Centrality-preserving exact reductions of Multi-Layer Networks

Tatjana Petrov and Stefano Tognazzi

University of Konstanz, Germany,
Centre for the Advanced Study of Collective Behaviour, University of Konstanz, Germany

Abstract. Multi-Layer Networks (MLN) generalise the traditional, single-layered networks, by allowing to simultaneously express multiple aspects of relationships in collective systems, while keeping the description intuitive and compact. As such, they are increasingly gaining popularity for modelling Collective Adaptive Systems (CAS), e.g. engineered cyber-physical systems or animal collectives. One of the most important notions in network analysis are centrality measures, which inform us about the relative importance of nodes. Computing centrality measures is often challenging for large and dense single-layer networks. This challenge is even more prominent in the multi-layer setup, and thus motivates the design of efficient, centrality-preserving MLN reduction techniques. Network centrality does not naturally translate to its multi-layer counterpart, since the interpretation of the relative importance of nodes and layers may differ across application domains. In this paper, we take a notion of eigenvector-based centrality for a special type of MLNs (multiplex MLNs), with undirected, weighted edges, which was recently proposed in the literature. Then, we define and implement a framework for exact reductions for this class of MLNs and accompanying eigenvector centrality. Our method is inspired by the existing bisimulation-based exact model reductions for single-layered networks: the idea behind the reduction is to identify and aggregate nodes (resp. layers) with the same centrality score. We do so via efficient, static, syntactic transformations. We empirically demonstrate the speed up in the computation over a range of real-world MLNs from different domains including biology and social science.

Keywords: Multi-Layer Networks, Centrality Measures, Model Reduction, Efficient Algorithms.

1 Introduction

Traditional network analysis has facilitated key developments in research on Collective Adaptive Systems (CAS). CAS are a focus of important research efforts of today, such as ensuring the safety of cyber-physical systems, planning for smart cities, or understanding animal collective behaviour. These systems consist of a large number of entities which continuously interact with each other and the environment, they self-organise and often give rise to a system-level dynamics, *emergent behaviours*, which can not be seen by studying individuals in isolation. Network representation of a collective system is intuitive, and it allows to reason over the different aspects of the modelled system, e.g. information flows, or its evolution over time. Network analysis often

centers around classification of network components – nodes, edges etc. – wrt. different importance notions. Importance is defined through a *centrality* measure, and different algorithms for computing them have been proposed over time. A centrality measure is a real-valued function which associates nodes to their importance and, therefore, allows to rank them accordingly. Historically, the Bonacich index [5,6] (most often referred to as *eigenvector centrality*) and other extensions inspired by the Bonacich index such as Katz centrality [28] and PageRank [39] played a prominent role in network analysis. Other measures of centrality are based accordingly to different factors such as shortest paths [25], diffusion capability [1] and nodes with high contagion potential [13]. Although each of these notions measure different features of the nodes, they share common mathematical traits [4].

However, the traditional, single-layered networks allow to capture only one type of interaction among nodes. In many real-world scenarios, relations among individuals have multiple facets: in social networks, the same individuals may communicate via multiple communication platforms (i.e., they can use different online social networks to spread and gather information [51]). During epidemics, individuals interact both in the physical world, in which they spread the infection, and in a virtual communication network, where awareness about the disease is spread [23]. Moreover, 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 [21,2], dolphins [22] and birds [20].

Any finite, discrete number of different communication aspects among a set of agents, can be formally captured by adding typed edges or edge colours to the network description. Enriching the network formalism with multiple views/layers results in a multi-layer network (MLN) [16]. MLNs offer a novel way to model interactions among the components of a system as connected layers of different types of interactions. General MLNs allow for stacking up a collection of graphs over possibly different node-sets, through arbitrary coupling relationships between pairs of layers. In this work, we focus on a class of MLNs called *multiplex networks*. A multiplex is a collection of graphs over the same set of nodes but different edge sets, each of which is modelling a different type of interaction. Single-layer Networks are conveniently represented as matrices and many tools from matrix analysis have proven to be useful in identifying important network components. Along these lines, multiplex MLNs can be represented using tensors.

Carrying over the theory from network analysis to MLNs is desirable but non-trivial: most of the notions and concepts that are fundamental for single-layer network centrality do not naturally translate to its multi-layer counterpart, since the interpretation of the relative importance of nodes and layers may differ across application domains. For instance, in an effort to extend the Bonacich index to MLNs, several eigenvector-based centrality measures have been defined for multiplexes in the last few years [3,43,17,18]. In this work, we focus on the extension presented in [47] which is based on eigenvector centrality for undirected and (potentially) weighted multiplex MLNs. Among the large variety of methodologies for single-layer network analysis [26] such as clustering [7] and role-equivalent partitioning [32,50], we here aim for exact, centrality-preserving network reduction. In general, model reduction techniques aim to provide a smaller

or simpler dynamical model from the original one. Reductions are *exact*, when they guarantee an exact, provable relationship between their respective solutions, without error (otherwise, the reductions are *approximate*, when the error is either guaranteed or estimated). Exact, centrality-preserving network reduction was proposed in the context of single-layer networks [45]. This method is based on efficient model reduction framework for more general dynamical systems [46,24]; The core of these frameworks is based on an efficient partition-refinement procedure of Paige and Tarjan [40]. More specifically, some model reduction techniques based on lumping states have shown to preserve centrality properties of single-layer networks (such as Eigenvector centrality, Katz centrality and PageRank centrality) while, at the same time, relating to a variety of notions from different fields: exact role assignment [50], equitable partitions [34,35] and bisimulation [48,41].

In this work, we define and implement a framework for exact model reduction of multiplex MLNs, by lumping states and layers. Reduction is designed so to preserve the eigenvector centrality for multiplex MLNs, defined in [47] (i.e., two nodes equivalent in the ODEs enjoy the same eigenvector centrality). While our proposed framework directly extends the concept used in [45] for single-layer networks, the major technical challenge arising in the multi-layer setup is that the iterative scheme for computing eigenvector centrality for MLNs contains non-linear terms. In addition, two real-valued exponents, introduced to guarantee convergence, require additional care when lifting from the reduced solution to the original one. The relevance of our framework is demonstrated by benchmarking over a number of real-world multiplex MLNs.

Paper outline Section 2 reviews the background notions, while Section 3 introduces the proposed model reduction framework. Section 4 features an experimental evaluation on real-world multiplex MLNs. Section 5 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), eigenvector centrality measure for MLNs, IDOL programs for specifying dynamical systems and model reduction techniques based on Backward Differential Equivalence.

Notation. Throughout this work, when clear from context, we will use x_i both to denote the i-th element of vector $\boldsymbol{x}$ or the value of the map $x(i)$ (following Definition 3). For a partition $\mathcal{H}$ over a variable set $V_p \subseteq \{\mathtt{x}_1, \mathtt{x}_2, \ldots\}$, 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 $\mathtt{x}_{H,1}, \mathtt{x}_{H,2}, \ldots, \mathtt{x}_{H,|H|}$. We denote by $\|\cdot\|_1$ the 1-norm. We will denote with $V_N = \{1, \ldots, N\}$, $V_L = \{1, \ldots, L\}$ the set of nodes and layers, respectively. Vectors will be assumed to be written in column notation.

2.1 Networks and Multiplex Multi-Layer Networks

Definition 1. A (weighted, directed) *graph* is a pair $G = (V_N, E)$, where V_N is a set of $N \geq 1$ nodes and $E : V_N \times V_N \to \mathbb{R}_{\geq 0}$ is an edge-weighting function, such that

$E(i,j) = 0$ reflects that there is no edge in the graphical representation of the network. In matrix notation, a graph is given by a non-negative *adjacency matrix* $\boldsymbol{A} = (A_{ij}) \in \mathbb{R}_{\geq 0}^{N \times N}$. Graph G is *undirected*, if the matrix $\boldsymbol{A}$ is symmetric.

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. This paper will focus on weighted, undirected multiplex networks.

Definition 2. A *multiplex network* with L layers is an ordered collection of L graphs over the same set of nodes:

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

where $E^{(l)} : V_N \times V_N \to \mathbb{R}_{\geq 0}$ are the edge weights at 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 3^{rd}-order *adjacency tensor*:

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

that is, $\mathcal{A}_{ijl}$ is the weight of the edge from node i to node j in layer l.

Example 1. The adjacency tensor for the multiplex depicted in Fig. 1 left is given by layers $\boldsymbol{\mathcal{A}}^{(1)} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ and $\boldsymbol{\mathcal{A}}^{(2)} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$.

Remark 1. While in this work we will focus on multiplex networks , they are a special case of a more general notion of interconnected *multilayer networks* (MLNs), where layers can have different node sets, and, moreover, they can be coupled across layers in arbitrary ways[1]. For example, modelling public transport by different means (e.g. bus, train or metro) requires such a model.

2.2 Centrality measures

Given an undirected graph $G = (V_N, E)$ and its adjacency matrix $\boldsymbol{A} \in \mathbb{R}_{\geq 0}^{N \times N}$, we first recall the definition of eigenvector centrality for single-layer networks [36].

Definition 3. *Eigenvector centrality* $x : V_N \to \mathbb{R}_{\geq 0}$ maps each node to the average 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 $\boldsymbol{x} \in \mathbb{R}_{\geq 0}^{V_N}$ is such that $\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}$, that is, $\boldsymbol{x}$ is the right eigenvector wrt. the adjacency matrix $\boldsymbol{A}$.

[1] We provide the definitions of interconnected MLNs in Section A of the Appendix

For a given graph with adjacency matrix $\boldsymbol{A}$, eigenvector centrality may not be well-defined, that is, there may exist no unique non-negative right eigenvector (up to linear scaling). By the famous Perron-Frobenius result, whenever the largest real eigenvalue of $\boldsymbol{A}$ is unique, eigenvector centrality is guaranteed to be well-defined, and it is the respective eigenvector, with all non-negative entries. When eigenvector centrality is well-defined, it can be efficiently computed with the power iteration scheme. We restate this well-known result, for the sake of transparent analogy with the case of MLN's, which we introduce next.

Theorem 1. *([36])* If there exists a unique, non-negative eigenvector centrality on $\boldsymbol{A}$, denoted by $\boldsymbol{x}^*$, and such that $\|\boldsymbol{x}^*\|_1 = 1$, it can be computed as a limit of the power iteration sequence $\boldsymbol{x}^{(k)} = \frac{\boldsymbol{A}\boldsymbol{x}^{(k-1)}}{\left\|\boldsymbol{A}\boldsymbol{x}^{(k-1)}\right\|_1}$ for $k \geq 0$ and initially $\boldsymbol{x}^{(0)} = \mathbb{1}^N$.

In this paper, we will use one possible extension of eigenvector centrality for multiplex MLNs, proposed in [47]. The authors propose a 2-map, $\boldsymbol{f}$*-eigenvector centrality*, in which the first component of the map represents the centrality associated to the *nodes*, while the second component is centrality associated to the *layers*.

Definition 4. *([47])* Let $\boldsymbol{\mathcal{A}} \in \mathbb{R}_{\geq 0}^{N \times N \times L}$ be the adjacency tensor of an MLN with weighted, undirected layers, and let $\alpha, \beta > 0$ be such that $\frac{2}{\beta} < (\alpha - 1)$. Then, define $\boldsymbol{f} = (f_1, f_2) : \mathbb{R}_{\geq 0}^N \times \mathbb{R}_{\geq 0}^L \to \mathbb{R}_{\geq 0}^N \times \mathbb{R}_{\geq 0}^L$ as follows:

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

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

In words, the centrality x_i of node i is a sum of the centralities of each of its neighbouring nodes, weighted by the product of the edge-weight and the centrality of the layer at which that connection lies. At the same time, the centrality of a layer t_l is a sum of the centrality of all edges at that layer, where an importance of an edge is, in addition to its own weight, weighted by the centrality of the two nodes which constitute it. The parameters α and β are introduced in order to guarantee convergence and respectively well-definedness in case of undirected MLNs. Further discussion is beyond the scope of this manuscript and we refer the interested reader to [47].

Similarly as in the case of single-layer networks, a power iteration scheme for computing $\boldsymbol{f}$-centrality is desired. Throughout the rest of the work, we will use a normalised version of $\boldsymbol{f}$-mapping, denoted by $\boldsymbol{g}$:

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{t}) = \left(\frac{f_1(\boldsymbol{x}, \boldsymbol{t})}{\|f_1(\boldsymbol{x}, \boldsymbol{t})\|_1}, \frac{f_2(\boldsymbol{x}, \boldsymbol{t})}{\|f_2(\boldsymbol{x}, \boldsymbol{t})\|_1} \right)$$

We now restate a result from [47], that, for a given MLN with undirected layers, $\boldsymbol{f}$-centrality is well-defined and it can be computed as a limit of a power iterative sequence.

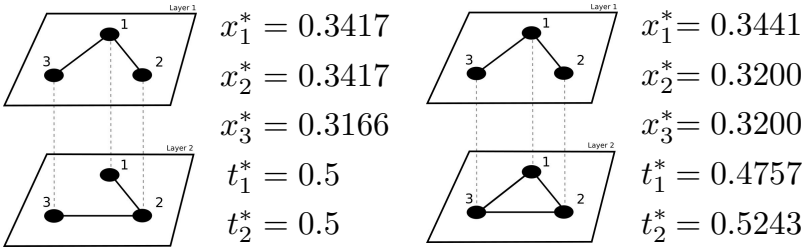

Fig. 1. An example with two MLNs and their respective f-eigenvector centralities.

Theorem 2. *([47])* There exists a unique, non-negative fixed point of the mapping $\boldsymbol{g}$. Moreover, this fixed point, denoted by $(\boldsymbol{x}^*, \boldsymbol{t}^*) \in \mathbb{R}^N_{\geq 0} \times \mathbb{R}^L_{\geq 0}$, is a limit of the following iterative scheme [2]:

$$(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)}) = \boldsymbol{g}(\boldsymbol{x}^{(k-1)}, \boldsymbol{t}^{(k-1)}) \text{ for } k \geq 1 \text{ and initially } (\boldsymbol{x}^{(0)}, \boldsymbol{t}^{(0)}) = (\mathbb{1}^N, \mathbb{1}^L) \quad (1)$$

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

Example 2. Consider the MLN depicted in Figure 1 (left). The iterative scheme to compute the $\boldsymbol{f}$-eigenvector centrality (Def. 4) is the following:

$$\begin{aligned}
x_1^{(k+1)} &= (1x_2^{(k)}t_1^{(k)} + 1x_3^{(k)}t_1^{(k)} + 1x_2^{(k)}t_2^{(k)})^{\frac{1}{\alpha}} / ||f_1(\boldsymbol{x}, \boldsymbol{t})||_1 \\
x_2^{(k+1)} &= (1x_1^{(k)}t_1^{(k)} + 1x_1^{(k)}t_2^{(k)} + 1x_3^{(k)}t_2^{(k)})^{\frac{1}{\alpha}} / ||f_1(\boldsymbol{x}, \boldsymbol{t})||_1 \\
x_3^{(k+1)} &= (1x_1^{(k)}t_1^{(k)} + 1x_2^{(k)}t_2^{(k)})^{\frac{1}{\alpha}} / ||f_1(\boldsymbol{x}, \boldsymbol{t})||_1 \\
t_1^{(k+1)} &= (2x_1^{(k)}x_2^{(k)} + 2x_1^{(k)}x_3^{(k)})^{\frac{1}{\beta}} / ||f_2(\boldsymbol{x}, \boldsymbol{t})||_1 \\
t_2^{(k+1)} &= (2x_1^{(k)}x_2^{(k)} + 2x_2^{(k)}x_3^{(k)})^{\frac{1}{\beta}} / ||f_2(\boldsymbol{x}, \boldsymbol{t})||_1
\end{aligned}$$

Example 3. In Figure 1 we show two different MLNs and their respective $\boldsymbol{f}$-eigenvector centralities. Adding an edge at Layer 2 changes both the node centrality and the layer centrality scores. More specifically, Node 1 and 3 gain importance while Node 2 loses importance. Moreover, if in the left example the two layers had equivalent centralities, in the right one, Layer 2 becomes more important because it contains more connections between high-centrality-nodes. This shows that when we choose $\boldsymbol{f}$-eigenvector centrality as the measure of choice, the role played by the nodes and layers is intertwined and therefore the two aspects of the $\boldsymbol{f}$ mapping can not be computed independently.

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

2.3 Intermediate Drift Oriented Language (IDOL)

The Intermediate Drift Oriented Language (IDOL) is a language for describing non-linear, first-order, autonomous and explicit finite systems of coupled ordinary differential equations (ODEs). We here report the fragment of the syntax and semantics of IDOL which is useful for presenting this work.

Syntax. An IDOL program p over a set of variables V_p is written in the following syntax:

$$p ::= \varepsilon | \mathrm{x}_i' = \eta, p$$
$$\eta ::= n | \mathrm{x}_i | \eta + \eta | \eta \cdot \eta$$

where $\mathrm{x}_i \in V_p$, $n \in \mathbb{Z}$ and ε is used to define the end of the program.

Semantics. We will consider conventional ODE semantics for a given IDOL program p, as the solution of the system of ODE's that it represents. The solution map $[\![\cdot]\!] : \mathbb{R}_{\geq 0}^{|V_p|} \rightarrow (V_p \rightarrow ([0, T) \rightarrow \mathbb{R}_{\geq 0}))$ will (deterministically) map each initial condition and a variable to a trace from the time domain with horizon $T \in \mathbb{R}_{\geq 0}$ to a value. For simplicity, we will denote the solution for variable x_i by $[\![\mathrm{x}_i]\!]_{\boldsymbol{x}_0}$, and we omit the dependency on initial condition $\boldsymbol{x}_0 \in \mathbb{R}_{\geq 0}^{|V_p|}$ when clear from context. Moreover, we will use $[\![\mathrm{x}]\!](t)$ to denote the vector of solutions for $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_{|Vp|}$.

Example 4. An example of a system of ODEs that can be written in IDOL:

$$\mathrm{x}_1' = 1\mathrm{x}_2\mathrm{x}_1 + 1\mathrm{x}_3\mathrm{x}_1 + 2\mathrm{x}_2$$
$$\mathrm{x}_2' = 2\mathrm{x}_1 + 1\mathrm{x}_1\mathrm{x}_2 + 1\mathrm{x}_3\mathrm{x}_2$$
$$\mathrm{x}_3' = 2\mathrm{x}_1 + 1\mathrm{x}_2$$

2.4 Backward Differential Equivalence

Backward differential equivalence (BDE) is a model reduction technique for dynamical systems written in IDOL [8,11]. BDE groups IDOL variables which are *exact fluid lumpable* - they have the same ODE semantics whenever they are given the same initial assignment. 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 IDOL program, using only one representative variable from each partition class. The reduction algorithms proposed in [8,11] are only syntactically manipulating the IDOL program (without executing its semantics), and they are of polynomial complexity in the number of variables of the program. We propose in this paper to use BDE reductions, to reduce the computation of $\boldsymbol{f}$-centrality measure for MLNs.

Definition 5. We call $\boldsymbol{x} \in \mathbb{R}_{\geq 0}^{|V_p|}$ *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 6. Let p be an IDOL program and $\mathcal{H}$ a partition over the variable set V_p. Then, the IDOL program p is *exact fluid lumpable wrt. partition* $\mathcal{H}$, if $[\![\mathrm{x}]\!]_{\boldsymbol{x}_0}(t)$ is constant on $\mathcal{H}$ for all $t \geq 0$, whenever $\boldsymbol{x}_0 = [\![\mathrm{x}]\!](0)$ is constant on $\mathcal{H}$. Then, we will call $\mathcal{H}$ a *BDE partition* of V_p.

Following [8], computing the coarsest BDE partition is possible to compute in polynomial-time worst-case complexity, for any IDOL program which corresponds to a set of chemical reactions with mass-action kinetics.

We now state the result which shows how to use a BDE to construct a reduced IDOL program, operating over only the representative variables (BDE quotient).

Theorem 3. *([9])* Let p be an IDOL program and $\mathcal{H}$ a BDE partition over its variable set V_p, and $T > 0$ a time horizon. The *backward reduced program of p with respect to $\mathcal{H}$*, denoted by $\tilde{p}_{\mathcal{H}}$, is defined over a set of variables $V_{p_{\mathcal{H}}} = \{\mathtt{y}_1, \ldots, \mathtt{y}_{|\mathcal{H}|}\}$, and it follows the update functions:

$$\mathtt{y}'_H = \eta_{H,1}[\mathtt{x}_{\bar{H},1}/\mathtt{y}_{\bar{H}}, \ldots, \mathtt{x}_{\bar{H},|\bar{H}|}/\mathtt{y}_{\bar{H}} : \bar{H} \in \mathcal{H}], \text{ for } H \in \mathcal{H},$$

where $\mathtt{x}_{\bar{H},i}/\mathtt{y}_{\bar{H}}$ denotes the action of renaming any variable from class $\mathtt{x}_{\bar{H},i}$ by its representative $\mathtt{y}_{\bar{H}}$.

Originally designed for reducing ODEs, BDE techniques have also been applied for reducing single-layer networks, continuous-time Markov chains (CTMCs) and differential algebraic equations. In particular, in [45], a property-preserving exact model reduction algorithm for networks is shown. The given network is first transformed into an IDOL program, 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.

Theorem 4. *([45])* Given a graph $G = (V_N, E)$ with adjacency matrix $\boldsymbol{A}$, let p_G be the IDOL program over the set of variables V_N: $\mathtt{x}'_i = \sum_{1 \leq j \leq n} A_{ij} \cdot \mathtt{x}_j$ for all $i \in V_N$. 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 IDOL program is such that the equation for the derivative of variable $\mathtt{x}_i$ is the weighted sum of its direct (outgoing) edges[3]. So, the key idea in that the transformation from the network to an IDOL program is that the equations in the IDOL program exactly match the iterative scheme for computing the centrality measure of interest.

Notice that the obtained IDOL program contains only linear transformations over its variables. We next propose an analogue of Thm. 4 for multiplex MLNs. The translation to an IDOL program will encode the iterative scheme for computing $\boldsymbol{f}$-centrality, which involves non-linear terms (of second order), as well as the positive exponents α and β.

3 Centrality-preserving MLN reduction

Given a multi-layered network, its $\boldsymbol{f}$-centrality can be computed with the iterative scheme $\boldsymbol{g}$ presented in Theorem 2 (Eq. 1) . Our aim is to bypass the direct computation and instead compute the $\boldsymbol{f}$-centrality indirectly, by first detecting sets of variables

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

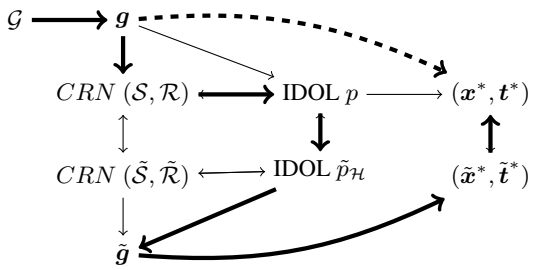

Fig. 2. An illustration of the proposed methodology. The arrows are used for illustrative purpose and they are not to be formally interpreted. For a given MLN $\mathcal{G}$, its $\boldsymbol{f}$-centrality vector $(\boldsymbol{x}^*, \boldsymbol{t}^*)$ can be computed directly through the iterative map $\boldsymbol{g}$ (dotted line). Alternatively, as depicted with thick full arrows, the equations in $\boldsymbol{g}$ can first be translated into an IDOL programme p with variables encoding nodes and layers of the MLN, and its BDE quotient program $p_{\mathcal{H}}$ is used to define a reduced iterative scheme $\tilde{\boldsymbol{g}}$ over a reduced set of variables, where more nodes and more layers are represented by only one, representative variable.The solution of the reduced iterative scheme, $(\tilde{\boldsymbol{x}}^*, \tilde{\boldsymbol{t}}^*)$, allows to exactly reconstruct the $\boldsymbol{f}$-centrality of the original MLN.

which evolve equivalently throughout the iterations, and then proposing a reduced iterative scheme $\tilde{\boldsymbol{g}}$, where only one representative variable is kept for each set of equivalent ones.

To do so, we first introduce an assignment of an IDOL program to a given MLN. Then, given an IDOL program, we compute its BDE-equivalent quotient, as described in Section 2. BDE equivalence guarantees that the original and smaller IDOL program have the same differential, continuous-step semantics. On the other hand, the iterative scheme $\boldsymbol{g}$ defines a discrete-step semantics over the variables of the MLN. Our proposal is to compute the centrality measure of the original MLN using the smaller IDOL program. To do so, we need to show that the reduced iterative scheme $\tilde{\boldsymbol{g}}$ over the BDE-quotient of the original IDOL program, preserves the solutions of the iterative scheme $\boldsymbol{g}$. A diagram of the workflow of the proposed framework is presented in Figure 2.

The following theorem shows which are the quantities that we should account for when we search for equivalences among the centrality scores.

Theorem 5. For $i \in V_N$, define the quantity of interest

$$\bar{x}_i^{(k)} := \sum_{a=1}^{N} \sum_{l=1}^{L} A_{ial} x_a^{(k)} t_l^{(k)},$$

which is the right hand side of the mapping $\boldsymbol{f}$ from Definition 4, without the exponential operator α. Then, for all pairs of nodes $i, j \in V_N$, it holds that

$$\text{if } \bar{x}_i^{(k)} = \bar{x}_j^{(k)} \text{ then } x_i^{(k+1)} = x_j^{(k+1)}, \text{ for all } k \geq 0.$$

We report the proofs in Section B of the Appendix.

The key idea is to go from the definition of the multilayer network eigenvector centrality obtained with the iterative scheme (1) to an IDOL program p such that there is a

correspondence between the node and the layer eigenvector centrality and the variables of the IDOL program.

Definition 7. (IDOL translation) Let $\mathcal{G}$ be a multiplex network and let $\boldsymbol{\mathcal{A}}_{ijl}$ be the 3-rd order adjacency tensor of the multiplex $\mathcal{G}$. We define an IDOL program p, with $V_p = V_N \cup V_L$, as follows:

$$\mathtt{x}'_i = \sum_{j=1}^{N}\sum_{l=1}^{L} A_{ijl}\mathtt{x}_j\mathtt{t}_l$$
$$\mathtt{t}'_l = \sum_{i=1}^{N}\sum_{j=1}^{N} A_{ijl}\mathtt{x}_i\mathtt{t}_j$$

for all $i \in V_N$ and for all $l \in V_L$. With $[\![\mathtt{x}]\!](0) = \mathbb{1}^N$ and $[\![\mathtt{t}]\!](0) = \mathbb{1}^L$.

We now want to identify which nodes in the MLN have identical $\boldsymbol{f}$-eigenvector centrality. This holds if they follow equivalent equations in the iterative scheme used for computing them. Similarly to the result presented in Theorem 4 which shows a similar translation for single layer networks, the iterative scheme equations used to compute the $\boldsymbol{f}$-eigenvector centralities on MLNs can be translated to an IDOL program. The major technical difference is in that the MLN translation contains non-linear terms, and the exponents α and β. Once we have obtained the corresponding IDOL program we can apply the general technique for computing the equivalences among its variables.

The next Theorem shows how to write an IDOL program, such that if two variables have the same semantics in the dynamical system of the IDOL program, 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 6. Let $\mathcal{G}$ be a multiplex network and let $\boldsymbol{\mathcal{A}}_{ijl}$ be the 3-rd order adjacency tensor of the multiplex network $\mathcal{G}$. Let $\boldsymbol{f}$ be the mapping as defined in Dfn. 4, and let $\boldsymbol{g}$ be its normalized version. Let $(\boldsymbol{x}^*, \boldsymbol{t}^*)$ be the unique solution (the centrality scores). Given any initial conditions $(\boldsymbol{x}^{(0)}, \boldsymbol{t}^{(0)}) \in \mathbb{R}^N_{\geq 0} \times \mathbb{R}^L_{\geq 0}$ and $(\boldsymbol{x}^{(k+1)}, \boldsymbol{t}^{(k+1)}) = \boldsymbol{g}(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})$, the following holds:

$$\lim_{k\to\infty} (\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)}) = (\boldsymbol{x}^*, \boldsymbol{t}^*)$$

Then, in the IDOL program p obtained via Definition 7, for chosen time horizon $T > 0$, some $i, j \in \{1, \ldots, N\}$ and $l, q \in \{1, \ldots, L\}$, the following holds:

- If $\forall t \in [0, T).\ [\![\mathtt{x}_i]\!](t) = [\![\mathtt{x}_j]\!](t)$ in the IDOL program p, then $\forall k \in \mathbb{N}\,.\, x_i^{(k)} = x_j^{(k)}$
- If $\forall t \in [0, T).\ [\![\mathtt{t}_l]\!](t) = [\![\mathtt{t}_q]\!](t)$ in the IDOL program p, then $\forall k \in \mathbb{N}\,.\, t_l^{(k)} = t_q^{(k)}$

From Theorem 6 we now know that we can build a non-linear IDOL program p such that if two variables are equal in the IDOL program then, the corresponding nodes (or layer) centrality are equal. Now, we can use the established results on the IDOL program and calculate the BDE partition $\mathcal{H}$ on the IDOL program p generated by the translation in Dfn. 7.

Up to this point, starting from a multiplex graph $\mathcal{G}$ we provided a procedure to translate it into an IDOL program and we provided a technique to calculate a BDE partition $\mathcal{H}$ on the IDOL program. Now, we introduce the following lemma and definition to formally translate partition $\mathcal{H}$, which is defined over the IDOL program's variables, to its counterpart $\boldsymbol{\mathcal{H}}^*$ defined over the nodes and the layers of the multiplex graph $\mathcal{G}$.

Corollary 1. For a multiplex network $\mathcal{G}$, let p be the IDOL program defined in Dfn. 7 with variable set $V_p = \{\mathtt{x}_1, \ldots, \mathtt{x}_N, \mathtt{t}_1, \ldots, \mathtt{t}_L\}$. Next, let $\boldsymbol{\mathcal{H}} = (\mathcal{H}_x, \mathcal{H}_t)$ be a BDE partition over V_p, such that $\mathcal{H}_x$ is a partition over the node variables and $\mathcal{H}_t$ is a partition over the layer variables. Furthermore, we will denote by $V_p^* = \{x_1, \ldots, t_1, \ldots\} \cong V_p$ the variables used in the iterative scheme $\boldsymbol{g}^{(\cdot)}$. Then, for all initial conditions, semantic equivalence from the IDOL programme carries over to the equivalence in the iterative scheme:

- $\forall H_x \in \mathcal{H}_x, \forall t \geq 0, \forall \mathtt{x}_i, \mathtt{x}_j \in H_x . \llbracket \mathtt{x}_i \rrbracket(t) = \llbracket \mathtt{x}_j \rrbracket(t) \implies \forall k \in \mathbb{N} . x_i^{(k)} = x_j^{(k)}$
- $\forall H_t \in \mathcal{H}_t, \forall t \geq 0, \forall \mathtt{t}_l, \mathtt{t}_q \in H_t . \llbracket \mathtt{t}_l \rrbracket(t) = \llbracket \mathtt{t}_q \rrbracket(t) \implies \forall k \in \mathbb{N} . t_l^{(k)} = t_q^{(k)}$

In the following, we will denote by $\boldsymbol{\mathcal{H}}^* = (\mathcal{H}_x^*, \mathcal{H}_t^*)$ a partition over V_p^*, such that any two variables are lumped in the BDE of the IDOL programme iff their respective symbols are lumped in the iterative scheme:

$$\forall i, j \in \{1, \ldots, N\}, H_{a,x} \in \mathcal{H}_x . \mathtt{x}_i, \mathtt{x}_j \in H_{a,x} \implies H_{a,x}^* \in \mathcal{H}^* . x_i, x_j \in H_{a,x}^*$$
$$\forall i, j \in \{1, \ldots, L\}, H_{a,t} \in \mathcal{H}_t . \mathtt{t}_i, \mathtt{t}_j \in H_{a,t} \implies H_{a,t}^* \in \mathcal{H}^* . t_i, t_j \in H_{a,t}^*$$

Example 5. If we go back to the running example presented in the left of Figure 1 and we apply Theorem 6 we obtain the following IDOL program p:

$$\begin{aligned} \mathtt{x}_1' &= 1\mathtt{x}_2\mathtt{t}_1 + 1\mathtt{x}_3\mathtt{t}_1 + 1\mathtt{x}_2\mathtt{t}_2 & \mathtt{t}_1' &= 2\mathtt{x}_1\mathtt{x}_2 + 2\mathtt{x}_1\mathtt{x}_3 \\ \mathtt{x}_2' &= 1\mathtt{x}_1\mathtt{t}_1 + 1\mathtt{x}_1\mathtt{t}_2 + 1\mathtt{x}_3\mathtt{t}_2 & \mathtt{t}_2' &= 2\mathtt{x}_1\mathtt{x}_2 + 2\mathtt{x}_2\mathtt{x}_3 \\ \mathtt{x}_3' &= 1\mathtt{x}_1\mathtt{t}_1 + 1\mathtt{x}_2\mathtt{t}_2 \end{aligned}$$

We consider the following partition $\mathcal{H} = \{\{\mathtt{x}_1, \mathtt{x}_2\}, \{\mathtt{x}_3\}, \{\mathtt{t}_1\}, \{\mathtt{t}_2\}\}$, which is a BDE of p and we shall use $\mathtt{y}_1$ as the representative of block $\{\mathtt{x}_1, \mathtt{x}_2\}$, $\mathtt{y}_2$ as the representative of block $\{\mathtt{x}_3\}$ and $\mathtt{r}_1, \mathtt{r}_2$ representatives of the blocks $\{\mathtt{t}_1\}, \{\mathtt{t}_2\}$, respectively. The IDOL quotient of p given $\mathcal{H}$ is the following:

$$\begin{aligned} \mathtt{y}_1' &= 1\mathtt{y}_1\mathtt{r}_1 + 1\mathtt{y}_2\mathtt{r}_1 + 1\mathtt{y}_1\mathtt{r}_2 & \mathtt{r}_1' &= 2\mathtt{y}_1\mathtt{y}_1 + 2\mathtt{y}_1\mathtt{y}_2 \\ \mathtt{y}_2' &= 1\mathtt{y}_1\mathtt{r}_1 + 1\mathtt{y}_1\mathtt{r}_2 & \mathtt{r}_2' &= 2\mathtt{y}_1\mathtt{y}_1 + 2\mathtt{y}_1\mathtt{y}_2 \end{aligned}$$

Now that we established the relationship between the partitions we proceed to define the proper reduced system to calculate the multiplex node and layer centrality as follows.

Definition 8. Let $(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)}) = \boldsymbol{g}(\boldsymbol{x}^{(k-1)}, \boldsymbol{t}^{(k-1)})$ be the iterative scheme and let $\boldsymbol{\mathcal{H}} = (\mathcal{H}_x, \mathcal{H}_t)$ be the BDE partition on the IDOL program p obtained using Theorem 6 and

Cor. 1. Let $\boldsymbol{\mathcal{H}}^* = (\mathcal{H}_x^*, \mathcal{H}_t^*)$ be the corresponding partition on the iterative scheme as defined in Cor. 1. We define $(\boldsymbol{y}^{(k)}, \boldsymbol{r}^{(k)}) = \tilde{\boldsymbol{g}}(\boldsymbol{y}^{(k-1)}, \boldsymbol{r}^{(k-1)})$ as the *Reduced iterative scheme* with respect to $\boldsymbol{\mathcal{H}}^*$:

$$\tilde{f}_1 = f_1[x_{H_x,1}/y_{H_x}, \ldots, x_{H_x,|H_x|}/y_{H_x}, t_{H_t,1}/r_{H_t}, \ldots, t_{H_t,|H_t|}/r_{H_t} : H_x \in \mathcal{H}_x \ , \ H_t \in \mathcal{H}_t]$$
$$\tilde{f}_2 = f_2[x_{H_x,1}/y_{H_x}, \ldots, x_{H_x,|H_x|}/y_{H_x}, t_{H_t,1}/r_{H_t}, \ldots, t_{H_t,|H_t|}/r_{H_t} : H_x \in \mathcal{H}_x \ , \ H_t \in \mathcal{H}_t]$$

Next, we define $\bar{y}_i^{(k)}$ similarly as we previously defined $\bar{x}_i^{(k)}$:

$$\bar{y}_i^{(k)} = \sum_{j=1}^{N} \sum_{l=1}^{L} A_{ijl} y_{H_{x,j}}^{(k-1)} r_{H_{t,l}}^{(k-1)}$$

where, $H_{x,j} = i$ if $x_j \in H_{x,i}$ and $H_{t,q} = l$ if $t_q \in H_{t,l}$. In order to guarantee that $\boldsymbol{x}^{(k)} = \boldsymbol{y}^{(k)}$ we shall use the following reduced computation:

$$\boldsymbol{x}^{(k)} = \frac{\bar{\boldsymbol{x}}^{(k)}}{\left\|\bar{\boldsymbol{x}}^{(k)}\right\|_1} = \frac{\bar{\boldsymbol{y}}^{(k)}}{\sum_{j=1}^{m} |H_{x,j}| \bar{y}_j^{(k)}} = \boldsymbol{y}^{(k)}$$

We can now focus on the second component of the mapping and we define $\bar{r}_l^{(k)}$ the nominator of $\tilde{\boldsymbol{f}}_2$ as follows:

$$\bar{r}_l^{(k)} = \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ijl} y_{H_{x,i}}^{(k-1)} y_{H_{x,j}}^{(k-1)}$$

where, $H_{x,j} = i$ if $x_j \in H_{x,i}$. In order to guarantee that $\boldsymbol{t}^{(k)} = \boldsymbol{r}^{(k)}$ we shall use the following reduced computation:

$$\boldsymbol{t}^{(k)} = \frac{\bar{\boldsymbol{t}}^{(k)}}{\left\|\bar{\boldsymbol{t}}^{(k)}\right\|_1} = \frac{\bar{\boldsymbol{r}}^{(k)}}{\sum_{j=1}^{Q} |H_{t,j}| \bar{r}_j^{(k)}} = \boldsymbol{r}^{(k)}$$

Example 6. If we consider the running example presented in the left of Figure 1, we know that the partition $\mathcal{H} = \{\{\mathtt{x}_1, \mathtt{x}_2\}, \{\mathtt{x}_3\}, \{\mathtt{t}_1\}, \{\mathtt{t}_2\}\}$ is a BDE of the IDOL program p and we obtained the following reduced IDOL program:

$$\begin{aligned} \mathtt{y}_1' &= 1\mathtt{y}_1\mathtt{r}_1 + 1\mathtt{y}_2\mathtt{r}_1 + 1\mathtt{y}_1\mathtt{r}_2 & \mathtt{r}_1' &= 2\mathtt{y}_1\mathtt{y}_1 + 2\mathtt{y}_1\mathtt{y}_2 \\ \mathtt{y}_2' &= 1\mathtt{y}_1\mathtt{r}_1 + 1\mathtt{y}_1\mathtt{r}_2 & \mathtt{r}_2' &= 2\mathtt{y}_1\mathtt{y}_1 + 2\mathtt{y}_1\mathtt{y}_2 \end{aligned}$$

In order to compute the original $\boldsymbol{f}$-eigenvector centrality values we set up the following iterative scheme:

$$\begin{aligned} \bar{y}_1^{(k)} &= 1y_1^{(k-1)} r_1^{(k-1)} + 1y_2^{(k-1)} r_1^{(k-1)} + 1y_1^{(k-1)} r_2^{(k-1)} \\ \bar{y}_2^{(k)} &= 1y_1^{(k-1)} r_1^{(k-1)} + 1y_1^{(k-1)} r_2^{(k-1)} \\ \bar{r}_1^{(k)} &= 2y_1^{(k-1)} y_1^{(k-1)} + 2y_1^{(k-1)} y_2^{(k-1)} \end{aligned}$$

$$\bar{r}_2^{(k)} = 2y_1^{(k-1)}y_1^{(k-1)} + 2y_1^{(k-1)}y_2^{(k-1)}$$
$$x_1^{(k)} = x_2^{(k)} = \bar{y}_1^{(k)}/(|2\bar{y}_1^{(k)}| + |\bar{y}_2^{(k)}|)$$
$$x_2^{(k)} = \bar{y}_2^{(k)}/(|2\bar{y}_1^{(k)}| + |\bar{y}_2^{(k)}|)$$

4 Experimental results

In this section we present the results of our experimental evaluation on some real world case studies. We measure the performance of our approach in terms of model reduction ratio and we measure the speed up in the computation of the desired centrality measures.

Implementation and Environment. The tools used for the experiments are MATLAB and ERODE [10], a state-of-the-art tool for model reduction for systems of ODEs and Chemical Reaction Networks. The input is the list of edges $E^{(l)}$ for all $l \in \{1, ..., L\}$ representing a multiplex network $\mathcal{G} = \{G^{(l)} = (V_N, E^{(l)})\}_{l \in V_L}$. ERODE accepts the input as a file that encodes an ODE system or a Chemical Reaction Networks (CRN). Due to a bottleneck in the processing of files in the ODE format we had to input the files in the CRN format. A MATLAB script translates the list of edges in the CRN using Definition 11 presented in Section C of the Appendix. ERODE then proceeds with the model reduction and provides the reduced CRN as its output. The centrality scores are computed with a MATLAB script and another MATLAB script is used to convert the reduced CRN into the reduced model and used to calculate the centrality score on the reduced model. All the experiments have been conducted on a MacBook Pro with a 2.6 GHz Intel Core i7 with 16 GB of RAM.

The instances. In order to provide some real-world case studies we ran our proposed reduction technique on multiplex MLNs retrieved from the CoMuNe Lab repository (https://comunelab.fbk.eu). The results for both undirected and directed instances are presented in Table 1. Columns show the name of the instance, the number of nodes, the number of layers, the number of nodes in the reduced model (Red. Nodes), the number of layers in the reduced model (Red. Layers), the time spent to compute the centrality measure using the original model, the time spent to do the BDE reduction and the time spent to compute the centrality measure using the reduced model. Moreover, the Reduction Ratio (size of the nodes of the reduced model divided by the size of the nodes of the original models) and the Speed Up Ratio (time spent to compute the centrality on the reduced model divided by the time spent to compute the centrality on the original model) are presented in the last two columns. We first present the *undirected graphs* instances. These instances are undirected in the repository.

- *Padgett-Florentine-Families*: this multiplex describes the relationships between Florentine families in the Renaissance, the two layers represent marriage alliances and business relationships, respectively [38].
- *CS-Aarhus*: this multiplex social network consists of five kinds of relationships between the employees of the Computer Science department at Aarhus university. The layers represent the following relationships: Facebook, Leisure, Work, Co-Authorship and Lunch [33].

- *London-Transport*: the nodes in this multiplex represent the train stations in London and edges encode existing routes between stations. The layers represent the Underground, Overground and DLR stations, respectively [15].
- *EUAirTrainsportation*: the multilayer network is composed by thirty-seven different layers each one corresponding to a different airline operating in Europe [12].
- *PierreAuger*: this instance represents the different working tasks carried between a two years span within the Pierre Auger Collaboration between the CoMuNe Lab and the Pierre Auger observatory. Each layer represents 16 different topics based on the keywords and the content of each submission [19].
- *arxiv-netscience*: this multiplex consists of layers corresponding to different arXiv categories. Nodes represent authors and edges represent the weighted co-authorship relationship [19].

Due to the fact that many of the undirected instances are small we do not obtain sensible reductions nor speed up in the computation. Despite this, we can observe a meaningful reduction for the largest of the undirected instances, namely the *arxiv-netscience* instance. We now present the instances that in the repository are *directed*. It is worth noting that, because of the fact that the centrality measure we considered throughout this paper is defined for the undirected case only, we modified these instances to make them undirected in order to prove the effectiveness of our proposed methodology. Another reason to do so is the fact that there is a small amount of undirected instances. Moreover, the undirected instances present a limited variety of nodes, edges and layer sizes.

- *Krackhardt-High-Tech*: this multiplex social network describes the relationships between managers of an high-tech company. The layers represent advice, friendship and "reports to" relationships, respectively [30].
- *Vickers-Chan-7thGraders*: this data was collected from 29 seventh grade students in a school in Victoria, Australia. Students were asked to nominate their classmates on a number of relations including the following three (layers): Who do you get on with in the class? Who are your best friends in the class? Who would you prefer to work with? [49]
- *Kapferer-Tailor-Shop*: this instance represents the interactions in a tailor shop in Zambia over a period of ten months. The layers represent two different types of interactions, recorded at two different times. The relationships captured by this multiplex are *instrumental* (work-related) and *sociational* (friendship, socio-emotional) interactions [27].
- *Lazega-Law-Firm*: this multiplex social network consists of three kinds of relationships between partners and associates of a corporate law partnership. The layers represent co-work, friendship and advice relationships, respectively [31,42].
- *Genetic interaction instances*: we consider a variety of genetic interactions networks that are present in the CoMuNe Lab repository [14]. In turn, these instances were taken from the Biological General Repository for Interaction Datasets (BioGRID) and represent different types of genetic interactions for organisms [44]. These instances are marked with the keyword *genetic* in Table 1.
- *Fao-Trade*: this multiplex describes different type of trade relationships among countries, it was originally obtained from the Food and Agriculture Organization

of the United Nations. The worldwide food import/export network is an economic network in which layers represent products, nodes are countries and edges at each layer represent import/export relationships of a specific food product among countries. It is worth pointing out that, due to the nature of this instance, it has the peculiarity that there are more layers than nodes [14].
- *MoscowAthletics2013*: this multiplex represents the different types of social relationships among Twitter users during the 2013 World Championships in Athletics. The three layers correspond to retweets, mentions and replies over the time frame of the event. These are the relationships that will be also used for the following Twitter instances [37].
- *NYClimateMarch2014*: this instance describes the Twitter interactions among the users during the People's Climate March in 2014 [37].
- *MLKing2013*: this instance describes the Twitter interactions among the users during the 50^{th} anniversary of Martin Luther King's speech in 2013 [37].
- *Cannes2013*: this instance describes the Twitter interactions among the users during the Cannes Film Festival in 2013 [37].

As expected, similarly to the undirected instances, the small instances do not provide much insight on the effectiveness of the methodology but, as the size of the instance increases we can see significant reductions and speed ups. Notably, when tackling the big instances the proposed methodology yields reductions that reduce the size of the network to half of its original size and, in the case of the largest instance we can obtain a reduction that at least provides some information about which nodes have the same centrality. Such information could not be retrieved by calculating the centrality directly on the original multiplex because of the computational cost.

5 Conclusions and future work

In this paper, we have shown a framework that allows to reduce MLNs while exactly preserving the $\boldsymbol{f}$-eigenvector centrality measure proposed in literature. In other words, we have related an extension of eigenvector centrality on undirected and (possibly) weighted multiplex MLNs to an exact model reduction technique for dynamical systems called backward differential equivalence (BDE). The relevance of the result was demonstrated by computing reduction of real-world MLNs and by showing a speed up in the computation of the centrality measure of interest. Immediate challenges for future work include extending the framework to other centrality measures and to other families of MLNs, as well as the extension of these results to multiplex MLNs that feature directed layers. The presented framework is a versatile cornerstone work that can be further tailored to deal with other types of network notions such as extensions of role equivalence on MLNs. Exact reductions might not yield significant reductions in typically asymmetric real-world case studies. It is therefore a practically important future research direction to study approximate reductions, some versions of which have already been proposed for the study of clustering in networks [7] and for agent-based models [29]. Thanks to the theory established in this paper, we hope to be able to design the approximate versions of this reduction technique.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2117 422037984.

References

1. Abhijit Banerjee, Arun G a, Esther Duflo, and Matthew Jackson. The diffusion of microfinance. *Science (New York, N.Y.)*, 341:1236498, 07 2013.
2. Louise Barrett, Peter Henzi, and David Lusseau. 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, 08 2012.
3. Federico Battiston, Vincenzo Nicosia, and Vito Latora. Structural measures for multiplex networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 89:032804, 03 2014.
4. Francis Bloch, Matthew Jackson, and Pietro Tebaldi. Centrality measures in networks. *SSRN Electronic Journal*, 08 2016.
5. Phillip Bonacich. Factoring and weighting approaches to status scores and clique identification. 1972.
6. Phillip Bonacich. Power and centrality: A family of measures. 1987.
7. Ulrik Brandes, Daniel Delling, Marco Gaertler, R. Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 20:172 – 188, 03 2008.
8. Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Efficient syntax-driven lumping of differential equations. In *TACAS'16*.
9. Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Symbolic computation of differential equivalences. In *43st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, 2016.
10. Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. ERODE: A tool for the evaluation and reduction of ordinary differential equations. In *TACAS*, pages 310–328, 2017.
11. Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Maximal aggregation of polynomial dynamical systems. *PNAS*, 114(38), 2017.
12. Alessio Cardillo, Jesus Gmez-Gardees, Massimiliano Zanin, Miguel Romance, David Papo, Francisco Del Pozo Guerrero, and Stefano Boccaletti. Emergence of network features from multiplexity. *Scientific reports*, 3:1344, 02 2013.
13. Nicholas Christakis and James Fowler. Social network sensors for early detection of contagious outbreaks. *PloS one*, 5:e12948, 09 2010.
14. Manlio De Domenico, Vincenzo Nicosia, Alex Arenas, and Vito Latora. Structural reducibility of multilayer networks. *Nature Communications*, 6:6864, 04 2015.
15. Manlio De Domenico, Albert Solé-Ribalta, Sergio Gómez, and Alex Arenas. Navigability of interconnected networks under random failures. *Proceedings of the National Academy of Sciences*, 2014.
16. Manlio De Domenico, Albert Sol-Ribalta, Emanuele Cozzo, Mikko Kivel, Yamir Moreno, Mason Porter, Sergio Gomez, and Alex Arenas. Mathematical formulation of multi-layer networks. *Physical Review X*, 3, 07 2013.
17. Manlio De Domenico, Albert Sol-Ribalta, Elisa Omodei, Sergio Gomez, and Alex Arenas. Centrality in interconnected multilayer networks. *Physica D: Nonlinear Phenomena*, 323, 11 2013.

18. Manlio De Domenico, Albert Sol-Ribalta, Elisa Omodei, Sergio Gomez, and Alex Arenas. Ranking in interconnected multilayer networks reveals versatile nodes. *Nature communications*, 6:6868, 04 2015.
19. Manlio De Domenico, Andrea Lancichinetti, Alexandre Arenas, and Martin Rosvall. Identifying modular flows on multilayer networks reveals highly overlapping organization in social systems. *ArXiv*, abs/1408.2925, 2015.
20. Damien Farine, Lucy Aplin, Ben Sheldon, and William Hoppitt. Interspecific social networks promote information transmission in wild songbirds. *Proceedings. Biological sciences / The Royal Society*, 282, 03 2015.
21. Mathias Franz, Jeanne Altmann, and Susan Alberts. Knockouts of high-ranking males have limited impact on baboon social networks. *Current Zoology*, 61:107–113, 02 2015.
22. Stefanie Gazda, Swami Iyer, Timothy Killingback, Richard Connor, and Solange Brault. The importance of delineating networks by activity type in bottlenose dolphins (tursiops truncatus) in cedar key, florida. *Royal Society Open Science*, 2:140263–140263, 03 2015.
23. Clara Granell, Sergio Gomez, and Alex Arenas. Dynamical interplay between awareness and epidemic spreading in multiplex networks. *Physical review letters*, 111:128701, 09 2013.
24. Giulio Iacobelli, Mirco Tribastone, and Andrea Vandin. Differential bisimulation for a markovian process algebra. In *MFCS*, pages 293–306, 2015.
25. Matthew Jackson. *Social and Economic Networks*. 12 2008.
26. Jeffrey Johnson, Steve Borgatti, and Martin Everett. *Analyzing Social Networks*. 01 2013.
27. Bruce Kapferer. Strategy and transaction in an african factory: African workers and indian management in a zambian town. *Manchester University Press*, 43(4):362363, 1972.
28. Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.
29. Wasiur R. KhudaBukhsh, Arnab Auddy, Yann Disser, and Heinz Koeppl. Approximate lumpability for markovian agent-based models using local symmetries. *Journal of Applied Probability*, 56, 04 2018.
30. David Krackhardt. Cognitive social structures. *Social Networks*, 9(2), 1987.
31. Emmanuel Lazega. The collegial phenomenon: The social mechanisms of cooperation among peers in a corporate law partnership. *Oxford University Press*, 2001.
32. Jürgen Lerner. Role assignments. In *Network Analysis: Methodological Foundations [outcome of a Dagstuhl seminar, 13-16 April 2004]*, pages 216–252, 2004.
33. Matteo Magnani, Barbora Micenková, and Luca Rossi. Combinatorial analysis of multiple networks. *CoRR*, abs/1303.4986, 2013.
34. Brendan McKay. Practical graph isomorphism. *Congressus Numerantium*, 30:45–87, 1981.
35. Brendan McKay and Adolfo Piperno. Practical graph isomorphism ii. *Journal of Symbolic Computation*, 09 2013.
36. Mark Newman. *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010.
37. Elisa Omodei, Manlio De Domenico, and Alex Arenas. Characterizing interactions in online social networks during exceptional events. *Frontiers in Physics*, 3, 06 2015.
38. John F. Padgett and Christopher K. Ansell. Robust action and the rise of the medici, 1400-1434. *American Journal of Sociology*, 98(6):1259–1319, 1993.
39. Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. In *WWW 1999*, 1999.
40. Robert Paige and Robert Endre Tarjan. Three partition refinement algorithms. *SIAM J. Comput.*, 16(6):973–989, 1987.
41. George J. Pappas. Bisimilar linear systems. *Automatica*, 39(12):2035–2047, 2003.
42. Tom A. B. Snijders, Philippa E. Pattison, Garry L. Robins, and Mark S. Handcock. New specifications for exponential random graph models. *Sociological Methodology*, 36(1):99–153, 2006.

43. Luis Sol Conde, Miguel Romance, Regino Herrero, Julio Flores, Alejandro Garca del Amo, and Stefano Boccaletti. Eigenvector centrality of nodes in multiplex networks. *Chaos (Woodbury, N.Y.)*, 23:033131, 09 2013.
44. Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. Biogrid: A general repository for interaction datasets. *Nucleic acids research*, 34:D535–9, 01 2006.
45. Stefano Tognazzi, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Differential equivalence yields network centrality. In Tiziana Margaria and Bernhard Steffen, editors, *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, 2018.
46. Max Tschaikowski and Mirco Tribastone. Exact Fluid Lumpability for Markovian Process Algebra. In *CONCUR*, pages 380–394, 2012.
47. Francesco Tudisco, Francesca Arrigo, and Antoine Gautier. Node and layer eigenvector centralities for multiplex networks. *SIAM Journal of Applied Mathematics*, 78(2):853–876, 2018.
48. A. J. van der Schaft. Equivalence of dynamical systems by bisimulation. *IEEE Transactions on Automatic Control*, 49, 2004.
49. M. Vickers and M. Chan. Representing classroom social structure. victoria institute of secondary education, melbourne. 1981.
50. Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.
51. Xuetao Wei, Nicholas Valler, B. Prakash, Iulian Neamtiu, Michalis Faloutsos, and Christos Faloutsos. Competing memes propagation on networks: A case study of composite networks. *ACM SIGCOMM Computer Communication Review*, 42:5–11, 10 2012.

A Interconnected MLNs

Definition 9. Let $V_L = \{1, \dots, L\}$ denote a set of L layers, $V^{(l)} = \{1, 2, \dots, N_l\}$ a set of nodes at layer $l \in V_L$, and $E^{(l)} : V^{(l)} \times V^{(l)} \to \mathbb{R}_{\geq 0}$ the edge weights at layer l. Moreover, for any two layers $l, l' \in V_L$, let $\Delta_{l,l'} : V^{(l)} \times V^{(l')} \to \mathbb{R}_{\geq} 0$ (resp. $\mathbf{\Delta}_{l,l'} \in \mathbb{R}_{\geq 0}^{N_l \times N_{l'}}$) be the *coupling function* (resp. *coupling matrix*) connecting the nodes between layers l and l'. Then, an *interconnected multilayer network* (MLN) $(\mathcal{G}, \mathcal{D})$ is a collection of graphs $\mathcal{G} = \{G^{(l)} = (V^{(l)}, E^{(l)})\}_{l \in V_L}$, paired with a non-negative family of *coupling* functions $\mathcal{D} = \{\Delta_{l,l'}\}_{l,l' \in V_L}$.

Definition 10. A *node-align MLN* is such that all layers have an identical set of nodes. An MLN has *diagonal couplings*, if for any two layers $l, l' \in V_L$, the coupling matrix $\mathbf{\Delta}_{l,l'}$ is diagonal.

Formally, multiplex networks are a special case of interconnected multilayer networks, which are *node-align* and have *diagonal couplings*. Notice that interconnected MLNs can be specified by a 4th order tensor.

B Proofs

B.1 Proof of Theorem 5

For some $\alpha > 0$, we start from $\bar{x}_i^{(k)} = \bar{x}_j^{(k)}$, because we have the same α on both sides we know that $(\bar{x}_i^{(k)})^{\frac{1}{\alpha}} = (\bar{x}_j^{(k)})^{\frac{1}{\alpha}}$ holds. From Definition 4 we know that $(\bar{x}_i^{(k)})^{\frac{1}{\alpha}} = f_1(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})_i$ and $(\bar{x}_j^{(k)})^{\frac{1}{\alpha}} = f_1(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})_j$ and therefore we know that $f_1(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})_i = f_1(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})_j$ holds. Therefore, we know that if we divide by the same quantity $\left\| \boldsymbol{f}_1(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)}) \right\|_1$ the following $\frac{f_1(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})_i}{\left\| \boldsymbol{f}_1(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)}) \right\|_1} = \frac{f_1(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})_j}{\left\| \boldsymbol{f}_1(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)}) \right\|_1}$ is true. We then apply Theorem 2 and we know that $\frac{f_1(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})_i}{\left\| f_1(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)}) \right\|_1} = g(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})_{1,i}$ and $\frac{f_1(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})_j}{\left\| \boldsymbol{f}_1(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)}) \right\|_1} = g(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})_{1,j}$ so $g(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})_{1,i} = g(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})_{1,j}$ holds. Finally, we apply Equation 1 and obtain that $x_i^{(k+1)} = g(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})_{1,i} = g(\boldsymbol{x}^{(k)}, \boldsymbol{t}^{(k)})_{1,j} = x_j^{(k+1)}$. The same reasoning applies to $\bar{t}_l^{(k)}$ and $\bar{t}_q^{(k)}$ for any pair of layers l, q.

B.2 Proof of Theorem 6

Let us assume that $\forall t \in \mathbb{R}_{\geq 0} \ . \ [\![\mathrm{x}_i]\!](t) = [\![\mathrm{x}_i]\!](t)$. Let us focus on those quantities. We know that, because the IDOL program p is differentiable, there exists $\delta > 0$ and functions $e_i(t)$, $e_j(t)$ such that $\lim_{t \to 0} e_i(t) = \lim_{t \to 0} e_j(t) = 0$ and

$$[\![\mathrm{x}_i]\!](t) = [\![\mathrm{x}_i]\!](0) + \left(\sum_{a=1}^{N} \sum_{l=1}^{L} A_{ial} [\![\mathrm{x}_a]\!](0) [\![\mathrm{t}_l]\!](0) \right) \cdot t + e_i(t)$$

$$[\![\mathrm{x}_j]\!](t) = [\![\mathrm{x}_j]\!](0) + \left(\sum_{a=1}^{N} \sum_{l=1}^{L} A_{jal} [\![\mathrm{x}_a]\!](0) [\![\mathrm{t}_l]\!](0) \right) \cdot t + e_j(t)$$

for all $0 \leq t < \delta$. We know that $[\![\mathtt{x}_i]\!](0) = [\![\mathtt{x}_j]\!](0)$. We also know that for all $a \in \{1, \ldots, N\}, l \in \{1, \ldots, L\}$. $[\![\mathtt{x}_a]\!](0)[\![\mathtt{t}_l]\!](0) = 1$ and therefore we know that

$$[\![\mathtt{x}_i]\!](t) = [\![\mathtt{x}_j]\!](t) \implies \sum_{a=1}^{N}\sum_{l=1}^{L} A_{ial} = \sum_{a=1}^{N}\sum_{l=1}^{L} A_{jal}$$

We also show the following equivalence that will prove useful in the latter part of the proof, where we know that there exists $\delta > 0$ and functions $e_i(h)$, $e_j(h)$ such that $\lim_{h\to 0} e_i(h) = \lim_{h\to 0} e_j(h) = 0$ and

$$[\![\mathtt{x}_i]\!](t+h) = [\![\mathtt{x}_i]\!](t) + \left(\sum_{a=1}^{N}\sum_{l=1}^{L} A_{ial}[\![\mathtt{x}_a]\!](t)[\![\mathtt{t}_l]\!](t)\right) \cdot h + e_i(h)$$
$$[\![\mathtt{x}_j]\!](t+h) = [\![\mathtt{x}_j]\!](t) + \left(\sum_{a=1}^{N}\sum_{l=1}^{L} A_{jal}[\![\mathtt{x}_a]\!](t)[\![\mathtt{t}_l]\!](t)\right) \cdot h + e_j(h)$$

We define $\mathtt{z}_{al}(t)$ as the following:

$$\mathtt{z}_{al}(t) = [\![\mathtt{x}_a]\!](t)[\![\mathtt{t}_l]\!](t)$$

Because $[\![\mathtt{x}_i]\!](t) = [\![\mathtt{x}_j]\!](t)$, h is the same in both equations and $\lim_{h\to 0} e_i(h) = \lim_{h\to 0} e_j(h) = 0$ we know that the following holds:

$$\forall \mathtt{z}_{al}(t) \geq 0 \,.\, \sum_{a=1}^{N}\sum_{l=1}^{L} A_{ial}\mathtt{z}_{al}(t) = \sum_{a=1}^{N}\sum_{l=1}^{L} A_{jal}\mathtt{z}_{al}(t) \tag{2}$$

We now proceed to show that $\forall k \in \mathbb{N}$. $x_i^{(k)} = x_j^{(k)}$. First, we need to explicitly state that, by definition of the iterative scheme presented in Theorem 2, this $x_i^{(0)} = x_j^{(0)} = 1$ holds. For all $k > 0$ we need to focus on the following:

$$x_i^{(k)} = \frac{(\bar{x}_i^{(k-1)})^{\frac{1}{\alpha}}}{\left\|\boldsymbol{f}_1(\boldsymbol{x}^{(k-1)}, \boldsymbol{t}^{(k-1)})\right\|_1}$$
$$x_j^{(k)} = \frac{(\bar{x}_j^{(k-1)})^{\frac{1}{\alpha}}}{\left\|\boldsymbol{f}_1(\boldsymbol{x}^{(k-1)}, \boldsymbol{t}^{(k-1)})\right\|_1}$$

From Theorem 5 we know that $\bar{x}_i^{(k-1)} = \bar{x}_j^{(k-1)} \implies x_i^{(k)} = x_j^{(k)}$. Therefore, it is sufficient to show that $\bar{x}_i^{(k-1)} = \bar{x}_j^{(k-1)}$ holds.

$$\begin{aligned}\bar{x}_i^{(k-1)} &= \sum_{a=1}^{N}\sum_{l=1}^{L} A_{ial} x_a^{(k-1)} t_l^{(k-1)} \\ \bar{x}_j^{(k-1)} &= \sum_{a=1}^{N}\sum_{l=1}^{L} A_{jal} x_a^{(k-1)} t_l^{(k-1)}\end{aligned} \tag{3}$$

If we define $z_{al}^{(k-1)}$ as:

$$z_{al}^{(k-1)} = x_a^{(k-1)} t_l^{(k-1)}$$

And when we plug its definition in Equation 3 we obtain:

$$\bar{x}_i^{(k-1)} = \sum_{a=1}^{N} \sum_{l=1}^{L} A_{ial} z_{al}^{(k-1)}$$
$$\bar{x}_j^{(k-1)} = \sum_{a=1}^{N} \sum_{l=1}^{L} A_{jal} z_{al}^{(k-1)}$$

From Equation 2 we know that for any quantity $z_{al}^{(k-1)}$ the following holds:

$$\sum_{a=1}^{N} \sum_{l=1}^{L} A_{ial} z_{al}^{(k-1)} = \sum_{a=1}^{N} \sum_{l=1}^{L} A_{jal} z_{al}^{(k-1)}$$

Thus, concluding the proof on the first component of the mapping. The same reasoning applies to prove that $[\![\mathtt{t}_l]\!](t) = [\![\mathtt{t}_q]\!](t) \Rightarrow t_l^{(k)} = t_q^{(k)}$.

C From a Multiplex to a CRN

Based on the work presented in [8] we present here the methodology to go directly from a multiplex MLN to a CRN that, under the assumption of mass-action kinetics, will have the underlying ODE system that we are interested in.

Definition 11. *Let $\mathcal{G} = \{G^{(l)} = (V_N, E^{(l)})\}_{l \in V_L}$ be a multiplex network and let $\mathcal{A} = (\mathcal{A}_{ijl})$ be its adjacency tensor. The set of species S is the set of all nodes and layers $S = \{x_1, \ldots, x_N, t_1, \ldots, t_L\}$. The set of reactions is inferred from the edges in the multiplex:*

$$\forall i, j \in \{1, \ldots, N\}, \forall l \in \{1, \ldots, L\} \,.\, A_{ijl} = w \;\Rightarrow\; x_i + t_l \to x_i + t_l + x_j, w \in R$$
$$\forall i < j \in \{1, \ldots, N\}, \forall l \in \{1, \ldots, L\} \,.\, A_{ijl} = w \;\Rightarrow\; x_i + x_j \to x_i + x_j + t_l, 2w \in R$$

Example 7. If we use the multiplex from the running example and we apply the transformation presented in Definition 11 we obtain the following CRN:

$$x_2 + t_1 \to x_2 + t_1 + x_1, 1 \qquad x_1 + t_1 \to x_1 + t_1 + x_3, 1$$
$$x_3 + t_1 \to x_3 + t_1 + x_1, 1 \qquad x_2 + t_2 \to x_2 + t_2 + x_3, 1$$
$$x_2 + t_2 \to x_2 + t_2 + x_1, 1 \qquad x_1 + x_2 \to x_1 + x_2 + t_1, 2$$
$$x_1 + t_1 \to x_1 + t_1 + x_2, 1 \qquad x_2 + t_2 \to x_1 + x_3 + t_1, 2$$
$$x_1 + t_2 \to x_1 + t_2 + x_2, 1 \qquad x_1 + x_2 \to x_1 + x_2 + t_2, 2$$
$$x_3 + t_2 \to x_3 + t_2 + x_2, 1 \qquad x_1 + x_3 \to x_2 + x_3 + t_2, 2$$

Table 1. Experimental results

Undirected Instances									
Instance	Nodes	Layers	Red. Nodes	Red. Layers	Centrality (s)	BDE Reduction (s)	Red. Centrality (s)	Reduction	Speed-up
Padgett-Florentine-Families	16	2	16	2	0.26	0.00	-	100 %	100 %
CS-Aarhus	61	5	61	5	0.07	0.01	-	100 %	100 %
London-transport	368	3	366	3	0.14	0.01	0.15	99 %	104 %
EUAirTransportation	450	37	374	37	0.67	0.03	0.55	84 %	83 %
pierreauger	514	16	351	16	1.28	0.18	0.52	69 %	41 %
arxiv-netscience	14488	13	8008	13	192.53	0.66	83.71	55 %	43 %
Directed instances									
Instance	Nodes	Layers	Red. Nodes	Red. Layers	Centrality (s)	BDE Reduction (s)	Red. Centrality (s)	Reduction	Speed-up
Krackhardt-High-Tech	21	3	21	3	0.05	0.00	-	100 %	100 %
Vickers-Chan-7thGraders	29	3	29	3	0.06	0.00	-	100 %	100 %
Kapferer-Tailor-Shop	39	4	39	4	0.07	0.01	-	100 %	100 %
Lazega-Law-Firm	71	3	71	3	0.14	0.02	-	100 %	100 %
hepatitusC-genetic	105	3	11	3	0.06	0.00	0.03	13 %	44 %
danioRerio-genetic	155	5	90	5	0.08	0.00	0.04	59 %	47 %
humanHerpes4-genetic	216	4	46	4	0.09	0.00	0.03	23 %	30 %
CKM-Physicians-Innovation	246	3	242	3	0.21	0.01	0.18	98 %	87 %
celegans-connectome	279	3	279	3	0.70	0.05	-	100 %	100 %
bos-genetic	325	4	162	4	0.12	0.01	0.09	50 %	76 %
candida-genetic	367	7	62	7	0.13	0.00	0.09	18 %	66 %
xenopus-genetic	461	5	276	5	0.18	0.01	0.08	60 %	42 %
humanHIV1-genetic	1005	5	137	5	0.51	0.02	0.13	14 %	25 %
plasmodium-genetic	1203	3	994	3	1.04	0.04	0.94	83 %	90 %
rattus-genetic	2640	6	1264	6	3.50	0.06	1.19	48 %	34 %
celegans-genetic	3879	6	2372	6	7.90	0.10	4.16	61 %	53 %
sacchpomb-genetic	4092	7	3613	7	53.71	0.75	59.47	88 %	111 %
sacchcere-genetic	6570	7	6087	7	494.21	10.14	428.68	93 %	87 %
arabidopsis-genetic	6980	7	4527	7	32.74	0.112	17.21	65 %	53 %
mus-genetic	7747	7	4887	7	35.99	0.30	19.58	63 %	54 %
drosophila-genetic	8215	7	7397	7	76.42	0.45	72.34	90 %	95 %
homo-genetic	18222	7	13978	7	747.52	1.969	562.42	77 %	75 %
fao-trade	214	364	214	364	49.17	0.50	-	100 %	100 %
MoscowAthletics2013	88804	3	37773	3	3359.26	6.68	1316.53	42 %	39 %
NYClimateMarch2014	102439	3	48018	3	6188.51	5.36	3296.76	47 %	53 %
MLKing2013	327707	3	63136	3	21289.98	2.51	1806.65	19 %	8 %
Cannes2013	438537	3	180443	3	>10 hr	12.05	>3 hr	69 %	