

# Production process networks: a trophic analysis

**Aurélien Hazan**

Université Paris-Est Créteil Val-de-Marne  
LISSI, IUTSF, F-77567 Lieusaint, France

E-mail: [aurelien.hazan@u-pec.fr](mailto:aurelien.hazan@u-pec.fr)

March 2022

**Abstract.** In this article, production process databases originating from environmental sciences, more specifically from Life Cycle Inventory (LCI), are considered as bipartite directed random networks. To model the observed directed hierarchical connection patterns, we turn to recent development concerning trophic coherence. Extending the scope to include bipartite networks, we compare several LCI networks to networks from other fields, and show empirically that they have high coherence and belong to the loopless regime, or close to its boundary.

## 1. Introduction

Production processes are central to organized societies and have thus been extensively studied in complex systems and complex networks literature from various viewpoints: interfirm supply-chain [1, 2, 3], trade [4, 5, 6], material flow analysis [7]. Recent works use block models to reconstruct an interfirm network [8, 9, 10, 11] from partial knowledge, or model manufacturing process below the factory level [12].

While they give deep insight into the structure of production processes, those works leave aside the technical details of production: inputs and outputs are usually aggregated, and detailed production processes are not considered. Furthermore, they neglect the ordering intrinsic to production processes and tend to focus on sectoral layers or communities [13].

In the field of Life-cycle Inventory (LCI), at the crossroads of engineering, environment sciences and economy, large and detailed databases of production processes have been built up for decades, and offer a more precise description. As explained in sec.2.2, two types of nodes are considered (processes and flows) and are typically used to perform Life Cycle Analysis, that is computing the emissions to the environment associated with the production of a unit of a specific product. LCI shares common methods with Input-Output Analysis [14], which uses monetary units and operates at the sector level.

In the LCI literature, a few articles consider complex networks tools to analyze production: in [15], the two-mode structure is projected onto a directed monopartite

network. Short mean path-length and power-law distribution of degrees are observed. This is explained by the authors by the existence of a hub such as utility sectors. The network is asymmetric ‡, and displays very low clustering and reciprocity. Furthermore, a comparison to networks corresponding to IO data is made and shows very different characteristics.

Similarly, several authors have analyzed IO datasets with network tools [16] or with some interest in topology [17], but those nets are highly aggregated, which leads to dense weighted networks where each sector seems connected to every other.

Conversely, little attention was paid to LCI databases in the complex networks community. While complex networks methods are not likely to improve performance of classical problems in LCI represented algebraically (see sec 2.2), they can help tackle problems more naturally expressed in a network way, and that so far remain unaddressed, for example: identifying central processes, probing the robustness to attacks of a given supply chain to avoid cascading failures, or examining dynamical phenomena such as propagation on the production network.

In the present article, we thus look at LCI databases from a network science standpoint and describe the properties of production networks at the process level, to propose new explanations for unexplained observations.

Firstly, the observed networks are directed: each process takes inputs and generates outputs. Then, in LCI databases, process nodes are connected to product (or “flow”) nodes, but are not directly connected, making the network bipartite or “two-mode”. (As explained in sec. 2.2 some LCI databases are built in a way such that they can be projected to monopartite networks, preserving some empirical properties).

Being directed networks, production processes share some of their properties. It has been remarked already in the literature [18, 19] that real directed networks were “underlooped”, that is they have less directed cycles than randomized versions of the network. Examples provided included a power grid, food webs, a metabolic network, the worldwide web, a neural network and a genetic transcription network.

In [20] the notion of inherent directionality is put forward, relying on a measure of hierarchical order inspired by trophic level [21]. A model is proposed and compared to empirical networks, and confirms the underlooped nature of empirical networks. As will be shown in sec. 3 it turns out that this property is also present in production processes depicted by LCI databases, which has not been reported so far, up to our knowledge.

The notion of hierarchical order was used in [2], and also in the random directed acyclic graphs (DAG) literature, which deals with random directed graphs without cycles. As explained in [22]: “it is the ordering of the vertices and not their acyclic structure that is the definitive property of the network. The acyclic structure is merely a corollary of the ordering”. Such ordering is domain-dependent and is often related to

‡ ‘the structure of the network is highly asymmetric. There are processes (transportation, electricity) that deliver to many other processes but that do not require inputs from these other processes. Conversely, there are processes (such as infrastructure-related processes) that have few customers, but that have many suppliers’ [15]

time ordering, as in citation networks. Production processes can be expected to display significant order since processes are partially ordered: some steps of the production process can be reversed, while others can't. For example in food production, plants first grow, then are processed, packed, shipped and sold in retail stores. Finding an underlying order in DAG requires solving the topological ordering problem, used in scheduling optimization. Strategies to randomize empirical DAG while preserving ordering were discussed for example in [23, 24] and rely on the topological ordering. As remarked in [24] the latter can be non-unique, which may bias the generated ensemble.

Furthermore, DAG random models do not allow the presence of a small number of cycles. Recent works explain low abundance of loops in directed networks by the notion of trophic coherence, which quantifies to what extent a network can be ordered into discrete levels [25, 26, 27, 28]. In these works, random graph ensembles are defined not from block-like assumptions but supposing that trophic coherence is constant, set to the empirically observed value. Some important properties of those random ensembles can be computed approximately and compared to empirical networks, for example the largest eigenvalue or the branching factor. Comparisons are made with life sciences (ecological food webs, genetic and metabolic), economic (international trade, input-output, supply-chains) [29, 30], linguistic and technical networks.

Nevertheless, trophic coherence theory concerns general unipartite directed networks, which disregards the particularities of bipartite networks encountered in LCI datasets. Conversely economic complexity measures such as country fitness and product complexity in [5] are adapted to bipartite undirected networks. We show below that some of the results of the trophic coherence theory can be extended to handle the bipartite case, providing the necessary correction to the bias induced by bipartivity. Then we analyze LCI datasets considered as networks, with a trophic coherence point of view and compare them to networks studied in other domains.

Section 2 introduces some elements of the theory as well as the datasets. Section 3 discusses our results and applications, sec. 4 concludes.

## 2. Methods and data

### 2.1. Trophic level and trophic coherence theory

In this section, we give a few definitions selected in the theory by Johnson and colleagues in [25, 27]. Let  $A$  be the adjacency of a graph  $G$  with the convention  $a_{ij} = 1$  if there is an edge§ from node  $j$  to node  $i$ , and  $k_i^{in} = \sum_j a_{ij}$  and  $k_i^{out} = \sum_j a_{ji}$  the in and out-degrees of node  $i$ .

A walk is a “sequence of nodes such that every consecutive pair of nodes in the sequence is connected by an edge” [31]. A directed walk respects the direction specified by the directed network. A path is a self-avoiding walk, that does not intersect itself.

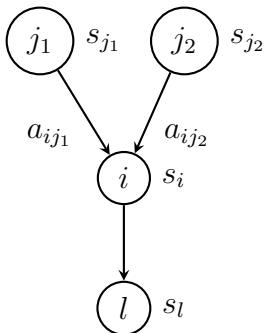
§ the opposite convention also exists in the literature.

The branching factor, whose role was evidenced in [18], is:

$$\alpha = \frac{\langle k^{in} k^{out} \rangle}{\langle k \rangle}. \quad (1)$$

In ecology [21], the notion of trophic level  $s_i$  of a node  $i$  was introduced to reveal the hierarchical ordering of species in food webs, as illustrated in Fig. 1:

$$s_i = 1 + \frac{1}{k_i^{in}} \sum_j a_{ij} s_j. \quad (2)$$



**Figure 1.** Trophic level of node  $i$ ,  $s_i$  has value  $s_i = 1 + \frac{1}{2}(s_{j_1} + s_{j_2})$ . The trophic difference  $x_{ij_1} = s_i - s_{j_1}$  has value  $1 + \frac{1}{2}(-s_{j_1} + s_{j_2})$ , which equals 1 if  $s_{j_1} = s_{j_2}$ .

$s_i$  is similar to the PageRank measure [32] defined for monopartite directed networks, with a normalization of the sum term by  $k_i^{in}$  instead of  $k_j^{out}$ . Thus  $s_i$  is also akin to PageRank-related measures such as countries' fitness and products' complexity, as discussed in Appendix E.

Equation (2) defines a set of linear equations that has a solution when each node belongs to a walk starting at a “basal” node, i.e. a node such that  $k_i^{in} = 0$ . Differences between trophic levels  $x_{ij} = s_i - s_j$  and their standard deviation  $q$ , named “trophic coherence”, were introduced in [25]:

$$q = std(x_{ij}) \quad (3)$$

In [27] the relationship between  $q$  and the propensity for a network to include loops was examined. The authors introduce the “coherence ensemble”, which is the set of random graphs with a specified number of nodes, degree sequence, and coherence  $q$ . They compute  $n_\nu$  the total number of walks of length  $\nu$  in  $G$ ,  $m_\nu$  the total number of cycles of length  $\nu$ , and  $c_\nu$  the expected proportion of walks of length  $\nu$  that are cycles, and their average values  $\bar{n}_\nu, \bar{m}_\nu, \bar{c}_\nu$  in the “coherence ensemble”. The authors show that the network ensemble belongs either to a “loopful” (resp. “loopless”) regime depending on a parameter  $\tau$  being positive (resp. negative), with the following definition:

$$\tau = \ln \alpha + \frac{1}{2\bar{q}^2} - \frac{1}{2q^2}, \quad (4)$$

where  $\alpha$  is the branching factor defined above, and  $\tilde{q}$  is the average value of coherence in the “basal ensemble” associated to  $G$ . This “basal ensemble” is a restriction of the directed configuration ensemble with the additional constraint that the proportion of in-neighbors connected to non-basal nodes is kept fixed with value  $k_i^{in} L_B/L$ , where  $L$  is the number of edges and  $L_B$  is the number of edges connected to basal nodes.

Notably, in the large  $\nu$  limit, the eigenvalue with leading real part noted  $\lambda_1$  can be related to  $\tau$ :

$$\bar{\lambda}_1 = e^\tau \quad (5)$$

with  $\bar{x}$  the average value of  $x$  in the ‘coherence ensemble’.

## 2.2. LCI, datasets and methods

In this section we discuss basic definitions and problems in the field of Life Cycle Inventory. Performing the LCI of a unit process consists in “the compilation and quantification of [its] inputs and outputs” [33]. Inputs and outputs are usually called flows and each unit process is associated to a reference flow, i.e. its main output. A product system is a collection of such processes and flows.

For example, production of electricity can be modeled as a unit process: in a simplified way, to produce 10 kWh of electricity it uses 2 litres of fuel, and outputs 1 kg of carbon dioxide  $CO_2$  and 0.1 kg of sulphur dioxide  $SO_2$  [34].

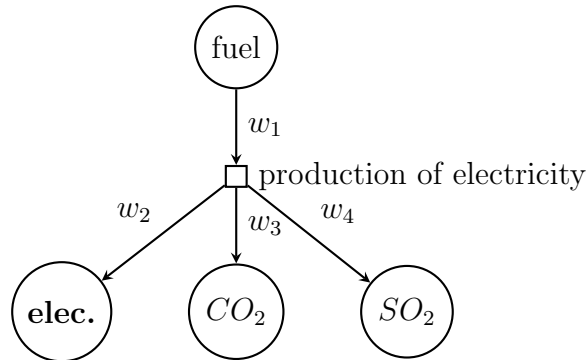
There are several ways to represent unit processes and product systems [15]. Firstly, a unit process can be represented as a process vector, for example  $\mathbf{p}_1 = (-2, 10, 1, 0.1)^T$  in the case of electricity production. The first dimension is associated to litres of fuel, the second one to kWh of electricity, etc. . . Negative values stand for inputs, and positive ones for outputs. A product system will then correspond to a set of  $m$ -dimensional vectors with  $m$  the number of flows, assembled in a rectangular matrix  $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_n]$ , with  $n$  the number of processes. Secondly, an equivalent graph representation of a unit process is given in Fig. 2. The graph is directed and bipartite, since processes are only connected to flows, and conversely. It is weighted by the amount of inputs and outputs necessary to produce a given quantity of reference output, but each weight has a specific unit (for examples liters, kg, kWh, etc. . .).

A few more definitions are needed: elementary flows go from a process to the environment or reversely, intermediate flows (like products and wastes) are generated by processes. Processes can take both elementary and intermediate flows as input and output.

With the matrix representation, scaling up all inputs and outputs of process 1 by factor  $s_1$ , of process 2 by factor  $s_2$ , etc. . . amounts to multiplying  $\mathbf{P}$  by a scale vector  $\mathbf{s} = [s_1, \dots, s_n]^T$ . This representation is relevant to address the inventory problem, i.e. finding the scale vector  $\mathbf{s}$  such that a demand flow  $\mathbf{f}$  is met, with  $\mathbf{f} \in \mathbb{R}^m$ . For example, if 100 kWh of electricity are necessary, then  $\mathbf{p}_1$  should be scaled by  $s_1 = 10$ . Then this unit process will output 10 kg of  $CO_2$  and 1 kg of  $SO_2$  to the environment. Similarly if a product system  $\mathbf{P}$  is scaled by vector  $\mathbf{s}$  for some reason (for example meeting the

demand in electricity, heat, and iron) then the total amount of environmental input and output flows can be readily computed. This constitutes Life Cycle Analysis (LCA), that is the assessment of the environmental impacts of a product system meeting a certain demand.

Performing LCI and LCA, requires adequate datasets, prepared by experts and made available to process engineers. To put it simply, such datasets are large  $\mathbf{P}$  matrices, with additional domain-specific metadata.



**Figure 2.** LCI example. Simplified bipartite graph representation of electricity production, from [34]. Reference flow is in bold case. Circles (resp. squares) represent flows (resp. processes).  $w_1 = 2L$  of fuel,  $w_2 = 10$  kWh of electricity,  $w_3 = 1$  kg of  $CO_2$ ,  $w_4 = 0.1$  kg of  $SO_2$ .

Then, we present briefly the common characteristics of LCI datasets and preprocessing steps required to build bipartite and monopartite graphs from them. Databases are selected because they’re free to download, and have the following description taken from [nexus.openlca.org](http://nexus.openlca.org):

- Agribalyse: “the French LCI database for the agriculture and food sector (...) comprises LCIs for 2 500 agricultural and food products produced and/or consumed in France”.
- ELCD: “(European reference Life Cycle Database) comprises Life Cycle Inventory (LCI) data from EU-level business associations and other sources for key materials, energy carriers, transport, and waste management”.
- Worldsteel: “This study contains global and regional LCI data for 16 steel products, from hot rolled coil to plate, rebar, sections, and coated steels”.
- Bioenergiedat: “Processes for bioenergy supply chains, with German background”.
- Ozlci: “The database inventory groups cover 958 [Australasian regional] supply chains”. It includes building products, chemicals, electric products, fabrics, farm and forest products, metal, minerals, as well as “utilities comprising use of freight, fuel, water and power by energy source and state grid”.

Some databases have a wide span, for example Agribalyse, which starts from the farm and goes to the distribution. Others have a narrower scope, for example Worldsteel.

In several databases, there is a convention that processes have a single product output, plus elementary flows. As in [15], we filter elementary flows since they do not connect vertices, which can result in processes having out-degree equal to one.

We do not perform monopartite projection because it loses information and can modify dramatically the networks’ properties [35]. In the case where process out-degree is equal to one, some properties (e.g. in/out degree) are conserved, but this is not the general case for other properties.

Additional preprocessing steps may be required, for example when modeling conventions result in spurious loops. This is documented for ELCD database in the case of transport $\parallel$ , and will be dealt with below by selective node removal.

### 2.3. Other datasets

Because of bipartivity, we can’t directly compare our results to those in [27]. For comparison purposes, a list of repositories of empirical networks was explored (see detailed list in Appendix D). Among all examined empirical networks tagged as directed and bipartite (2-mode), most have a trivial trophic structure. This is mostly because in those cases all edges from mode A to mode B have the same direction (for example in a plant/pollinator foodweb, all edges go from plants to pollinators). Those networks were thus excluded from our comparison basis. On the contrary, metabolic networks from the BiGG database [37] that represent chemical reactions on one hand and metabolites on the other, are bipartite and directed and don’t have a trivial trophic structure. Instead of considering the whole set of reactions and metabolites which would not correspond to a physiological phenomenon, we follow the standard practice in flux balance analysis (FBA) which consists in maximizing an objective function (for example biomass production), which can be provided with the BiGG dataset. Optimizing the objective function leads to turning off some reactions, and getting rid of metabolites not involved at the selected operating point. After this preprocessing step, a bipartite network can be built from the remaining reactions and metabolites. Further preprocessing is added optionally, for comparison: thermodynamically infeasible cycles are removed using CycleFreeFlux [38], whose authors define cycles as “sets of reactions that together carry a flux that does not influence on the exchange reactions of the model (...). These are metabolic ‘perpetual motion machines’ and do not occur in biological reality”.

Lastly we mention the topic of finite size random graph samplers, that usually serve as a point of comparison in network science. Two of them are of interest in our case. Firstly, sampling from the basal ensemble in sec. 2.1 can be realized approximately using an off-the-shelf edge-rewiring algorithm, using a specific network  $g_0$  as a seed (for example an empirically observed network, or a random graph with a degree sequence sampled from a specific distribution, such as the power law). Indeed, ensemble equivalence was exemplified in the monopartite case by [27, SI Appendix §1.3]

$\parallel$  e.g. ”cargo” appears as both an input and an output to transport processes [36].

with finite size Erdős-Rényi and scale-free random graphs. When network size increases, this leads to  $\langle q \rangle_{\text{samp}} = \tilde{q}$ , where  $\langle \cdot \rangle_{\text{samp}}$  is the average under random sampling. Thus  $\tau = \ln \alpha$ , from eq.(4). Hence  $\tau$  in the initial graph  $g_0$  is not preserved in general by such a random rewiring algorithm. The same can be observed in the bipartite case, with a rewiring algorithm that preserves both direction and bipartiteness, such as graph-tool [39]. Secondly, a sampler was proposed in [26] that allows to generate a random graph with a fixed value for  $q$ . But this sampler does not take a seed graph as an input, nor any degree sequence. Thus it does not allow to generate a random graph with specified  $q$  and degree sequence, which would allow to control  $\alpha$ , and  $\tau$ . Further, it has not been extended to the bipartite case. To conclude, random graph generators with a controllable degree sequence and trophic parameters at the same time do not exist in the literature so far, up to our knowledge, and this constitutes an interesting research direction.

### 3. Results

In this section, we adapt to the bipartite case the results of [27], partly summed-up in sec. 2.1, more particularly eq.(4-5).

#### 3.1. Bipartite directed configuration model

In this section, we express  $m_\nu$  and  $n_\nu$  in the bipartite case, more specifically their average value  $\hat{m}_\nu$  and  $\hat{n}_\nu$  in the directed configuration model. Following the convention in [27],  $\hat{m}_\nu$  does not concern simple cycles but includes multiple counts with different starting points in loops. Only the results are shown, and the full derivation is in Appendix A. Using the properties of adjacency matrices of bipartite networks, and the bipartite version of a directed configuration model, with an expected number of directed edges  $p_{ij} = \frac{k_i^{\text{in}} k_j^{\text{out}}}{L}$  from node  $j$  to node  $i$ , we put  $A$  in the form:

$$A = \begin{pmatrix} 0 & \mathbf{y}\mathbf{v}^T \\ \mathbf{u}\mathbf{x}^T & 0 \end{pmatrix} \quad (6)$$

and derive the average number of walks of length  $2\nu$ :

$$\hat{n}_{2\nu} = (\alpha_{xy}\alpha_{uv})^\nu \left( \frac{L_{xy}}{\alpha_{xy}} + \frac{L_{uv}}{\alpha_{uv}} \right) \quad (7)$$

with  $\mathbf{x}^T \mathbf{y} = \alpha_{xy}$ ,  $\mathbf{v}^T \mathbf{u} = \alpha_{uv}$ ,  $L_{xy} = \sum_{ij} y_i x_j$  and  $L_{uv} = \sum_{ij} u_i v_j$ . It can be compared to the expression of the monopartite directed configuration model in [27]:

$$\hat{n}_\nu^{\text{mono}} = L\alpha^{\nu-1} \quad (8)$$

Then the average number of cycles of length  $2\nu$  is:

$$\hat{m}_{2\nu} = 2(\alpha_{xy}\alpha_{uv})^\nu \quad (9)$$



which is the counterpart of the following expression in the monopartite case:

$$\hat{m}_\nu^{mono} = \alpha^\nu \quad (10)$$

Interestingly  $\hat{m}_{2\nu}$  is even: this is because it does not count the number of simple cycles but includes the same loops at different starting points, and each loop contains an even number of vertices.

Following [27] we finally define:

$$\hat{c}_{2\nu} = \frac{\hat{m}_{2\nu}}{\hat{n}_{2\nu}} = \frac{2}{\frac{L_{xy}}{\alpha_{xy}} + \frac{L_{uv}}{\alpha_{uv}}} \quad (11)$$

### 3.2. Basal ensemble

Adapting eq.(4) in the bipartite case requires an expression for  $\tilde{q}$ , the coherence in the basal ensemble. In short, we show that the expression found in [27]  $\tilde{q} = \sqrt{\frac{L}{L_B} - 1}$  is only slightly modified, and the leading term is still  $\sqrt{\frac{L}{L_B}}$  in the common situation in which  $L$  is much larger than  $L_B$ .

To see that, we keep the setting with  $N$  nodes,  $L$  edges and  $L_B$  basal edges, and add bipartiteness. Left nodes include  $N_b^L$  basal and  $N_{nb}^L$  non-basal nodes. Right nodes are exclusively non-basal and can connect only to left nodes, either basal or non-basal, as represented in Fig.3. Basal edges go only from left basal nodes to right non-basal nodes. Non-basal edges are established between left non-basal and right nodes, in both directions. For consistency, left non-basal nodes must have in-degree greater than 1, which writes:

$$\tilde{k}_{nb}^{in,L} = \frac{L - L_B}{2N_{nb}^L} \geq 1 \quad (12)$$

In the basal ensemble, the proportion of in-neighbors connected to right nodes is kept fixed with value  $k_i^{in} L_B / L$ . Right nodes receive a total of  $\frac{L-L_B}{2} + L_B = \frac{L+L_B}{2}$  edges. Noting  $\tilde{s}^R$  the average trophic level of right nodes, and  $\tilde{s}_b^L$  (resp.  $\tilde{s}_{nb}^L$ ) the average trophic level of left basal (resp. non-basal) nodes, we get from the definition in eq.(2):

$$\tilde{s}_b^L = 1 \quad (13)$$

$$\tilde{s}^R = 1 + \frac{2L_B}{L + L_B} + \frac{L - L_B}{L + L_B} \tilde{s}_{nb}^L \quad (14)$$

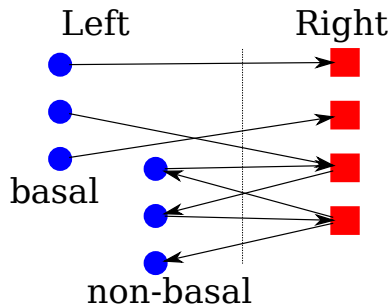
Similarly, non-basal left nodes receive edges from right nodes only and we get:

$$\tilde{s}_{nb}^L = 1 + \tilde{s}^R \quad (15)$$

This yields:

$$\tilde{s}^R = \frac{L}{L_B} + 1 \quad (16)$$

$$\tilde{s}_{nb}^L = \frac{L}{L_B} + 2 \quad (17)$$



**Figure 3.** Bipartite basal ensemble network.

We remark that  $\tilde{s}^R$  has the same value as  $\tilde{s}_{nb}$  in the monopartite case [27], which explains why  $\tilde{q}_b$  is to leading order close to  $\sqrt{\frac{L}{L_B}}$  when  $L$  is much larger than  $L_B$ . This is treated in detail in Appendix B, and compared to numerical simulations.

Lastly, we explored the possibility that the numbers of edges leaving non-basal left layer and leaving the right layer are unbalanced, to reflect what is observed empirically with the LCI dataset. We report that indeed this modulates measured values for  $q$ . However, we were not able to find a simple yet accurate enough model for the observed behavior. Therefore in first approximation, we keep the balanced model below.

### 3.3. Leading eigenvalue

From eq.(9) we have an expression for  $\hat{m}_{2\nu}$  the average number of cycles of length  $2\nu$  in the bipartite directed configuration model. In sec. 3.2 we noticed that  $\tilde{q}$  the coherence in the bipartite ensemble had approximately the same expression as in the monopartite case.

Following [27], and taking only loops with even length into account, we derive an expression similar to that in the monopartite case eq.(4-5)  $\lambda_1 = \max_i \{Re(\lambda_i)\}$ :

$$\overline{\lambda_1} = e^\tau \tag{18}$$

$$\tau = \log \sqrt{\alpha_{xy}\alpha_{uv}} + \frac{1}{2\tilde{q}^2} - \frac{1}{2q^2} \tag{19}$$

The full derivation can be found in Appendix C. As noted in sec. 3.1,  $\sqrt{\alpha_{xy}\alpha_{uv}}$  has values that can be directly related to the monopartite branching factor  $\alpha$  in certain particular cases, but not in general. As remarked in sec. 3.2 the bipartite definition  $\tilde{q}$  can be well approximated by the usual monopartite value, in the particular case of balanced left and right layers.

### 3.4. Application to empirical graphs

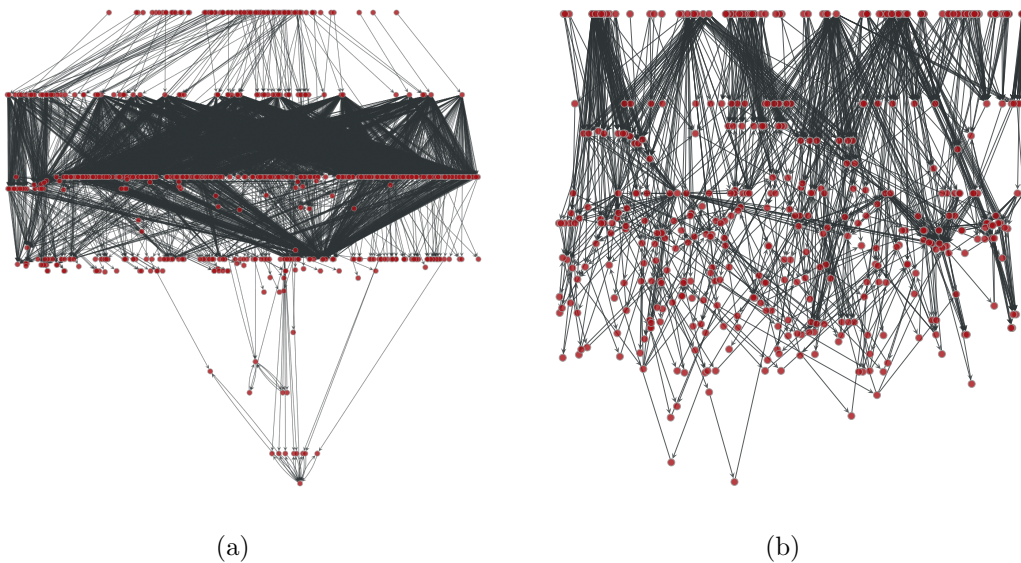
In this section, the datasets presented in sec.2.2-2.3 are analyzed using the tools depicted in sec. 3.1-3.3, adapted to the bipartite setting.

Tab. 1 shows that LCI networks are coherent, with an average  $q$  much lower than for other considered datasets, and are more likely in the loopless regime than other

|           | $q$  | $\tau$ |
|-----------|------|--------|
| LCI       | 0.37 | -1.5   |
| Metabolic | 10   | 1.8    |

**Table 1.** Bipartite network median characteristics, by dataset category.

datasets. This leaves room for fluctuation inside the LCI category. For example, the Ozlci dataset has a trivial structure, since it occupies only 2 layers, and hence  $q = 0$ . Fig. 4 presents two LCI networks embedded in 2d space<sup>¶</sup>, with differing behaviors: the ELCD network is very coherent ( $q = 0.21$ ), and more coherent than the corresponding average randomized networks ( $q/\tilde{q} = 0.03$ ). The Bioenergiedat network is less coherent with value  $q = 0.99$ , and  $q/\tilde{q}$  ratio just above 1.



**Figure 4.** LCI networks embedded in 2d space.  $y$ -axis represents trophic level with inverted axis, low  $s_i$  at the top.  $x$ -axis. (left) ELCD; (right) Bioenergiedat.

From the LCI literature, we expect a large number of feedback loops in the examined databases. Citing [34]: “Feedback loops occur frequently in industrial systems. For instance, mining of coal needs electricity, while production of electricity needs coal”. However, the majority of datasets are acyclic. Only Agribalyse and ELCD contain a small number of cycles compared to the edge number. In Agribalyse their occurrence is mostly related to seeds, which are both an input and output in plant growing. In ELCD, loops are associated with a few nodes, and filtering them is enough to remove all cycles, as explained in sec. 2.2.

This low number of loops has not been reported so far, up to our knowledge. It may arise from modeling bias, restricted scope, or from dataset selection bias. Also,

<sup>¶</sup> this is done using ForceAtlas2 [40]

|               | $N$   | $N_B$ | $\langle k \rangle$ | $q$  | $\tilde{q}$ | $\alpha$ | $q/\tilde{q}$ | $\tau$    | cycle | $\lambda_1$ |
|---------------|-------|-------|---------------------|------|-------------|----------|---------------|-----------|-------|-------------|
| Agribalyse    | 31698 | 1630  | 6.9                 | 1.28 | 8.15        | 2.46     | 0.16          | 0.60      | 117   | 1.52        |
| ELCD          | 894   | 104   | 11.5                | 0.21 | 6.72        | 3.42     | 0.03          | -9.91     | 62    | 3.00        |
| ELCD filtered | 883   | 99    | 11.4                | 0.14 | 6.80        | 3.29     | 0.02          | -23.46    | 0     | 0.03        |
| Worldsteel    | 63    | 5     | 12.7                | 0.53 | 1.32        | 0.99     | 0.40          | -1.51     | 0     | $8.8e^{-4}$ |
| Bioenergiedat | 457   | 112   | 5.1                 | 0.99 | 0.92        | 1.42     | 1.08          | 0.43      | 0     | 0.22        |
| Ozlc          | 1914  | 957   | 1.0                 | 0.00 |             |          |               | $-\infty$ | 0     | $1.9e^{-1}$ |

**Table 2.** Bipartite network trophic characteristics, LCI datasets. ‘Cycle’ is the number of unique elementary directed cycles, computed with graph-tool [39].

recurring loops challenge numerical solvers in Life Cycle Analysis not relying on matrix inversion. In the complex network literature, arguments of improved dynamic stability [18, 19] and transport [18] have been put forward to explain the lack of loops. In [20] the authors hypothesize that “instead the absence of feedback loops is a byproduct of a more inherent feature of networks: the existence of a preferred directionality”. In comparison, non-LCI networks introduced in sec.2.3, contain a large number of cycles, not reported in Tab. 3 because of prohibitive computational cost.

Also the consistency of formulas in sec.3.3, adapted from [27], that relate  $\tau$  and  $\lambda_1$ , can be discussed. In Fig. 5, the leading eigenvalue  $\lambda_1$  is plotted as a function of  $\tau$ , for both datasets (LCI and non-LCI). Circles representing non-LCI data are well fitted by the dashed curve. Crosses representing LCI datasets are close to the exponential, except for the outlier value at  $(\tau, \lambda_1) = (-10, 3)$  which represents the ELCD dataset, and seems inconsistent with the curve. After filtering as explained in sec. 2.2 it is mapped to  $(\tau, \lambda_1) = (-23, 0.03)$ , which gives a satisfactory fit. This case is reminiscent of a remark in [20]: “typically loops are not independent as they can share some nodes. In particular, hubs are statistically more likely than other nodes to take part in loops”. Indeed removing just a few nodes changed dramatically the behavior from a coherence point of view. However such cycle-removing is of course not relevant for acyclic datasets, while in Agribalyse cycles are scattered across the dataset rather than concentrated. This raises the question of the robustness of coherence measures and will be discussed in sec.4.

In sec.2.2 the topic of monopartite projection was evoked as it is easier to use off-the-shelf tools than to extend a theory to the bipartite case. This is the approach taken for example in [15], that may be justified by the particular nature of LCI databases, as noted in sec.2.2. Several numerical experiments were run to try to find patterns in the effect of monopartite projection. First, this imposes the constraint that basal nodes must all belong to the same layer, more specifically to the layer chosen for projection. This problem may be mitigated using the new definitions of coherence in [29]. In the easier case where  $k_{out} = 1$  for processes and projection is done on the flow layer, we were not able to evidence a predictable behavior for coherence quantities. Sometimes

|              | $N$ | $N_B$ | $\langle k \rangle$ | $q$   | $\tilde{q}$ | $\alpha$ | $q/\tilde{q}$ | $\tau$ | $\lambda_1$ |
|--------------|-----|-------|---------------------|-------|-------------|----------|---------------|--------|-------------|
| iJR904       | 657 | 13    | 4.72                | 11.65 | 10.87       | 5.97     | 1.07          | 1.79   | 5.87        |
| iJR904*      | 733 | 15    | 4.63                | 9.06  | 10.59       | 6.43     | 0.86          | 1.86   | 6.49        |
| iSB619       | 623 | 12    | 4.70                | 11.90 | 11.00       | 5.92     | 1.08          | 1.78   | 6.04        |
| iSB619*      | 658 | 17    | 4.70                | 8.46  | 9.48        | 6.27     | 0.89          | 1.83   | 6.54        |
| iAF692       | 744 | 10    | 4.54                | 16.11 | 12.96       | 6.18     | 1.24          | 1.82   | 5.56        |
| iAF692*      | 781 | 16    | 4.56                | 7.77  | 10.51       | 6.46     | 0.74          | 1.86   | 6.03        |
| iND750       | 718 | 8     | 4.45                | 8.16  | 13.28       | 5.23     | 0.61          | 1.65   | 5.95        |
| iND750*      | 824 | 20    | 4.42                | 10.70 | 9.49        | 5.53     | 1.13          | 1.71   | 6.47        |
| iYO844       | 700 | 12    | 4.78                | 11.82 | 11.77       | 6.30     | 1.00          | 1.84   | 6.30        |
| iYO844*      | 741 | 17    | 4.77                | 9.02  | 10.15       | 6.60     | 0.89          | 1.89   | 6.56        |
| iAB_RBC_283  | 168 | 12    | 2.95                | 3.75  | 4.43        | 2.32     | 0.85          | 0.83   | 3.10        |
| iAB_RBC_283* | 285 | 21    | 3.39                | 3.79  | 4.69        | 3.68     | 0.81          | 1.29   | 4.42        |
| iIT341       | 637 | 14    | 4.57                | 14.79 | 10.15       | 5.87     | 1.46          | 1.77   | 5.52        |
| iIT341*      | 643 | 15    | 4.61                | 14.19 | 9.89        | 5.98     | 1.43          | 1.79   | 5.66        |
| iNJ661       | 845 | 7     | 5.09                | 12.97 | 17.49       | 7.47     | 0.74          | 2.01   | 7.27        |
| iNJ661*      | 925 | 19    | 5.04                | 10.07 | 11.03       | 7.99     | 0.91          | 2.08   | 7.89        |

**Table 3.** Bipartite network trophic characteristics, other datasets. The number of cycles is not reported for this dataset because it is very high, and too time-consuming to compute exhaustively. Asterisks identify CycleFreeFlux preprocessing as explained in sec. 2.3

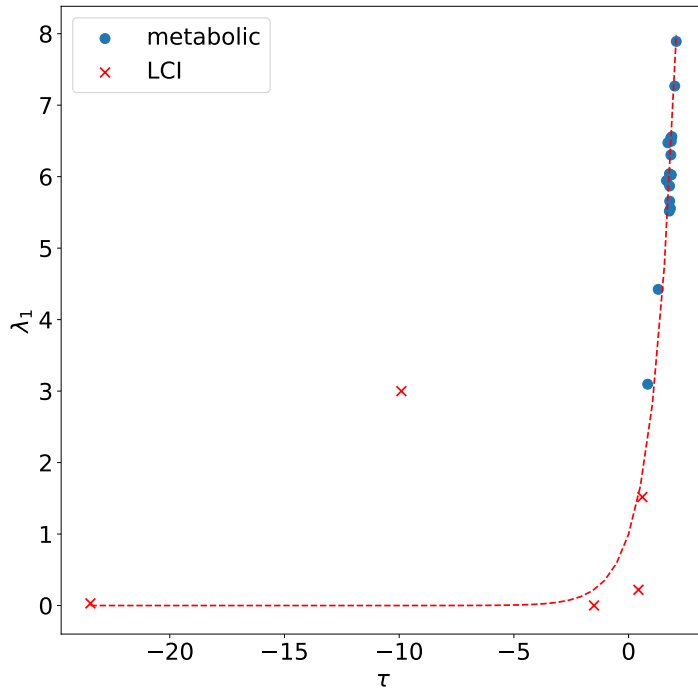
the projected network has only 2 trophic values, which results in  $q_{mono} = 0$ , although  $q$  was nonzero in the bipartite network. The case where  $k_{out}$  can take any value for processes is even harder to deal with.

#### 4. Conclusion

Starting from the observation that little work had been devoted to production processes at the fine-grain level depicted by LCI in the complex networks community, we proposed a first original contribution in that direction. First, unlike earlier works, we proposed i) to keep the bipartite structure to avoid loss of information, ii) to look for a random model, iii) able to reproduce hierarchical features of the datasets.

This was done by building on existing theory by Johnson and colleagues, upon extending some of their tools to the bipartite case. We report that:

- the studied empirical networks built from LCI databases have high coherence compared to other existing datasets and low loop number.
- the random “coherence ensemble” satisfactorily reproduces an important property of empirical dataset (the largest real part of the set of eigenvalues), which is closely related to other important aspects such as behavior of dynamical systems defined on networks as shown by several contributions in the literature.



**Figure 5.** Leading eigenvalue  $\lambda_1$  as a function of  $\tau$ ,  $\lambda_1 = f(\tau)$  in the bipartite case. The outlier value at  $(\tau, \lambda_1) = (-10, 3)$  represents the ELCD dataset before filtering. After filtering it is mapped to  $(\tau, \lambda_1) = (-23, 0.03)$ .

In further work we plan to extend the number of studied LCI databases, to consider how these observations can extend to other classical properties (such as clustering, diameter, ...). Unbalanced bipartite basal model will be explored, as well as potential useful applications for LCI that can be derived from those findings. Furthermore, the new definitions of trophic coherence in [29] will be tested in the particular case of LCI networks.

### Appendix A. Loop count in the bipartite directed configuration model

Two properties of adjacency matrices of bipartite networks are used below. Let  $A$  be the adjacency matrix of a directed bipartite network  $G$ , and  $B_l, B_r$  the associated biadjacency matrices:

$$A = \begin{pmatrix} 0 & B_r \\ B_l & 0 \end{pmatrix}$$

Following [41], we notice that even and odd powers of  $A$  have different expressions.

Since odd cycles are absent in a bipartite network, we focus on even powers of  $A$ :

$$A^{2\nu} = \begin{pmatrix} (B_r B_l)^\nu & 0 \\ 0 & (B_l B_r)^\nu \end{pmatrix} \quad (\text{A.1})$$

Further, in the monopartite directed configuration model, the expected number of directed edges from node  $j$  to node  $i$  has the following form:

$$p_{ij} = \frac{k_i^{\text{in}} k_j^{\text{out}}}{L} \quad (\text{A.2})$$

Being rank 1 matrices, the biadjacency matrices  $B_l$  and  $B_r$  can be written as outer products of vectors:  $B_l = \mathbf{u}\mathbf{x}^T$  and  $B_r = \mathbf{y}\mathbf{v}^T$ .

$A$  and its even powers can thus be written:

$$A = \begin{pmatrix} 0 & \mathbf{y}\mathbf{v}^T \\ \mathbf{u}\mathbf{x}^T & 0 \end{pmatrix} \quad (\text{A.3})$$

$$A^{2\nu} = \begin{pmatrix} (\mathbf{y}\mathbf{v}^T \mathbf{u}\mathbf{x}^T)^\nu & 0 \\ 0 & (\mathbf{u}\mathbf{x}^T \mathbf{y}\mathbf{v}^T)^\nu \end{pmatrix} \quad (\text{A.4})$$

Summing all terms in eq.(A.4) we get the total number of walks of length  $2\nu$  :

$$\begin{aligned} n_{2\nu} &= (\mathbf{v}^T \mathbf{u})^\nu (\mathbf{x}^T \mathbf{y})^\nu \left( \frac{\sum_{ij} y_i x_j}{\mathbf{x}^T \mathbf{y}} + \frac{\sum_{ij} u_i v_j}{\mathbf{v}^T \mathbf{u}} \right) \\ &= (\alpha_{xy} \alpha_{uv})^\nu \left( \frac{L_{xy}}{\alpha_{xy}} + \frac{L_{uv}}{\alpha_{uv}} \right) \end{aligned} \quad (\text{A.5})$$

with  $\mathbf{x}^T \mathbf{y} = \alpha_{xy}$ ,  $\mathbf{v}^T \mathbf{u} = \alpha_{uv}$ ,  $L_{xy} = \sum_{ij} y_i x_j$  and  $L_{uv} = \sum_{ij} u_i v_j$ .

Summing all diagonal terms in eq.(A.4) we get the total number of cycles of length  $2\nu$ :

$$\begin{aligned} m_{2\nu} &= (\mathbf{v}^T \mathbf{u})^\nu (\mathbf{x}^T \mathbf{y})^\nu \left( \frac{\sum_i y_i x_i}{\mathbf{x}^T \mathbf{y}} + \frac{\sum_i u_i v_i}{\mathbf{v}^T \mathbf{u}} \right) \\ &= 2(\alpha_{xy} \alpha_{uv})^\nu \end{aligned} \quad (\text{A.6})$$

## Appendix B. Basal ensemble

In this section, we compute  $\tilde{q}$  in the bipartite case. From sec. 3.2 we have:

$$\begin{aligned} \tilde{s}^R &= \frac{L}{L_B} + 1 \\ \tilde{s}_{nb}^L &= \frac{L}{L_B} + 2 \end{aligned}$$

Three types of edges will be observed in the bipartite basal ensemble:

- from basal left nodes to right nodes: there are  $L_B$  such edges, with  $x_{ij} = s_{nb}^R - 1 = \frac{L}{L_B}$ .
- from non-basal left nodes to right nodes: there are  $\frac{L-L_B}{2}$  such edges, with  $x_{ij} = -1$
- from right nodes to non-basal left nodes: there are  $\frac{L-L_B}{2}$  such edges, with  $x_{ij} = 1$

From those values, the variance  $\tilde{q}^2$  is:

$$\tilde{q}^2 = \frac{L_B}{L} \left( \frac{L}{L_B} - 1 \right)^2 + \frac{L - L_B}{2L} \left( -1 - 1 \right)^2 \quad (\text{B.1})$$

Which yields:

$$\tilde{q} = \sqrt{\frac{L}{L_B} - \frac{L_B}{L}} \quad (\text{B.2})$$

### Appendix C. Leading eigenvalue

We replicate the steps in [27] to get an expression for  $\tau$  in the bipartite case. In the coherence ensemble with coherence  $q$ , the sum  $S = \sum_k x_k$  along a cycle is equal to zero. Modeling the  $x_k$  as random variables, the authors notice that  $S$  has approximately a Gaussian distribution, and  $Pr(S = 0)$  is proportional to  $\bar{c}_\nu$ . In the bipartite case we consider closed walks with length  $2\nu$ , since odd cycles are not allowed. The random variable along such a walks  $S$  has mean  $2\nu$  and variance  $2\nu q^2$ . It follows that:

$$\bar{c}_{2\nu} = B_{2\nu} \frac{1}{\sqrt{q\nu}} \exp\left(\frac{-\nu}{q^2}\right) \quad (\text{C.1})$$

where  $B_{2\nu}$  is unknown. Taking the particular case of the basal ensemble, and supposing as in [27] that  $\hat{c}_{2\nu} = \bar{c}_{2\nu}$  we get:

$$\bar{c}_{2\nu} = \hat{c}_{2\nu} \frac{\tilde{q}}{q} \exp\left(\nu \left(\frac{1}{\tilde{q}^2} - \frac{1}{q^2}\right)\right) \quad (\text{C.2})$$

Since  $\bar{c}_{2\nu} = \frac{\bar{m}_{2\nu}}{\bar{n}_{2\nu}}$  and supposing that  $\hat{n}_{2\nu} \approx \bar{n}_{2\nu}$  we have:

$$\begin{aligned} \bar{m}_{2\nu} &= 2(\alpha_{xy}\alpha_{uv})^\nu \frac{\tilde{q}}{q} \exp\left(\nu \left(\frac{1}{\tilde{q}^2} - \frac{1}{q^2}\right)\right) \\ &= 2 \frac{\tilde{q}}{q} \exp\left(2\nu \left(\log \sqrt{\alpha_{xy}\alpha_{uv}} + \frac{1}{2\tilde{q}^2} - \frac{1}{2q^2}\right)\right) = 2 \frac{\tilde{q}}{q} \exp(2\nu\tau) \end{aligned} \quad (\text{C.3})$$

with  $\tau$  defined as in eq.(19).

In parallel the trace of the  $2n$ -th power of the adjacency matrix still can be expressed:

$$Tr(A^{2\nu}) = \sum_i \lambda_i^{2\nu} = m_{2\nu} \quad (\text{C.4})$$

Taking the expectation in the coherence ensemble:

$$\bar{m}_{2\nu} = \sum_i \bar{\lambda}_i^{2\nu}$$

Taking to the power  $1/\nu$  then the large  $\nu$  limit as in [27]:

$$\lim_{+\infty} \left( \sum_i \bar{\lambda}_i^{2\nu} \right)^{\frac{1}{\nu}} = \bar{\lambda}_1^2 = \exp(2\tau) \quad (\text{C.5})$$

And the same form  $\bar{\lambda}_1 = \exp(\tau)$  as in the monopartite case is recovered.



## Appendix D. Data and code availability

The repositories evoked in sec.2.3 are the Stanford Large Network Dataset Collection [42], ICON [43], and Netzschleuder [44]. They were selected because of their coverage and ability to filter by directedness and bipartiteness. The BiGG database [37] was also used for metabolic networks.

Several software packages were used in this work: Python, graph-tool [39], Igraph [45], Scikit-network [46], NetworkX [47], Cobrapy [48].

LCI datasets are freely available from nexus.openlca.org.

Code will we made available on gitlab.

## Appendix E. Comparison between trophic level and economic complexity measure

In this section we propose a quick comparison between trophic level in eq.(2) and economic complexity measures introduced for example in [5].

Suppose the starting point is a country/product network, bipartite and undirected. Then economic complexity measures such as country fitness or product complexity [5] can be computed. The trophic levels, however, are not defined because there are no basal nodes in the corresponding network (see sec. 2.1. Also note that another definition of trophic levels that does not require basal nodes was proposed in [29]). To sum up, in that case,  $s_i$  and measures of economic complexity can't be compared.

Conversely, if the starting point is a bipartite directed network with basal nodes, then the trophic levels are defined and can be computed. Further, upon transforming this network into an undirected one (which discards important information), economic complexity measures can also be computed.

For example we focus on simple directed bipartite 3-motifs in see Fig.E1. First we compute the trophic levels  $s_i$  using eq.(2) and write them next to each node, with basal node trophic levels set to 1.

Then, following [5], we compute  $F_c$  the fitness of a country  $c$  and the complexity  $Q_p$  of a product  $p$ , as indicated by the following equations:

$$\begin{cases} \tilde{F}_c^{(n)} = \sum_{p'} M_{cp'} Q_{p'}^{(n-1)} & \text{with } 1 \leq c \leq \mathcal{C} \\ \tilde{Q}_p^{(n)} = \left( \sum_{c'} M_{c'p} / F_{c'}^{(n-1)} \right)^{-1} & \text{with } 1 \leq p \leq \mathcal{P}, \end{cases} \quad (\text{E.1})$$

with  $\mathcal{C}$  the total number of countries and  $\mathcal{P}$  the total number of products. Moreover:

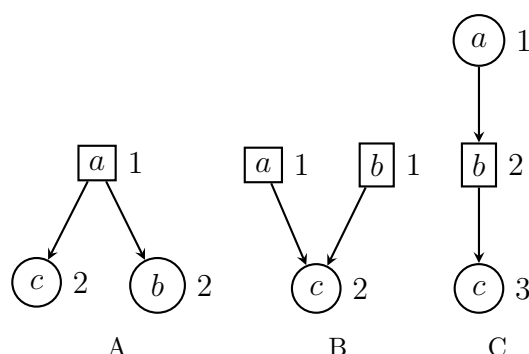
$$\begin{cases} F_c^{(n)} = \frac{\tilde{F}_c^{(n)}}{\langle \tilde{F}_c^{(n)} \rangle_c} \\ Q_p^{(n)} = \frac{\tilde{Q}_p^{(n)}}{\langle \tilde{Q}_p^{(n)} \rangle_p} \end{cases} \quad (\text{E.2})$$

with initial values  $\tilde{F}_c^{(0)} = \tilde{Q}_p^{(0)} = 1, \forall c, p$ .

We notice that upon transforming the directed bipartite motifs into undirected networks, motifs A and C are mapped to the same motif. Then the recursions for motifs A and B lead to a fixed point that can be easily found by hand:  $F_c^{(\infty)} = 1, Q_p^{(\infty)} = 1$ .

To summarize, in this simple case it appears that from the point of view of the economic complexity measures, the three motifs look the same, whereas the trophic levels preserve hierarchicalness.

A broader range of behaviors is expected if  $s_a$  and  $s_b$  are distinct. Also, a similar study could be conducted with 4-motifs, but this is left for further works.



**Figure E1.** Example of directed bipartite 3-motifs, trophic level  $s_i$  is given next to each node.

- [1] T. Watanabe, I. Uesugi, A. Ono (eds.), *The Economics of Interfirm Networks, Advances in Japanese Business and Economics*, vol. 4 (Springer Japan, Tokyo, 2015). DOI 10.1007/978-4-431-55390-8. URL <http://link.springer.com/10.1007/978-4-431-55390-8>
- [2] E. Letizia, F. Lillo, EPJ Data Science **8**(1), 21 (2019). DOI 10.1140/epjds/s13688-019-0197-5. URL <https://epjdatascience.springeropen.com/articles/10.1140/epjds/s13688-019-0197-5>
- [3] C. Diem, A. Borsos, T. Reisch, J. Kertész, S. Thurner, Available at SSRN 3826514 (2021)
- [4] C.A. Hidalgo, R. Hausmann, Proceedings of the National Academy of Sciences **106**(26), 10570 (2009). DOI 10.1073/pnas.0900943106. URL <https://pnas.org/doi/full/10.1073/pnas.0900943106>
- [5] A. Tacchella, M. Cristelli, G. Caldarelli, A. Gabrielli, L. Pietronero, Scientific Reports **2**(1), 723 (2012). DOI 10.1038/srep00723. URL <http://www.nature.com/articles/srep00723>
- [6] T. Squartini, G. Fagiolo, D. Garlaschelli, Physical Review E **84**(4), 046117 (2011). DOI 10.1103/PhysRevE.84.046117. URL <https://link.aps.org/doi/10.1103/PhysRevE.84.046117>
- [7] P. Nuss, W.Q. Chen, H. Ohno, T.E. Graedel, Environmental Science & Technology **50**(7), 4091 (2016). DOI 10.1021/acs.est.5b05094. URL <https://pubs.acs.org/doi/10.1021/acs.est.5b05094>
- [8] S. Hooijmaaijers, G. Buiten, A methodology for estimating the dutch interfirm trade network, including a breakdown by commodity. Tech. rep., Technical report, Statistics Netherlands (2019)
- [9] A. Rachkov, F. Pijpers, D. Garlaschelli, Potential Biases in Network Reconstruction Methods Not Maximizing Entropy. Tech. rep., Statistics Netherlands (2020)
- [10] C.E. Mattsson, F.W. Takes, E.M. Heemskerk, C. Diks, G. Buiten, A. Faber, P.M. Sloot, Frontiers in big Data **4** (2021). Publisher: Frontiers Media SA
- [11] A. Hazan, Physica A: Statistical Mechanics and its Applications **519**, 1 (2019). DOI <https://doi.org/10.1016/j.physa.2018.12.020>. URL <https://www.sciencedirect.com/science/article/pii/S0378437118315322>
- [12] T. Funke, T. Becker, Journal of Manufacturing Systems **56**, 296 (2020). DOI 10.1016/j.jmsy.2020.06.015. URL <https://linkinghub.elsevier.com/retrieve/pii/S0278612520301035>

- [13] M. Barigozzi, G. Fagiolo, G. Mangioni, *Physica A: Statistical Mechanics and its Applications* **390**(11), 2051 (2011). DOI 10.1016/j.physa.2011.02.004. URL <https://linkinghub.elsevier.com/retrieve/pii/S0378437111001129>
- [14] R.E. Miller, P.D. Blair, *Input-output analysis: foundations and extensions*, 2nd edn. (Cambridge University Press, Cambridge [England] ; New York, 2009). OCLC: ocn311310287
- [15] R. Heijungs, *Journal of Environmental Accounting and Management* **3**, 151 (2015). DOI 10.5890/JEAM.2015.06.005
- [16] F. Cerina, Z. Zhu, A. Chessa, M. Riccaboni, *PLOS ONE* **10**(7), e0134025 (2015). DOI 10.1371/journal.pone.0134025. URL <https://dx.plos.org/10.1371/journal.pone.0134025>
- [17] R. Wood, M. Lenzen, *Journal of Industrial Ecology* **13**(2), 264 (2009). DOI 10.1111/j.1530-9290.2009.00113.x. URL <https://onlinelibrary.wiley.com/doi/10.1111/j.1530-9290.2009.00113.x>
- [18] G. Bianconi, N. Gulbahce, A.E. Motter, *Phys. Rev. Lett.* **100** (2008). DOI 10.1103/physrevlett.100.118701. URL <https://doi.org/10.1103/physrevlett.100.118701>
- [19] A. Ma'ayan, G.A. Cecchi, J. Wagner, A.R. Rao, R. Iyengar, G. Stolovitzky, *Proceedings of the National Academy of Sciences* **105**(49), 19235 (2008). DOI 10.1073/pnas.0805344105. URL <https://pnas.org/doi/full/10.1073/pnas.0805344105>
- [20] V. Domínguez-García, S. Pigolotti, M.A. Muñoz, *Sci. Rep.* **4**, 7497 (2014). DOI 10.1038/srep07497. URL <https://doi.org/10.1038/srep07497>
- [21] S. Levine, *J. Theor. Biol.* **83**, 195 (1980). DOI 10.1016/0022-5193(80)90288-x. URL [https://doi.org/10.1016/0022-5193\(80\)90288-x](https://doi.org/10.1016/0022-5193(80)90288-x)
- [22] B. Karrer, M.E.J. Newman, *Physical Review E* **80**(4), 046110 (2009). DOI 10.1103/PhysRevE.80.046110. URL <https://link.aps.org/doi/10.1103/PhysRevE.80.046110>
- [23] J. Goñi, B. Corominas-Murtra, R.V. Solé, C. Rodríguez-Caso, *Physical Review E* **82**(6), 066115 (2010). DOI 10.1103/PhysRevE.82.066115. URL <https://link.aps.org/doi/10.1103/PhysRevE.82.066115>
- [24] C.J. Carstens, *Journal of Complex Networks* **2**(4), 419 (2014). DOI 10.1093/comnet/cnu028. URL <https://academic.oup.com/comnet/article-lookup/doi/10.1093/comnet/cnu028>
- [25] S. Johnson, V. Domínguez-García, L. Donetti, M.A. Muñoz, *Proc. Natl Acad. Sci. USA* **111**, 17923 (2014). DOI 10.1073/pnas.1409077111. URL <https://doi.org/10.1073/pnas.1409077111>
- [26] J. Klaise, S. Johnson, *Chaos* **26** (2016). DOI 10.1063/1.4953160. URL <https://doi.org/10.1063/1.4953160>
- [27] S. Johnson, N.S. Jones, *Proc. Natl Acad. Sci. USA* **114**, 5618 (2017). DOI 10.1073/pnas.1613786114. URL <https://doi.org/10.1073/pnas.1613786114>
- [28] S. Johnson, *Journal of Physics: Complexity* **1**(1), 015003 (2020). DOI 10.1088/2632-072x/ab8e2f. URL <https://doi.org/10.1088/2632-072x/ab8e2f>. Publisher: IOP Publishing
- [29] R.S. MacKay, S. Johnson, B. Sansom, *Royal Society open science* **7**(9), 201138 (2020). Publisher: The Royal Society
- [30] B. Sansom, S. Johnson, R. MacKay, *Trophic incoherence drives systemic risk in financial exposure networks*. Tech. rep., Working Paper (2021)
- [31] M. Newman, *Networks* (Oxford university press, 2018)
- [32] S. Brin, L. Page, *Computer networks and ISDN systems* **30**(1-7), 107 (1998). Publisher: Elsevier
- [33] ISO14040, *Environmental management — Life cycle assessment — Principles and framework* (CEN, Brussels, Belgium, 2006). URL <https://www.iso.org/obp/ui/#iso:std:iso:14040:ed-2:v1:en>
- [34] R. Heijungs, S. Suh, *The computational structure of life cycle assessment* (Springer, Dordrecht; London, 2011). OCLC: 751543198
- [35] M. Latapy, C. Magnien, N. Del Vecchio, (2006). DOI 10.48550/ARXIV.COND-MAT/0611631. URL <https://arxiv.org/abs/cond-mat/0611631>. Publisher: arXiv Version Number: 1
- [36] Rodriguez, *ELCD 3.2 in openLCA*. Tech. rep., OpenLCA (2016). URL <https://nexus.openlca.org/ws/files/9566>

- [37] Z.A. King, J. Lu, A. Dräger, P. Miller, S. Federowicz, J.A. Lerman, A. Ebrahim, B.O. Palsson, N.E. Lewis, *Nucleic Acids Research* **44**(D1), D515 (2015). DOI 10.1093/nar/gkv1049. URL <https://doi.org/10.1093/nar/gkv1049>. eprint: <https://academic.oup.com/nar/article-pdf/44/D1/D515/16661243/gkv1049.pdf>
- [38] A.A. Desouki, F. Jarre, G. Gelius-Dietrich, M.J. Lercher, *Bioinformatics* **31**(13), 2159 (2015). DOI 10.1093/bioinformatics/btv096. URL <https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btv096>
- [39] T.P. Peixoto, figshare (2014). DOI 10.6084/m9.figshare.1164194. URL [http://figshare.com/articles/graph\\_tool/1164194](http://figshare.com/articles/graph_tool/1164194)
- [40] M. Jacomy, T. Venturini, S. Heymann, M. Bastian, *PLoS ONE* **9**(6), e98679 (2014). DOI 10.1371/journal.pone.0098679. URL <https://dx.plos.org/10.1371/journal.pone.0098679>
- [41] J. Kunegis, *Internet Mathematics* **11**(3), 201 (2015). Publisher: Taylor & Francis
- [42] J. Leskovec, A. Krevl. SNAP Datasets: Stanford Large Network Dataset Collection (2014). URL <http://snap.stanford.edu/data>
- [43] Clauset, Tucker, Sainz. The Colorado Index of Complex Networks (2016). URL <https://icon.colorado.edu/>
- [44] T.P. Peixoto. The Netzschleuder network catalogue and repository (2020). URL <https://networks.skewed.de/>
- [45] G. Csardi, T. Nepusz, others, *InterJournal, complex systems* **1695**(5), 1 (2006)
- [46] T. Bonald, N.d. Lara, Q. Lutz, B. Charpentier, *Journal of Machine Learning Research* **21**(185), 1 (2020). URL <http://jmlr.org/papers/v21/20-412.html>
- [47] D.A. Schult, in *In Proceedings of the 7th Python in Science Conference (SciPy)* (Citeseer, 2008)
- [48] A. Ebrahim, J.A. Lerman, B.O. Palsson, D.R. Hyduke, *BMC systems biology* **7**(1), 1 (2013). Publisher: Springer