

Exact and Approximate Role Assignment for Multi-Layer Networks

TATJANA PETROV

Universitätsstraße 10, 78457 Konstanz, Germany.

STEFANO TOGNAZZI*

Universitätsstraße 10, 78457 Konstanz, Germany.

*Corresponding author: stefano.tognazzi@uni-konstanz.de

[Received on 19 July 2021]

The concept of role equivalence has been applied in social network analysis for decades. Early definitions recognized two social actors as role equivalent, if they have identical relationships to the same other actors. Although this rather strong equivalence requirement has been relaxed in different ways, it is often challenging to detect interesting, non-trivial role equivalences, especially for social networks derived from empirical data. Multi-Layer Networks (MLNs) are increasingly gaining popularity for modelling collective adaptive systems, e.g. engineered cyber-physical systems or animal collectives. Multiplex networks, a special case of MLNs, transparently and compactly describe such complex interactions (social, biological, transportation), where nodes can be connected by links of different types. In this work, we first propose a novel notion of exact and approximate role equivalence for multiplex MLNs. Then, we implement and experimentally evaluate the algorithm on a suite of real-world case studies. Results demonstrate that our notion of approximate role assignment not only obtains non-trivial partitions over nodes and layers as well, but it provides a fine-grained hierarchy of role equivalences, which is impossible to obtain by (combining) the existing role detection techniques. We demonstrate the latter by interpreting in detail the case study of Florence families, a classical benchmark from literature.

Keywords: multi-layer networks, role assignment, exact and approximate model reduction

1. Introduction

Large amounts of data created by social interactions among physical and virtual actors open up novel challenges for (i) representing and modelling these interactions, as well as for (ii) automated extraction of useful information from models and data-sets alike. Traditional network analysis has facilitated these tasks by using (single-layer) networks to model interaction between agents; A variety of analysis methods have been developed, aiming to discover the social role of a node within the context of the network [30, 37], the network's structural properties [38], the relative importance of a node with respect to the rest of the network [4, 26, 42], the emergence of different communities (or clusters) [5, 6, 20]. However, all these techniques are applicable to single-layer networks, which allow to describe only one type of interaction between agents. In many real-world scenarios, relations among individuals have multiple facets and it is often advantageous to represent the data by a multiplex network [15]. A multiplex network is a special case of multi-layer network (MLNs) [27], where the same set of nodes is connected through different interaction *types* (colors), each type (color) visible at a different *layer*. Multiplex networks were shown useful to model a variety of real-world scenarios, including physical and virtual social interactions [23], infection spreading processes [49], biological interactions [3, 39], transportation networks [11, 14]. For instance, human users of online social networks can both interact

and share information either in the real-world and/or using a variety of online social networks [23]. Animals belonging to the same collective (herd, fish school, etc.) can relate to each other differently through different activities such as grooming, social aggregation, foraging, as shown for baboons [1, 21], dolphins [22] and birds [19]. Two locations in a transportation network can be connected by different means of transportation [14].

Multiplex networks have a richer structure than single-layer networks, and carrying over the theory and analysis from classical network analysis to Multiplex MLNs is desirable but non-trivial. Many of the concepts and algorithms do not naturally translate to multiplex networks, since the interpretation of nodes and layers may differ across application domains. Notable efforts have been done for community detection algorithms [35].

In this work, we focus on centrality notions for Multiplex MLNs (i.e., the importance of a node in the network) and, more specifically, on role equivalence (the detection of social roles of individuals [2, 16, 46, 52]). It is known that role discovery in single layer networks has points in common with centrality measures [30], bisimulation notions [32], equitable partitions [47] and graph isomorphism [33, 34]. For instance: regular equivalence, which is one of the simplest form of role equivalence, can be related to bisimulation [32]. Exact role assignment, instead, is shown to share properties with centrality measures, equitable partitions and efficient model reduction techniques for dynamical systems [51].

One of the main challenges for algorithmic role discovery is that the existing notions usually discover trivial partitions. A notion based on strong equivalence criterion typically leads to a trivial result in which each node has a role of his own. On the other hand, a notion with weaker requirements leads to a trivial partition in which all the nodes collapse in a single role.

The contributions of this paper are following. First, we propose a novel notion of exact role assignment to Multiplex MLNs, based on an extension of eigenvector centrality and on Backward Differential Equivalence, an efficient model reduction technique for dynamical systems. This approach offers multiple advantages: the proposed extension of eigenvector centrality simultaneously takes into account nodes and layers, different to many of the related works, e.g. for community detection [35] or aggregation [12], which focus on either nodes or layers separately. The algorithm for detecting exact role equivalence relies on a polynomial algorithm based on Paige-Tarjan celebrated partition-refinement technique. However, as mentioned above, in real-world cases which rarely exhibit symmetries, this algorithm often does not provide partitions useful in practice.

Our second contribution is a novel notion of approximate role assignment for Multiplex MLNs, inspired by an approximate version of Backward Differential Equivalence. The respective algorithm for detecting the approximate role equivalence is NP-hard, however it allows us to trade-off the runtime complexity for the benefit of obtaining a fine-grained hierarchy of role-equivalences. Finally, we implement and experimentally evaluate the algorithm on a suite of real-world case studies. Results confirm that our notion of approximate role assignment not only obtains non-trivial partitions over nodes and layers alike, but it provides a fine-grained hierarchy of role equivalences, which is impossible to obtain by (combining) the existing role detection techniques. We demonstrate the latter by interpreting in detail the case study of Florence families, a classical benchmark from literature.

PAPER OUTLINE Section 2 reviews the background notions, while Section 3 introduces the novel exact and approximate role assignment for Multiplex MLNs. Section 4 provides an overview of the implementation and the results of our experimental evaluation on real-world multiplex MLNs. Section 5 presents a case-study on the Florence data-set. In Section 6 we discuss and compare the proposed notion with other notions from the literature. Section 7 concludes the paper.

2. Background

In this section we provide an overview of the notions that will be used throughout the paper: single- and multi-layer networks (MLNs), role equivalence for single-layer networks, eigenvector centrality measures for MLNs and exact and approximate model reduction techniques based on Backward Differential Equivalence (BDE).

Notation. Throughout this work, when clear from context, we will use x_i both to denote the i -th element of vector \mathbf{x} or the value of the map $x(i)$ (following Definition 2.3). For a partition \mathcal{H} over a variable set $V_p \subseteq \{x_1, x_2, \dots\}$, induced by an equivalence relation $\sim_{\mathcal{H}} \subseteq V_p \times V_p$, we will denote elements of a partition class $H \in \mathcal{H}$ by $x_{H,1}, x_{H,2}, \dots, x_{H,|H|}$. We denote by $\|\cdot\|_1$ the 1-norm. We will denote with $V_N = \{1, \dots, N\}$, $V_L = \{1, \dots, L\}$ the set of nodes and layers, respectively. We will assume $N > 0$ and $L > 1$ (MLNs with $L = 1$ are a special case that corresponds with traditional single-layer networks). Vectors will be assumed to be written in column notation.

2.1 Role Assignment in single layer networks

The problem of discovering the role of each node in a network has been studied in different fields and many notions of *role equivalence* have been given. Regular equivalence, one of the first notions of role assignment, is an attempt to capture the sociological notion of a relational or structural role. Two nodes are said to play the same role (i.e., are regularly equivalent) if they have ties to the same roles. Regular equivalence is the weakest form of similarity between nodes and it has been shown in [32] that it can be related to the concept of bisimulation. Many variants of regular equivalence have been defined in the literature. In this work we will focus on the definition of *exact role assignment* on undirected graphs introduced in [30, 54].

DEFINITION 2.1 Given a undirected and unweighted graph (single-layer network) $G = (V, E)$ where $V \neq \emptyset$ denotes the set of nodes and $E \subseteq V \times V$ denotes the set of edges. Let $\mathbf{A} \in \{0, 1\}^{V \times V}$ symmetric adjacency matrix be the adjacency matrix of G and $i \in V$, let $N(i)$ denote the neighbours of i , that is $N(i) := \{j \in V \mid (i, j) \in E\}$. A surjective mapping $r : V \rightarrow \hat{V}$ with $\hat{V} \subseteq V$ is an exact role assignment if, for all $i, j \in V$

$$r(i) = r(j) \Rightarrow r(N(i)) = r(N(j))$$

where the last equation is an equation of multi-sets, i.e., vertices, that have the same role, must have the same number of each of the other roles in their neighborhoods.

2.2 Multiplex Multi Layer Networks

In this paper, we will work with a generalisation of networks called *multiplex networks* or *edge-colored-graphs*, which are useful for simultaneously representing different kinds of relationships over the same set of nodes [15]. This paper will focus on undirected multiplex networks.

DEFINITION 2.2 A *multiplex network* with N nodes and L layers is an ordered collection of L undirected graphs over the same set of nodes:

$$\mathcal{G} = \{G^{(l)} = (V_N, E^{(l)})\}_{l \in V_L},$$

where $E^{(l)} : V_N \times V_N \rightarrow \mathbb{R}_{\geq 0}$ are the edges on layer $l \in V_L$. For every layer l , we denote the non-negative adjacency matrix of the graph $G^{(l)}$ by $\mathbf{A}^{(l)} = (A_{ij}^{(l)}) \in \mathbb{R}_{\geq 0}^{N \times N}$. Then, the multiplex network can

be represented by a 3rd-order *adjacency tensor*:

$$\mathcal{A} = (\mathcal{A}_{ijl}) \in \mathbb{R}_{\geq 0}^{N \times N \times L}, \text{ such that } \mathcal{A}_{ijl} := A_{ij}^{(l)} = E^{(l)}(i, j),$$

that is, \mathcal{A}_{ijl} represents the presence of an edge between nodes i and j on layer l . Throughout this work we will consider unweighted networks unless stated otherwise.

2.3 Centrality and Versatility measures

Historically, centrality measures have been used to identify which nodes in a network should be considered important. In this work we start from *eigenvector centrality* on single-layered networks based on the following definition given in [52]:

DEFINITION 2.3 *Eigenvector centrality* $x : V_N \rightarrow \mathbb{R}_{\geq 0}$ maps each node to the weighted sum of eigenvector centralities of all nodes directly reachable from it: for $i \in V_N$, $x(i) = \frac{1}{\lambda} \sum_{j \in V_N} A_{ij} x(j)$, where $\frac{1}{\lambda}$ is some positive constant. In vector notation, the eigenvector centrality vector $\mathbf{x} \in \mathbb{R}_{\geq 0}^N$ is such that $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$, that is, \mathbf{x} is the right eigenvector wrt. the adjacency matrix \mathbf{A} .

Many extensions of centrality measures have been proposed for multiplex MLNs [2, 16, 17, 44, 46] and they are often referred as *versatility measures* because they tend to highlight nodes that play an important (central) role simultaneously across layers. In this paper, we will use one possible extension of eigenvector centrality for multiplex MLNs, proposed in [52]. The authors propose a 2-map, *f-eigenvector centrality*, in which the first component of the map $\mathbf{x} \in \mathbb{R}^N$ represents the centrality associated to the *nodes*, while the second component $\mathbf{t} \in \mathbb{R}^L$ is the centrality associated to the *layers*.

DEFINITION 2.4 ([52]) Let $\mathcal{A} \in \mathbb{R}_{\geq 0}^{N \times N \times L}$ be the adjacency tensor of a Multiplex MLN with undirected layers (i.e., each layer is an undirected graph), and let $\alpha, \beta > 0$ be such that $\frac{2}{\beta} < (\alpha - 1)^1$. Then, define $\mathbf{f} = (f_1, f_2) : \mathbb{R}_{\geq 0}^N \times \mathbb{R}_{\geq 0}^L \rightarrow \mathbb{R}_{\geq 0}^N \times \mathbb{R}_{\geq 0}^L$ as follows:

$$f_1(\mathbf{x}, \mathbf{t})_i = \left(\sum_{j=1}^N \sum_{l=1}^L A_{ijl} x_j t_l \right)^{\frac{1}{\alpha}} \text{ for } i \in V_N,$$

$$f_2(\mathbf{x}, \mathbf{t})_l = \left(\sum_{i=1}^N \sum_{j=1}^N A_{ijl} x_i x_j \right)^{\frac{1}{\beta}} \text{ for } l \in V_L.$$

In the case of single layer networks a power iteration scheme is used to calculate the centrality defined in Definition 2.3. Similarly, a power iteration scheme for computing \mathbf{f} -centrality is desired. Throughout the rest of the work, we will use a normalised version of \mathbf{f} -mapping, denoted by \mathbf{g} :

$$\mathbf{g}(\mathbf{x}, \mathbf{t}) = \left(\frac{\mathbf{f}_1(\mathbf{x}, \mathbf{t})}{\|\mathbf{f}_1(\mathbf{x}, \mathbf{t})\|_1}, \frac{\mathbf{f}_2(\mathbf{x}, \mathbf{t})}{\|\mathbf{f}_2(\mathbf{x}, \mathbf{t})\|_1} \right)$$

We now restate a result from [52], that, for a given MLN with undirected layers, \mathbf{f} -centrality is well-defined and it can be computed as a limit of a power iterative sequence.

¹We refer the interested reader to the original reference, for a discussion on the choice of α and β .

THEOREM 2.5 ([52]) There exists a unique, non-negative fixed point of the mapping \mathbf{g} . Moreover, this fixed point, denoted by $(\mathbf{x}^*, \mathbf{t}^*) \in \mathbb{R}_{\geq 0}^N \times \mathbb{R}_{\geq 0}^L$, is a limit of the following iterative scheme ²:

$$(\mathbf{x}^{(k)}, \mathbf{t}^{(k)}) = \mathbf{g}(\mathbf{x}^{(k-1)}, \mathbf{t}^{(k-1)}) \text{ for } k \geq 1 \quad (2.1)$$

$$(\mathbf{x}^{(0)}, \mathbf{t}^{(0)}) = (\mathbf{1}^N, \mathbf{1}^L) \quad (2.2)$$

Notice that, from the definition of \mathbf{g} , independently of $k \geq 0$, it holds that $\|\mathbf{x}^{(k)}\|_1 = \|\mathbf{t}^{(k)}\|_1 = 1$, including the limit value $(\mathbf{x}^*, \mathbf{t}^*)$.

2.4 Backward Differential Equivalence

Backward differential equivalence (BDE) is a model reduction technique for dynamical systems [7, 9]. In this work we will focus on PIVP (Polynomial Initial Value Problems) which are a class of Ordinary Differential Equations (ODEs). In general PIVP are defined by the ODEs $x'_i = q_i$, $1 \leq i \leq n$ where q_i is a multivariate polynomial over the set of variables $\mathcal{S} = \{x_1, \dots, x_n\}$. **Variables that have the same ODE solution whenever they are given the same initial assignment are called *exact fluid lumpable*. BDE is a characterisation for the condition of being exact fluid lumpable (i.e., BDE groups variables which are exact fluid lumpable).** Finding the (largest) BDE amounts to finding the coarsest partition over the variable set, which ensures that the semantic criterion is met. This criterion allows to construct a smaller PIVP, using only one representative variable from each partition class. The reduction algorithms proposed in [7, 9] are only syntactically manipulating the ODEs, and they are of polynomial complexity in the number of variables of the program.

DEFINITION 2.6 We call $\mathbf{x} \in \mathbb{R}_{\geq 0}^{|V_p|}$ *constant on \mathcal{H}* if for all $H \in \mathcal{H}$ and all $x_i, x_j \in H$, it holds that $x_i = x_j$.

DEFINITION 2.7 Let p be an PIVP and \mathcal{H} a partition over the variable set V_p . Then, the PIVP p is *exact fluid lumpable wrt. partition \mathcal{H}* , if $\mathbf{x}(t)$ is constant on \mathcal{H} for all $t \geq 0$, whenever \mathbf{x}_0 is constant on \mathcal{H} . **We denote the solution of the PIVP at time t with $\mathbf{x}(t)$.** Then, we will call \mathcal{H} a *BDE partition* of V_p .

Following [7], the coarsest BDE partition can be computed in polynomial time complexity, for any PIVP which corresponds to a set of chemical reactions with mass-action kinetics.

Originally designed for reducing ODEs, BDE techniques have also been applied for reducing single-layer networks [51], continuous-time Markov chains (CTMCs) [9] and differential algebraic equations [50]. In particular, in [51], a property-preserving exact model reduction algorithm for networks is shown. The given network is first transformed into a PIVP, and then a BDE reduction ensuring exact fluid lumpability is applied. We restate a Theorem showing that BDE reduction also preserves the measure of eigenvector centrality. **Throughout the rest of the work, to avoid ambiguity, we will denote with x_i and t_i the variables associated to the centrality iterative scheme and with \mathbf{x}_i and \mathbf{t}_i the variables of the PIVP.**

THEOREM 2.8 ([51]) Given a graph $G = (V_N, E)$ with adjacency matrix \mathbf{A} , let p_G be the PIVP over the set of variables V_N :

$$x'_i = \sum_{1 \leq j \leq n} A_{ij} \cdot x_j, \text{ for all } i \in V_N.$$

²We refer the interested reader to the original reference, for a discussion on the error and rate of convergence.

Let x_i^* denote the eigenvector centrality of node i . Then, \mathcal{H} is a BDE of p_G if and only if, for all $H \in \mathcal{H}$ and for all $x_i, x_j \in H$, it holds that $x_i^* = x_j^*$.

In words, the transformation from network to PIVP is such that the equation for the derivative of variable x_i is the weighted sum of its direct (outgoing) edges³. So, the key idea in the transformation from the network to an PIVP is that the equations in the PIVP exactly match the iterative scheme for computing the centrality measure of interest. Notice that the obtained PIVP contains only linear transformations over its variables. The following result allows one to express exact role assignment as a BDE.

THEOREM 2.9 Given a symmetric adjacency matrix $\mathbf{A} \in \{0, 1\}^{V \times V}$ and a partition \mathcal{H} of V , set $r(i) = \hat{i}_H$ when $i \in H$ for $H \in \mathcal{H}$ (i.e., \hat{i}_H is the representative of block H).

- r is an exact role assignment if and only if for all $H, H' \in \mathcal{H}$ and $i, j \in H$, it holds that $\sum_{k \in H'} A_{ik} = \sum_{k \in H'} A_{jk}$.

THEOREM 2.10 Let $\mathbf{A} \in \mathbb{R}^{V \times V}$ (in particular, \mathbf{A} may have negative entries) and \mathcal{H} a partition of V .

1. \mathcal{H} is a BDE of $\mathbf{x}^{(k+1)} = \mathbf{A}\mathbf{x}^{(k)}$ if and only if, for all $H, H' \in \mathcal{H}$ and $i, j \in H$, it holds that $\sum_{k \in H'} A_{ik} = \sum_{k \in H'} A_{jk}$.
2. In the case when \mathbf{A} is an adjacency matrix of an undirected graph, \mathcal{H} is a BDE of $\mathbf{x}^{(k+1)} = \mathbf{A}\mathbf{x}^{(k)}$ if and only if, for all $H, H' \in \mathcal{H}$ and $i, j \in H$, it holds that $\sum_{k \in H'} A_{ik} = \sum_{k \in H'} A_{jk}$.
3. In the case when \mathbf{A} is an adjacency matrix of an undirected graph, \mathcal{H} is a BDE of $\mathbf{x}^{(k+1)} = \mathbf{A}\mathbf{x}^{(k)}$ if and only if \mathcal{H} is an exact role assignment.

Proof. see, [51] (Theorem 10) □

Theorem 2.10 yields the following known result [30, 51]:

COROLLARY 2.1 Exact role assignment yields eigenvector centrality.

2.5 Approximate BDE

In [10] the notion of ε -BDE has been introduced on PIVP and we shall discuss how these systems relate to the graphs that we are considering. Let $\mathcal{V} = \{x_1, \dots, x_n\}$ be the set of variables of a PIVP defined by the ODEs $\dot{x}_i = q_i$, $1 \leq i \leq n$, where q_i is a multivariate polynomial over \mathcal{V} . We can consider polynomials in the normal form $x^\alpha = \prod_{x_i \in \mathcal{V}} x_i^{\alpha_{x_i}}$ where α is a multi-index and the monomial x^α appears in q_i at most once. For a polynomial q_i in normal form with variables in \mathcal{V} , let $c(q_i, x^\alpha)$ denote the coefficient of the monomial x^α . Armed with this definition we can recast the definition of BDE as the following.

DEFINITION 2.11 Fix a PIVP, a partition \mathcal{H} of \mathcal{V} and write $x_i \sim_{\mathcal{H}} x_j$ if all coefficients of the following polynomial are zero, where $[x_i/x_j]$ denotes the variable substitution of x_i with x_j . $x_{H',k}$ denotes the k -th element of block H' and $x_{H'}$ denotes the representative of block H' .

$$\mathcal{P}_{i,j}^{\mathcal{H}} = (q_i - q_j)[x_{H',1}/x_{H'}, \dots, x_{H',|H'|}/x_{H'} : H' \in \mathcal{H}]$$

³in case of symmetric graphs, ingoing and outgoing edges will be indistinguishable and overall neighbours are accounted for.

i.e., when

$$\sum_{\alpha \in \mathbb{N}_0^3} |c(\mathcal{P}_{i,j}^{\mathcal{H}}, \mathbf{x}^\alpha)| = 0 \quad (2.3)$$

A partition \mathcal{H} is a BDE if $\mathcal{H} = \mathcal{V} / (\sim_{\mathcal{H}}^* \cup \sim_{\mathcal{H}})$, where \sim^* denotes the transitive closure of a relation \sim .

If we relax the constraint of the difference between the coefficients to be exactly zero and allow for a tolerance ε we obtain the definition of *approximate BDE*.

DEFINITION 2.12 Fix a PIVP, a partition $\mathcal{H} = \{H_1, \dots, H_m\}$ of \mathcal{V} , and $\varepsilon \geq 0$. We write $\mathbf{x}_i \sim_{\mathcal{H}, \varepsilon} \mathbf{x}_j$ if

$$\sum_{\alpha \in \mathbb{N}_0^3} |c(\mathcal{P}_{i,j}^{\mathcal{H}}, \mathbf{x}^\alpha)| \leq \varepsilon$$

where $\mathcal{P}_{i,j}^{\mathcal{H}}$ is as in Definition 2.11. A partition \mathcal{H} is an ε -BDE if $\mathcal{H} = \mathcal{V} / (\sim_{\mathcal{H}, \varepsilon}^* \cap \sim_{\mathcal{H}})$.

The following result from [10] establishes the fact that the coarsest ε -BDE partition is unique. Discovering the coarsest ε -BDE partition is an NP-hard problem [10].

THEOREM 2.13 Fix a PIVP, a partition \mathcal{G} of \mathcal{V} , and $\varepsilon \geq 0$. Then, there exists a unique coarsest ε -BDE partition refining \mathcal{G} .

2.6 Exact reductions for MLNs

Given a multi-layered network, its \mathbf{f} -centrality can be computed with the iterative scheme \mathbf{g} presented in Theorem 2.5 (Eq. 2.1). Next we present how the definition of the multilayer network eigenvector centrality obtained with the iterative scheme (2.1) can be translated to a PIVP p such that there is a correspondence between the node and the layer eigenvector centrality and the variables of the PIVP.

DEFINITION 2.14 (PIVP translation) Let \mathcal{G} be a multiplex network and let \mathcal{A}_{ijl} be the 3-rd order adjacency tensor of the multiplex \mathcal{G} . We define a PIVP p , with $V_p = V_N \cup V_L$, as follows:

$$\mathbf{x}'_i = \sum_{j=1}^N \sum_{l=1}^L \mathcal{A}_{ijl} \mathbf{x}_j \mathbf{t}_l \quad \mathbf{t}'_l = \sum_{i=1}^N \sum_{j=1}^N \mathcal{A}_{ijl} \mathbf{x}_i \mathbf{x}_j$$

for all $i \in V_N$ and for all $l \in V_L$. With $\mathbf{x}_0 = \mathbf{1}^N$ and $\mathbf{t}_0 = \mathbf{1}^L$.

We provide an example of the PIVP translation presented in Definition 2.14 in Section A of the Appendix. The next Theorem shows how to write an PIVP, such that if two variables have the same semantics in the dynamical system, then, the respective nodes in the iterative scheme of a given MLN have identical centrality scores over all the steps of the computation, provided the equivalence over initial conditions.

THEOREM 2.15 Let \mathcal{G} be a multiplex network and let \mathcal{A}_{ijl} be the 3-rd order adjacency tensor of the multiplex network \mathcal{G} . Let \mathbf{f} be the mapping as defined in Definition. 2.4, and let \mathbf{g} its normalized version. Let $(\mathbf{x}^*, \mathbf{t}^*)$ be the unique solution (the centrality scores). Given any initial conditions $(\mathbf{x}^{(0)}, \mathbf{t}^{(0)}) \in \mathbb{R}_{\geq 0}^N \times \mathbb{R}_{\geq 0}^L$ and $(\mathbf{x}^{(k+1)}, \mathbf{t}^{(k+1)}) = \mathbf{g}(\mathbf{x}^{(k)}, \mathbf{t}^{(k)})$, the following holds:

$$\lim_{k \rightarrow \infty} (\mathbf{x}^{(k)}, \mathbf{t}^{(k)}) = (\mathbf{x}^*, \mathbf{t}^*)$$

Then, in the PIVP p with time horizon $T > 0$ obtained via Definition 2.14, for all $i, j \in \{1, \dots, N\}$ and $l, q \in \{1, \dots, L\}$, the following holds:

- If $\forall t \in [0, T) . x_i(t) = x_j(t)$ in PIVP p , then $\forall k \in \mathbb{N} . x_i^{(k)} = x_j^{(k)}$
- If $\forall t \in [0, T) . t_l(t) = t_q(t)$ in PIVP p , then $\forall k \in \mathbb{N} . t_l^{(k)} = t_q^{(k)}$

Similarly to the case of single layer networks this allows to extend the concept presented in Theorem 2.8 to MLNs. Now, we accompany Theorem 2.15 with the following lemma and definition to formally translate partition \mathcal{H} , which is defined over the PIVP's variables, to its counterpart \mathcal{H}^* defined over the nodes and the layers of the multiplex graph \mathcal{G} .

LEMMA 2.1 Let \mathcal{G} be a multiplex network and let p be the PIVP defined in Theorem 2.15 and let $\mathcal{H} = (\mathcal{H}_x, \mathcal{H}_t)$ be a BDE partition over the set of variables such that there is no overlap between the nodes and the layers, i.e. \mathcal{H}_x is a partition over the node variables $\{x_1, \dots, x_N\}$ and \mathcal{H}_t is a partition over the layer variables $\{t_1, \dots, t_L\}$. Then, for all initial conditions the following holds:

$$\forall t \in [0, T), \forall x_i, x_j \in H_x, \forall H_x \in \mathcal{H}_x . x_i(t) = x_j(t) \implies \forall k \in \mathbb{N} . x_i^{(k)} = x_j^{(k)}$$

and

$$\forall t \in [0, T), \forall t_l, t_q \in H_t, \forall H_t \in \mathcal{H}_t . t_l(t) = t_q(t) \implies \forall k \in \mathbb{N} . t_l^{(k)} = t_q^{(k)}$$

Moreover, let \mathcal{G} be the corresponding multiplex graph and we define $\mathcal{H}^* = (\mathcal{H}_x^*, \mathcal{H}_t^*)$ as the corresponding partition over the node and layer variables $\{x_1, \dots, x_N\}, \{t_1, \dots, t_L\}$ of \mathcal{G} . We define \mathcal{H}^* as follows:

$$\begin{aligned} \forall i, j \in \{1, \dots, N\}, H_{a,x} \in \mathcal{H}_x . x_i, x_j \in H_{a,x} &\implies H_{a,x}^* \in \mathcal{H}^* . x_i, x_j \in H_{a,x}^* \\ \forall i, j \in \{1, \dots, L\}, H_{a,t} \in \mathcal{H}_t . t_i, t_j \in H_{a,t} &\implies H_{a,t}^* \in \mathcal{H}^* . t_i, t_j \in H_{a,t}^* \end{aligned}$$

SUMMARY In this section, we reviewed the following notions and existing results. First, we introduced *Exact Role Assignment* for traditional single-layer networks, followed by *Multiplex MLNs*, and an extension of *Eigenvector Centrality* to MLNs. We then introduced Ordinary Differential Equations (ODEs), the problem of *exact* model reduction based on *Backward Differential Equivalence* (BDE), and the problem of *approximate* model reduction based on ε -BDE. Finally, we present the two major results that inspired the contributions shown in this manuscript: Theorem 2.10 and its corollary illustrate the relationship between BDE, eigenvector centrality and exact role assignment for single layer networks; In Theorem 2.15, we recall the result that binds BDE to the notion of eigenvector centrality for Multiplex MLNs.

3. Approximate Role Assignment for Multiplex MLNs

As we shown in the Section 2 eigenvector centrality can be related to exact role assignment and BDE in the case of single layer networks. To the best of our knowledge, there has been little to no effort to extend the notions of role assignments to Multiplex MLNs. This can be justified by the fact that the problems of extending centrality measures and the problems of clustering and community detection play a bigger role in the literature of network analysis. In this work we propose to use a reasoning similar to the one presented in Theorem 2.10 to extend the notion of Exact Role Assignment to Multiplex MLNs and we introduce an approximate version of it based on ε -BDE.

Similarly to the case of single-layered networks in which exact role assignment identifies as equivalent *which nodes share the same exact of neighbors in the same block* and mirrors the definition of eigenvector centrality one can enrich this notion with the extension of eigenvector centrality for multiplex MLNs.

DEFINITION 3.1 Let $\mathcal{A} \in \mathbb{R}_{\geq 0}^{N \times N \times L}$ be the adjacency tensor of an MLN with undirected layers, $i, j \in V_N$ and $l, q \in V_L$. Let $N(i, l)$ denote the neighbors of node i in layer l , that is $N(i, l) := \{j \in V_N \mid (i, j) \in \mathbf{A}^{(l)}\}$, $N : V_N \times V_L \rightarrow \mathbb{N}^{V_N}$. Let $M(i, \hat{l})$ denote the neighbors of node i in the set of layers \hat{l} , that is $M(i, \hat{l}) := \{j \in V_N \mid (i, j) \in \mathbf{A}^{(l)} \wedge r_2(l) = \hat{l}\}$, $M : V_N \times \hat{V}_L \rightarrow \mathbb{N}^{V_N}$. Let $W(\hat{i}, l)$ denote the neighbors of the set of nodes \hat{i} in layer l , that is $W(\hat{i}, l) := \{j \in V_N \mid (i, j) \in \mathbf{A}^{(l)} \wedge r_1(i) = \hat{i}\}$, $W : \hat{V}_N \times V_L \rightarrow \mathbb{N}^{V_N}$. Let $r_1 : V_N \rightarrow \hat{V}_N$ with $\hat{V}_N \subseteq V_N$ be a surjective mapping from a node to a representative of a set of nodes. Let $r_2 : V_L \rightarrow \hat{V}_L$ with $\hat{V}_L \subseteq V_L$ be a surjective mapping from a layer to a representative of a set of layers. Then, we define a surjective 2-map $r = (r_1, r_2)$, $r : V_N \times V_L \rightarrow \hat{V}_N \times \hat{V}_L$ with $\hat{V}_N \subseteq V_N$ and $\hat{V}_L \subseteq V_L$ as a *multiplex exact role assignment* if, for all $i, j \in V_N$ and $l, q \in V_L$ the following two properties hold:

$$\begin{aligned} r_1(i) = r_1(j) &\Rightarrow \forall l \in V_L \cdot r_1(M(i, r_2(l))) = r_1(M(j, r_2(l))) \\ r_2(l) = r_2(q) &\Rightarrow \forall i \in V_N \cdot r_1(W(r_1(i), l)) = r_1(W(r_1(i), q)) \end{aligned}$$

where the right hand side equations are equations of multi-sets, i.e., nodes (resp. layers) that have the same role, must have the same number of each of their other roles in their neighborhoods. **We show an example of multiplex exact role assignment as proposed in Definition 3.1 in Section B of the Appendix.**

THEOREM 3.2 Let $\mathcal{A} \in \mathbb{R}_{\geq 0}^{N \times N \times L}$ be the adjacency tensor of an MLN with undirected layers and a partition $\mathcal{H} = (\mathcal{H}_x, \mathcal{H}_l)$ over the set of variables (V_N, V_L) such that there is no overlap between the nodes and layers. Set $r_1(i) = \hat{i}_{H_x}$ when $i \in H_x$ for $H_x \in \mathcal{H}_x$ (i.e., \hat{i}_{H_x} is the representative of block H_x) and set $r_2(l) = \hat{l}_{H_l}$ when $l \in H_l$ for $H_l \in \mathcal{H}_l$ (i.e., \hat{l}_{H_l} is the representative of block H_l).

- r_1 is a *node exact role assignment* if and only if for all $H_x, H'_x \in \mathcal{H}_x$, for all $H'_l \in \mathcal{H}_l$ and $i, j \in H_x$ it holds that:

$$\sum_{l \in H'_l} \sum_{k \in H'_x} A_{ikl} = \sum_{l \in H'_l} \sum_{k \in H'_x} A_{jkl}$$

- r_2 is a *layer exact role assignment* if and only if for all $H_l \in \mathcal{H}_l$, for all $H_x, H'_x \in \mathcal{H}_x$ and $l, q \in H_l$ it holds that:

$$\sum_{i \in H_x} \sum_{j \in H'_x} A_{ijl} = \sum_{i \in H_x} \sum_{j \in H'_x} A_{ijq}$$

Proof. see, Appendix D □

COROLLARY 3.1 Let $\mathcal{A} \in \mathbb{R}_{\geq 0}^{N \times N \times L}$ be the adjacency tensor of an MLN with undirected layers and a partition $\mathcal{H} = (\mathcal{H}_x, \mathcal{H}_l)$ over the set of variables of the PIVP such that there is no overlap between the nodes and the layers, i.e. \mathcal{H}_x is a partition over the node variables $\{x_1, \dots, x_N\}$ and \mathcal{H}_l is a partition over the layer variables $\{t_1, \dots, t_L\}$.

- \mathcal{H} is a BDE of the PIVP translation from Definition 2.14 if and only if

1. for all $H_x, H'_x \in \mathcal{H}_x$ and $H'_l \in \mathcal{H}_l$ and $x_i, x_j \in H_x$ it holds that

$$\sum_{l \in H'_l} \sum_{k \in H'_x} A_{ikl} = \sum_{l \in H'_l} \sum_{k \in H'_x} A_{jkl}$$

2. and for all $H_l \in \mathcal{H}_l$ and $H_x, H'_x \in \mathcal{H}_x$ and $\tau_l, \tau_q \in H_l$ it holds that

$$\sum_{i \in H_x} \sum_{j \in H'_x} A_{ijl} = \sum_{i \in H_x} \sum_{j \in H'_x} A_{ijq}$$

- In the case \mathcal{A} is an adjacency tensor of a multiplex, \mathcal{H} is a BDE of the PIVP obtained with Definition 2.14 if and only if \mathcal{H}^* from Lemma 2.1 is an exact role assignment.

This allows to extend the notion of exact role assignment to multiplex networks. The main advantages are the following: it is obtainable in polynomial time, the partition preserves the centrality scores and it allows to define some notion of equivalence that does not only apply at the level of the nodes but it can simultaneously relate equivalent layers (i.e., layers that play the same role in the multiplex). The main disadvantage of this approach is that the constraints outlined in Definition 3.1 can not be satisfied easily by real-world and very asymmetric MLNs as these constraints are quite strict and require precisely the same amount of neighbors from the other class. Moreover, as the experimental evaluation shows in [43], because of this reason, no equivalent layers have been found on any of the benchmark MLNs. In order to relax these strict constraints we now introduce an *approximate* counterpart that allows a tolerance ε used to loosen those constraints.

DEFINITION 3.3 Let $\mathcal{A} \in \mathbb{R}_{\geq 0}^{N \times N \times L}$ be the adjacency tensor of an MLN with undirected layers and let \mathcal{H} be a ε -BDE of the PIVP obtained from Definition 2.14, then we call \mathcal{H}^* a *ε -approximate role assignment* for the MLN.

We illustrate an example of 4-approximate role assignment in Section C of the Appendix. This definition enables the ability to choose the perturbation parameter ε in such a way that it can capture different degrees of granularity of the partition. It is worth noting that the approximation ε acts at the level of the *coefficients* of the monomials of the PIVP. Throughout this work the coefficients of such monomials are the entries of the 3-rd order adjacency tensor, i.e. if there exists an edge between a pair of nodes in a given layer. Intuitively, two nodes are exact role equivalent if they have the same *exact* number of neighbors in any other block on any other layer. The parameter ε lifts this strict constraint by allowing a tolerance on this notion. For instance, with $\varepsilon = 1$ two nodes are 1-approximate role equivalent if they have the same *exact* number of neighbors in any other block *or* if the number of neighbors in any other block differs by a maximum of 1. On one hand, if we define ε -approximate role assignment in such a way we can avoid using ε -BDE. On the other hand, the theoretically sound definition of ε -BDE presented in Definition 2.12 coupled with the transitive closure of the equivalence relation guarantees the uniqueness of the coarsest partition. The main advantages are: it allows to discover more non-trivial partitions, ε can be used as an indicator of how much layers and nodes relate to each other, it allows to have an hierarchy of partitions and it is unique, differently to the ε -equitable partitions and approximate role assignments in the literature[25]. The main disadvantage is that, compared to its exact counterpart, the problem is NP-hard.

SUMMARY In this Section, we proposed the notion of *Approximate Role Assignment* for Multiplex MLNs. The workflow is as follows: the input is a Multiplex MLNs; Using Definition 2.14, we rewrite the input to its associated PIVP; At this point, the user can obtain the partition corresponding to *Exact Role Assignment* via Definition 3.1, using the polynomial procedure to obtain BDE on the PIVP. Alternatively, (s)he can obtain the partition corresponding to *Approximate Role Assignment* via Definition 3.3, by choosing a value of ε and using the NP-hard procedure to obtain ε -BDE on the PIVP.

4. Practical aspects and experimental evaluation

In this section we present the results of our experimental evaluation on a number of real-world case studies. The tools used for the experiments are MATLAB and ERODE [8], a state-of-the-art tool for automated model reduction for systems of ODEs and Chemical Reaction Networks. The input is the list of edges $E^{(l)}$ for all $l \in \{1, \dots, L\}$ representing a multiplex network $\mathcal{G} = \{G^{(l)} = (V_N, E^{(l)})\}_{l \in V_L}$. A MATLAB script translates the list of edges to a PIVP.⁴ ERODE then proceeds with the computation of BDE and ε -BDE and provides the partitions as its output. All experiments have been conducted on a MacBook Pro with a 2.6 GHz Intel Core i7 with 16 GB of RAM.

We ran our novel role detection technique on multiplex MLNs retrieved from the CoMuNe Lab repository (<https://comunelab.fbk.eu>). We present results on the following undirected instances: Padgett-Florentine-Families [40], CS-Aarhus [31], London Transport [14], EUAirTransportation [11], PierreAuger [18]. In order to test our algorithm on *directed* instances, we first had to transform these instances into undirected multiplex MLNs, since the centrality measure we considered in this work is not defined for directed networks. We present the results on the following modified directed instances: Krackhardt-High-Tech [28], Vickers-Chan-7thGraders [53], Kapferer-Tailor-Shop [24], Lazega-Law-Firm [29, 45], hepatitisC-genetic, danioRerio-genetic, humanHerpes4-genetic, celegans-connectome, bos-genetic, candida-genetic, xenopus-genetic, celegans-genetic [13, 48], CKM-Physicians-Innovation.

4.1 Experimental evaluation

In Table 1 we show the result for the (originally) undirected instances of multiplex MLNs from the repository. For each instance (rows) the Table displays the number of blocks of the coarsest partition

Table 1: Results of ε -partitioning on undirected MLN instances. Observe a gradual decrease of the number of blocks on all the instances, revealing a fine-grained hierarchy of partitions. While we see that the granularity of hierarchy depends on network topology, a general semantic interpretation of obtained partitions is beyond the scope of this manuscript. Yet, we show that a domain expert can do so, as shown in the example network presented in Section 5.

Undirected Instances		Number of ε -partition classes (distinct roles identified)																				
Instance	Nodes	$\varepsilon = 0$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Florence	16	16	14	9	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
CS-Aarhus	61	61	61	61	60	56	54	51	51	46	45	34	28	22	9	1	1	1	1	1	1	1
London Transport	368	364	84	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
EU Air	450	374	295	243	214	191	172	168	157	146	133	127	119	117	107	99	92	88	81	78	76	74
pierreauger	514	351	204	70	37	24	18	9	7	5	1	1	1	1	1	1	1	1	1	1	1	1
Instance	Layers	$\varepsilon = 0$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Florence	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1
CS-Aarhus	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	4
London Transport	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
EU Air	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37
pierreauger	16	16	16	16	16	16	16	14	12	12	7	5	5	5	5	5	5	5	5	5	5	5

found using the notions presented in Section 3, for different values of ε (columns). Concretely, we show the number of distinct roles found for the *exact* role assignment ($\varepsilon = 0$), and for *approximate* role assignment with values of ε spanning from 1 to 20. The results are presented for both nodes and layers.

⁴The code is available at <https://github.com/stefanotognazzi/ApproximateRoleAssignmentForMLNs>.

Similarly, we present results for the modified directed instances in Table 2. We can see in both Tables that our proposed notion can find non-trivial partitions for a variety of instances and, as we have hoped for, it accomplishes a gradual decrease of the number of blocks as ε increases.

Table 2: Results of ε -partitioning on the modified directed MLN instances. Observe a gradual decrease of the number of blocks on the majority of the instances, revealing a fine-grained hierarchy of partitions. While we see that the granularity of hierarchy depends on network topology, a general semantic interpretation of obtained partitions is beyond the scope of this manuscript.

Directed Instances			Number of ε -partition classes (distinct roles identified)																			
Instance	Nodes	$\varepsilon = 0$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Krackhardt-High-Tech	21	21	21	21	21	21	21	21	21	21	21	20	2	1	1	1	1	1	1	1	1	1
Vickers-Chan-7thGraders	29	29	29	29	29	29	28	28	28	28	27	26	25	25	25	23	3	1	1	1	1	1
Kapferer-Tailor-Shop	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39
Lazega-Law-Firm	71	71	71	71	71	71	71	71	71	71	71	71	71	71	71	71	71	71	71	71	71	71
hepatitusC-genetic	105	11	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
danioRerio-genetic	155	79	18	9	6	3	3	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1
humanHerpes4-genetic	216	46	29	13	8	6	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
CKM-Physicians-Innovation	246	242	239	232	223	214	138	11	4	3	3	3	2	1	1	1	1	1	1	1	1	1
celegans-connectome	279	279	279	279	278	277	277	276	276	274	274	273	272	262	261	257	252	242	235	222	216	202
bos-genetic	325	162	20	8	6	4	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
candida-genetic	367	62	25	13	7	4	4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3
xenopus-genetic	461	276	104	37	19	13	8	6	4	3	3	3	3	3	3	2	2	2	2	2	2	2
celegans-genetic	3879	2372	1287	709	446	288	179	82	49	42	37	33	28	26	22	20	19	19	18	17	16	14
Instance	Layers	$\varepsilon = 0$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Krackhardt-High-Tech	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Vickers-Chan-7thGraders	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Kapferer-Tailor-Shop	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Lazega-Law-Firm	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
hepatitusC-genetic	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
danioRerio-genetic	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4
humanHerpes4-genetic	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
CKM-Physicians-Innovation	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
celegans-connectome	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
bos-genetic	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3
candida-genetic	7	7	7	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
xenopus-genetic	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4
celegans-genetic	6	6	6	6	6	6	6	6	6	6	6	6	6	5	5	5	5	5	5	5	5	5

In Table 3 we show, for the undirected instances, the smallest value of ε which generates the trivial partition with only one block for nodes (second column) and layers (third column). The collapse point indicates what is the value of ε such that all the nodes (resp. layers) collapse into a single block. For instance, the first line of Table 3 that describes the collapse points for the Florence case study states that when $\varepsilon = 4$ all the nodes are considered 4-Approximate Role Assignment equivalent and therefore there is only one block. However, with $\varepsilon = 4$, Layer 1 and Layer 2 are not considered equivalent. Moreover, for all ε such that $0 \leq \varepsilon \leq 9$ Layer 1 and Layer 2 are not considered equivalent. Then, when $\varepsilon = 10$ (the value reported in the Table), Layer 1 and Layer 2 are considered 10-Approximate Role Assignment Equivalent and therefore all the Layers (in this case 2) collapse to a single block. These collapse point are interesting because, ideally, they can be used as a couple of scalars to describe a structural property of the multiplex, for instance, with a small ε as a collapse point the multiplex MLN is more easily collapsed into one node than with a larger value of ε .

From a performance standpoint, all the partitions were found in the order of 10 seconds for all but two instances, namely *pierreauger* and *celegans-genetic*, which took approximately one minute of runtime. In Table 4 we present an analysis of the scalability of the proposed approach. As previously mentioned, in the case of exact role assignment we can use a polynomial procedure and we refer to the results presented in [43], in which, the authors show that such reduction can be obtained in the order of seconds for instances that feature up to half a million nodes. In this work we shift our focus

Table 3: ε -collapse points

Instance	Nodes	Layers
Florence	4	10
CS-Aarhus	14	138
London Transport	3	582
EU Air	74	588
pierreauger	9	400

Table 4: Scalability analysis

Nodes	Edges	Time	Nodes	Edges	Time
100	304	< 1 s	10000	31843	35 s
250	752	< 1 s	12500	38421	63 s
500	1366	< 1 s	15000	49141	79 s
1000	2988	< 1 s	17500	55579	121 s
2500	7638	2 s	20000	64673	T.O.
5000	15081	9 s			

on the approximate notion: to analyse the limitations of the approach we synthesise MLNs composed by 2 layers. The first layer is built using a power-law degree distribution network generated with a configuration model with exponent 2.5. The second layer is a copy of the first layer with 20 percent of randomly added edges (non-overlapping with previous edges)⁵. The results show that, on a machine designed for personal use, the bound in terms of the number of nodes is 20000 nodes, which is coherent with the results presented for the real-world benchmarks.

A semantic interpretation of the found roles for each of the case studies we experimented with is beyond the scope of this paper. Nevertheless, in the next Section, we present how an expert can provide a semantic interpretation of the discovered roles based on the case study of Florence families.

5. Case study: Roles in Florence families

In this section, we highlight that our notion of approximate role assignment over multiplex MLN detects a useful, fine-grained hierarchy of roles, which is impossible to obtain by (combining) the existing role detection techniques (exact/approximate, single-layer/multi-layer). For this illustration, we use a network of Florence families - a classic benchmark dataset from the literature. In [40] the authors collected the data about the interactions of different families in Florence during the rise of the Medici family to power. Throughout this section we will use the mapping between nodes and families presented in Table 5. The dataset is comprised of two layers: the first layer (Fig.1a, 3a, 3b) represents the marriage

Table 5: Node Mapping

Node ID	Family Name	Node ID	Family Name
1	Acciaiuoli	9	Medici
2	Albizzi	10	Pazzi
3	Barbadori	11	Peruzzi
4	Bischeri	12	Pucci
5	Castellani	13	Ridolfi
6	Ginori	14	Salviati
7	Guadagni	15	Strozzi
8	Lamberteschi	16	Tornabuoni

relationships among the families, an edge between two nodes means that there has been at least one

⁵This approach used to synthesise MLNs was inspired by [23]

marriage between the members of those families. The second layer (Fig.1b, 4) represents the business relationships between the families. One of the most celebrated results on this dataset is that the Medici family enjoyed success and empowerment within this framework because of their strategic positioning in these social networks. This has been shown in multiple works from the literature via different means. For instance in [40] this claim has been made by showing that the Medici family has the highest Eigen-vector centrality score on the single-layer network of the marriages. Only few works tried to exploit the multiplex nature of this instance: in [41] the claim that the Medici family was the most important at that time in Florence is made by defining a game-theory inspired metric on a network that aggregates the information of the two layers. Similarly, in [35] **a similarity measure between nodes is developed using the information from the same 2-layer multiplex that we use in this Section. Then, the authors run a hierarchical clustering algorithm on what they refer as *multilinks* and use its output to detect communities. They claim the importance of the Medici family by showing that their proposed community detection algorithm places them as part of 3 communities while all the other families appear in at most 2 communities.** We first discuss what we obtain using both exact and approximate role assignment on the single-layer networks. Then, we shall show the results using the proposed notions on the Multiplex MLN. Let us first start by using the *exact role assignment* using only the marriage single-layer network: the result is shown in Fig. 1a and as we can see there is no reduction. Therefore, no meaningful information is provided by this notion on this network. Then, we use the same notion on the business single-layer network. The result is presented in Fig. 1b and we can see that, in this case, we discover that nodes 10, 14 and 16 are exact role equivalent.

We now apply the extension of exact role assignment to the multiplex network and the results obtained are depicted in Fig 2. As in the case of the marriage network we do not find any role equivalent nodes meaning that we can not infer any information on this instance.

We have seen that the notion of exact role assignment can not provide any meaningful insight on the social roles covered by the families in this dataset. We now proceed to address the following two questions. First, we saw that exact role assignment can not provide meaningful partitions for the single-layer networks, what is the result of applying the proposed approximate extension on the single-layer networks? Second, why we need the multiplex representation in combination with the approximate extension?

To address the first question we apply the proposed approximate role assignment extension on the marriage network using $\varepsilon = 1$ (Fig 3a) and $\varepsilon = 2$ (Fig 3b). As we can see from Fig. 3b all the nodes collapse in the same block, therefore resulting in a trivial partition. A different result is obtained when we use $\varepsilon = 1$. The resulting partition of 1-approximate role equivalent nodes is the following: $\mathcal{H} = \{\{1, 2, 3, 6, 8, 10, 13, 14, 16\}, \{4\}, \{5, 11, 15\}, \{7\}, \{9\}\}$. This partition suggests that nodes 5, 11 and 15 engage in the same social role in the network and we will discuss this thoroughly later in the section. If we apply the approximate role assignment on the business network using $\varepsilon = 1$ (Fig 4) we can see that we obtain the trivial partition. To sum up, if we use the proposed approximate extension on single-layer networks we find trivial partitions even for a small value of ε and therefore it does not prove as useful as expected. It is worth noting that this, in general is not true, for instance the combination of the marriage network with $\varepsilon = 1$ provides a non-trivial partition of the nodes. From an omitted experimental evaluation of this notion on single-layer networks we can see that the phenomenon of obtaining the trivial partition for small values of ε is persistent on many benchmark networks.

To address the second question we now take the multiplex representation of the instance and apply the proposed notion. If we use the exact role assignment (Fig 2) we find the trivial partition and we do not obtain any information. We then apply the approximate version with $\varepsilon = 1$ (Fig 5a) and we observe that we find the first non trivial block composed by nodes 10 and 14. We previously saw that with exact

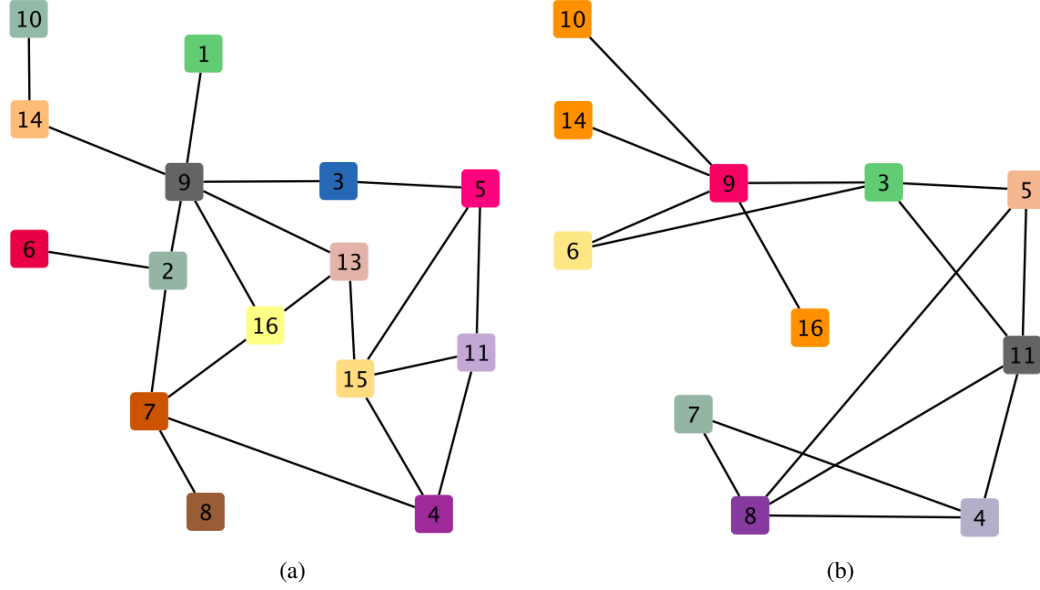


FIG. 1: (a) Results of exact role assignment on the single layer marriage network. Each node represents a family according to the mapping presented in Table 5 and each edge encodes the fact that there has been at least a marriage from between the members of the families linked. Each color represents a block of the exact role assignment partition. (b) Results of exact roles assignment on the single layer business network. Each node represents a family according to the mapping presented in Table 5 and each edge encodes the fact that there has been at least a business relationship between the families. Each color represents a block of the exact role assignment partition.

role assignment on the business network we found the block $\{10, 14, 16\}$, then we saw that with $\varepsilon = 1$ on the marriage network we found the block $\{1, 2, 3, 6, 8, 10, 13, 14, 16\}$. In other words, except for the trivial partition in which every node is a singleton, we were never able to discriminate between these nodes thus justifying the use of all the information available because in order to discriminate between these three nodes we need to fully exploit the expressiveness of the multiplex.

Then, from Fig. 5b we can observe that if we loosen the constraints we are able to capture another non-trivial partition. For instance with $\varepsilon = 2$ we can observe the following block $\{1, 2, 6, 10, 13, 14, 16\}$ and if we compare it with the block that we previously found on the marriage network only we can see that we have been able to single out node 8, meaning that, when taking in account both marriage and business relationships that node plays a different social role in the network. Moreover if we compare with the previous block that made nodes 5, 11 and 15 equivalent we can now see that we are able to discriminate among them because of the differences that emerge by considering both layers.

Finally, when we use $\varepsilon = 3$ (Fig 6a) we observe another interesting phenomenon: we are able to discriminate node 9, which represents the Medici family, with respect to the rest of the network thus adding our proposed methodology to the long line of techniques that manage to distinguish this fact. It is worth noting that this is not the case neither if we use exact role assignment nor approximate role assignment on any of the single layer networks.

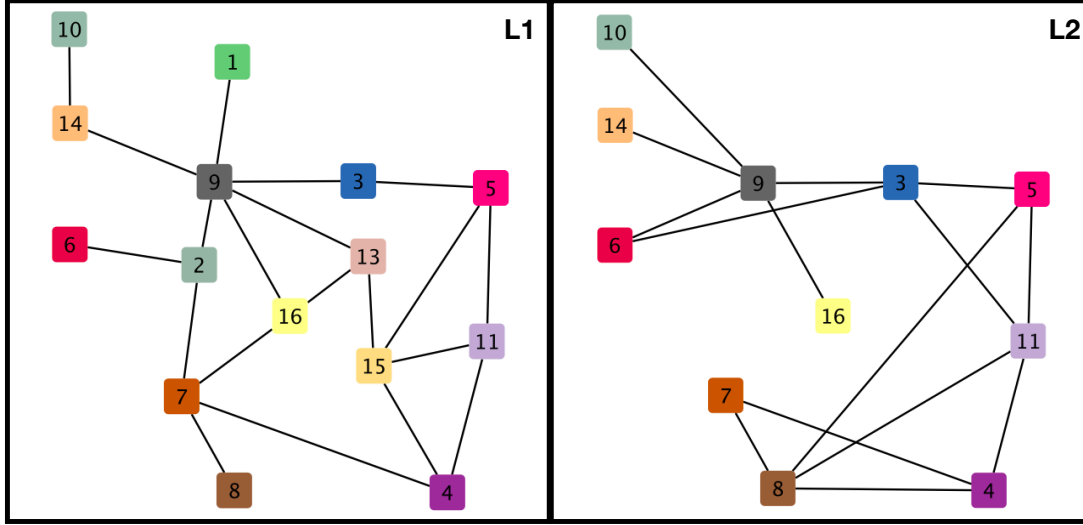


FIG. 2: Results of exact role assignment on the MLN. Layer 1 (L1) represents the marriage connections. Layer 2 (L2) represents the business connections. Each color represents a block of the exact role assignment partition.

The partitions depicted in Figs. 5a, 5b, 6a can be illustrated using a dendrogram as well (Fig 6b) which is a common way to represent the partitions obtained using hierarchical clustering. We would like to stress that role equivalence and hierarchical clustering can not be compared to one another (see Discussions 6.2), nor it can be compared to community detection (see Discussions 6.1). It is worth pointing out that in [35] using community detection nodes 3 and 7 participate in the same communities on the same number of layers. Nodes 3 and 7 can be discriminated using the proposed methodology from [35] only by going in-depth in the link based dendrogram. Using our technique we are able to discriminate between those nodes because, despite they share some traits in common, they play a different social role in the Florence environment. In this case, we can clearly see how node 3 shares a direct marriage connection and a direct business connection with the Medici family (node 9) while node 7 does not.

6. Discussions

6.1 Role Equivalence vs Community Detection

Although role equivalence and community detection problem share some common traits, they are fundamentally different. The goal of community detection is to split the set of nodes in a certain number of communities based on their similarity, local vicinity or clustering. On the other hand, role equivalence aims at identifying which nodes cover the same social role. For example, let us imagine that there are two universities: University A and University B, and let us assume that employees of each university can be either Professor, Postdoc or PhD student. Then, assume that all the relationships among them are modeled with either a network or with a Multiplex MLN. The goal of a community detection algorithm in this hypothetical example would be to correctly split who is an employee of University A from the

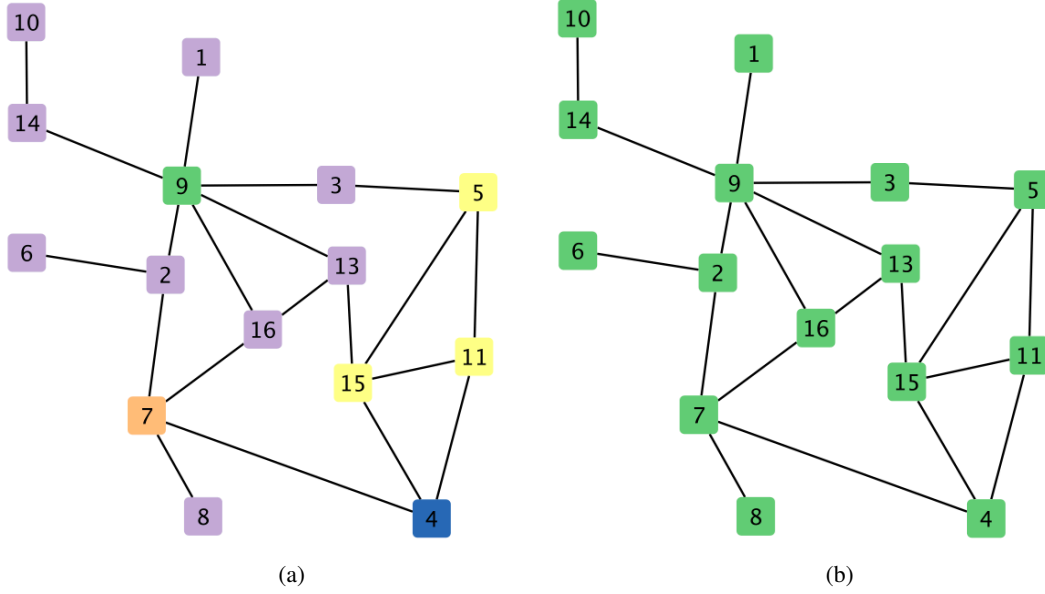


FIG. 3: (a) Results of 1-approximate role assignment on the single layer marriage network. Each node represents a family according to the mapping presented in Table 5 and each edge encodes the fact that there has been at least a marriage from between the members of the families linked. Each color represents a block of the 1-approximate role assignment partition. (b) Results of 2-approximate role assignment on the single layer marriage network. Each node represents a family according to the mapping presented in Table 5 and each edge encodes the fact that there has been at least a marriage from between the members of the families linked. Each color represents a block of the 2-approximate role assignment partition.

employees of University B. On the other hand, Role Equivalence aims at partitioning the nodes in three sets: Professors, Postdocs and PhD students regardless of which university employs them. It comes at no surprise at this point that the two problems give rise to different results and, in general, they can not be compared to one another. Moreover, in the examples and in the proposed methodology for role equivalence we always use a partition and each node can only appear in one block of the partition. Community detection allows both the usage of partitions or *set coverage*. In the latter, a node can appear in more than one community. Another remark is that in [35], the communities discovered depend on two parameters ϵ and z . The choice of those parameters, in general, depends on the network under consideration. This highlights another perk of the approach presented in this paper, as the notions proposed in Section 3 do not depend nor require a parameter search for the underlying instance.

6.2 Approximate Role Equivalence vs Hierarchical Clustering

Another problem that can be related to Role Equivalence is Hierarchical Clustering. In this work, we used *dendrograms* to illustrate the hierarchy among partitions discovered using different values of ϵ . Dendrograms are frequently used to present results of Hierarchical Clustering algorithms (see [35]) but,

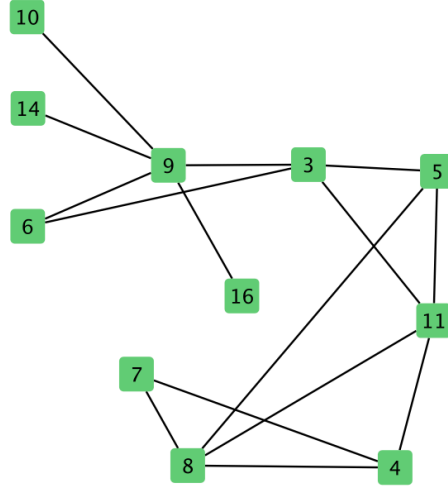


FIG. 4: Results of 1-approximate role assignment on the single layer business network. Each node represents a family according to the mapping presented in Table 5 and each edge encodes the fact that there has been at least a business relationship between the families. Each color represents a block of the exact role assignment partition.

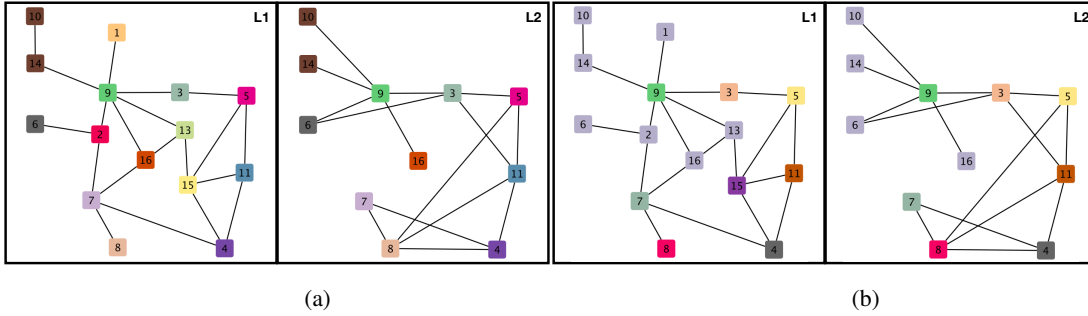


FIG. 5: (a) Results of 1-approximate role assignment on the MLN. Layer 1 (L1) represents the marriage connections. Layer 2 (L2) represents the business connections. Each color represents a block of the 1-approximate role assignment partition. (b) Results of 2-approximate role assignment on the MLN. Layer 1 (L1) represents the marriage connections. Layer 2 (L2) represents the business connections. Each color represents a block of the 2-approximate role assignment partition.

as we saw from the case-study about Florence, the results are in general incomparable and substantially different. In [35] the authors define a similarity measure between the edges and then run their hierarchical clustering approach based on this measure. In general, when using hierarchical clustering approaches, one would have to define a measure of similarity between nodes and this is the main difference with our approach. With the proposed approach presented in Section 3 we do not require the modeler to define such a measure and our approach relies entirely on the topological structure of the

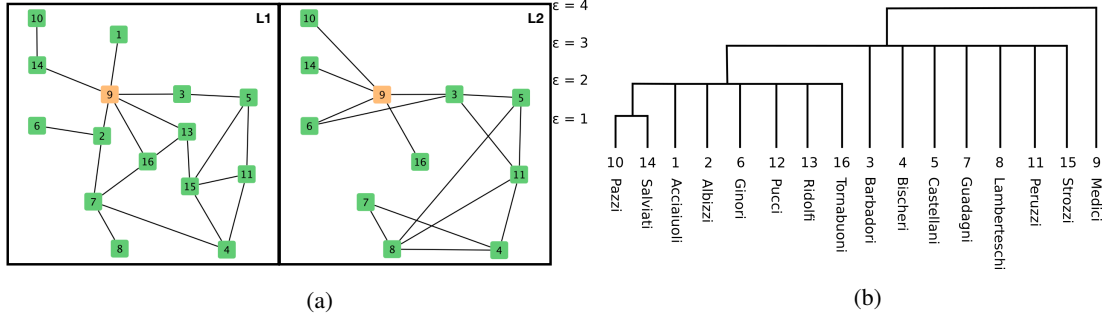


FIG. 6: (a) Results of 3-approximate role assignment on the MLN. Layer 1 (L1) represents the marriage connections. Layer 2 (L2) represents the business connections. Each color represents a block of the 3-approximate role assignment partition. (b) A dendrogram representation of the hierarchy of ϵ -approximate role assignment partitions discovered by the methodologies presented in Section 3.

MLN. Such similarity measures often rely on parameters that need tuning of their own as presented in [35] binding the results to the choice of parameters. Although it is a compelling idea to define a similarity measure such that an hierarchical clustering algorithm outputs the same dendrogram as the one presented in Figure 6b we could not discover any and finding such a measure is one of the possible future directions.

6.3 Static Networks vs Dynamic Networks

Some of the techniques mentioned in the related works of this paper can be applied on dynamic networks [36]. The work proposed in this paper is currently limited to static multiplex MLNs meaning that one can do the computation of the partitions once the multiplex is fixed. If, for instance, one or multiple edges are added or removed, one would have to recompute the partitions.

7. Conclusions and future works

In this paper, we proposed a novel notion of exact role assignment for multiplex MLNs, inspired by BDE, an exact model reduction technique for dynamical systems. Moreover, we presented an approximate role assignment, inspired by the approximate counterpart of BDE. The relevance of the result was demonstrated by computing a variety of non-trivial role assignment partitions on real-world multiplex benchmarks. Throughout this work, we developed the notions of role assignment for undirected, unweighted networks. An immediate future direction includes considering weighted networks and the effect of approximate ϵ -perturbations on weighted instances. Moreover, we focused on static networks, and one of the possible directions is to investigate the possible extensions in an *online* fashion, so that our framework can deal with dynamic time-evolving multiplex MLNs.

The framework we presented in this paper can be tailored to arbitrary extensions of centrality measures for multiplex MLNs. As such, it represents a versatile, cornerstone contribution to automated, flexible detection of role equivalence in complex networks.

Funding

This work was supported by the Ministry of Science, Research and the Arts of the state of Baden-Württemberg, and the DFG Centre of Excellence 2117 ‘Centre for the Advanced Study of Collective Behaviour’ (ID: 422037984).

Acknowledgment

The authors would like to thank Giacomo Rapisardi for the inspiring discussions on the topic and Andrea Vandin for the support and the insights on the use of the tool ERODE.

REFERENCES

1. Barrett, L., Henzi, P. & Lusseau, D. (2012) Taking Sociality Seriously: The Structure of Multi-Dimensional Social Networks as a Source of Information for Individuals. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, **367**, 2108–18.
2. Battiston, F., Nicosia, V. & Latora, V. (2014) Structural measures for multiplex networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, **89**, 032804.
3. Bentley, B., Branicky, R., Barnes, C. L., Chew, Y. L., Yemini, E., Bullmore, E. T., Vértés, P. E. & Schafer, W. R. (2016) The Multilayer Connectome of *Caenorhabditis elegans*. *PLOS Computational Biology*, **12**(12), 1–31.
4. Bonacich, P. (1972) Factoring and weighting approaches to status scores and clique identification. .
5. Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z. & Wagner, D. (2008) On Modularity Clustering. *Knowledge and Data Engineering, IEEE Transactions on*, **20**, 172 – 188.
6. Brandes, U. & Lerner, J. (2010) Structural Similarity: Spectral Methods for Relaxed Blockmodeling. *Publ. in : Journal of Classification* 27 (2010), 3, pp. 279-306, **27**.
7. Cardelli, L., Tribastone, M., Tschaikowski, M. & Vandin, A. Efficient Syntax-driven Lumping of Differential Equations. In *TACAS’16*.
8. Cardelli, L., Tribastone, M., Tschaikowski, M. & Vandin, A. (2017a) ERODE: A Tool for the Evaluation and Reduction of Ordinary Differential Equations. In *TACAS*, pages 310–328.
9. Cardelli, L., Tribastone, M., Tschaikowski, M. & Vandin, A. (2017b) Maximal aggregation of polynomial dynamical systems. *PNAS*, **114**(38).
10. Cardelli, L., Tribastone, M., Tschaikowski, M. & Vandin, A. (2018) Guaranteed Error Bounds on Approximate Model Abstractions through Reachability Analysis. *CoRR*, **abs/1807.06888**.
11. Cardillo, A., Gómez-Gardeñes, J., Zanin, M., Romance, M., Papo, D., Del Pozo Guerrero, F. & Boccaletti, S. (2013) Emergence of network features from multiplexity. *Scientific reports*, **3**, 1344.
12. De Domenico, M., Nicosia, V., Arenas, A. & Latora, V. (2014a) Layer aggregation and reducibility of multilayer interconnected networks. .
13. De Domenico, M., Nicosia, V., Arenas, A. & Latora, V. (2015a) Structural reducibility of multilayer networks. *Nature Communications*, **6**, 6864.
14. De Domenico, M., Solé-Ribalta, A., Gómez, S. & Arenas, A. (2014b) Navigability of interconnected networks under random failures. *Proceedings of the National Academy of Sciences*.
15. De Domenico, M., Solé-Ribalta, A., Cozzo, E., Kivelä, M., Moreno, Y., Porter, M., Gomez, S. & Arenas, A. (2013a) Mathematical Formulation of Multi-Layer Networks. *Physical Review X*, **3**.
16. De Domenico, M., Solé-Ribalta, A., Omodei, E., Gomez, S. & Arenas, A. (2013b) Centrality in Interconnected Multilayer Networks. *Physica D: Nonlinear Phenomena*, **323**.
17. De Domenico, M., Solé-Ribalta, A., Omodei, E., Gomez, S. & Arenas, A. (2015b) Ranking in interconnected multilayer networks reveals versatile nodes. *Nature communications*, **6**, 6868.
18. Domenico, M. D., Lancichinetti, A., Arenas, A. & Rosvall, M. (2015) Identifying modular flows on multilayer networks reveals highly overlapping organization in social systems. *ArXiv*, **abs/1408.2925**.

19. Farine, D., Aplin, L., Sheldon, B. & Hoppitt, W. (2015) Interspecific social networks promote information transmission in wild songbirds. *Proceedings. Biological sciences / The Royal Society*, **282**.
20. Fortunato, S. (2009) Community Detection in Graphs. *Physics Reports*, **486**.
21. Franz, M., Altmann, J. & Alberts, S. (2015) Knockouts of high-ranking males have limited impact on baboon social networks. *Current Zoology*, **61**, 107–113.
22. Gazda, S., Iyer, S., Killingback, T., Connor, R. & Brault, S. (2015) The importance of delineating networks by activity type in bottlenose dolphins (*Tursiops truncatus*) in Cedar Key, Florida. *Royal Society Open Science*, **2**, 140263–140263.
23. Granell, C., Gomez, S. & Arenas, A. (2013) Dynamical Interplay between Awareness and Epidemic Spreading in Multiplex Networks. *Physical review letters*, **111**, 128701.
24. Kapferer, B. (1972) Strategy and Transaction in an African Factory: African Workers and Indian Management in a Zambian Town. *Manchester University Press*, **43**(4), 362–363.
25. Kate, K. & Ravindran, B. (2009) Epsilon Equitable Partition: A positional analysis method for large social networks. In *COMAD*.
26. Katz, L. (1953) A new status index derived from sociometric analysis. *Psychometrika*, **18**(1), 39–43.
27. Kivelä, M., Arenas, A., Barthélemy, M., Gleeson, J. P., Moreno, Y. & Porter, M. A. (2014) Multilayer networks. *Journal of Complex Networks*, **2**(3), 203–271.
28. Krackhardt, D. (1987) Cognitive social structures. *Social Networks*, **9**(2).
29. Lazega, E. (2001) The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership. *Oxford University Press*.
30. Lerner, J. (2004) Role Assignments. In *Network Analysis: Methodological Foundations [outcome of a Dagstuhl seminar, 13-16 April 2004]*, pages 216–252.
31. Magnani, M., Micenková, B. & Rossi, L. (2013) Combinatorial Analysis of Multiple Networks. *CoRR*, **abs/1303.4986**.
32. Marx, M. & Masuch, M. (2003) Regular equivalence and dynamic logic. *Social Networks*, **25**(1), 51 – 65.
33. McKay, B. (1981) Practical Graph Isomorphism. *Congressus Numerantium*, **30**, 45–87.
34. McKay, B. & Piperno, A. (2013) Practical Graph Isomorphism II. *Journal of Symbolic Computation*.
35. Mondragon, R. J., Iacovacci, J. & Bianconi, G. (2018) Multilink communities of multiplex networks. *PLOS ONE*, **13**(3), 1–15.
36. Mucha, P., Richardson, T., Macon, K., Porter, M. & Onnela, J.-P. (2010) Community Structure in Time-Dependent, Multiscale, and Multiplex Networks. *Science (New York, N.Y.)*, **328**, 876–8.
37. Müller, J. & Brandes, U. (2019) The evolution of roles. pages 406–413.
38. Newman, M. (2003) The Structure and Function of Complex Networks. *Computer Physics Communications*, **147**, 40–45.
39. Nicosia, V. & Latora, V. (2015) Measuring and modeling correlations in multiplex networks. *Physical Review E*, **92**, 032805.
40. Padgett, J. F. & Ansell, C. K. (1993) Robust Action and the Rise of the Medici, 1400-1434. *American Journal of Sociology*, **98**(6), 1259–1319.
41. Pagan, N. & Dörfler, F. (2019) Game theoretical inference of human behavior in social networks. *Nature Communications*, **10**.
42. Page, L., Brin, S., Motwani, R. & Winograd, T. (1999) The PageRank Citation Ranking: Bringing Order to the Web. In *WWW 1999*.
43. Petrov, T. & Tognazzi, S. (2020) Centrality-Preserving Exact Reductions of Multi-Layer Networks. In *Leveraging Applications of Formal Methods, Verification and Validation: Engineering Principles*, pages 397–415, Cham. Springer International Publishing.
44. Rahmede, C., Iacovacci, J., Arenas, A. & Bianconi, G. (2017) Centralities of nodes and influences of layers in large multiplex networks. *Journal of Complex Networks*, **6**(5), 733–752.
45. Snijders, T. A. B., Pattison, P. E., Robins, G. L. & Handcock, M. S. (2006) NEW SPECIFICATIONS FOR EXPONENTIAL RANDOM GRAPH MODELS. *Sociological Methodology*, **36**(1), 99–153.
46. Solá Conde, L., Romance, M., Herrero, R., Flores, J., García del Amo, A. & Boccaletti, S. (2013) Eigenvector

- centrality of nodes in multiplex networks. *Chaos (Woodbury, N.Y.)*, **23**, 033131.
47. Staples-Moore, A. (2008) Equitable Partitions in Graph Theory. .
 48. Stark, C., Breitzkreutz, B.-J., Reguly, T., Boucher, L., Breitzkreutz, A. & Tyers, M. (2006) BioGRID: A general repository for interaction datasets. *Nucleic acids research*, **34**, D535–9.
 49. Stella, M., Andreazzi, C. S., Selakovic, S., Goudarzi, A. & Antonioni, A. (2016) Parasite spreading in spatial ecological multiplex networks. *Journal of Complex Networks*, **5**(3), 486–511.
 50. Tognazzi, S., Tribastone, M., Tschaikowski, M. & Vandin, A. (2018a) Backward Invariance for Linear Differential Algebraic Equations. pages 3771–3776.
 51. Tognazzi, S., Tribastone, M., Tschaikowski, M. & Vandin, A. (2018b) Differential Equivalence Yields Network Centrality. In *Leveraging Applications of Formal Methods, Verification and Validation. Distributed Systems - 8th International Symposium, ISoLA 2018, Limassol, Cyprus, November 5-9, 2018, Proceedings, Part III*, volume 11246 of *Lecture Notes in Computer Science*, pages 186–201. Springer.
 52. Tudisco, F., Arrigo, F. & Gautier, A. (2018) Node and Layer Eigenvector Centralities for Multiplex Networks. *SIAM Journal of Applied Mathematics*, **78**(2), 853–876.
 53. Vickers, M. & Chan, M. (1981) Representing classroom social structure. Victoria Institute of Secondary Education, Melbourne. .
 54. Wasserman, S. & Faust, K. (1994) *Social network analysis: Methods and applications*, volume 8. Cambridge university press.

Appendix

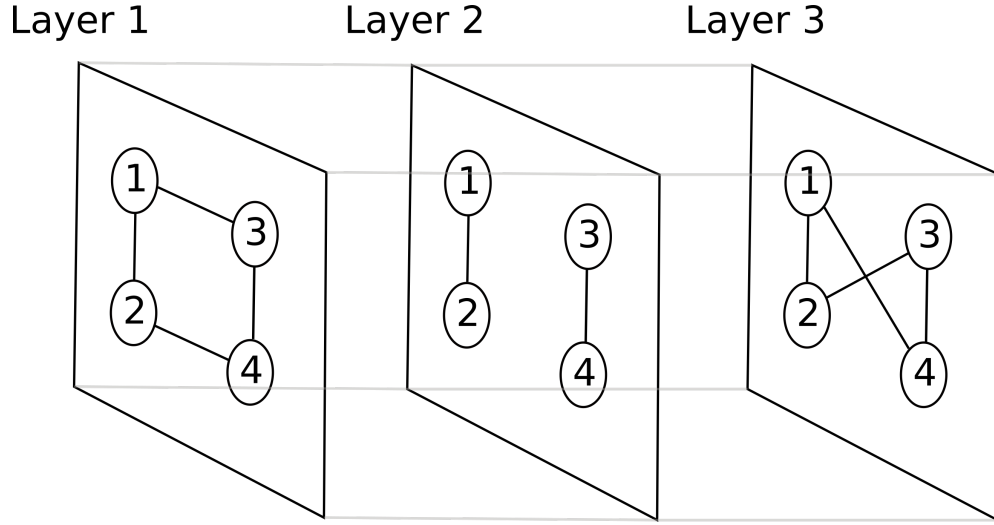


FIG. A.7: Multiplex MLN with 4 nodes and 3 layers.

A. Translation from MLN to PIVP

Consider the Multiplex in Figure A.7. The corresponding PIVP obtained via Definition 2.14 is the following:

$$\begin{aligned}
 \dot{x}_1 &= x_2 t_1 + x_3 t_1 + x_2 t_2 + x_2 t_3 + x_4 t_3 & \dot{t}_1 &= 2x_1 x_2 + 2x_1 x_3 + 2x_2 x_4 + 2x_3 x_4 \\
 \dot{x}_2 &= x_1 t_1 + x_4 t_1 + x_1 t_2 + x_1 t_3 + x_4 t_3 & \dot{t}_2 &= 2x_1 x_2 + 2x_3 x_4 \\
 \dot{x}_3 &= x_1 t_1 + x_4 t_1 + x_4 t_2 + x_2 t_3 + x_4 t_3 & \dot{t}_3 &= 2x_1 x_2 + 2x_1 x_4 + 2x_2 x_3 + 2x_3 x_4 \\
 \dot{x}_4 &= x_2 t_1 + x_3 t_1 + x_3 t_2 + x_1 t_3 + x_4 t_3
 \end{aligned}$$

B. Example of Multiplex Exact Role Assignment

Consider the Multiplex in Figure A.7 with $V_N = \{1, 2, 3, 4\}$ and $V_L = \{1, 2, 3\}$. For the sake of readability we will use $\hat{V}_N = \{a, b\}$ and $\hat{V}_L = \{s, p\}$ instead of $\hat{V}_N \subseteq V_N$ and $\hat{V}_L \subseteq V_L$

Let us use the following 2-map:

- $r_1(1) = r_1(2) = a, r_1(3) = r_1(4) = b$
- $r_2(1) = r_2(3) = s, r_2(2) = p$

The first step is to extract the set of neighbors for each node and layer:

- $N(1,1) = \{2,3\}$, $N(2,1) = \{1,4\}$, $N(3,1) = \{1,4\}$ and $N(4,1) = \{2,3\}$.
- $N(1,2) = \{2\}$, $N(2,2) = \{1\}$, $N(3,2) = \{4\}$ and $N(4,2) = \{3\}$.
- $N(1,3) = \{2,4\}$, $N(2,3) = \{1,3\}$, $N(3,3) = \{2,4\}$ and $N(4,3) = \{1,4\}$.

Then, we apply M to each combination of nodes and representative of layers and obtain the multi-sets of neighbors:

- $M(1,s) = \{2,3,2,4\}$, $M(2,s) = \{1,4,1,3\}$, $M(3,s) = \{1,4,2,4\}$ and $M(4,s) = \{2,3,1,3\}$.
- $M(1,p) = \{2\}$, $M(2,p) = \{1\}$, $M(3,p) = \{4\}$ and $M(4,p) = \{3\}$.

Then, we apply r_1 to each of the multi-sets the we obtained in the previous step in order to identify the multi-sets of the representatives of the nodes:

- $r_1(M(1,s)) = r_1(M(2,s)) = r_1(M(3,s)) = r_1(M(4,s)) = \{a,b,a,b\}$.
- $r_1(M(1,p)) = r_1(M(2,p)) = \{a\}$ and $r_1(M(3,p)) = r_1(M(4,p)) = \{b\}$.

We will now shift our attention to the layers and the first thing we will do is apply W to each combination of layer and representative of the nodes:

- $W(a,1) = W(b,1) = W(a,3) = W(b,3) = \{1,2,3,4\}$.
- $W(a,2) = \{1,2\}$ and $W(b,2) = \{3,4\}$.

Then, similarly to what we did at the nodes level, we apply r_1 to each of the sets that we obtained in the previous step:

- $r_1(W(a,1)) = r_1(W(b,1)) = r_1(W(a,3)) = r_1(W(b,3)) = \{a,a,b,b\}$.
- $r_1(W(a,2)) = \{a,a\}$ and $r_1(W(b,2)) = \{b,b\}$.

We now have all the pieces in place to check the property we are interested in. Let us start with the nodes:

- $r_1(1) = r_1(2) = a$: we need to check that for all $l \in V_l$ the following $r_1(M(1, r_2(l))) = r_1(M(2, r_2(l)))$ holds:
 - $r_1(M(1, r_2(1))) = r_1(M(1, r_2(3))) = r_1(M(1, s)) = \{a,b,a,b\}$
 $r_1(M(2, r_2(1))) = r_1(M(2, r_2(3))) = r_1(M(2, s)) = \{a,b,a,b\}$
 - $r_1(M(1, r_2(2))) = r_1(M(1, p)) = \{a\}$
 $r_1(M(2, r_2(2))) = r_1(M(2, p)) = \{a\}$
- $r_1(3) = r_1(4) = b$:
 - $r_1(M(3, r_2(1))) = r_1(M(3, r_2(3))) = r_1(M(3, s)) = \{a,b,a,b\}$
 $r_1(M(4, r_2(1))) = r_1(M(4, r_2(3))) = r_1(M(4, s)) = \{a,b,a,b\}$
 - $r_1(M(3, r_2(2))) = r_1(M(3, p)) = \{b\}$
 $r_1(M(4, r_2(2))) = r_1(M(4, p)) = \{b\}$

Similarly, we check the property on the layers:

- $r_2(1) = r_2(3) = s$: we need to check that for all $i \in V_n$ the following $r_1(W(r_1(i), 1)) = r_1(W(r_1(i), 3))$ holds:

$$\begin{aligned}
& - r_1(W(r_1(1), 1)) = r_1(W(r_1(2), 1)) = r_1(W(a, 1)) = \{a, a, b, b\} \\
& \quad r_1(W(r_1(1), 3)) = r_1(W(r_1(2), 3)) = r_1(W(a, 3)) = \{a, a, b, b\} \\
& - r_1(W(r_1(3), 1)) = r_1(W(r_1(4), 1)) = r_1(W(b, 1)) = \{a, a, b, b\} \\
& \quad r_1(W(r_1(3), 3)) = r_1(W(r_1(4), 3)) = r_1(W(b, 3)) = \{a, a, b, b\}
\end{aligned}$$

Therefore, we can conclude that r is a MLN exact role assignment.

C. BDE and Approximate-BDE

Consider the PIVP presented in Section A. We show that $\mathcal{H} = \{\{x_1, x_2, x_3, x_4\}, \{t_1, t_2, t_3\}\}$ is not a BDE. We first re-write q according to the partition \mathcal{H} by substituting the occurrences of variables with the representative of their block:

$$\begin{aligned}
q_{x_1} &= x_1 t_1 + x_1 t_1 + x_1 t_1 + x_1 t_1 + x_1 t_1 = 5x_1 t_1 & q_{t_1} &= 2x_1 x_1 + 2x_1 x_1 + 2x_1 x_1 + 2x_1 x_1 = 8x_1 x_1 \\
q_{x_2} &= x_1 t_1 + x_1 t_1 + x_1 t_1 + x_1 t_1 + x_1 t_1 = 5x_1 t_1 & q_{t_2} &= 2x_1 x_1 + 2x_1 x_1 = 4x_1 x_1 \\
q_{x_3} &= x_1 t_1 + x_1 t_1 + x_1 t_1 + x_1 t_1 + x_1 t_1 = 5x_1 t_1 & q_{t_3} &= 2x_1 x_1 + 2x_1 x_1 + 2x_1 x_1 + 2x_1 x_1 = 8x_1 x_1 \\
q_{x_4} &= x_1 t_1 + x_1 t_1 + x_1 t_1 + x_1 t_1 + x_1 t_1 = 5x_1 t_1
\end{aligned}$$

We first study $x_1 \sim_{\mathcal{H}} x_2$, from Definition we need to check the difference between the polynomials:

$$\mathcal{P}_{x_1, x_2}^{\mathcal{H}} = q_{x_1} - q_{x_2} = 5x_1 t_1 - 5x_1 t_1 = 0x_1 t_1$$

We conclude that $c(\mathcal{P}_{x_1, x_2}^{\mathcal{H}}, x_1 t_1) = 0$. An analogous reasoning applies to prove $x_1 \sim_{\mathcal{H}} x_3$, $x_1 \sim_{\mathcal{H}} x_4$, $x_2 \sim_{\mathcal{H}} x_3$, $x_2 \sim_{\mathcal{H}} x_4$ and $x_3 \sim_{\mathcal{H}} x_4$. We now proceed to analyse $t_1 \sim_{\mathcal{H}} t_2$:

$$\mathcal{P}_{t_1, t_2}^{\mathcal{H}} = q_{t_1} - q_{t_2} = 8x_1 x_1 - 4x_1 x_1 = 4x_1 x_1$$

We show that $c(\mathcal{P}_{t_1, t_2}^{\mathcal{H}}, x_1 x_1) = 4 \neq 0$ and we can therefore conclude that \mathcal{H} is not a BDE. However, if we consider the approximate variant ε -BDE and we fix $\varepsilon = 4$ we can conclude that \mathcal{H} is a 4-BDE.

D. Proof of Theorem 3.2

In order to prove Theorem 3.2 we need to show that under the assumption that $i, j \in V_N$ belong to the same block $H_x \in \mathcal{H}_x$ the following holds:

$$\forall l \in V_L . r_1(M(i, r_2(l))) = r_1(M(j, r_2(l))) \Leftrightarrow \forall H'_x \in \mathcal{H}_x, \forall H'_l \in \mathcal{H}_l . \sum_{l \in H'_l} \sum_{k \in H'_x} A_{ikl} = \sum_{l \in H'_l} \sum_{k \in H'_x} A_{jkl}$$

First, we focus on $M(i, r_2(l))$ and let us denote with $\hat{l} \in \hat{V}_L$ the representative of the block $H_l \in \mathcal{H}_l$ s.t. $r_2(l) = \hat{l}$. Then, by definition of M :

$$M(i, \hat{l}) = \{k \in V_N \mid (i, k) \in \mathbf{A}^{(l)} \wedge r_2(l) = \hat{l}\}$$

let us denote with $m_{M(i,\hat{l})}(k)$ the multiplicity of element k in the multiset $M(i,\hat{l})$ and notice that $m_{M(i,\hat{l})}(k) = \sum_{l \in H_t} A_{ikl}$. Similarly, for $M(j, r_2(l))$ by definition of M we observe that:

$$M(j, \hat{l}) = \{k \in V_N \mid (j, k) \in \mathbf{A}^{(l)} \wedge r_2(l) = \hat{l}\}$$

let us denote with $m_{M(j,\hat{l})}(k)$ the multiplicity of element k in the multiset $M(j,\hat{l})$ and notice that $m_{M(j,\hat{l})}(k) = \sum_{l \in H_t} A_{jkl}$.

We now shift our focus on $r_1(M(i, r_2(l)))$ where $r_2(l) = \hat{l}$, by definition of r_1 and M we write:

$$r_1(M(i, \hat{l})) = \{\hat{k} \in \hat{V}_N \mid (i, k) \in \mathbf{A}^{(l)} \wedge r_2(l) = \hat{l} \wedge r_1(k) = \hat{k}\}$$

Let us denote with $m_{r_1(M(i,\hat{l}))}(\hat{k})$ the multiplicity of element \hat{k} in the multiset $r_1(M(i,\hat{l}))$ and notice that $m_{r_1(M(i,\hat{l}))}(\hat{k}) = \sum_{k \in H_x} \sum_{l \in H_t} A_{ikl}$. Similarly, for $r_1(M(j, r_2(l)))$ where $r_2(l) = \hat{l}$ we observe that:

$$r_1(M(j, \hat{l})) = \{\hat{k} \in \hat{V}_N \mid (j, k) \in \mathbf{A}^{(l)} \wedge r_2(l) = \hat{l} \wedge r_1(k) = \hat{k}\}$$

if we denote with $m_{r_1(M(j,\hat{l}))}(\hat{k})$ the multiplicity of element \hat{k} in the multiset $r_1(M(j,\hat{l}))$ and notice that $m_{r_1(M(j,\hat{l}))}(\hat{k}) = \sum_{k \in H_x} \sum_{l \in H_t} A_{jkl}$.

Let us start with direction (\Rightarrow): we know that in order for equation $\forall l \in V_l \cdot r_1(M(i, r_2(l))) = r_1(M(j, r_2(l)))$ to hold all the elements of the two multisets need to have the same multiplicity, i.e. the following needs to hold:

$$\forall \hat{k} \in r_1(M(i, r_2(l))) \cdot m_{r_1(M(i, r_2(l)))}(\hat{k}) = m_{r_1(M(j, r_2(l)))}(\hat{k})$$

We know that, fixed a block H_x s.t. $\hat{k} \in H_x$ (i.e. \hat{k} is the representative of block H_x) the following holds:

$$m_{r_1(M(i, r_2(l)))}(\hat{k}) = \sum_{k \in H_x} \sum_{l \in H_t} A_{ikl} = \sum_{k \in H_x} \sum_{l \in H_t} A_{jkl} = m_{r_1(M(j, r_2(l)))}(\hat{k})$$

When we consider that we know that this holds for all $\hat{k} \in r_1(M(i, r_2(l)))$ (i.e., it holds for all the representatives of all the blocks) and that for all $l \in V_L$ it trivially holds that $r_2(l) = r_2(l)$ we can conclude that the following must hold:

$$\forall H'_x \in \mathcal{H}_x, \forall H'_t \in \mathcal{H}_t \cdot \sum_{l \in H'_t} \sum_{k \in H'_x} A_{ikl} = \sum_{l \in H'_t} \sum_{k \in H'_x} A_{jkl}$$

A similar reasoning applies to prove direction (\Leftarrow) and to prove the same for the second part of the 2-map, namely r_2 , that considers a similar claim on the layers' variables rather than the nodes' variables.