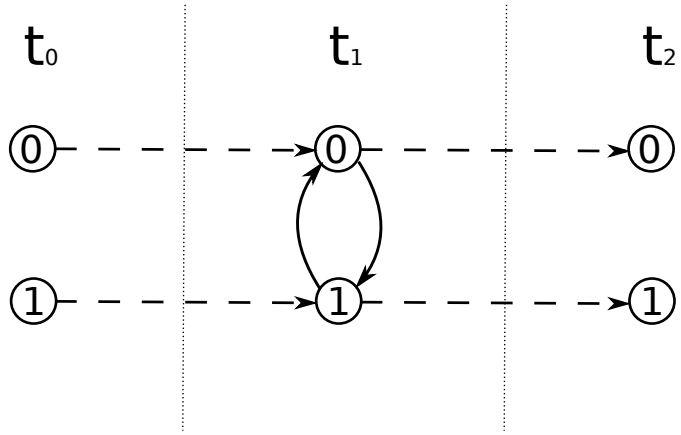


Figure 2: Time-variyng graph example.

2.10 Strongly connected TVG

We define a strongly connected TVG as a TVG H in which for all pairs of nodes $u, v \in V(H)$ there is a path $u \to v$ and also a path $v \to u$. Nevertheless, this does not mean that any pair of nodes is capable of communicating at all times. It only means that each node is capable of eventually accessing any other node at some time instant. Note that in the temporal node representation a strongly connected TVG is not necessarily a strongly connected graph. For the

graph to be strongly connected, every pair of temporal nodes would have to communicate, which is not required for the TVG to be strongly connected.

Figure 3 shows an example of a strongly connected TVG. The TVG shown in Figure 3 is strongly connected because there is a path from node 0 to node 1, a path from node 0 to node 2, a path from node 1 to node 0, a path from node 1 to node 2, a path from node 2 to node 1, and a path from node 2 to node 0. However, the temporal node representation of the TVG is clearly not strongly connected. Additionally, in the TVG case, strong connectivity has to be considered over time. If no time restriction is stated, we consider that the connectivity is defined over all time instants present on the TVG. However, connectivity can also be defined over a subset of the time instants of the TVG. When this happens, the connectivity becomes stronger as the time interval over which it is defined becomes smaller. For instance, the TVG shown in Figure 3 is strongly connected if all time instants are considered. However, the sub-TVG obtained when only instants t_0 and t_1 are considered is not strongly connected, since in this sub-TVG node 2 is an isolated node, i.e. it has no dynamic edges incident to it.

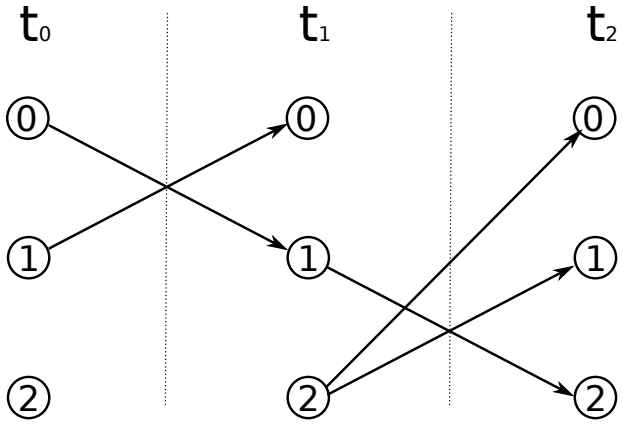


Figure 3: Example of a strongly connected TVG.

2.10.1 Closed strongly connected TVG

We define a closed strongly connected TVG as a TVG H on which for all pairs of nodes in H there is a path from each node to itself that passes through the other node of the pair. This means that for all pair of nodes $u, v \in V(H)$ there is a path $P_u = u \to u$, such that there is a temporal node $v_t \in V(P_u)$ for which $v = \pi_1(v_t)$, and also a path $P_v = v \to v$ such that there is a temporal node $u_t \in V(P_v)$ for which $u = \pi_1(u_t)$. Note that a path $P = u \to u$ is not necessarily a cycle, since the temporal nodes (u, t_a) where the path start and (u, t_b) where it ends may have time instants $t_a \neq t_b$. Figure 4 shows an example of a closed strongly connected TVG. There is a path $P_0 = 0 \to 0$ which passed through node 1 and also a path $P_1 = 1 \to 1$ which passes through node 0. Note that neither $P_0 = [(0, t_0), (1, t_1), (0, t_2)]$ nor $P_1 = [(1, t_1), (0, t_2), (1, t_3)]$ are cycles.

In this kind of connectivity, we also have that the temporal node representation of a closed strongly connected TVG is not necessarily a strongly connected directed graph. This can easily be seen in Figure 4. This sort of connectivity also depends on the considered set of time instants. For instance, in Figure 4, the sub-TVG having only time instants t_0, t_1, and t_2 is not closed strongly connected because since t_3 is not present, there is no path from node 1 to node 1 passing through node 0. The closed strong connectivity can also be made stronger by imposing tighter

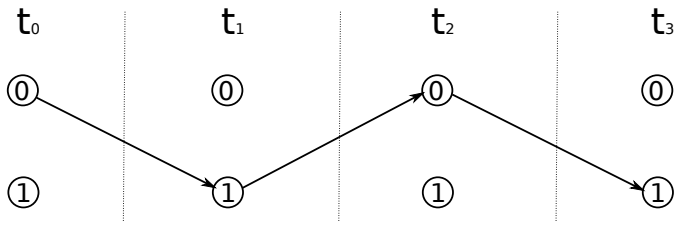


Figure 4: Closed strongly connected TVG.

time constraints to it. For instance, the TVG shown in Figure 5 is closed strongly connected, however, in this case the sub-TVGs containing only time instants t_0, t_1, t_2 and t_1, t_2, t_3 are also closed strongly connected.

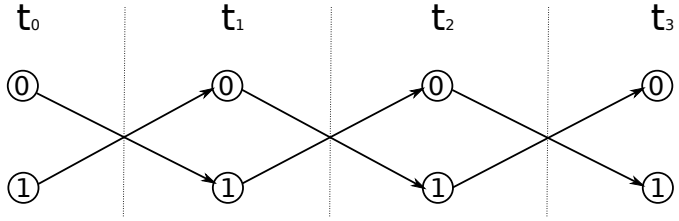


Figure 5: Closed strongly connected TVG with stronger time constraints.

2.10.2 Temporal node strongly connected TVG

This is the strongest form of connectivity. This kind of connectivity happens when the temporal node representation of the TVG is also strongly connected. This means that for a given TVG H to be temporal node strongly connected, for every pair of temporal nodes $u_t, v_t \in VT(H)$ there is a path $P_{u_t} = u_t \to v_t$ and a path $P_{v_t} = v_t \to u_t$. This form of connectivity on a given TVG H happens if and only if its temporal node representation $g(H)$ is also a strongly connected directed graph. Further, unless the TVG is trivial, having only one time instant, this form of strong connectivity implies the existence of temporal cycles.

3 Representations and algebraic structures of a TVG

In this section, we discuss ways to properly represent a TVG using our proposed model. Similarly to static graphs, a TVG can be fully represented by an algebraic structure. We thus present forms of TVG representation based on adjacency and incidence. The algebraic representations are in the form of tensors and matrices. The tensor form is regarded as the canonical form of representing the TVG, while the matrix form is based on the natural isomorphism (i.e. temporal node representation) stated in Sections 2.2 and 2.4. The TVG adjacency tensor and its matrix form are discussed in Sections 3.1 and 3.2, respectively. Likewise, the TVG incidence tensor and its matrix form are discussed in Sections 3.3 and 3.4, respectively. In order to illustrate such representations, we use the TVG W presented in Figure 6, where spatial edges are represented by solid arrows and temporal edges by dashed arrows. Finally, in Section 3.5, we analyze the memory complexity for storing a TVG using our model.

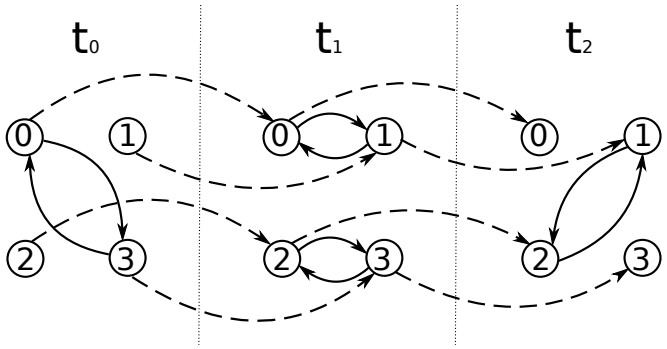


Figure 6: TVG W: Example for algebraic representation.

3.1 TVG adjacency tensor

The adjacency tensor follows from the adjacency matrix widely used to represent static graphs. However, since the dynamic edges used in the TVG are ordered quadruples, this representation has to be done by means of a 4th order tensor.

We then define the adjacency tensor of a TVG H as a 4th order tensor $\mathbb{A}(H)$ with dimension $|V(H)| \times |T(H)| \times |V(H)| \times |T(H)|$ that has an entry for every possible dynamic edge in H. Each dynamic edge present in the TVG H is represented by a non-zero entry in the adjacency tensor $\mathbb{A}(H)$, while all other entries have a zero value. The non-zero entries represent the weight of the corresponding dynamic edge in the represented TVG. In the case of a unweighted TVG, the non-zero entries corresponding to the dynamic edges present in the TVG have value 1. The notation $\mathbb{A}(H)^{u,t_a}_{v,t_b}$ is used to identify the entry corresponding to the dynamic edge (u, t_a, v, t_b) of the TVG H.

As an example, the adjacency tensor of the TVG W depicted in Figure 6 has dimension $4 \times 3 \times 4 \times 3$, having a total of 144 entries. For instance, the pair of dynamic spatial edges connecting node 0 at time t_0 to node 3 at time t_0 is represented by the entries $\mathbb{A}(W)^{0,t_0}_{3,t_0}$ and $\mathbb{A}(W)^{3,t_0}_{0,t_0}$, where both carry value 1 as the TVG W is unweighted.

Note that, even though the adjacency tensor $\mathbb{A}(H)$ has dimension $|V(H_t)|\times|T(H_t)|\times|V(H_t)|\times|T(H_t)|$, only the entries corresponding to dynamic edges present in the TVG H have non-zero values.

3.2 Matrix form of the TVG adjacency tensor

Since Theorem 2.1 ensures that a TVG has a directed static graph that is isomorphic to it, it follows that the TVG can be represented by an adjacency matrix. Therefore, we choose the adjacency matrix of the directed static graph obtained from the temporal node representation to be the matrix form of the TVG adjacency tensor.

To get the matrix representation of the TVG adjacency tensor using the temporal node representation (i.e. natural isomorphism) presented in Section 2.4, we only need to consider that each temporal node (u, t_a) can be thought of as a node in a static graph. This static graph has $|V| \times |T|$ nodes and, as a consequence, its adjacency matrix has $|V| \times |T| \times |V| \times |T| = |V|^2 \times |T|^2$ entries. Since the non-zero entries of this matrix correspond to the dynamic edges of the TVG, further analysis show that this matrix is usually sparse and can therefore be stored in an efficient way.

Figure 7 shows the matrix representation obtained for the illustrative TVG W shown in

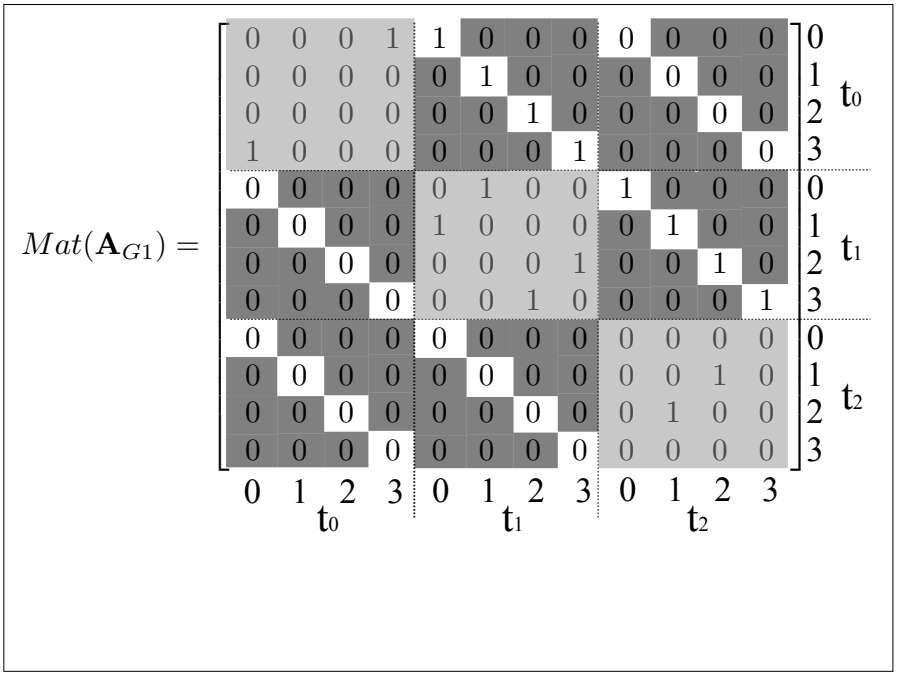


Figure 7: The matrix form of the adjacency tensor of TVG W.

Figure 6. From Figure 7, we highlight that the matrix form of the TVG adjacency tensor has interesting structural properties. First, each one of the four nodes (identified as 0, 1, 2, and 3) of the TVG W clearly appears as a separate entity in each of the three time instants (t_0, t_1, and t_2) that compose the TVG W. Second, the main block diagonal (lightly shaded) contains the entries corresponding to the spatial edges at each time instant. In these three blocks the entries corresponding to the spatial edges of the TVG carry value 1. Third, the unshaded entries at the off-diagonal blocks correspond to the temporal edges. The eight progressive temporal edges present at the TVG W are indicated by the value 1 on the first superior diagonal. Finally, the dark shaded entries are the ones that correspond to the mixed edges. Since no mixed edges are present in the example TVG W, all these entries contain value 0. Further, we remark that the entries corresponding to progressive (mixed and temporal) edges are above the main block diagonal, whereas the edges corresponding to regressive edges appear below the main block diagonal. All these structural properties derive from the order adopted for representing the nodes and time instants present in the TVG and can be readily verified in the matrix form in a convenient way.

Note that the procedure used to obtain the matrix form of the adjacency tensor is the well-known matricization or unfolding of a tensor [Kol06]. In fact, the matricization of the adjacency tensor can be seen as an equivalent form of Theorem 2.1, since it shows that a 4th order tensor of dimension $|V| \times |T| \times |V| \times |T|$ can be written as a square matrix with the same number of entries. Further, since each entry of the matrix corresponds to an entry in the tensor, the process can be reversed, obtaining the corresponding tensor from its matrix form. Hence, in agreement with the stated in Section 2.2, while the process of obtaining the matrix form from the higher order tensor is straightforward, the process of obtaining the tensor from the matrix form can only be done if the dimensions of the tensor form are known.

3.3 TVG incidence tensor

The representation used on the incidence tensor relates the dynamic edges present in the TVG with the nodes and time instants defined in the TVG, leading to a 3rd order tensor.

We define the incidence tensor of a TVG H as a 3rd order tensor $\mathbb{C}(H)$ with dimension $|E(H)| \times |V(H)| \times |T(H)|$. Each entry of this incidence tensor corresponds to a dynamic edge, node, and time instant combination. An entry has value -1 if the dynamic edge corresponding to this entry is originated at the node and time instant represented by the entry. Conversely, an entry has value 1 if the dynamic edge corresponding to the entry has as destination the node and time instant represented by the entry. An entry has value 0 if there is no dynamic edge related to the node and time instant represented by the entry. This representation usually does not carry the weight of the edges, just an indication of their origin and destination. For instance, assuming that the dynamic edge $(0, 0, 0, 1)$ of the TVG W depicted in Figure 7 is labeled as e_0, then the entry $\mathbb{C}(H)^{t_0}_{0,e_0}$ has value -1, the entry $\mathbb{C}(H)^{t_1}_{0,e_0}$ has value 1, and all other entries $\mathbb{C}(H)^{t}_{n,e_0}$, except the previous two, have value 0. Further, the incidence tensor representing the TVG W shown in Figure 7 has $16 \times 4 \times 3 = 192$ entries. Note that the incidence tensor is by construction a sparse tensor, since its dimension is $|E(H)| \times |V(H)| \times |T(H)|$ and the number of non-zero entries is $2 \times |E(H)|$.

3.4 Matrix form of the TVG incidence tensor

The matrix form of the incidence tensor of a TVG is the incidence matrix corresponding to the directed static graph obtained though the temporal node representation presented in Section 2.2. For instance, considering the TVG W shown in Figure 6, there is a temporal edge connecting node 0 at time t_0 to node 0 at time t_1. Without loss of generality, we can label this edge as e_0 and assign it the coordinate 0 for the edge dimension of the incidence tensor. As a consequence, this edge is represented in the incidence tensor by setting $\mathbb{C}^0_{0,0} = -1$ and $\mathbb{C}^1_{0,0} = 1$. Taking the temporal edge connecting node 0 at time t_1 to node 0 at time t_2 and labeling it as e_1, its representation on the incidence tensor is $\mathbb{C}^1_{0,1} = -1$ and $\mathbb{C}^2_{0,1} = 1$. By repeating this procedure for each edge in the TVG, the corresponding incidence tensor is created. Since the TVG W has 16 edges, being 8 temporal edges and 8 spatial edges, 4 nodes, and 3 time instants, the corresponding incidence tensor has a total of $4 \times 16 \times 3 = 192$ entries, from which 160 carry value 0, 16 carry value -1, and 16 value 1. Note that the resulting matrix has 12 rows (for 3 times and 4 nodes) and 16 columns, one for each edge in the TVG, totalizing $12 \times 16 = 192$ entries.

Figure 8 shows the matrix representation of the incidence tensor corresponding to the TVG W shown in Figure 6. In this representation, the eight temporal edges present in the TVG are labeled as edges e_0 to e_7 and the 8 spatial edges as edges e_8 to e_{15}. Considering this labeling, it is straightforward to verify the correspondence of the incidence tensor in matrix form to the TVG.

The matrix form representation of the incidence tensor is compatible with the isomorphism between the TVG and a static graph stated in Theorem 2.1. Indeed, the matrix form of the incidence tensor can also be seen as the incidence matrix of the temporal node representation of the TVG shown in Figure 6. The matrix form of the incidence tensor can also be seen as a matricization of this tensor, where the unfolding is done in a way that the resulting matrix coincides with the incidence matrix of the temporal node representation.

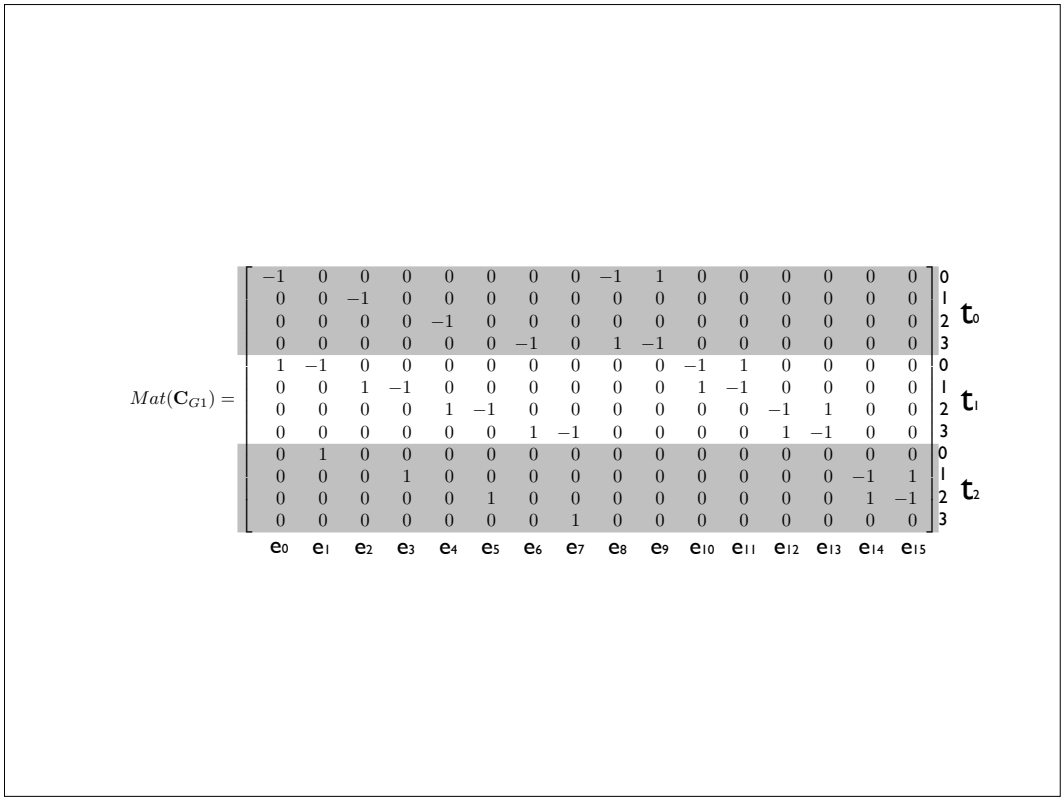


Figure 8: The matrix form of the incidence tensor of TVG W.

3.5 Memory complexity for storing a TVG

In this subsection, we analyze the memory complexity for storing a TVG. This analysis is valid for the TVG representations based on adjacency and incidence as well as their corresponding algebraic structures, which have all been discussed in Sections 3.1 to 3.4.

In the general case, TVGs may have disconnected nodes or unused time instants. A disconnected node is defined as a node that has no dynamic edge incident to it, i.e. a node with no connection at any time instant. In a similar way, an unused time instant in a TVG is defined as a time instant at which there is no dynamic edge originated from or destined to it. In most practical cases, however, a TVG is expected to have none or very few disconnected nodes as well as a unused time instants as compared with the total number of nodes or time instants, respectively.

We show that when a TVG has no disconnected nodes and unused time instants, or if the number of disconnected nodes and unused time instants is significantly lower than the number of dynamic edges, then the amount of storage needed to represent a TVG is determined by the number of dynamic edges in the TVG. In other words, we show in this section that in most practical cases the memory complexity $M_C(H)$ for storing a given TVG H is $M_C(H) = \Theta(|E(H)|)$, where $E(H)$ is the set of dynamic edges in the TVG H.

The first step to achieve this is to show that the set of connected nodes and the set of used time instants of a TVG can be recovered from the set of dynamic edges. Lemma 3.1 shows this by using the dynamic edge definition and the canonical projections defined in Section 2.1.

Lemma 3.1 *In any given TVG H, the set of connected nodes and the set of used time instants can be recovered from the set $E(H)$ that contains the dynamic edges of H.*

Proof 3.1 *Let $V_C(H)$ be the set of connected nodes on TVG H. We now show that $V_C(H)$ can be constructed from the set $E(H)$. Since $V_C(H)$ contains only connected nodes, for every $u \in V_C(H)$ there is at least one dynamic edge incident to u. Let $e_u \in E(H)$ be a dynamic edge incident to node u. Then, either e_u is of the form $(u, \cdot, \cdot, \cdot)$ or of the form $(\cdot, \cdot, u, \cdot)$, and therefore, either $u = \pi_1(e_u)$ or $u = \pi_3(e_u)$. Therefore, we can write $V_C(H)$ as*

$$V_C(H) = \bigcup_{e \in E(H)} \{\pi_1(e), \pi_3(e)\}.$$

We now use a similar reasoning to recover the set of used time instants from the set $E(H)$. Let $T_U(H)$ be the set of used time instants in H and let t_u be a used time instant. Then, there is at least one dynamic edge $e_u \in E(H)$ of the form $(\cdot, t_u, \cdot, \cdot)$ or $(\cdot, \cdot, \cdot, t_u)$ such that $t_u = \pi_2(e_u)$ or $t_u = \pi_4(e_u)$. Hence, we can write $T_U(H)$ as

$$T_U(H) = \bigcup_{e \in E(H)} \{\pi_2(e), \pi_4(e)\}.$$

Since Lemma 3.1 shows that the both sets of connected nodes and used time instants in a given TVG H can be recovered from the set of dynamic edges $E(H)$, we can conclude that such information is redundant and does not need to be stored to represent a TVG. Therefore, to represent any given TVG H, it suffices to store the set of dynamic edges $E(H)$ as well as the sets of disconnected nodes and unused time instants of the TVG.

We now demonstrate in Theorem 3.1 that when the sets of disconnected nodes and unused time instants of a TVG are significantly smaller than the set of dynamic edges, which is actually expected in most practical cases, then the asymptotic memory complexity of the TVG representation is determined by the cardinality of the set of dynamic edges.

Theorem 3.1 *If the set of disconnected nodes and the set of unused time instants are significantly smaller than the set of dynamic edges, then the memory complexity $M_C(H)$ for storing a given TVG H is determined by the size of the set of dynamic edges, i.e. $M_C(H) = \Theta(|E(H)|)$.*

Proof 3.2 *Let H be an arbitrary TVG, $E(H)$ its set of dynamic edges, $V_C(H)$ its set of connected nodes, $V_N(H)$ its set of disconnected nodes, $T_U(H)$ its set of used time instants, $T_N(H)$ its set of unused time instants, and $M_C(H)$ the memory complexity for storing the TVG H. Note that $V(H) = V_C(H) \cup V_N(H)$ and $T(H) = T_U(H) \cup T_N(H)$, whereas $V_C(H) \cap V_N(H) = \emptyset$ and $T_U(H) \cap T_N(H) = \emptyset$.*

Since Lemma 3.1 shows that $V_C(H)$ and $T_U(H)$ can be recovered from $E(H)$, it follows that to store a representation of H in memory, it suffices to store $E(H)$, $V_N(H)$, and $T_N(H)$. We now analyze the asymptotic bounds for the memory complexity for storing these sets, together with the assumption that $|V_N(H)| + |T_N(H)| \ll |E(H)|$.

- *Lower bound for $M_C(H)$: Since for storing the set $E(H)$ it is necessary at least to store an ordered quadruple for each dynamic edge $e \in E(H)$, the memory needed is $c_1 \times |E(H)|$, where c_1 is an integer constant. To store the set $V_N(H)$ it is necessary to store all nodes $u \in V_N(H)$ and therefore the memory needed is $c_2 \times |V_N(H)|$, while to store the set $T_N(H)$ it is necessary to store all time instants $u \in T_N(H)$, leading to a memory need of $c_3 \times |T_N(H)|$. Therefore, the total memory need is at least $c_1 \times |E(H)| + c_2 \times |V_N(H)| + c_3 \times |T_N(H)|$, and $M_C(H) = \Omega(|E(H)| + |V_N(H)| + |T_N(H)|)$. Finally, from our assumption that $|V_N(H)| + |T_N(H)| \ll |E(H)|$, we conclude that $M_C(H) = \Omega(|E(H)|)$.*

- *Upper bound for $M_C(H)$: On the upper bound analysis, we see that the information to be stored is at most the same three sets $E(H)$, $V_N(H)$, and $T_N(H)$, which had to be stored in the lower bound analysis. Therefore, we conclude that $M_C(H) = O(|E(H)|)$.*

Since $M_C(H) = \Omega(|E(H)|)$ and also $M_C(H) = O(|E(H)|)$, we finally conclude that $M_C(H) = \Theta(|E(H)|)$, if $|V_N(H)| + |T_N(H)| \ll |E(H)|$.

Corollary 3.1 *The complete expression of the memory complexity for a given TVG H is $M_C(H) = \Theta(|V_N(H)| + |T_N(H)| + |E(H)|)$.*

Proof 3.3 *It follows from the proof of Theorem 3.1 that the formal and complete expression of the lower bound memory complexity for storing a given TVG H is $M_C(H) = \Omega(|V_N(H)| + |T_N(H)| + |E(H)|)$, where $V_N(H)$ its set of disconnected nodes, $T_N(H)$ its set of unused time instants, and $E(H)$ the set of dynamic edges, while the upper bound is $M_C(H) = \Omega(|V_N(H)| + |T_N(H)| + |E(H)|)$. We therefore conclude that complete form of the memory complexity is $M_C(H) = \Theta(|V_N(H)| + |T_N(H)| + |E(H)|)$.*

Since in most practical cases the amount of disconnected nodes and unused time instants is significantly smaller than the number of dynamic edges (i.e., $|V_N(H)| + |T_N(H)| \ll |E(H)|$), we can conclude that the expected amount of memory needed to store a representation of a given TVG H is determined by the size of the set of dynamic edges $|E(H)|$, i.e. $M_C(H) = \Theta(|E(H)|)$ as states Theorem 3.1. Furthermore, any given TVG H for most practical applications can typically be expected to be *sparse*, i.e. the number of dynamic edges is significantly smaller than the squared number of temporal nodes ($|E(H)| \ll |VT(H)|^2$), thus allowing its storage in a compact form, similar to the compressed forms used for sparse matrices [BBC+94].

4 Unifying the representation of previous models

In this section, we show that the TVG model we propose can be used to represent many previous models found in the literature. Further, these models are not always capable of representing each other and none of them has the same representation range of the unifying model we propose.

In order to assert that all the considered models can be directly represented by our model, we show that each of these models is in fact a subset of what is representable in our model and can therefore be easily represented in our model, maintaining any previously obtained result as at least valid on a special case. To achieve this, we show that each of the studied models can be represented in our TVG model by using only a subset of the types of dynamic edges available on our model (see Section 2.1). We achieve this by presenting the general structure of the matrix form of the adjacency tensor of the TVG that represents the model under study. We establish a set of four nodes $V = \{0, 1, 2, 3\}$ and a set of three time instants $T = \{t_0, t_1, t_2\}$ to be used to illustrate all of the following analyses. This is done without loss of generality, since the same procedure could be applied to any set of nodes and time instants present in a TVG.

To show the representation in our model of the different previous TVG models found in the literature, we group them into classes that can be represented in our unifying model in similar ways and analyze these classes in the remaining of this section.

4.1 Models based on snapshots

Some snapshot models for TVGs adopt an aggregate graph and a sequence of successive state subgraphs that represent the network in a discrete way as time passes. Some examples are the models proposed by Xuan et al. [BXFJ03] and Ferreira [Fer04]. We also consider in this same class of

snapshot models the models proposed by Holme [Hol05] and Holme and Saramäki [HS12], where edges are represented as triples of the form (i, j, t) meaning the existence of a contact between nodes i and j at time t. Still under this same class, we also consider the models proposed by Tang et al. [TMML09, TSM$^+$10, TMM$^+$10], as well as all other models in which the TVG is proposed as a sequence of static graphs (i.e., the snapshots), each of them representing the TVG at a given time instant.

Snapshot models are widely used in the literature and in general give an intuitive and straightforward notion of TVGs. Nevertheless, the snapshot models also demand some assumptions to be made without having them explicitly constructed in the model. For instance, it is usually assumed in such snapshot models that the nodes have a sort of memory that allows the transitivity induced by the edges to propagate on each node over time. This means that a path can be constructed passing through a given node even if this node is disconnected from all others during a period of time, meaning that the node is capable of retaining the edge transitivity during the period of disconnection. Even though this behavior is straightforward and intuitively expected in many cases, it is not constructed within the model and has to be assumed as an additional (external) property of the model, resulting that this assumption has also to be incorporated into the algorithms used with those model and thus making these algorithms dependent of these external assumptions.

From the analysis of the snapshot models, we remark that clearly they only use edges connecting nodes in the same given time instant. This is consistent with the concept of spatial dynamic edges proposed in our model. In this way, all the TVGs constructed in this class of snapshot models can be represented in our model by using only spatial dynamic edges, given that all the necessary nodes and time instants are present in our model. Thus, a TVG with four nodes and three time instants in this class of snapshot model would be represented in our model by a TVG whose adjacency tensor in matrix form is as the one presented in Figure 9. The entries containing "*" may have non-zero values, indicating the potential presence of a spatial edges. Note that all the entries of this kind are located in the main block diagonal of the adjacency tensor in matrix form, which actually corresponds to the snapshots.

In our general representation, we allow spatial and temporal self-loops (i.e. dynamic edges connecting a node to itself at the same time instant). This kind of edge is represented on the main diagonal of the matrix, just as it would be on an adjacency matrix of a static graph.

Any TVG of the snapshot class can be represented in this straightforward way in our proposed unifying model. Actually, snapshot models are in fact a sub-space of the representable TVGs in our model, making clear that this whole class of snapshot models is rather a subset of the representable TVGs in our model. Further, in snapshot models, each snapshot is formally disconnected from each other (although arguably implicitly connected), since the temporal connections are not explicitly constructed in the snapshot models.

The number of dynamic edges used to represent a TVG of this kind is the same as the number of edges present in the original snapshot-based representation. Therefore, the memory complexity for storing TVGs in our model is the same found on the original snapshot-based representation, which is compatible with the memory complexity discussed in Section 3.5.

4.2 Models based on continuous time intervals (CTI)

The class discussed in this subsection includes models that use a presence function defined over continuous time intervals (i.e. $t \in \mathbb{R}^+$), such as the continuous time version of TVG proposed by Casteigts et al. [CFQS12]. We further assume that the presence function used in such models is constructed in such a way that every time interval (or their union) has a non-zero and finite measure. Although this assumption is not explicitly stated in the original paper, it is consistent

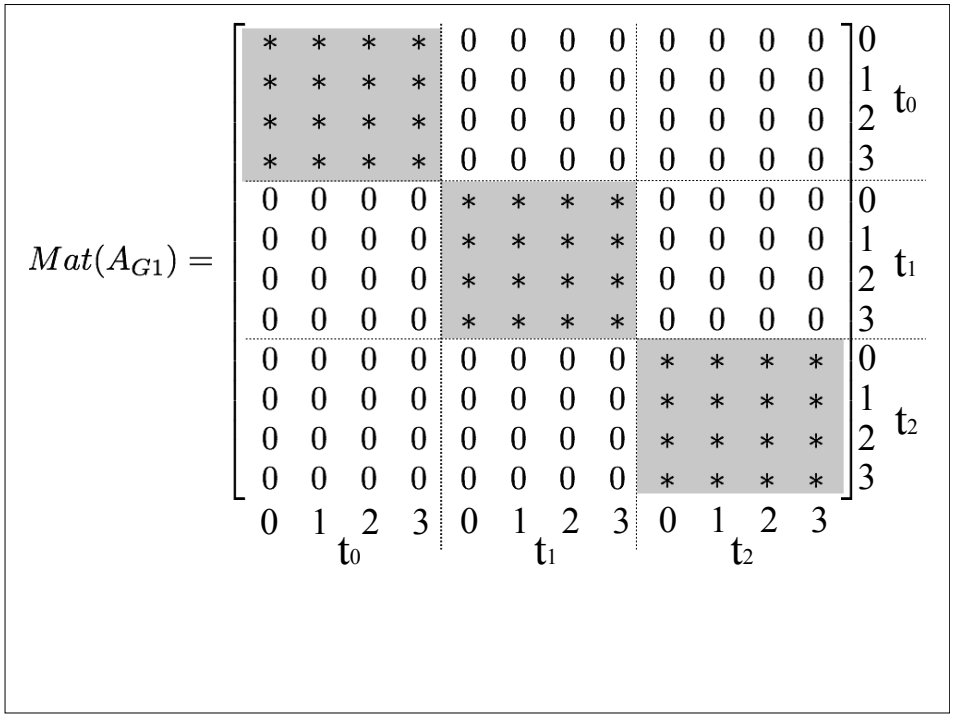


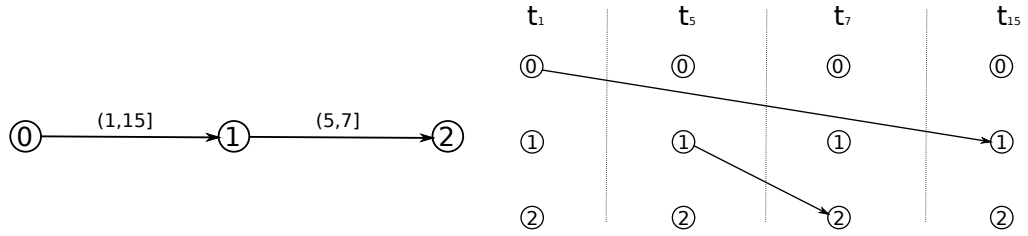
Figure 9: Snapshot models represented by our unifying TVG model.

with all examples and the reasoning present therein. It is also important to remind that the model we construct is a discrete version of the continuous time interval model that nonetheless retains all information present on the original model based on continuous time intervals.

In order to represent TVGs based on continuous time intervals in our unifying model, some representations are possible depending on the target application of the model that defines the semantics associated with an edge in the TVG. An example with three possible representations is presented in Figure 10 and each of the these three representations is explained in the following.

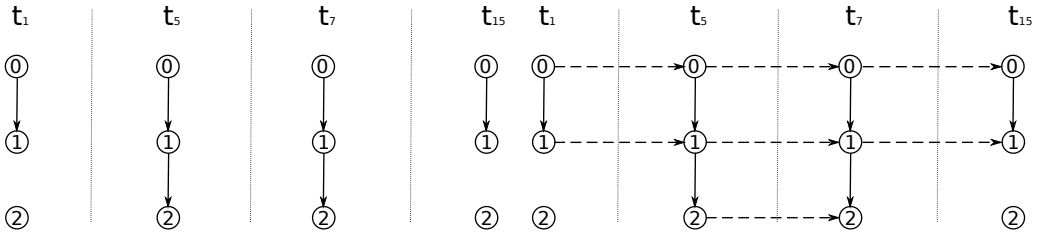
If the semantic of a time interval $(t_a, t_b]$ associated with an edge e between two nodes u and v is that the edge exists from the beginning of the interval (i.e. from time t_a) until the end of the interval (i.e. until time t_b), it is possible to represent the existence of such an edge by using a mixed dynamic edge. This is consistent with the original semantics, where the edge e exists in the interval delimited by the mixed edge (i.e. from t_a until t_b). A concrete example of this first representation is shown comparing Figures 10(a) and 10(b). The edge between nodes 0 and 1 present at the time interval $(1, 15]$ shown in Figure 10(a) is represented by the mixed dynamic edge $(0, t_1, 1, t_{15})$ in Figure 10(b). Similarly, the edge between nodes 1 and 2 at the time interval $(5, 7]$ is represented by the mixed dynamic edge $(1, t_5, 2, t_7)$.

In the case an edge is present at different time intervals, we represent this edge by having one mixed edge for each time interval. Further, if the edge between nodes u and v is bidirectional, this edge is represented at each time interval by the pair of edges (u, t_a, v, t_b) and (v, t_a, u, t_b). Note that once the edges present in the TVG are defined for all time intervals, it is possible to construct the set T of time instants based on the dynamic edges present in the set E. From this, we conclude that the TVGs in this class modeled by continuous time intervals can be represented in our model using only progressive mixed dynamic edges.



(a) TVG using a model based on continuous time intervals.

(b) First representation: Use of mixed edges.



(c) Second representation: Snapshots.

(d) Third representation: Use of temporal and spatial edges.

Figure 10: Converting continuous time intervals into a discrete TVG.

Figure 11 shows the possible non-zero entries for a TVG with four nodes and three time instants based on the model of continuous time intervals using the first representation we are describing. Clearly from Figure 11, this kind of TVG is a subspace (and therefore a subset) of the TVGs representable by our proposed unifying model. Thus, we conclude that TVGs of this class are particular cases of the TVGs representable in our model.

Even though this first representation by our model carries all information present in the original TVG based on continuous time intervals, it still relies on an assumption that is not explicitly present in this first representation. Namely, this assumption concerns that the time instants present on each mixed dynamic edge represent the time intervals on which the edge exists. Note that the same happens in the original model based on continuous time intervals. For example, using this assumption, a path between nodes 0 and 2 exists at any given time instant during the interval between t_5 and t_7. To understand that this assumption is made in a manner that does not directly follows as a property of the structure of the TVG, notice that it makes the transitivity of the relation implied by an edge more difficult and cumbersome to determine. This happens because an external processing is needed to verify the presence of the edge (either by the time intervals indicated on the mixed dynamic edge or by the set of intervals on the continuous time model) in order to be able to establish its transitivity to another edge. To establish the existence of a path connecting two nodes, it is necessary to compute the intersections of the time intervals on which each edge exists. The resulting path then exists on the time interval obtained by this intersection. Therefore, the existence of a path connecting nodes in this kind of representation can be non-intuitive and computationally expensive to determine, thus impacting considerations about connectivity, reachability, communicability, and any other property derived from the transitivity of edges in a TVG.

A second representation (Figure 10(c)) comes from the realization that the continuous time

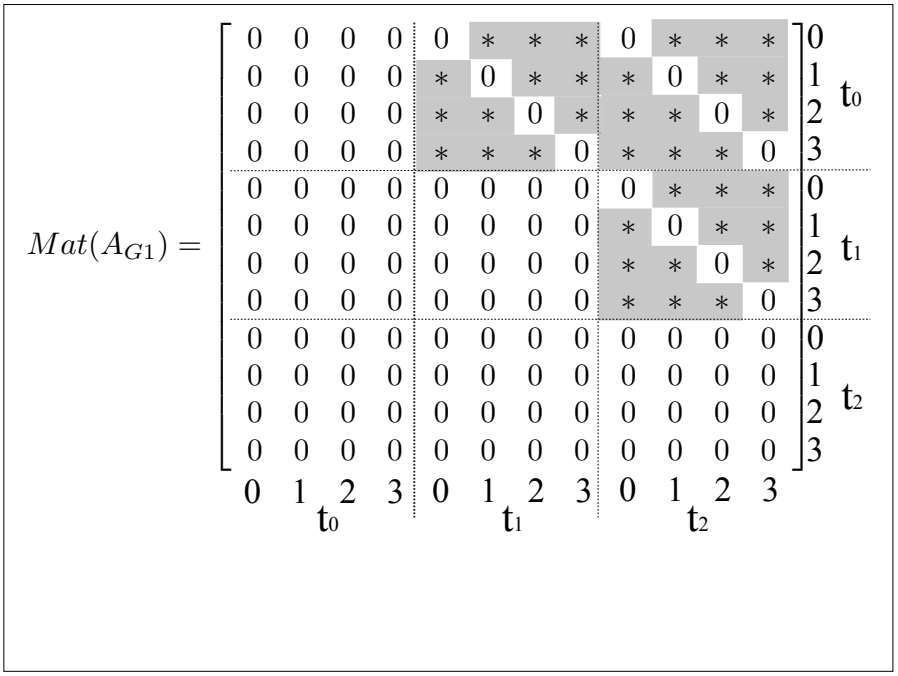


Figure 11: The first representation of TVGs based on continuous time intervals.

intervals model is in fact a continuos form of the snapshot model, in which edges are present at infinite (uncountable) time instants. Therefore, a natural way to represent the CTI model in a discrete form is to have the time instants set T formed by the beginning and end instants of each interval present on the CTI model, and then placing the corresponding edge at each time instant for which an edge is present in the CTI model. The result of this is a snapshot representation of the CTI model, as shown in Figure 10(c), where the edge between nodes 0 and 1 is present at instants t_0, t_5, t_7, and t_{15}, while the edge between nodes 1 and 2 is present at instants t_5 and t_7. This representation has the same characteristics we highlighted for the snapshot class of models. Although the TVG connectivity can be readily determined at each time instant, the snapshots (i.e. time instants) are formally disconnected from each other. Hence, if any sort of connectivity between time instants is to be considered, it has to be stated as an additional assumption, which is not directly encoded in the representation.

A third representation (Figure 10(d)) can be derived from the second one (snapshots), by formally placing temporal dynamic edges where the connectivity between successive time instants is desired. In our example, this leads to the representation shown in Figure 10(d), where temporal nodes are used to connect nodes 0 and 1 between time instants t_0, t_5, t_7, and t_{15} and to connect nodes 1 and 2 from time instant t_5 to t_7. This reflects the fact that in the original CTI model the edge $(0, 1)$ was present at the interval $(t_0, t_{15}]$, while the edge $(1, 2)$ was present at the interval $(t_5, t_7]$. Figure 12 shows the possible non-zero entries on the matrix form of the adjacency tensor of a TVG of this class using this third representation.

In order to determine the memory complexity of representing a TVG based on continuous time intervals in our proposed unifying model, we state the following proposition, based on the third representation, presented in Figure 10(d), which uses the largest number of dynamic edges:

Proposition 4.1 *A TVG based on continuous time intervals with n nodes, m edges, and η*

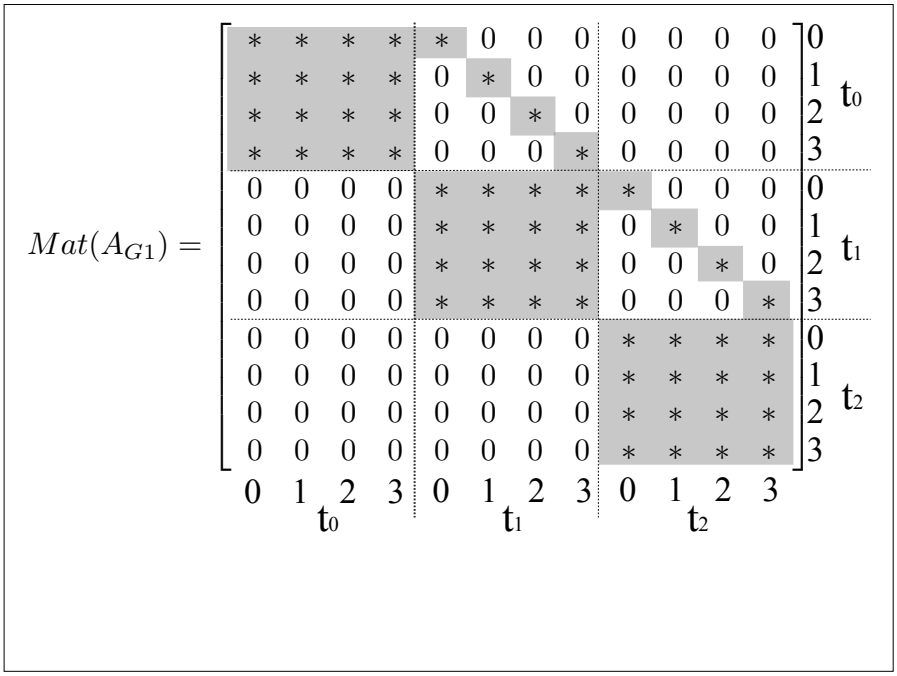


Figure 12: The third representation of TVGs based on continuous time intervals.

continuous time intervals defining these edges can be represented in our model with $O(\eta^2)$ dynamic edges.

Proof 4.1 *For each time interval in the continuous time representation, a mixed dynamic edge is created on our model. Since each mixed edge requires two time instants to be defined, it follows that the representation in our model requires at most 2η time instants. Therefore, in our model we have $|T| = 2\eta$ in the worst case.*

To decompose a mixed dynamic edge into spatial and temporal dynamic edges, at most $|T|$ spatial dynamic edges are needed to connect the two nodes on the mixed edge in all possible time instants. Further, at most $2(|T|-1)$ temporal dynamic edges are needed to connect the nodes over all possible time instants. Therefore, to fully decompose a mixed edge, at most $2(|T|-1)+|T| = 3|T|-2$ dynamic edges are needed.

Hence, as η mixed dynamic edges are needed in the representation, to fully decompose them into spatial and temporal edges, at most $\eta(3|T|-2)$ dynamic edges are needed. Expanding and substituting $|T| = 2\eta$ for the worst case, we have

$$\begin{aligned}\eta(3|T|-2) &= \eta(6\eta-2)\\ &= 6\eta^2-2\eta.\end{aligned}$$

We therefore conclude that the number of dynamic edges needed is $O(\eta^2)$.

From this, we further conclude that, for a TVG with η continuous time intervals, in our model we have $|E| = O(\eta^2)$. Therefore, the memory complexity of the representation based on continuous time instants is $O(\eta^2)$, which is compatible with the memory complexity discussed in Section 3.5.

4.3 Models based on spatial and temporal edges (STE)

Some models like the one proposed by Kostakos [Kos09] are based on the idea that a class of links represent instantaneous iterations between distinct nodes while other class represent a waiting state of a given node. These concepts are formalized in our proposed unifying model by spatial and temporal dynamic edges. In our model, these dynamic edges are fully formalized and can be used to make a unambiguous representation of this kind of TVG. Figure 13 depicts the entries that may be non-zero on the matrix form of the adjacency tensor of a TVG of this class having four nodes and three time instants. It is again clear that this class of TVGs can be thus represented as a subset of the TVGs that can be represented by the unifying model we propose.

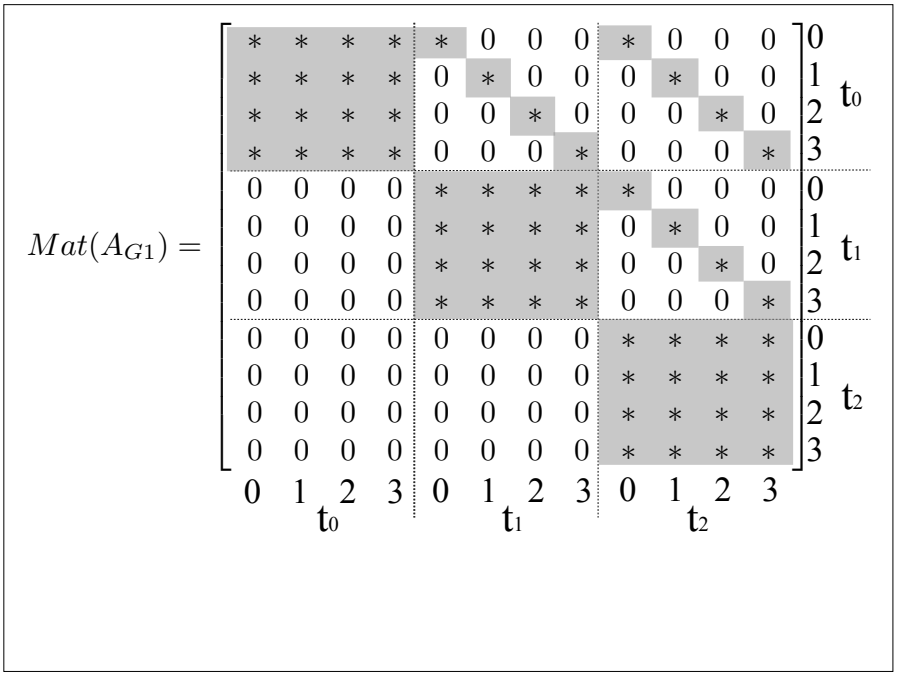


Figure 13: Representation of TVGs based on spatial and temporal edges.

4.4 Models based on temporal and mixed edges (TME)

Some works found in the recent literature, such as the one by Kim and Anderson [KA12], loosely suggest the use of edges connecting nodes at different time instants. TVGs of this class can be represented in our model using only temporal and mixed dynamic edges. Figure 14 shows the matrix form of the adjacency tensor of a TVG of this class. It can be seen that this is also a particular case of the TVGs that can be represented using our unifying model.

4.5 Overview on the unifying representation of previous models

Table 1 shows a representation map that indicates if a (class of) TVG model(s) is able to represent another (class of) TVG model(s). We compare the unifying representation model we propose with the models based on (i) snapshots, (ii) continuous time intervals (CTI), (iii) spatial and

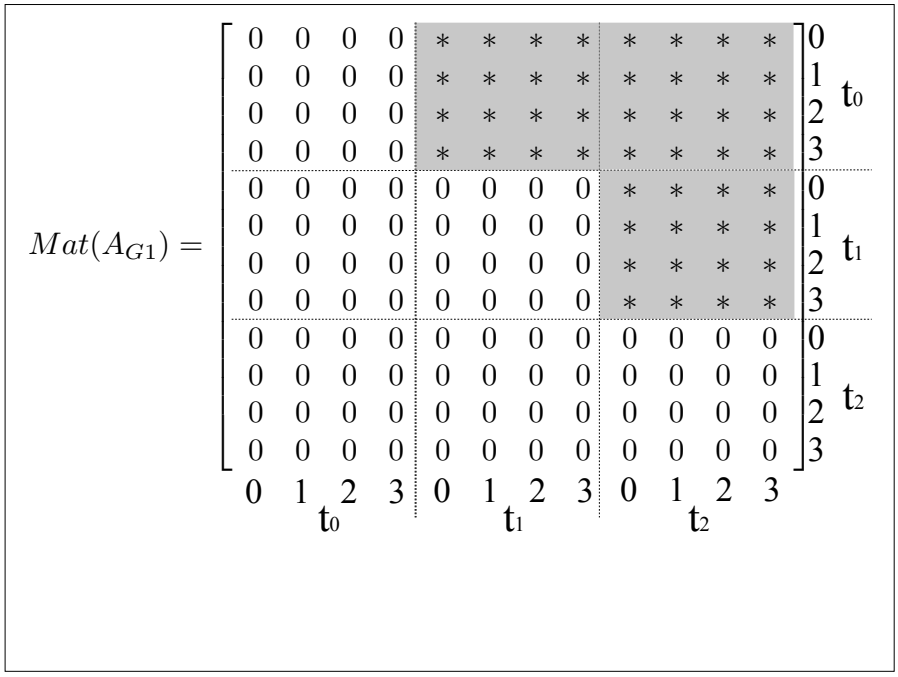


Figure 14: Representation of TVGs based on temporal and mixed edges.

temporal edges (STE), and (iv) temporal and mixed edges (TME), which have been presented in Sections 4.1 to 4.4, respectively.

Snapshot models can only represent TVGs of the continuous time intervals (CTI) class, since they are in fact a continuos time version of the snapshot model. Direct representation of other models by snapshots is not possible because the snapshot model lacks the notion of temporal edges. Models based on continuous time intervals (CTI) and spatial and temporal edges (STE) can represent snapshot models because they support the notion of spatial edges as well as they are mutually able to represent each other. Models based on temporal and mixed edges (TME) are able to represent models based on continuous time intervals (CTI) in a discretized way. Remark that the converse is not true, i.e. models based on CTI are unable to represent models based on TME as they lack the notion of mixed edges. Furthermore, note that models based on TME are unable to represent in all cases models with spatial edges, such as the ones based on snapshots and STE. This is because, although a mixed edge could be seen as a composition of a spatial and a temporal edge, a mixed edge is unable to represent a single spatial edge, thus preventing the representation of these cases.

Overall, we remark that considering the previous classes of TVG models, none is able to represent all others. This basically happens because the kind of edges present in one model not necessarily can be transformed into the kind of edges present in another model or class of models. In contrast, we have shown along this section that all classes of previous TVG models we consider can be represented in the unifying model we propose, whereas these previous TVG models not necessarily are able to represent each other, as shown in Table 1. Additionally, none of the previous TVG models makes use of regressive edges, which could be used in our proposal to intrinsically model cyclic (i.e. periodic) behavior in dynamic networks. Therefore, we highlight that none of the previous analyzed models have all the representation capabilities and thus the

Table 1: Representation map between TVG models: An entry is checked if the (class of) model(s) in the row is able to represent the (class of) model(s) in the column.

	Snapshots	CTI	STE	TME	Unifying model
Snapshots	✓	✓			
CTI	✓	✓	✓		
STE	✓	✓	✓		
TME		✓		✓	
Unifying model	✓	✓	✓	✓	✓

same expressive power of our proposed unifying model for TVGs.

5 Summary and outlook

In this paper, we have proposed a novel model for representing finite discrete Time-Varying Graphs (TVGs). We have shown that our model is simple, yet flexible and efficient for the representation and modeling of dynamic networks. The proposed model preserves the discrete nature of the basic graph abstraction and has algebraic representations similar to the ones used on a regular graph. Moreover, we have also shown that our unifying model has enough expressive power to represent several previous (classes of) models for dynamic networks found in the recent literature, which in general are unable to represent each other, and also to intrinsically represent cyclic (i.e. periodic) behavior of dynamic networks. To further illustrate the flexibility of our model, we remark that our model can also be used to represent time schedules using a TVG. For example, in this case, the nodes can be thought of as locations and mixed edges as the amount of time taken to move between locations. Such representation can model scheduled arrivals and departures in transportation systems allowing, for instance, the evaluation whether a connection is feasible or the delivery time for logistics management.

We have analyzed the proposed model proving that if the TVG nodes can be considered as independent entities at each time instant, the analyzed TVG is isomorphic to a directed static graph. This basic theoretical result has provided the ground for achieving other theoretical results that show that some properties of the analyzed TVG can be inferred from the temporal node representation of that TVG. This is an important set of theoretical results because this allows the use of the isomorphic directed graph as a tool to analyze both the properties of a TVG and the behavior of dynamic processes over a TVG. We have also demonstrated that, for most practical cases, the asymptotic memory complexity of our TVG model is determined by the cardinality of the set of edges. Further, from the basic definition of the model we proposed for representing TVGs, we have derived some basic properties such as communicability and connectivity using only properties that follow directly from the underlying structure of the model, such as the transitivity of the relation induced by the dynamic edges. As a consequence, in contrast to previous works, our model can be used without the need of external assumptions, meaning that all properties are derived from explicit properties of the TVG, in the same way that happens with static graphs.

As future work, we intend to build upon our proposed model, using it to analyze the behavior of dynamic processes over TVGs, such as random walks or diffusion processes, and also to model the dynamics of different networks. Moreover, given the encouraging features shown by our unifying model for representing TVGs, we believe it is promising to further study and explore the potential of this model. In particular, we intend to further investigate the combinational

and algebraic properties of this model, as a step forward while aiming at the development of a fundamental theory for TVGs parallel to the existing one for static graphs.

Acknowledgment

This work was partially funded by the Brazilian funding agencies CAPES, CNPq, and FAPERJ as well as the Brazilian Ministry of Science, Technology, and Innovation (MCTI).

References

[AHFVZ12] José Ignacio Alvarez-Hamelin, Éric Fleury, Alessandro Vespignani, and Artur Ziviani. Complex dynamic networks: Tools and methods (guest editorial). *Computer Networks*, 56(3):967–969, December 2012.

[BBC+94] Richard Barrett, Michael Berry, Tony F. Chan, James Demmel, Junde M. Donato, Jack Dongarra, Victor Eijkhout, Roldan Pozo, Charles Romine, and Henk Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, PA, 1994.

[BBV08] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. *Dynamical Processes on Complex Networks*. Cambridge University Press, 2008.

[BJG08] Jürgen Bang-Jensen and Gregory Z. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer Publishing Company, Incorporated, 2nd edition, 2008.

[BXFJ03] Bình Minh Bùi-Xuân, Afonso Ferreira, and Aubin Jarry. Evolving graphs and least cost journeys in dynamic networks. In *Proceedings of WiOpt – Modeling and Optimization in Mobile, Ad-Hoc and Wireless Networks*, pages 141–150, Sophia Antipolis, France, 2003. INRIA Press.

[CFQS12] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, October 2012.

[CKN08] Ioannis Chatzigiannakis, Athanasios Kinalis, and Sotiris Nikoletseas. Efficient data propagation strategies in wireless sensor networks using a single mobile sink. *Computer Communications*, 31(5):896–914, March 2008.

[Fal03] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the ACM SIGCOMM*, pages 27–34, New York, NY, USA, 2003. ACM Press.

[Fer04] Afonso Ferreira. Building a reference combinatorial model for MANETs. *IEEE Network*, 18(5):24–29, September 2004.

[FNR+12] Daniel Figueiredo, Philippe Nain, Bruno Ribeiro, Edmundo de Souza e Silva, and Don Towsley. Characterizing continuous time random walks on time varying graphs. *ACM SIGMETRICS Performance Evaluation Review*, 40(1):307–318, June 2012.

[GaVSZ13] Abraão Guimarães, Alex B. Vieira, Ana Paula C. Silva, and Artur Ziviani. Fast centrality-driven diffusion in dynamic networks. In *Proceedings of the Workshop on Simplifying Complex Networks for Practitioners (SIMPLEX), World Wide Web conference (WWW)*, pages 821–828, New York, NY, USA, 2013. ACM Press.

[GDM09] Niloy Ganguly, Andreas Deutsch, and Animesh Mukherjee, editors. *Dynamics On and Of Complex Networks*. Springer, 2009.

[Hol05] Petter Holme. Network reachability of real-world contact sequences. *Physical Review E*, 71(4):1–8, April 2005.

[HS12] Petter Holme and Jari Saramäki. Temporal networks. *Physics Reports*, 519(3):97–125, October 2012.

[HS13] Petter Holme and Jari Saramäki, editors. *Temporal Networks*. Springer, 2013.

[IFM09] Marios Iliofotou, Michalis Faloutsos, and Michael Mitzenmacher. Exploiting Dynamicity in Graph-based Traffic Analysis: Techniques and Applications. In *Proceedings of the ACM CoNEXT*, pages 241–252, New York, NY, USA, 2009. ACM Press.

[KA12] Hyoungshick Kim and Ross Anderson. Temporal node centrality in complex networks. *Physical Review E*, 85(2):026107, February 2012.

[KKT03] David Kempe, Jon Kleinberg, and Eva Tardos. Maximizing the Spread of Influence through a Social Network. In *Proceedings of the ACM SIGKDD*, pages 137–146, New York, NY, USA, 2003. ACM Press.

[Kol06] Tamara G Kolda. Multilinear operators for higher-order decompositions. Technical Report SAND2006-2081, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, USA, 2006.

[Kos09] Vassilis Kostakos. Temporal graphs. *Physica A: Statistical Mechanics and its Applications*, 388(6):1007–1023, March 2009.

[PSV01] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic spreading in scale-free networks. *Physical Review Letters*, 86(14):3200–3203, 2001.

[SVZ12] Cigdem Sengul, Aline Carneiro Viana, and Artur Ziviani. A survey of adaptive services to cope with dynamics in wireless self-organizing networks. *ACM Computing Surveys*, 44(4):23:1–23:35, September 2012.

[TMM+10] John Tang, Mirco Musolesi, Cecilia Mascolo, Vito Latora, and Vincenzo Nicosia. Analysing information flows and key mediators through temporal centrality metrics. In *Proceedings of the Workshop on Social Network Systems (SNS)*, page Article no. 3, New York, NY, USA, 2010. ACM Press.

[TMML09] John Tang, Mirco Musolesi, Cecilia Mascolo, and Vito Latora. Temporal distance metrics for social network analysis. In *Proceedings of the ACM workshop on Online social networks (WOSN)*, pages 31–36, New York, NY, USA, 2009. ACM Press.

[TSM+10] J. Tang, S. Scellato, M. Musolesi, C. Mascolo, and V. Latora. Small-world behavior in time-varying graphs. *Physical Review E*, 81(5):81–84, May 2010.

[VVZF10] Massimo Vecchio, Aline Carneiro Viana, Artur Ziviani, and Roy Friedman. DEEP: Density-based proactive data dissemination protocol for wireless sensor networks with uncontrolled sink mobility. *Computer Communications*, 33(8):929–939, May 2010.

Contents

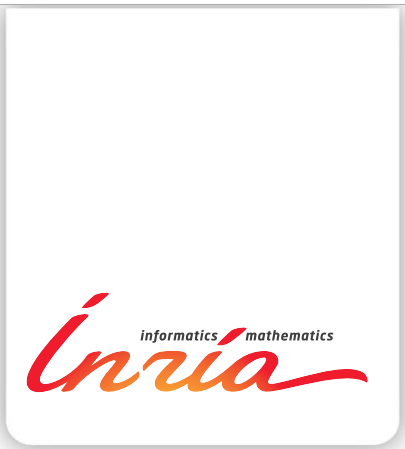
1 Introduction **3**

2 Proposed model for representing TVGs and its main properties **4**
2.1 Proposed model for representing TVGs 4
2.2 TVG isomorphism 5
2.3 Isomorphism between TVGs and directed graphs 6
2.3.1 Order preserving isomorphism 8
2.3.2 Natural isomorphism 8
2.3.3 Generalized isomorphism 9
2.4 TVG representation by temporal nodes 9
2.5 Aggregated directed graph 11
2.6 Degree 11
2.7 Adjacency 12
2.8 Walks, trails, paths, and cycles 14
2.8.1 Walk 14
2.8.2 Trail 16
2.8.3 Path 16
2.8.4 Cycle 17
2.9 Shortest paths on a TVG 18
2.10 Strongly connected TVG 19
2.10.1 Closed strongly connected TVG 20
2.10.2 Temporal node strongly connected TVG 21

3 Representations and algebraic structures of a TVG **21**
3.1 TVG adjacency tensor 22
3.2 Matrix form of the TVG adjacency tensor 22
3.3 TVG incidence tensor 24
3.4 Matrix form of the TVG incidence tensor 24
3.5 Memory complexity for storing a TVG 25

4 Unifying the representation of previous models **27**
4.1 Models based on snapshots 27
4.2 Models based on continuous time intervals (CTI) 28
4.3 Models based on spatial and temporal edges (STE) 33
4.4 Models based on temporal and mixed edges (TME) 33
4.5 Overview on the unifying representation of previous models 33

5 Summary and outlook **35**



RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399