Mathematical Analysis of a Network's Asymptotic Behaviour Based on its Strongly Connected Components

Treur, Jan

published in
Complex Networks and Their Applications VII
2019

DOI (link to publisher)
10.1007/978-3-030-05411-3_53

document version
Publisher's PDF, also known as Version of record

document license
Article 25fa Dutch Copyright Act

citation for published version (APA)
Treur, J. (2019). Mathematical Analysis of a Network's Asymptotic Behaviour Based on its Strongly Connected Components. In L. M. Ailello, C. Cherifi, H. Cherifi, R. Lambiotte, P. Lió, & L. M. Rocha (Eds.), *Complex Networks and Their Applications VII: Volume 1 Proceedings The 7th International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2018* (Vol. 1, pp. 663-679). (Studies in Computational Intelligence; Vol. 812). Springer - Verlag. https://doi.org/10.1007/978-3-030-05411-3_53

Mathematical Analysis of a Network's Asymptotic Behaviour Based on Its Strongly Connected Components

Jan Treur(✉)

Behavioural Informatics Group, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
j.treur@vu.nl

Abstract. In this paper a general theorem is presented that relates asymptotic behaviour of a network to the network's characteristics concerning the network's strongly connected components and their mutual connections. The theorem generalises existing theorems for specific cases such as acyclic networks, fully and strongly connected networks, and theorems addressing only linear functions.

1 Introduction

This paper analyses the relation between network structure and emerging asymptotic diffusion behaviour of a network. In many cases this relation is only studied by performing simulation experiments. However, as shown in this paper, within a certain context it is also possible to analyse mathematically how certain asymptotic behaviours relate to certain properties of the network structure. In [12] this question was only addressed for two special cases: the case of an acyclic network, and the case of a strongly connected network; the general case remained open. The current paper develops a mathematical analysis for the general case. To achieve that, tools were adopted from the area of Graph Theory, in particular the manner to identify the connectivity structure within a graph by decomposition of the graph according to its (maximal) strongly connected components and the resulting (acyclic) condensation graph [6], Chap. 3, and the notion of stratification of an acyclic directed graph; e.g., [3].

In addition to the connectivity structure, the main theorem also takes into account the combination functions by which the impacts from multiple incoming connections are aggregated. It applies not to just one type (for example, linear or scaled sum functions), but to a whole class of functions: those that are characterised as being monotonic, scalar-free and normalised. This class does include the often used linear functions, but also, for example, *n*th order Euclidean combination functions involving squares or higher powers of values in the aggregation. Moreover, it explains which exactly are the relevant characteristics that make that these functions contribute to certain asymptotic behaviour. It will be shown how using the above mentioned tools from Graph Theory, together with the characteristics of combination functions mentioned, enable to address the general case and obtain a main theorem about it.

To apply this main theorem to a given network, first the decomposition of the network into its strongly connected components is determined. A variety of efficient

L. M. Aiello et al. (Eds.): COMPLEX NETWORKS 2018, SCI 812, pp. 663–679, 2019.
https://doi.org/10.1007/978-3-030-05411-3_53

algorithms are available to determine these strongly connected components; for example, see [1, 4, 5, 8, 9, 13]. Next, the connections between these components are identified, as shown in an acyclic condensation graph, and a stratification of it. Based on this acyclic and stratified structure added to the original network, the main theorem will indicate whether and which states within the network will end up in a common equilibrium value, and determine bounds for the equilibrium values of the states.

In the paper, first in Sect. 2 the basic definition of network used as a vehicle is briefly discussed. Next, in Sect. 3 asymptotic behaviour is discussed, illustrated for an example network with an example simulation. In Sect. 4 the definitions of the Graph Theory tools on connectivity are discussed; in Sect. 5 the identified characteristics of combination functions are defined. In Sect. 6 the main theorem is formulated and it is shown how it was proved. Finally, Sect. 7 is a discussion.

2 Temporal-Causal Networks

A temporal-causal network model is based on three notions, connection weight, combination function, and speed factor, which define the network structure; see Table 1, upper part. Note that the word temporal in temporal-causal refers to the causality, not to the network. To provide sufficient flexibility, a number of standard combination functions are available as options, but also own-defined functions can be used. In Table 1, lower part it is shown how a conceptual representation describing a network structure defines a numerical representation of the network's dynamics; see also [10], Chap. 2, or [11]. Here $X_1, \ldots, X_k$ are the states with outgoing connections to state Y. This defines the detailed dynamic semantics of a temporal-causal network.

Table 1. Conceptual and numerical representations of a temporal-causal network

Concepts	*Notation*	*Explanation*
States and connections	X, Y, $X \rightarrow Y$	Describes the nodes and links of a network structure (e.g., in graphical or matrix format)
Connection weight	$\boldsymbol{\omega}_{X,Y}$	*Connection weight* $\boldsymbol{\omega}_{X,Y} \in [-1, 1]$ represents the strength of the impact of state X on state Y through connection $X \rightarrow Y$
Aggregating multiple impacts	$\mathbf{c}_Y(..)$	For each state Y a *combination function* $\mathbf{c}_Y(..)$ is chosen to combine the causal impacts of other states on state Y
Timing of the causal effect	$\boldsymbol{\eta}_Y$	For each state Y a *speed factor* $\boldsymbol{\eta}_Y \geq 0$ is used to represent how fast a state is changing upon causal impact
Concepts	*Numerical representation*	*Explanation*
State values over time t	$Y(t)$	At each time point t each state Y has a real number value in [0, 1]
Single causal impact	$\mathbf{impact}_{X,Y}(t) = \boldsymbol{\omega}_{X,Y} X(t)$	At t state X with connection to state Y has an impact on Y, using weight $\boldsymbol{\omega}_{X,Y}$
Aggregating multiple impacts	$\mathbf{aggimpact}_Y(t) = \mathbf{c}_Y(\mathbf{impact}_{X1,Y}(t), \ldots, \mathbf{impact}_{Xk,Y}(t)) = \mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y} X_1(t), \ldots, \boldsymbol{\omega}_{Xk,Y} X_k(t))$	The aggregated impact of multiple states X_i on Y at t, is determined using combination function $\mathbf{c}_Y(..)$
Timing of the causal effect	$Y(t+\Delta t) = Y(t) + \boldsymbol{\eta}_Y[\mathbf{aggimpact}_Y(t) - Y(t)]\Delta t = Y(t) + \boldsymbol{\eta}_Y[\mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y} X_1(t), \ldots, \boldsymbol{\omega}_{Xk,Y} X_k(t)) - Y(t)]\Delta t$	The impact on Y is exerted over time gradually, using speed factor $\boldsymbol{\eta}_Y$

The difference equations in the last row in Table 2 can be used for simulation and mathematical analysis, and can also be written in differential equation format: $\mathbf{d}Y(t)/\mathbf{d}t = \boldsymbol{\eta}_Y[\mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{Xk,Y}X_k(t)) - Y(t)]$. Note that combination functions are functions on the 0–1 interval within the real numbers: $[0, 1]^k \rightarrow [0, 1]$. Examples of combination functions often used (see also [10], Chap. 2, Table 2.10) are the *identity* **id**(.) for states with only one impact, the minimum function **min**(..), the *advanced logistic sum* combination function $\mathbf{alogistic}_{\sigma,\tau}$(..) with steepness σ and threshold τ, defined by

$$\mathbf{alogistic}_{\sigma,\tau}(V_1, \ldots, V_k) = [1/(1 + \mathrm{e}^{-\sigma(V_1 + \cdots + V_k - \tau)}) - 1/(1 + \mathrm{e}^{\sigma\tau})](1 + \mathrm{e}^{-\sigma\tau}),$$

and the Euclidean combination function of nth order with scaling factor λ (generalising the scaled sum $\mathbf{ssum}_\lambda$(..) for $n = 1$) defined by

$$\mathbf{eucl}_{n,\lambda}(V_1, \ldots, V_k) = \left(\left(V_1^n + \cdots + V_k^n\right)/\lambda\right)^{1/n}$$

3 Asymptotic Network Behaviour

The asymptotic behaviour will be explored by analysing the possible equilibria. First a definition of stationary points and equilibria.

Definition 1 (stationary point and equilibrium). A state Y has a *stationary point* at t if $\mathbf{d}Y(t)/\mathbf{d}t = 0$. The network is in *equilibrium* at t if every state Y of the model has a stationary point at t.

Considering the differential equation format for a temporal-causal network model, and assuming a nonzero speed factor a more specific criterion can be found.

Lemma 1 (Criterion for a stationary point in a temporal-causal network). Let Y be a state and $X_1, \ldots, X_k$ the states with outgoing connections to state Y. Then Y has a stationary point at t if and only if $\mathbf{c}_Y(\omega_{X1,Y}X_1(t), \ldots, \omega_{Xk,Y}X_k(t)) = Y(t)$.

The example network shown in Fig. 1 is used as illustration. For the connection weights shown in Table 2 the simulation outcome (for $\Delta t = 0.5$) shown in Fig. 2 was obtained.

In this simulation state X_1 has initial value 0.9 and this stays constant due to having speed factor 0. The other states have initial value 0, except X_5 which has 0.9 as initial value. The speed factor of states X_2 to X_{10} is 0.5, and their combination function is a normalised scaled sum function. The simulation outcome can be seen in Fig. 2. It turns out that states X_1 to X_4 all end up at value 0.9, states X_5 to X_7 all at value 0.3 and states X_8 to X_{10} at different individual values 0.681, 0.490, and 0.389, respectively. So, there is some clustering, but also some states end up in their own unique value, and it can be observed that these unique values are in between the cluster values.

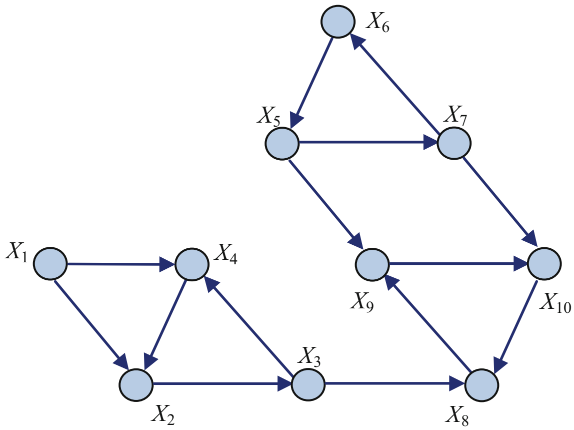

Fig. 1. Example network

Table 2. Connection weights for the example simulation

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
X_1		0.8		0.5						
X_2			1							
X_3				0.2				0.8		
X_4		0.6								
X_5							0.6		0.8	
X_6					0.7					
X_7						0.8				0.8
X_8									0.8	
X_9										0.7
X_{10}								0.6		

How can such an emerging pattern be explained? This question will be addressed in the next sections. It will be found out how the pattern depends on the network's characteristics, and in particular on the connectivity within the network and the characteristics of the combination functions. Each of these factors will be discussed first in different sections, after which it will be analysed how they relate to the emerging pattern.

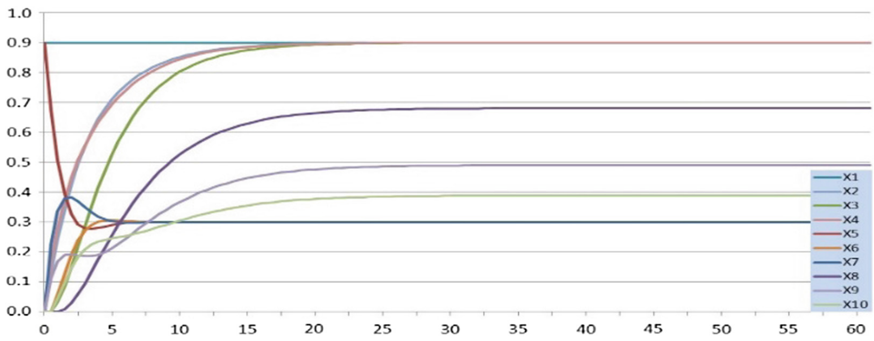

Fig. 2. Example simulation

4 Connectivity and Strongly Connected Components

To analyse connectivity within Graph Theory the notion of strongly connected component has been identified. The main parts of the following definitions can be found, for example, in [6], Chap. 3, or [7], Sect. 6. Note that in the current paper only nonnegative connection weights are considered.

Definition 2 (reachability and strongly connected components).

(a) State Y is *forward reachable* from state X if there is a directed path from X to Y with nonzero connection weights and speed factors.
(b) A network N is *strongly connected* if every two states are mutually forward reachable within N.
(c) A state is called *independent* if it is not forward reachable by any other state.
(d) A *subnetwork* of a network N is a network whose states and connections are states and connections of N.
(e) A *strongly connected component* C of a network N is a strongly connected subnetwork of N such that no larger strongly connected subnetwork of N contains it as a subnetwork.

Strongly connected components C can be identified by choosing any node X of N and adding all nodes that are on any cycle through X. Note also that when a node X is not on any cycle, then it will form a singleton strongly connected component C by itself; this applies in particular to all nodes of N with indegree or outdegree zero. There are efficient algorithms available to determine the strongly connected components of a network or graph; for example, see [1, 4, 5, 8, 9, 13]. The strongly connected components of the example network shown in Fig. 1 are depicted in Fig. 3.

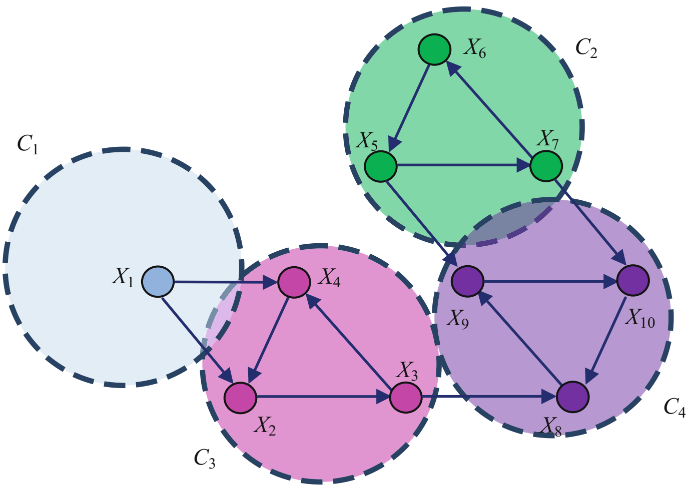

Fig. 3. The strongly connected components within the example network

Having identified the strongly connected components, allows to obtain a kind of abstracted picture of the network, called the condensation graph; see Fig. 4.

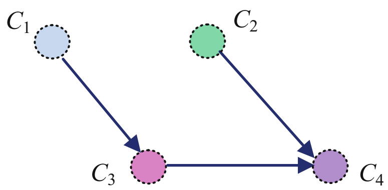

Fig. 4. Condensation of the example network by its strongly connected components: the directed acyclic condensation graph C(N)

Definition 3 (condensation graph). The *condensation* C(N) of a network N with respect to its strongly connected components is a graph whose nodes are the strongly connected components of N and whose connections are determined as follows: there is a connection from node C_i to node C_j in C(N) if and only if in N there is at least one connection from a node in the strongly connected component C_i to a node in the strongly connected component C_j.

An important result is that a condensation graph C(N) is always an acyclic graph. The following theorem summarizes this; see also [6], Chap. 3, Theorems 3.6 and 3.8, or [7], Sect. 6.

Theorem 1 (acyclic condensation graph).

(a) For any network N its condensation graph C(N) is acyclic, and has at least one state of outdegree zero and at least one state of indegree zero.

(b) The network N is acyclic itself if and only if it is graph-isomorphic to C(N). In this case the nodes in C(N) all are singleton sets $\{X\}$ containing one state X from N.
(c) The network N is strongly connected itself if and only if C(N) only has one node; this node is the set of all states of N.

As a next step, for any acyclic directed graph a stratification structure is defined; for example, see [3]. Here a similar construction is applied in particular to the condensation graph C(N) thus obtaining a stratified condensation graph SC(N); see Fig. 5.

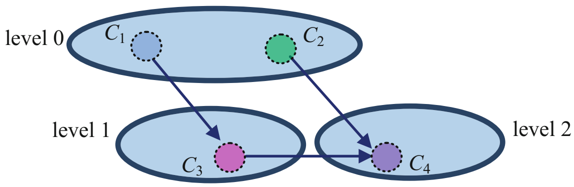

Fig. 5. Stratified condensation graph SC(N) for the example network

Definition 4 (stratified condensation graph). The *stratified condensation graph* for network N, denoted by SC(N), is the condensation graph C(N) together with a leveled partition $S_0, \ldots, S_{h-1}$ in strata S_i such that $S_0 \cup \cdots \cup S_{h-1}$ is the set of all nodes of C (N) and the S_i are mutually disjoint, which is defined inductively as follows. Here h is the height of C(N), i.e., the length of the longest path in C(N).

(i) The stratum S_0 is the set of nodes in C(N) without incoming connections in C(N)
(ii) For each $i > 0$ the stratum S_i is the set of nodes in C(N) for which all incoming connections in C(N) come only from nodes in $S_0, .., S_{i-1}$.

If node X is in stratum S_i, its *level* is i.

5 Characteristics of Combination Functions

It has been found out (see Sect. 6) how the following characteristics of combination functions relate to asymptotic behaviour as discussed in Sect. 3. Note that for combination functions it is (silently) assumed that $\mathrm{c}(V_1, \ldots, V_k) = 0$ iff $V_i = 0$ for all i.

Definition 5 (monotonic, scalar-free, and additive for a combination function).

(a) A function c(..) is called *monotonically increasing* if for all values U_i, V_i it holds

$$U_i \leq V_i \text{ for all } i \Rightarrow \mathrm{c}(U_1, \ldots, U_k) \leq \mathrm{c}(V_1, \ldots, V_k)$$

(b) A function c(..) is called *strictly monotonically increasing* if

$$U_i \leq V_i \text{ for all } i, \text{ and } U_j < V_j \text{ for at least one } j \Rightarrow \mathrm{c}(U_1, \ldots, U_k) < \mathrm{c}(V_1, \ldots, V_k)$$

(c) A function c(..) is called *scalar-free* if for all $\alpha > 0$ and all $V_1, \ldots, V_k$ it holds

$$c(\alpha V_1, \ldots, \alpha V_k) = \alpha c(V_1, \ldots, V_k)$$

(d) A function c(..) is called *additive* if for all $U_1, \ldots, U_k$ and $V_1, \ldots, V_k$ it holds

$$c(U_1 + V_1, \ldots, U_k + V_k) = c(U_1, \ldots, U_k) + c(V_1, \ldots, V_k)$$

(e) A function c(..) is called *linear* if it is both scalar-free and additive.

Note that (nth order) Euclidean combination functions satisfy (a), (b), and (c), and the first order ones (= scaled sum functions) satisfy also (d) and (e).

Definition 6 (normalised). A network is *normalised* if for each state Y it holds $c_Y(\omega_{X1,Y}, \ldots, \omega_{Xk,Y}) = 1$,, where $X_1, \ldots, X_k$ are the states with outgoing connections to Y.

Note that for a Euclidean combination function of nth order the scaling parameter choice $\lambda_Y = \omega^n_{X1,Y} + \cdots + \omega^n_{Xk,Y}$ will provide a normalised network. This can be done in general:

(1) normalising a combination function
If any combination function $\mathbf{c}_Y(..)$ is replaced by $\mathbf{c}'_Y(..)$ defined as

$$\mathbf{c}'_Y(V_1, \ldots, V_k) = \mathbf{c}_Y(V_1, \ldots, V_k)\,/\mathbf{c}_Y(\omega_{X1,Y}, .., \omega_{Xk,Y})$$

(note $\mathbf{c}_Y(\omega_{X1,Y}, .., \omega_{Xk,Y}) > 0$ since $\omega_{Xi,Y} > 0$), then the network becomes normalised.

(2) normalising the connection weights (for scalar-free combination functions)
For scalar-free combination functions also normalisation is possible by adapting the connection weights; define $\omega'_{Xi,Y} = \omega_{Xi,Y}/\mathbf{c}_Y(\omega_{X1,Y}, .., \omega_{Xk,Y})$, then indeed it holds:

$$\mathbf{c}_Y(\omega'_{X1,Y}, .., \omega'_{Xk,Y}) = \mathbf{c}(\omega_{X1,Y}/c_Y(\omega_{X1,Y}, .., \omega_{Xk,Y}), .., \omega_{Xk,Y}/c(\omega_{X1,Y}, .., \omega_{Xk,Y})) = 1$$

The following proposition illustrates some of the implications of the above-defined characteristics. This will be used in Sect. 6.

Proposition 1. Suppose the network is normalised.

(a) If the combination functions are scalar-free and $X_1, \ldots, X_k$ are the states with outgoing connections to Y, and $X_1(t) = \cdots = X_k(t) = V$ for some common value V, then also $\mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y} X_1(t), \ldots, \boldsymbol{\omega}_{Xk,Y} X_k(t)) = V$.

(b) If the combination functions are scalar-free and $X_1, \ldots, X_k$ are the states with outgoing connections to Y, and for $U_1, \ldots, U_k$, $V_1, \ldots, V_k$ and $\alpha \geq 0$ it holds $V_i = \alpha\ U_i$, then $\mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y} V_1, \ldots, \boldsymbol{\omega}_{Xk,Y} V_k) = \alpha \mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y} U_1, \ldots, \boldsymbol{\omega}_{Xk,Y} U_k)$

If in this situation in two different simulations, state values $X_i(t)$ and $X'_i(t)$ are generated then $X_i'(t) = \alpha X_i(t) \quad \Rightarrow \quad X_i'(t+\Delta t) = \alpha X_i(t+\Delta t)$.

(c) If the combination functions are additive and $X_1, \ldots, X_k$ are the states with outgoing connections to Y, then for values $U_1, \ldots, U_k, V_1, \ldots, V_k$ it holds

$$\mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}(U_1+V_1), \ldots, \boldsymbol{\omega}_{Xk,Y}(U_k+V_k)) = \mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}U_1, \ldots, \boldsymbol{\omega}_{Xk,Y}U_k) + \mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}V_1, \ldots, \boldsymbol{\omega}_{Xk,Y}V_k)$$

If in this situation in three different simulations, state values $X_i(t)$, $X'_i(t)$ and $X''_i(t)$ are generated then

$$X_i''(t) = X_i(t) + X_i'(t) \quad \Rightarrow \quad X_i''(t+\Delta t) = X_i(t+\Delta t) + X_i'(t+\Delta t)$$

(d) If the combination functions are scalar-free and monotonically increasing, and $X_1, \ldots, X_k$ are the states with outgoing connections to Y, and $V_1 \leq X_1(t), \ldots, X_k(t) \leq V_2$ for some values V_1 and V_2, then also

$$V_1 \leq \mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{Xk,Y}X_k(t)) \leq V_2$$

and if $\eta_Y\,\Delta t \leq 1$ and $V_1 \leq Y(t) \leq V_2$ then $V_1 \leq Y(t+\Delta t) \leq V_2$.

Proof. (a) This follows from

$$\begin{aligned}\mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{Xk,Y}X_k(t)) &= \mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}V, \ldots, \boldsymbol{\omega}_{Xk,Y}V) \\ &= V\mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}, \ldots, \boldsymbol{\omega}_{Xk,Y}) = V\end{aligned}$$

(b) and (c) can be verified easily.
(d) This follows from

$$\begin{aligned}V_1 &= V_1\mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}, \ldots, \boldsymbol{\omega}_{Xk,Y}) = \mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}V_1, \ldots, \boldsymbol{\omega}_{Xk,Y}V_1) \\ &\leq \mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{Xk,Y}X_k(t)) \leq \mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}V_2, \ldots, \boldsymbol{\omega}_{Xk,Y}V_2) \\ &= V_2\mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}, \ldots, \boldsymbol{\omega}_{Xk,Y}) = V_2\end{aligned}$$

and the second part from

$$\begin{aligned}&Y(t) + \eta_Y[\mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{Xk,Y}X_k(t)) - Y(t)]\Delta t \\ &= \mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{Xk,Y}X_k(t))\eta_Y\Delta t + Y(t)(1-\eta_Y\Delta t) \leq V_2\eta_Y\Delta t + V_2(1-\eta_Y\Delta t) = V_2\end{aligned}$$

and similarly for V_1

$$\mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{Xk,Y}X_k(t))\eta_Y\Delta t + Y(t)(1-\eta_Y\Delta t) \geq V_1\eta_Y\Delta t + V_1(1-\eta_Y\Delta t) = V_1$$

6 Asymptotic Network Behaviour and Network Characteristics

In this section it is shown how the network structure characteristics concerning connectivity and combination function characteristics as discussed in Sects. 4 and 5 relate to emerging network behaviour. As a first case, consider a network without cycles, for example, a hierarchical population following a leader. Then the following theorem has been proven using Lemma 1 from Sect. 3 and Proposition 1; see [12].

Theorem 2 (common state values provide equilibria). Suppose a network with nonnegative connections is based on normalised and scalar-free combination functions. Then the following hold.

(a) Whenever all states have the same value V, the network is in an equilibrium state.
(b) If for every state for its initial value V it holds $V_1 \leq V \leq V_2$, then for all t for every state Y it holds $V_1 \leq Y(t) \leq V_2$. In an achieved equilibrium for every state for its equilibrium value V it holds $V_1 \leq V \leq V_2$.

Theorem 3 (Common equilibrium state values; acyclic case). Suppose an acyclic network with nonnegative connections is based on normalised and scalar-free combination functions.

(a) If in an equilibrium state the independent states all have the same value V, then all states have the same value V.
(b) If, moreover, the combination functions are monotonically increasing, and in an equilibrium state the independent states all have values V with $V_1 \leq V \leq V_2$, then all states have values V with $V_1 \leq V \leq V_2$.

Next, a basic lemma for dynamics of normalised networks with combination functions that are (strictly) monotonically increasing and scalar-free; see [12].

Lemma 2. Let a normalised network with nonnegative connections be given and its combination functions are monotonically increasing and scalar-free; then the following hold:

(a) (i) If for some node Y at time t for all nodes X with $\omega_{X,Y}>0$ it holds $X(t) \leq Y(t)$, then $Y(t)$ is decreasing at t: $\mathbf{d}Y(t)/\mathbf{d}t \leq 0$.
(ii) If the combination functions are strictly increasing and a node X exists with $X(t) < Y(t)$ and $\omega_{X,Y}>0$, and the speed factor of Y is nonzero, then $Y(t)$ is strictly decreasing at t: $\mathbf{d}Y(t)(t)/\mathbf{d}t <0$.

(b) (i) If for some node Y at time t for all nodes X with $\omega_{X,Y}>0$ it holds $X(t) \geq Y(t)$, then $Y(t)$ is increasing at t: $\mathbf{d}Y(t)/\mathbf{d}t \geq 0$.
(ii) If, the combination function is strictly increasing and a node X exists with $X(t) > Y(t)$ and $\omega_{X,Y}>0$, and the speed factor of Y is nonzero, then $Y(t)$ is strictly increasing at t: $\mathbf{d}Y(t)/\mathbf{d}t > 0$.

Using Lemmas 1 and 2 the following proposition has been proven for strongly connected networks with cycles; see [12].

Theorem 4 (Common equilibrium state values; strongly connected cyclic case). Suppose the combination functions of the normalised network N are scalar-free and strictly monotonically increasing. Then the following hold.

(a) If the network is strongly connected itself, then in an equilibrium state all states have the same value.
(b) Suppose the network has one or more independent states and the subnetwork without these independent states is strongly connected. If in an equilibrium state all independent states have values V with $V_1 \leq V \leq V_2$, then all states have values V with $V_1 \leq V \leq V_2$. In particular, when all independent states have the same value V, then all states have this same value V.

Next, the main, general theorem is formulated, consisting of two Theorems 5 and 6.

Theorem 5 (main theorem on equilibrium state values, part I). Suppose the network N is normalised and its combination functions are scalar-free and strictly monotonic. Let SC(N) be the stratified condensation graph of N. Then in an equilibrium state of N the following hold.

(a) For each strongly connected component $C \in$ SC(N) of N of level 0 the following hold:
(i) All states in N belonging to C have the same equilibrium value V.
(ii) If for the initial values V of all states in N belonging to C it holds $V_1 \leq V \leq V_2$, then also for the equilibrium values V of all states in C it holds $V_1 \leq V \leq V_2$.
(iii) In particular, when all initial values of states in N belonging to C are equal to one value V, then the equilibrium value of all states in C is also V.
(b) Let $C \in$ SC(N) be a strongly connected component of N of level $i > 0$. Let $C_1, .., C_k \in \mathrm{SC}(N)$ be the strongly connected components of N with an outgoing connection to C within the condensation graph SC(N). Then the following hold:
(i) If for the equilibrium values V of all states in N belonging to $C_1 \cup \cdots \cup C_k$ it holds $V_1 \leq V \leq V_2$, then for all states in N belonging to C for their equilibrium value V it holds $V_1 \leq V \leq V_2$.
(ii) In particular, when all equilibrium values of all states in N belonging to $C_1 \cup \cdots \cup C_k$ are equal to one value V, then also the equilibrium values of all states in N belonging to C are equal to the same V.

Proof.

(a) (i) This follows from Theorem 3(a).
(ii) This follows from Proposition 1(b).
(iii) This follows from (ii) with $V_1 = V_2 = V$.

(b) (i) This follows from Theorem 3(b) applied to C augmented with (as independent states) the states in $C_1 \cup \cdots \cup C_k$ with outgoing connections to states in C, with their values and these connections.
(ii) This follows from (i) with $V_1 = V_2 = V$.

Theorem 6 (main theorem on equilibrium state values, part II). Suppose the network N is normalised and its combination functions are scalar-free and strictly monotonic. Let SC(N) be the stratified condensation graph of N. Then in an equilibrium state of N the following hold.

(a) If the equilibrium values of all states in every strongly connected component of level 0 in SC(N) are equal to one value V, then the equilibrium state values of all states in N are equal to the same value V.
(b) If for the equilibrium values V of all states in every strongly connected component of level 0 in SC(N) it holds $V_1 \leq V \leq V_2$, then for the equilibrium state values V of all states in N it holds $V_1 \leq V \leq V_2$.
(c) If the initial values of all states in every strongly connected component of level 0 in SC(N) are equal to one value V, then for the equilibrium state values of all states in N are equal to the same value V.
(d) If for the initial values V of all states in every strongly connected component of level 0 in SC(N) it holds $V_1 \leq V \leq V_2$, then for the equilibrium state values V of all states in N it holds $V_1 \leq V \leq V_2$.

Proof. Use induction over the number of strata in SC(N) and apply Theorem 4(a) for the level 0 stratum and Theorem 4(b) for the induction step from the strata of level $j < i$ to the stratum of level $i > 0$.

These theorems are in accordance with the example simulation shown in Fig. 2 and other simulations that were conducted. As an illustration, the strongly connected components of level 0 for the example are the subnetworks based on $\{X_1\}$ and $\{X_5, X_6, X_7\}$ (see Figs. 3 and 5). The initial values of X_1 and X_5 are 0.9, and all other initial values are 0. From Theorems 5(a)(i) and 4(a)(ii), it follows that the equilibrium value of X_1 is 0.9, which indeed is the case, and those of X_5, X_6, X_7 are the same and ≤ 0.9; this is indeed the case in Fig. 2, as these three equilibrium values of X_5, X_6, X_7 are all 0.3. This specific value 0.3 in principle depends on the initial values of the states and the connection weights, which are not taken into account in the theorems. However, see also Theorem 7.

Next, consider the level 1 component C_3, based on $\{X_2, X_3, X_4\}$. The only incoming connection here is from X_1, which has equilibrium value 0.9 (implied by Theorem 5(a)(ii)). Now by Theorem 5(b)(ii) it follows that all of X_2, X_3, X_4 also have the same equilibrium value 0.9; this is indeed the case in Fig. 2. Finally, consider level 2 component C_4, based on $\{X_8, X_9, X_{10}\}$. It has two incoming connections, one from X_3 in and one from X_5 in C_2. Now their equilibrium values are not equal: they are 0.9 and 0.3, respectively. Therefore it is not implied by the above theorems that the equilibrium values of X_8, X_9, X_{10} are the same; and indeed in Fig. 2 they are different: 0.681, 0.490, and 0.389, respectively. Yet an implication from Theorem 5(b)(i) is that these equilibrium values should be ≥ 0.3 and ≤ 0.9. This is indeed the case in Fig. 2. This illustrates how these theorems can be applied. Note that the specific equilibrium values 0.681, 0.490, and 0.389 cannot be predicted in this way. They also depend on the connection weights for the states X_8, X_9, X_{10} within component C_4, and these are not taken into account in the general theorem. However see also below, in the last paragraph of this section

As a variation, if the initial value of X_1 is set at 0.3 instead of 0.9, then all equilibrium values turn out to become the same 0.3; see Fig. 6. In this case the values of all states in the level 0 components C_1 and C_2 have the same value 0.3. Similar to the first case above, also the states in C_3 have the equilibrium value 0.3 because they are only affected by X_1 which has value 0.3. But now the equilibrium values of both X_3 in C_3 and X_5 in C_2 are the same 0.3, so Theorem 5(b)(ii) can be applied to derive that all states in C_4 also have that same equilibrium value 0.3.

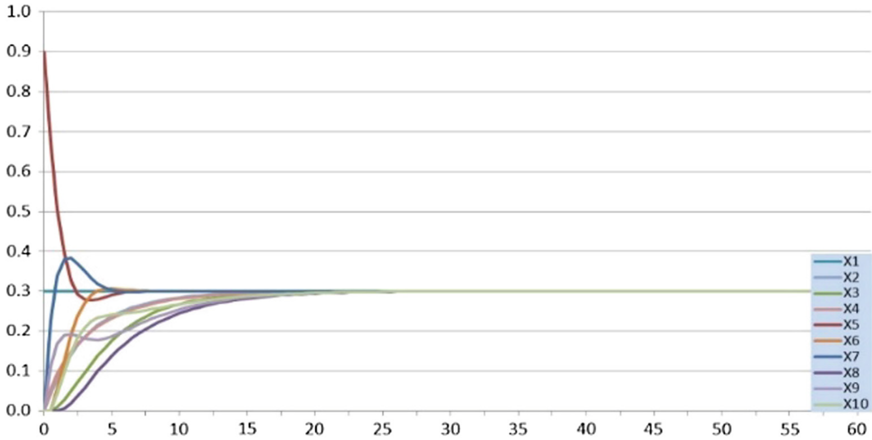

Fig. 6. Variation of the example simulation for initial value 0.3 of X1

This proves that all states of the network have to have value 0.3 in the equilibrium. Alternatively, just apply Theorem 6(a) for this case. Then from the equal equilibrium values in the level 0 components C_1 and C_2 it immediately follows that all states in the network have that same equilibrium value.

In the main theorem the level 0 components play an important role, as initial nodes in the stratified condensation graph SC(N). Therefore it can be useful to know more about their state values in an equilibrium. A result on this is the following.

Theorem 7 (equilibrium state values for level 0 components). Suppose the network N with states $X_1, \ldots, X_n$ is normalised and strongly connected. Then the following hold.

(a) If the combination functions of the network N are scalar-free, then for given connection weights and speed factors, for any value $V \in [0, 1]$ there are initial values such that V is the common state value in an equilibrium achieved from these initial values.

(b) For given connection weights and speed factors, let eq: $[0, 1]^n \rightarrow [0, 1]$ be the function such that $\text{eq}(V_1, \ldots, V_n)$ is the common state value for an equilibrium achieved from initial values $X_i(0) = V_i$ for all i. Then $\text{eq}(0, \ldots, 0) = 0$, $\text{eq}(1, \ldots, 1) = 1$, and the following hold:
(i) If the combination functions of the network are scalar-free, then eq is scalar-free
(ii) If the combination functions of the network are additive, then eq is additive

(c) Suppose the combination functions of the network N are linear. For given connection weights and speed factors for each i let e_i be the achieved common equilibrium value for initial values $X_i(0) = 1$ and $X_j(0) = 0$ for all $j \neq i$, i.e., $e_i = \text{eq}(0, \ldots, 0, 1, 0, \ldots, 0)$ with 1 as ith argument. Then the sum of the e_i is 1, i.e., $e_1 + \cdots + e_n = 1$ and in the general case for these given connection weights and speed factors, the common equilibrium value eq(…) is a linear, monotonically increasing, continuous and differentiable function of the initial values $V_1, \ldots, V_n$ satisfying the following linear relation: $\text{eq}(V_1, \ldots, V_n) = e_1 V_1 + \cdots + e_n V_n$. If the combination functions of N are strictly increasing, then $e_i > 0$ for all i, and eq is also strictly increasing.

Proof. (a) This follows from Proposition 1(a) or (d) with $V_1 = V_2 = V$.

(b and c) This follows from Proposition 1(b) and (c), and Lemma 2.

Based on this theorem, in particular for the case of linear combination functions, for level 0 components after each base equilibrium value e_i is determined, any equilibrium value can be predicted from the initial values by the identified linear expression.

Note that in particular for the case of linear combination functions the equilibrium equations are linear and could be solved algebraically. However, this does not provide additional information for level 0 components. They have an infinite number of solutions as every common value V is a solution; apparently the linear equations always have a mutual dependency in this case. But for components of level $i > 0$ solving the linear equations can provide specific values, due to the specific input values they get from one or more lower level components. In this way the specific equilibrium values of the states X_8, X_9 and X_{10} in C_4 can be determined algebraically from the values of the states X_3, X_5, and X_7 in the lower level components C_2 and C_3 (repetitive digits in italics):

$$X_8 = 0.680952380952381,\ X_9 = 0.4904761904761905,\ X_{10} = 0.388888888888889$$

which indeed is in accordance with the values found in the simulation.

Finally a similar theorem that is applicable for components of level $i > 0$.

Theorem 8 (equilibrium state values for components of level i > 0). Suppose the network is normalised, and consists of a strongly connected component plus a number of independent states $A_1, \ldots, A_p$ with outgoing connections to this strongly connected component. Then the following hold

(a) Suppose the combination functions are scalar-free and $X_1, \ldots, X_k$ are the states with outgoing connections to Y. If for $U_1, \ldots, U_k, V_1, \ldots, V_k$ and $\alpha \geq 0$ it holds $V_i = \alpha\, U_i$ for all i, then $\mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}V_1, \ldots, \boldsymbol{\omega}_{Xk,Y}V_k) = \alpha\mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}U_1, \ldots, \boldsymbol{\omega}_{Xk,Y}U_k)$

(b) Suppose the combination functions are additive and $X_1, \ldots, X_k$ are the states with outgoing connections to Y. Then if for values $U_1, \ldots, U_k, V_1, \ldots, V_k, W_1, \ldots, W_k$ it holds $W_i = U_i + V_i$ for all i, then $\mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}W_1, \ldots, \boldsymbol{\omega}_{Xk,Y}W_k) = \mathbf{c}_Y(\boldsymbol{\omega}\, X1, YU_1, \ldots, \boldsymbol{\omega}_{Xk,Y}U_k) + \mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}V_1, \ldots, \boldsymbol{\omega}_{Xk,Y}V_k)$

(c) Suppose all combination functions of the network N are linear. Then for given connection weights and speed factors, for each state Y the achieved equilibrium value for Y only depends on the equilibrium values $V_1, \ldots, V_p$ of states $A_1, \ldots, A_p$; the function $\mathrm{eq}_Y(V_1, \ldots, V_p)$ denotes this achieved equilibrium value for Y.

(d) Suppose the combination functions of the network N are linear. For the given connection weights and speed factors for each i let $d_{i,Y}$ be the achieved equilibrium value for state Y in a situation with equilibrium values $A_i = 1$ and $A_j = 0$ for all $j \neq i$, i.e.,$d_{i,Y} = \mathrm{eq}_Y(0, \ldots, 0, 1, 0, .., 0)$ with 1 as ith argument. Then in the general case for these given connection weights and speed factors, for each Y in the strongly connected component its equilibrium value is a linear, monotonically increasing, continuous and differentiable function $\mathrm{eq}_Y(\ldots)$ of the equilibrium values $V_1, \ldots, V_p$ of $A_1, \ldots, A_p$ satisfying the following linear relation: $\mathrm{eq}_Y(V_1, \ldots, V_p) = d_{1,Y}V_1 + \cdots + d_{p,Y}V_p$. Here the sum of the $d_{i,Y}$ is 1 : $d_{1,Y} + \cdots + d_{p,Y} = 1$. In particular, the equilibrium values are independent of the initial values for all states Y different from $A_1, \ldots, A_p$. If the combination functions of N are strictly increasing, then $d_{i,Y} > 0$ for all i, and $\mathrm{eq}_Y(..)$ is also strictly increasing.

Proof. (a and b) follow from Proposition 1

(c) From (a) and (b) it follows that the equilibrium value of Y is a linear function of the initial values of all states of N. Therefore the function is a linear combination of $e_i = \mathrm{eq}_Y(0, \ldots, 0, 1, 0, \ldots, 0)$ where only one state has initial value 1 and all other 0. However, when all independent states have (constant) value 0, from Theorem 5(b)(ii) it follows that all states will have equilibrium value 0. In particular, this holds for cases that only one of the states that are not independent have initial value 1 and all other states have initial value 0. This shows that from the linear combination the coefficient e_i of these terms are 0. Therefore $\mathrm{eq}_Y(\ldots)$ is a function of $V_1, \ldots, V_p$ only. From (a) and (b) it follows that $\mathrm{eq}_Y(V_1, \ldots, V_p)$ is linear, as indicated above. Therefore

$$\begin{aligned}
\mathrm{eq}_Y(V_1, \ldots, V_p) &= \mathrm{eq}_Y(V_1, 0, \ldots, 0) + \cdots + \mathrm{eq}_Y(0, \ldots, 0, V_i, 0, \ldots, 0) \\
&\quad + \cdots + \mathrm{eq}_Y(0, \ldots, V_p) \\
&= \mathrm{eq}_Y(1, 0, \ldots, 0)V_1 + \cdots + \mathrm{eq}_Y(0, \ldots, 0, 1, 0, \ldots, 0)V_i \\
&\quad + \cdots + \mathrm{eq}_Y(0, \ldots, 1)V_p \\
&= d_{1,Y}V_1 + \cdots + d_{i,Y}V_i + \cdots + d_{p,Y}V_p
\end{aligned}$$

This theorem can be applied by only determining the effect of the independent states on the equilibrium values; this is easily applicable especially in cases that there are just a few of them.[1] Note that by using in the above proof Theorem 3 instead of Theorem 5(b)(ii), a similar theorem is obtained for the case of an acyclic network: then the equilibrium values of all states are linear combinations of the values of the initial states.

7 Discussion

To analyse and predict what asymptotic behaviour a given network will show is in general a challenging issue. In this paper a general theorem was presented that relates asymptotic network behaviour in terms of an equilibrium state to the network characteristics: in two parts described by Theorems 5 and 6 in Sect. 6. The relevant network characteristics concern on the one hand connectivity in terms of the network's strongly connected components and their mutual connections as shown in the network's condensation graph, and on the other hand characteristics of the combination functions used to aggregate the effects of multiple incoming connections (in particular, monotonicity, scalar freeness and normalisation). The theorem subsumes and generalises existing theorems for specific cases such as similar theorems for acyclic networks, fully connected networks and strongly connected networks (e.g., Theorems 3 and 4 in Sect. 6), and theorems addressing only scaled sum combination functions as one fixed type of combination function (e.g., Theorem 3 at p. 120 of [2]).

The main theorem can be applied to predict behaviour of a given network, or to set initial values in order to get some expected behaviour. It also can be used as a form of verification to check correctness of the implementation of a network. If simulation outcomes contradict the theorem, then this suggests that some debugging of the implementation may be needed.

References

1. Bloem, R., Gabow, H.N., Somenzi, F.: An algorithm for strongly connected component analysis in n log n symbolic steps. Form. Meth. Syst. Des. **28**, 37–56 (2006)
2. Bosse, T., Duell, R., Memon, Z.A., Treur, J., van der Wal, C.N.: Agent-based modelling of emotion contagion in groups. Cogn. Comput. **7**(1), 111–136 (2015)
3. Chen, Y.: General spanning trees and reachability query evaluation. In: Desai, B.C. (ed.) Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering, C3S2E'09, pp. 243–252. ACM Press (2009)
4. Fleischer, L.K., Hendrickson, B., Pınar, A.: On identifying strongly connected components in parallel. In: Rolim J. (ed) Parallel and Distributed Processing. IPDPS 2000. Lecture Notes in Computer Science, vol. 1800, pp. 505–511. Springer (2000)
5. Gentilini, R., Piazza, C., Policriti, A.: Computing strongly connected components in a linear number of symbolic steps. In: Proceedings of the SODA'03, pp. 573–582 (2003)
6. Harary, F., Norman, R.Z., Cartwright, D.: Structural models: an introduction to the theory of directed graphs. Wiley, New York (1965)
7. Kuich, W.: On the entropy of context-free languages. Inf. Control **16**, 173–200 (1970)

[1] At http://www.few.vu.nl/~treur/linearsolvingv04.pdf further analysis of the example network illustrates Theorem 8, and also for another example network model.

8. Li, G., Zhu, Z., Cong, Z., Yang, F.: Efficient decomposition of strongly connected components on GPUs. J. Syst. Architect. **60**(1), 1–10 (2014)
9. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM J. Comput. **1**(2), 146–160 (1972)
10. Treur, J.: Network-Oriented Modeling: Addressing the Complexity of Cognitive, Affective and Social Interactions. Springer Publishers (2016)
11. Treur, J.: The Ins and Outs of Network-Oriented Modeling: from Biological Networks and Mental Networks to Social Networks and Beyond. Paper for Keynote Lecture at ICCCI'18. Springer Publishers (2018)
12. Treur, J.: Relating emerging network behaviour to network structure. In: Proceedings of the 7th International Conference on Complex Networks and their Applications, ComplexNetworks'18. Studies in Computational Intelligence, Springer (2018)
13. Wijs, A., Katoen, J.P., Bošnacki, D.: Efficient GPU algorithms for parallel decomposition of graphs into strongly connected and maximal end components. Form. Methods Syst. Des. **48**, 274–300 (2016)