Equilibrium analysis for linear and nonlinear aggregation in network models applied to mental model aggregation in multilevel organisational learning

Canbaloğlu, Gülay; Treur, Jan

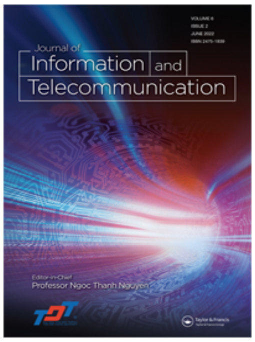
DOI
10.1080/24751839.2022.2043594

Publication date
2022

Document Version
Final published version

Published in
Journal of Information and Telecommunication

Citation (APA)
Canbaloğlu, G., & Treur, J. (2022). Equilibrium analysis for linear and nonlinear aggregation in network models: applied to mental model aggregation in multilevel organisational learning. *Journal of Information and Telecommunication*, *6*(3), 289-340. https://doi.org/10.1080/24751839.2022.2043594


Journal of Information and Telecommunication
Editor-in-Chief Professor Ngoc Thanh Nguyen

Taylor & Francis
Taylor & Francis Group

Journal of Information and Telecommunication

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tjit20

Equilibrium analysis for linear and nonlinear aggregation in network models: applied to mental model aggregation in multilevel organisational learning

Gülay Canbaloğlu & Jan Treur

JOURNAL OF INFORMATION AND TELECOMMUNICATION
https://doi.org/10.1080/24751839.2022.2043594

OPEN ACCESS Check for updates

Equilibrium analysis for linear and nonlinear aggregation in network models: applied to mental model aggregation in multilevel organisational learning

Gülay Canbaloğlu[a,b] and Jan Treur[b,c]

[a]Department of Computer Engineering, Koç University, Istanbul, Turkey; [b]Center for Safety in Healthcare, Delft University of Technology, Delft, The Netherlands; [c]Social AI Group, Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

ABSTRACT

In this paper, equilibrium analysis for network models is addressed and applied in particular to a network model of multilevel organisational learning. The equilibrium analysis addresses properties of aggregation characteristics and connectivity characteristics of a network. For aggregation characteristics, it is shown how certain classes of nonlinear functions enable equilibrium analysis of the emerging dynamics within the network like linear functions do. For connectivity characteristics, by using a form of stratification for the network's strongly connected components, it is shown how equilibrium analysis results can be obtained relating equilibrium values in any component to equilibrium values in (independent) components without incoming connections. In addition, concerning aggregation characteristics, two specific types of nonlinear functions for aggregation in networks (weighted euclidean functions and weighted geometric functions) are analysed. It is illustrated in detail how by using certain function transformations also methods for equilibrium analysis based on a symbolic linear equation solver, can be applied to make predictions about equilibrium values for them. All these results are applied to a network model for organisational learning. Finally, it is analysed in some depth how the function transformations applied can be described by the more general notion of function conjugate relation, also often used for coordinate transformations.

ARTICLE HISTORY
Received 25 December 2021
Accepted 15 February 2022

KEYWORDS
Equilibrium analysis; adaptive network; connectivity; aggregation; organisational learning

1. Introduction

Dynamics in network models described by node states that change over time (for example, for individuals' opinions, intentions, emotions, beliefs, …) depend on network characteristics for the connectivity between nodes, the aggregation of impacts from different nodes on a given node, and the timing of the node updates; e.g. (Treur,

CONTACT Jan Treur j.treur@vu.nl Center for Safety in Healthcare, Delft University of Technology, Delft, The Netherlands; Social AI Group, Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

2020b). As pictures of networks usually only show connectivity characteristics, the roles of the aggregation and timing characteristics are sometimes neglected. The aggregation and timing characteristics also play an important role in the dynamics within a network; for example, whether or not within a well-connected group in the end a common opinion, intention, emotion or belief is reached (a common value for all node states) also, or even mainly depends on them. Often, the tradition is that silent asumptions are made about the aggregation and timing characteristics. For timing characteristics, often it is silently assumed that the nodes are updated in a synchronous manner, although in application domains this assumption is usually not fulfilled. For aggregation, in social network models usually linear forms are applied; this fixed choice makes that it is not investigated how a variation of this choice of aggregation would affect the dynamics in the network. Only for the fixed linear aggregation type some theorems exist specifying connectivity conditions under which all node states converge to the same value, in particular when the network is strongly connected: from every node there is a path to every other node. In contrast, for (artificial) neural network models traditionally often some type of logistic sum format (also often not varied) is applied for aggregation.

In this paper, a more diverse landscape is explored which is not limited by the fixed conditions on connectivity, aggregation or timing as are so often imposed either silently or explicitly. For connectivity, both acyclic and cyclic networks are considered here, and for cyclic networks both strongly connected networks and networks that are not strongly connected. For aggregation, both networks with linear and nonlinear aggregation are considered and for networks with nonlinear aggregation, networks with logistic aggregation are addressed but also networks with other forms of nonlinear aggregation that can be analysed similarly to how networks with linear aggregation can be analysed. Finally, for timing both synchronous and asynchronous timing are covered.

The often occurring use of linear functions for aggregation for social network models may be based on a more general belief that dynamical system models can be analysed better for linear functions than for nonlinear functions. Although there may be some truth in this if specifically logistic nonlinear functions are compared to linear functions, in the current paper it is shown that such a belief is not correct in general. It is shown that also classes of nonlinear functions exist that enable good analysis possibilities when it comes to the emerging dynamics within a network model. Such classes and the dynamics they enable are analysed here in some depth, thereby among others not using any conditions on the connectivity but instead exploiting for any network its structure of strongly connected components (Bloem et al., 2006; Fleischer et al., 2000; Harary et al., 1965; Łacki, 2013; Wijs et al., 2016).

As an example, following (Treur, 2020a) in the current paper a theorem is discussed specifying conditions under which all node states converge to the same value (e.g. achieving a common decision or belief within a group). This theorem does not impose any conditions on connectivity and for aggregation applies to some class of nonlinear functions as well as it applies to linear functions. Moreover, for some (but not all) of this class of 'well-behaving' nonlinear functions it is found out that they can be (indirectly) related to linear functions by some form of function transformation, which then enables application of linear analysis methods such as symbolically solving sets of linear equations including parameters.

In this paper, as application domain in particular, the domain of multilevel organizational learning is addressed (Crossan et al., 1999; Kim, 1993; Wiewiora et al., 2019, 2020). It is shown how the equilibrium analysis methods addressed in the current paper can be applied to adaptive self-modeling network models for multilevel organizational learning (Canbaloğlu, Treur, & Roelofsma, 2022a, 2022b; Canbaloğlu, Treur, & Wiewiora, 2022c). Predictions are obtained on the eventually achieved learning results in terms of the mental models learnt.

In this paper, in Section 2 the basics of the modeling and analysis approach used from (Treur, 2020b) are briefly introduced. In Sections 3–6 a number of mathematically proven results are presented on equilibrium analysis of network models. These results cover many variations concerning connectivity, aggregation and timing. In particular, these results address both linear and nonlinear types of aggregation. Section 3 addresses equilibrium analysis for a specific condition on connectivity (and no condition on aggregation or timing), namely the case of acyclic networks. It does so by introducing a form of stratification for acyclic networks, thus obtaining Theorem 1 and Corollary 1, which indeed do not require any condition on the functions used for aggregation or on the timing. In Section 4, equilibrium analysis is addressed for some specific conditions on aggregation (on no conditions on connectivity and timing). Some (comparative equilibrium analysis) results are obtained for cases of monotonicity of functions used for aggregation and comparison relations between a number of often considered specific types of such functions (scaled sum, Euclidean, geometric, logistic, minimium and maximum functions); here in addition to a number of propositions for different cases, Theorem 2 and Corollaries 2 and 3 are obtained. Section 5 addresses another condition on aggregation (and again no condition on connectivity or timing); it in addition focuses on the role of being scalar-free for the functions used for aggregation, leading to results for the more general class of monotonic and scalar-free functions formulated as Theorem 3.

In Section 6, again only conditions on aggregation are considered and no conditions on connectivity or timing. Following (Treur, 2020a) the general connectivity structure is analysed in some more depth by taking into account the strongly connected components of a network with their mutual connections and the condensation graph based on them, which is always acyclic; e.g. (Harary et al., 1965). By introducing a stratification of this condensation graph similar to the stratification that is introduced in Section 3 for acyclic networks, results are obtained that are to a certain extent similar to the results for acyclic networks: Theorem 4 and Corollaries 4 and 5. In contrast to Section 3, these results do assume some conditions on the aggregation: the functions for aggregation have to be strictly monotonic, scalar-free and normalized.

The main results presented in Section 3–6, are applied in Section 7 to obtain equilibrium analysis results for network models of multilevel organizational learning processes. Moreover, in Section 8 two other examples are discussed showing how equilibrium analysis based on some types of nonlinear functions can be done in practice by function transformations to linear functions and subsequently using a (symbolic) solver for linear equations. This function transformation has some similarity to the notion of coordinate transformation and can be described by conjugate relations between functions. This type of transformation is investigated in some more depth in Section 9, where a number of further results are obtained, described by Theorems 5 and 6 addressing further steps in the characterization of scalar-free functions that can be used for aggregation and Theorems 7 and 8 addressing conjugate relations between scalar-free functions and linear functions. Finally, Section 10 is a discussion and Section 11 is an Appendix with proofs of the presented results.

2. Modeling and analysis of dynamics and adaptation for networks

In this section, the underlying network-oriented modelling approach used is briefly discussed and in relation to this the basic concepts used for equilibrium analysis.

2.1. Modeling by dynamic and adaptive networks

Following (Treur, 2020b), a temporal-causal network model is specified by the following types of network characteristics (here *X* and *Y* denote nodes of the network, also called states, which have state values *X(t)* and *Y(t)* over time *t*):

- **Connectivity characteristics.**
  Connections from a state *X* to a state *Y* and *weights* $\boldsymbol{\omega}_{X,Y}$
- **Aggregation characteristics.**
  For any state *Y*, some *combination function* $\mathbf{c}_Y(V_1\textbf{, \dots, } V_k)$ defines the aggregation that is applied to the *single impacts* $V_i = \boldsymbol{\omega}_{X_i,Y}X_i(t)$ on *Y* from its incoming connections from states $X_1, \dots, X_k$.
- **Timing characteristics.**
  Each state *Y* has a *speed factor* $\boldsymbol{\eta}_Y$ defining how fast it changes.

The following canonical difference equation used for simulation and analysis purposes incorporates these network characteristics $\boldsymbol{\omega}_{X,Y}$, $\mathbf{c}_Y$, $\boldsymbol{\eta}_Y$ in a numerical format:

$$Y(t + \Delta t) = Y(t) \ + \boldsymbol{\eta}_Y[\mathbf{aggimpact}_Y(t) \ - \ Y(t)]\Delta t \tag{1}$$

where $\mathbf{aggimpact}_Y(t) \ = \ \mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t), \ \dots, \boldsymbol{\omega}_{X_k,Y}X_k(t))$ for any state *Y* and X_1 to X_k are the states from which *Y* gets its incoming connections. A combination function is called *normalised* if this aggregated impact is 1 if all state values in it are 1. This expresses the general principle that network dynamics is implied (or entailed) by the network's structure characteristics.

The timing characteristics specified by speed factors $\boldsymbol{\eta}_Y$ enable to model more realistic processes for which not all states change in a synchronous manner. Network models that do not possess this option are less flexible as they silently impose synchronous processing as an artefact. The aggregation characteristics specified by the choice of combination functions $\mathbf{c}_Y$ and their parameters provide another form of flexibility to fit better to specific realistic applications. Also in this case, network models that do not possess such an option are less flexible and also silently impose artefacts that may make them fit less to specific applications. For example, for aggregation in social networks often only linear functions are used for aggregation.

The above concepts enable to design network models and their dynamics in a declarative manner, based on mathematically defined functions and relations. Realistic network models are usually adaptive: often some of their network characteristics change over time. By using *self-modeling networks* (or *network reification*), a similar network-oriented conceptualization can also be applied to *adaptive* networks to obtain a declarative description using mathematically defined functions and relations for them as well; see (Treur, 2020b). This works through the addition of new states to the network (called *self-model states* or *reification states*) which represent network characteristics by network states. If such self-model states are dynamic, they describe adaptive network characteristics. In a graphical 3D-format (e.g. see Secion 7), such self-model states are depicted at a next level (*self-model level* or *reification level*), where the original network is at a *base level*. As an example, the weight $\boldsymbol{\omega}_{X,Y}$ of a connection from state X to state Y can be represented (at a next reification level) by a self-model state named $\mathbf{W}_{X,Y}$. During processing based on the canonical difference equation (1), the value of this state $\mathbf{W}_{X,Y}$ is used as the connection weight $\boldsymbol{\omega}_{X,Y}$ it represents. Similarly, all other network characteristics from $\boldsymbol{\omega}_{X,Y}$, $\mathbf{c}_Y(..)$, $\boldsymbol{\eta}_Y$ can be made adaptive by including self-model states for them. As a self-modeling network model is also a temporal-causal network model itself, as has been shown in (Treur, 2020b), Ch 10, this self-modeling construction can easily be applied iteratively to obtain multiple self-model levels.

This self-modeling network construction can provide higher-order adaptive network models, and has turned out quite useful to model, for example, plasticity and metaplasticity in the form of a second-order adaptive mental network with three levels, one base level and a first-order self-model level for adaptation of connections and a second-order self-model level for control over such adaptation; e.g. (Abraham & Bear, 1996) and (Treur, 2020b), Ch 4. Recently, a three-level self-modeling network architecture has also been adopted to successfully model adaptation of internal mental models and its control (Treur & Van Ments, 2022; Van Ments & Treur, 2021) and to model organizational learning and its control (Canbaloğlu, Treur, & Roelofsma, 2022a, 2022b; Canbaloğlu, Treur, & Wiewiora, 2022a, 2022b). For the latter, see also Section 7.

2.2. Basic concepts for equilibrium analysis of dynamic and adaptive networks

The following types of properties are often considered for equilibrium analysis of dynamical systems in general.

Definition (stationary point, increasing, decreasing, equilibrium)

Let Y be a network state.

- Y has a *stationary point* at t if $\mathbf{d}Y(t)/\mathbf{d}t = 0$
- Y is *increasing* at t if $\mathbf{d}Y(t)/\mathbf{d}t > 0$
- Y is *decreasing* at t if $\mathbf{d}Y(t)/\mathbf{d}t < 0$
- The network model is in *equilibrium* at t if every state Y of the model has a stationary point at t.

For network models, the following criteria in terms of the network characteristics $\boldsymbol{\omega}_{X,Y}$, $\mathbf{c}_Y$, $\boldsymbol{\eta}_Y$ can be derived from the generic difference equation (1); see also (Treur, 2016, 2018):

Criteria for network model dynamics

Let Y be a state and $X_1, \ldots, X_k$ the states connected toward Y. For nonzero speed factors $\boldsymbol{\eta}_Y$ the following criteria in terms of network characteristics for connectivity and aggregation apply; here $\mathbf{aggimpact}_Y(t) = \mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t), \ \ldots, \ \boldsymbol{\omega}_{X_k,Y}X_k(t))$:

- Y has a stationary point at t $\Leftrightarrow$ $\mathbf{aggimpact}_Y(t) = Y(t)$
- Y is increasing at t $\Leftrightarrow$ $\mathbf{aggimpact}_Y(t) \ > Y(t)$

- Y is decreasing at t ⇔ $\mathbf{aggimpact}_Y(t) < Y(t)$
- The network model is in equilibrium a t ⇔ $\mathbf{aggimpact}_Y(t) = Y(t)$ for every state Y

The above criteria for a network being in an equilibrium (assuming nonzero speed factors) depend both on the connections weights $\boldsymbol{\omega}_{X,Y}$ used for connectivity and on the combination function $\mathbf{c}_Y$ used for aggregation. Note that in a self-modeling network, these criteria can be applied not only to base states but also to self-model states. In the latter case they can be used for equilibrium analysis of learning processes, as will be illustrated for organizational learning in Section 7.

In subsequent sections the equilibrium analysis is addressed not at the level of specific network structures and implied dynamics but at a more abstract level of properties of network structures and properties of dynamics implied by them. More specifically, in the remainder of this paper, it will be analysed how the criteria relate to certain properties of the connectivity characteristics and aggregation characteristics:

- For connectivity characteristics:
  how the criteria relate to properties of paths based on connections, such as
  ○ whether the network is acyclic or cyclic
  ○ for cyclic networks, the way in which the network is composed of its strongly connected components (the condensation graph of the network)
- For aggregation characteristics:
  how the criteria relate to properties of the combination functions defining the network's aggregation, such as
  ○ monotonicity
  ○ being scalar-free
  ○ comparison relations between combination functions

These properties will not only apply to linear functions but also to a wider class of functions extending the class of linear functions beyond the border with the class of nonlinear functions. Exploring nonlinear functions in this class and how some of them still may relate to linear functions is one of the main aims of the current paper.

3. Equilibrium analysis under connectivity conditions: acyclic networks

In the current section a relatively simple case will be addressed where a condition on the connectivity in the network (but no conditions on aggregation in the network) is considered: the case of acyclic networks.

3.1. Stratification for acyclic graphs or networks

A relatively simple but still very useful structure that can be added to any acyclic graph or network is the following form of stratification.

Definition (stratification for an acyclic graph or network)

For an acyclic graph or network, *stratification levels* 0, 1, .. are (inductively) assigned to the nodes such that the following hold:

- For a node Y without incoming connections from other nodes: level(Y) = 0
- For a node Y with incoming connections from nodes $X_1, \ldots, X_k$: level(Y) = 1 + max$_i$(level(X_i))

A simple example of an acyclic network with 7 states is shown in Figure 1. Based on their connectivity, the four indicated stratification levels are obtained. Note that for each state Y, the longest path from any level 0 state to Y determines its stratification level. For example, in Figure 1 state X_6 has level 2 since its longest path from any level 0 state is from X_2 via X_5, and X_7 has level 3 since its longest path from a level 0 state is from X_2 via X_5 and X_6.

3.2. Using stratification for equilibrium analysis of acyclic networks

Stratification is a useful instrument to analyse equilibria of acyclic networks; the following theorem can easily be obtained. It shows how for acyclic networks equilibrium values of states (with nonzero speed factor) for all levels $i > 0$ depend on equilibrium values of states at a lower level < i. This dependency across levels can directly be expressed by a mathematical function expression using the network characteristics for connectivity (the connection weights $\boldsymbol{\omega}_{X,Y}$) and aggregation (the combination functions $\mathbf{c}_Y(\ldots)$); see Section 11 for proofs of all results in this paper.

Theorem 1 (relating equilibrium values for an acyclic network from different stratification levels)

Suppose a network is acyclic and all states with incoming connections from other states have nonzero speed factors. Then the following hold.

a) In any equilibrium for each state Y of any stratification level $i > 0$, its equilibrium value $\mathbf{Y}$ depends by some mathematical function on the equilibrium values $\mathbf{X}$ of states X of level < i.

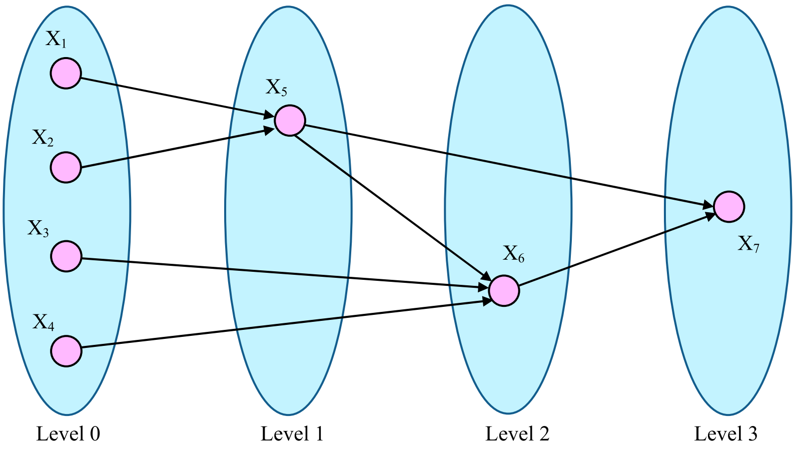


Figure 1. Example acyclic network with connectivity that induces the indicated stratification levels.

b) More specifically, in any equilibrium for any state Y of stratification level $i > 0$, its equilibrium value $\mathbf{Y}$ can be determined from equilibrium values $\mathbf{X}_j$ of states X_j at lower levels $< i$ by:

$$\mathbf{Y} = \mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}\mathbf{X}_1, \ \ldots, \ \boldsymbol{\omega}_{X_k,Y}\mathbf{X}_k)$$

By iterating the dependency relations across stratification levels described in Theorem 1, the equilibrium values of all states from all levels can be related to equilibrium values of states at level 0. This dependency can be described again by a mathematical function expression using the network characteristics for connectivity (the connection weights $\boldsymbol{\omega}_{X,Y}$) and aggregation (the combination functions $\mathbf{c}_Y(\ \ldots\ )$). This is expressed in Corollary 1.

Corollary 1 (relating all equilibrium values for an acyclic network to those of the level 0 states)

Suppose a network is acyclic and all states with incoming connections from other states have nonzero speed factors. Then the following hold.

a) By applying Theorem 1b) iteratively according to the stratification levels, in a straightforward manner for each state Y of the network, a mathematical expression can be obtained showing how its equilibrium value depends on the equilibrium values of states of level 0.
b) The mathematical expression in a) defines a mathematical function for $\mathbf{Y}$ in terms of the equilibrium values $\mathbf{X}$ of some states X of level 0 with as parameters connectivity and aggregation characteristics $\boldsymbol{\omega}_{Z1,Z2}$ and $\mathbf{c}_Z(..)$ of the network relating to states Z, Z_1, Z_2 on the paths from the involved level 0 states X to state Y. This mathematical function essentially is based on an iterated composition of combination functions of the states on the paths to Y in the network, nested according to the (inverse) branching structure of these paths to Y.

Note that although the mathematical functions to describe the dependencies for equilibrium values still can be expressed directly based on the connectivity and aggregation characteristics $\boldsymbol{\omega}_{X,Y}$ and $\mathbf{c}_Y(\ \ldots\ )$ of the network, in Corollary 1 they get a more complex, nested structure. First, for the example acyclic network of Figure 1 (assuming the combination function **alogistic**; for a definition of this function, see Section 4.2), by Theorem 1 the following relations between the equilibrium values of states at stratification levels 0–3 are obtained (here the $\mathbf{X}_i$ are the equilibrium values of states X_i):

Level 3 equilibrium value $\mathbf{X}_7$: dependence on Level 1 and 2 equilibrium values $\mathbf{X}_5$ and $\mathbf{X}_6$

$$\mathbf{X}_7 = \mathbf{alogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(\mathbf{X}_5, \mathbf{X}_6)$$

Level 2 equilibrium value $\mathbf{X}_6$: dependence on Level 0 and Level 1 equilibrium values $\mathbf{X}_3$, $\mathbf{X}_4$, and $\mathbf{X}_5$

$$\mathbf{X}_6 = \mathbf{alogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(\mathbf{X}_3, \mathbf{X}_4, \mathbf{X}_5)$$

Level 1 equilibrium value $\mathbf{X}_5$: dependence on Level 0 equilibrium values

$$\mathbf{X}_5 = \mathbf{alogistic}_{\sigma,\tau}(\mathbf{X}_1, \mathbf{X}_2)$$

Next, applying the iteration indicated in Corollary 1, this leads to the following functions for how the equilibrium values for the level 2 and 3 states X_6 and X_7 depend on the ones of the level 0 states:

$$\begin{aligned}
\mathbf{X}_6 &= \mathbf{alogistic}_{\sigma,\tau}(\mathbf{X}_3, \mathbf{X}_4, \mathbf{X}_5)\\
&= \mathbf{alogistic}_{\sigma,\tau}(\mathbf{X}_3, \mathbf{X}_4, \mathbf{alogistic}_{\sigma,\tau}(\mathbf{X}_1, \mathbf{X}_2))\\
\mathbf{X}_7 &= \mathbf{alogistic}_{\sigma,\tau}(\mathbf{X}_5, \mathbf{X}_6)\\
&= \mathbf{alogistic}_{\sigma,\tau}(\mathbf{alogistic}_{\sigma,\tau}(\mathbf{X}_1, \mathbf{X}_2), \mathbf{alogistic}_{\sigma,\tau}(\mathbf{X}_3, \mathbf{X}_4, \mathbf{X}_5))\\
&= \mathbf{alogistic}_{\sigma,\tau}(\mathbf{alogistic}_{\sigma,\tau}(\mathbf{X}_1, \mathbf{X}_2), \mathbf{alogistic}_{\sigma,\tau}(\mathbf{X}_3, \mathbf{X}_4, \mathbf{alogistic}_{\sigma,\tau}(\mathbf{X}_1, \mathbf{X}_2)))
\end{aligned}$$

Note that these are indeed nested combination functions according to the paths in the network to X_6 and X_7. This illustrates how in an acyclic network, the equilibrium values of all states of the entire network are determined by the equilibrium values of the level 0 states. In Section 7, other examples of expressions of nested combination functions as indicated in Corollary 1 will be shown for the application to a network model for organizational learning.

Note also that for realistic domains, networks are often not acyclic: usually they include at least some cycles or even many of them. Then the above Theorem 1 and Corollary 1 are not applicable to the network as a whole. However, even for such cyclic networks, sometimes it can be useful to consider subnetworks that still are acyclic and apply the above Theorem 1 and Corollary 1 to them. As an example, this will be illustrated for the application to organizational learning addressed in Section 7.

Moreover, following (Treur, 2020a) in Section 6 it will be shown how the approach based on stratification applied for Theorem 1 and Corollary 1 to the nodes of the (acyclic) network can also be applied not to the nodes but to (the condensation graph of) the strongly connected components of any network. In that section, some further results are obtained for networks with any type of (possibly cyclic) connectivity. The results there also show relations between equilibrium values of states from different stratification levels (and with the states at level 0) and in that sense are to a certain extent similar to those of Theorem 1 and Corollary 1 but much more general.

4. Equilibrium analysis under aggregation conditions: monotonicity and comparison for combination functions

In this section, some conditions on the aggregation in the network (but no conditions on the connectivity in the network) are considered. More specifically, it is explored how specific properties of the type of aggregation used in a network model enable to derive some further results for equilibrium analysis. As aggregation characteristics of a network model are defined by combination functions, this means that certain properties of these functions are considered here. In particular, it is discussed how monotonicity of combination functions and comparison (order) relations between them can be used to obtain specific (comparative) equilibrium analysis results. As the obtained results do not assume any conditions on the connectivity of the network, they apply both to acyclic and cyclic networks.

4.1. Equilibrium analysis using monotonicity and comparison relations for aggregation

The following monotonicity and comparison relations for the functions used for aggregation are considered.

Definition (monotonicity and comparison of functions)

Let a subset $R \subseteq \mathbb{R}$ be given.

a) A function f: $R^k \rightarrow \mathbb{R}$ is called *(monotonically) increasing* if for all $U_1, \ldots, U_k, V_1, \ldots, V_k \in R$ such that $U_i \leq V_i$ for all i it holds $f(U_1, \ldots, U_k) \leq f(V_1, \ldots, V_k)$.

b) A function f: $R^k \rightarrow \mathbb{R}$ is called *strictly (monotonically) increasing* if for all $U_1, \ldots, U_k, V_1, \ldots, V_k \in R$ such that $U_i \leq V_i$ for all i and $U_j < V_j$ for at least one j it holds $f(U_1, \ldots, U_k) < f(V_1, \ldots, V_k)$.

c) For two functions f,g: $R^k \rightarrow \mathbb{R}$, by $f \leq g$ the *function comparison relation* is denoted that for all $V_1, \ldots, V_k \in R$ it holds $f(V_1, \ldots, V_k) \leq g(V_1, \ldots, V_k)$

When these general properties of mathematical functions are applied in particular to the combination functions defining the aggregation characteristics of a network model, for any network model with any type of connectivity characteristics, the following theorem on equilibria can be derived.

Theorem 2 (preservation of comparison relations over time and for equilibria)

Suppose X_i are the states of a network model (with only positive connection weights and at least some nonzero speed factors) and all are using monotonically increasing combination functions $\mathbf{c}_i$. Assume $0 < \Delta t \leq 1/\max_Y(\boldsymbol{\eta}_Y)$; e.g. assume $\boldsymbol{\eta}_Y \leq 1$ for all Y and $0 < \Delta t \leq 1$. Then the following hold.

a) Suppose two simulation traces $X_i(t)$ and $X'_i(t)$ are given with initial values $X_i(0) \leq X'_i(0)$. Then it holds $X_i(t) \leq X'_i(t)$ for all t and i and for any achieved equilibrium, for the equilibrium values $\mathbf{X}_i$ and $\mathbf{X'}_i$ of X_i and X'_i it holds $\mathbf{X}_i \leq \mathbf{X'}_i$ for all i.

b) Moreover, suppose X'_i are again the states of the same network model but this time using monotonically increasing combination functions $\mathbf{c'}_i$. Then the following hold:
   (i) If $\mathbf{c}_i \leq \mathbf{c'}_i$ for all i and for the initial values it holds $X_i(0) \leq X'_i(0)$ for all i, then it holds $X_i(t) \leq X'_i(t)$ for all t and i.
   (ii) If $\mathbf{c}_i \leq \mathbf{c'}_i$ for all i and for the initial values it holds $X_i(0) \leq X'_i(0)$ for all i, then for any achieved equilibrium for all i for the equilibrium values $\mathbf{X}_i$ and $\mathbf{X'}_i$ of X_i and X'_i it holds $\mathbf{X}_i \leq \mathbf{X'}_i$.

4.2. Equilibrium analysis based on monotonicity and comparison for specific functions

Next, for a number of often used types of combination functions, which all are monotonically increasing, their comparison (order) relations are identified, so that it becomes clear how Theorem 2 can be applied to them for equilibrium analysis.

Definition (weighted euclidean functions, weighted geometric functions, logistic functions, and max and min functions)

a) A function g is a *weighted euclidean function* of order n if

$$g(V_1, \ldots, V_k) = \sqrt[n]{w_1 V_1^n + \ldots + w_k V_k^n}$$

for some weights w_1, .., w_k. If the sum of its weights is 1, it is called a *weighted euclidean average function*. A weighted euclidean function of order $n = 1$ is called a *linear function*.

b) A function g is a *weighted geometric function* if

$$g(V_1, \ldots, V_k) = V_1^{w_1} \; \ldots . \; V_k^{w_k}$$

for some weights w_1, .., w_k. If the sum of its weights is 1, it is called a *weighted geometric mean function*.

c) The *scaled euclidean function* $\mathbf{eucl}_{n,\boldsymbol{\lambda}}$ of order n is defined by

$$\mathbf{eucl}_{n,\boldsymbol{\lambda}}(V_1, \ldots, V_k) = \sqrt[n]{\frac{V_1^{\boldsymbol{n}} + \ldots + V_k^{\boldsymbol{n}}}{\boldsymbol{\lambda}}}$$

and the *scaled geometric mean function* $\mathbf{sgeomean}_{\boldsymbol{\lambda}}$ is defined by

$$\mathbf{sgeomean}_{\boldsymbol{\lambda}}(V_1, \ldots, V_k) = \sqrt[k]{\frac{V_1 * \ldots * V_k}{\boldsymbol{\lambda}}}$$

Moreover, the *scaled sum function* is defined as

$$\mathbf{ssum}_{\boldsymbol{\lambda}}(V_1, \ldots, V_k) = \mathbf{eucl}_{1,\boldsymbol{\lambda}}(V_1, \ldots, V_k)$$

When $\lambda = 1$, the latter two are also denoted by **geomean** and **sum**.

d) The simple and advanced logistic functions $\mathbf{slogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}$ and $\mathbf{alogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}$ are defined by

$$\mathbf{slogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(V_1, \ldots, V_k) = \frac{1}{1 + \mathrm{e}^{-\boldsymbol{\sigma}(V_1 + \ldots + V_k - \boldsymbol{\tau})}}$$

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

e) The scaled *minimum* and *maximum functions* **smin** and **smax** are defined by

$$\mathbf{smin}_{\boldsymbol{\lambda}}(V_1, \ldots, V_k) = min(V_1, \ldots, V_k)/\boldsymbol{\lambda}$$

$$\mathbf{smax}_{\boldsymbol{\lambda}}(V_1, \ldots, V_k) = max(V_1, \ldots, V_k)/\boldsymbol{\lambda}$$

When $\lambda = 1$, they are also denoted by **min** and **max**.

All above-defined functions are monotonically increasing in $V_1, \ldots, V_k$, as they are built in a suitable way as compositions of basic monotonic functions such as sum, product and division functions, power functions and exponential functions. Note that as $\mathbf{alogistic}_{\sigma,\tau}\mathbf{(}0, \ldots, 0\mathbf{)} = 0$, from this it follows in particular that $\mathbf{alogistic}_{\sigma,\tau}\mathbf{(}V_1, \ldots, V_k\mathbf{)} \geq 0$ for all $V_1, \ldots, V_k \geq 0$.

Next, in subsequent propositions some comparison relations between these functions are identified, first for the logistic functions.

Proposition 1 (comparison for logistic functions)

a) Suppose $\tau' < \tau$ and $\sigma > 0$. Then for any $V_1, \ldots, V_k \geq 0$ it holds

$$0 \leq \mathbf{alogistic}_{\sigma,\tau}(V_1, \ldots, V_k) < \mathbf{slogistic}_{\sigma,\tau}(V_1, \ldots, V_k) < \mathbf{slogistic}_{\sigma,\tau'}(V_1, \ldots, V_k) < 1$$

b) Moreover, for any $\sigma > 0$, and $V_1, \ldots, V_k \geq 0$ it holds

$$\lim_{\tau \to \infty} \mathbf{alogistic}_{\sigma,\tau}(V_1, \ldots, V_k) = \lim_{\tau \to \infty} \mathbf{slogistic}_{\sigma,\tau}(V_1, \ldots, V_k) = 0$$

$$\lim_{\tau \to -\infty} \mathbf{slogistic}_{\sigma,\tau}(V_1, \ldots, V_k) = 1$$

Next comparison relations between Euclidean functions and minimum and maximum functions are identified.

Proposition 2 (comparison between euclidean and min and max functions)

a) Suppose the scaling factor is set at $\lambda = k$, then for any $V_1, \ldots, V_k \geq 0$ it holds

$$\mathbf{min}(V_1, \ldots, V_k) \leq \mathbf{eucl}_{n,k}(V_1, \ldots, V_k) \leq \mathbf{max}(V_1, \ldots, V_k)$$

and

$$\lim_{n \to \infty} \mathbf{eucl}_{n,k}(V_1, \ldots, V_k) = \mathbf{max}(V_1, \ldots, V_k)$$

b) More in general, for any $\lambda > 0$ it holds

$$\mathbf{min}(V_1, \ldots, V_k).\sqrt[n]{\frac{k}{\lambda}} \leq \mathbf{eucl}_{n,\lambda}(V_1, \ldots, V_k) \leq \mathbf{max}(V_1, \ldots, V_k).\sqrt[n]{\frac{k}{\lambda}}$$

and

$$\lim_{n \to \infty} \mathbf{eucl}_{n,\lambda}(V_1, \ldots, V_k) = \mathbf{max}(V_1, \ldots, V_k)$$

Similarly, comparison relations between geometric functions and minimum and maximum functions are identified.

Proposition 3 (comparison between geometric mean and min and max functions)

a) Suppose the scaling factor is set at $\lambda = 1$, then for any $V_1, \ldots, V_k \geq 0$ it holds

$$\mathbf{min}(V_1, \ldots, V_k) \leq \mathbf{sgeomean}_{\lambda}(V_1, \ldots, V_k) \leq \mathbf{max}(V_1, \ldots, V_k)$$

b) More in general, for any $\lambda > 0$ and any $V_1, \ldots, V_k \geq 0$ it holds

$$\mathbf{min}(V_1, \ldots, V_k)\sqrt[k]{\frac{1}{\lambda}} \leq \mathbf{sgeomean}_{\lambda}(V_1, \ldots, V_k) \leq \mathbf{max}(V_1, \ldots, V_k).\sqrt[k]{\frac{1}{\lambda}}$$

Based on the comparison relations for combination functions identified in the three above propositions, Theorem 2 can be applied. As an example, in this way the following two corollaries of Theorem 2 are obtained on comparative equilibrium analysis: they compare equilibrium values obtained for various types of functions.

Corollary 2 (comparison relations for equilibrium values: logistic and sum functions)

Assume only positive connection weights and at least some nonzero speed factors and $\Delta t \leq 1/\text{max}_Y(\boldsymbol{\eta}_Y)$ (e.g. assume $\boldsymbol{\eta}_Y \leq 1$ for all Y and $\Delta t \leq 1$).

a) Suppose X_i are the states for a network model using advanced logistic combination functions $\mathbf{c}_i = \mathbf{alogistic}$ and X'_i for the same network model using simple logistic combination functions $\mathbf{c}'_i = \mathbf{slogistic}$ with the same parameters $\boldsymbol{\sigma}_i$ and $\boldsymbol{\tau}_i$ for each state X_i. Moreover, suppose two simulation traces $X_i(t)$ and $X'_i(t)$ are given with initial values $X_i(0) \leq X'_i(0)$ for all i, then for any achieved equilibrium with equilibrium values $\mathbf{X}_i$ and $\mathbf{X'}_i$ it holds $\mathbf{X}_i \leq \mathbf{X'}_i$ for all i.

b) Suppose X_i are the states for a network model using simple logistic combination functions $\mathbf{c}_i = \mathbf{slogistic}$ with parameters $\boldsymbol{\sigma}_i$ and $\boldsymbol{\tau}_i$ and X'_i for the same network model using simple logistic combination functions $\mathbf{c}'_i = \mathbf{slogistic}$ with the parameters $\boldsymbol{\sigma}_i$ and $\boldsymbol{\tau}'_i$ for each state X_i such that $\boldsymbol{\tau}'_i \leq \boldsymbol{\tau}_i$. Moreover, suppose two simulation traces $X_i(t)$ and $X'_i(t)$ are given with initial values $X'_i(0) \leq X_i(0)$ for all i, then for any achieved equilibrium with equilibrium values $\mathbf{X}_i$ and $\mathbf{X'}_i$ it holds $\mathbf{X}_i \leq \mathbf{X'}_i$ for all i.

c) Suppose X_i are the states for a network model using *advanced* logistic combination functions $\mathbf{c}_i = \mathbf{alogistic}$ with parameters $\boldsymbol{\sigma}_i$ and $\boldsymbol{\tau}_i$ and X'_i for the same network model using *scaled sum* combination functions $\mathbf{c}'_i = \mathbf{ssum}$ with the parameters λ'_i for each state X_i. Moreover, suppose two simulation traces $X_i(t)$ and $X'_i(t)$ are given with initial values $X'_i(0) \leq X_i(0)$ for all i. If $0 \leq \lambda'_i \leq \boldsymbol{\tau}_i$ or $\lambda'_i \leq 2\ \mathbf{min}(\boldsymbol{\sigma}_i, \boldsymbol{\tau}_i)$, and an equilibrium is achieved with equilibrium values $\mathbf{X}_i$ and $\mathbf{X'}_i$ then it holds $\mathbf{X}_i \leq \mathbf{X'}_i$ for all i.

Corollary 3 (comparison relations for equilibrium values: Euclidean, geometric, minimum and maximum functions)

Assume only positive connection weights and at least some nonzero speed factors and $\Delta t \leq 1/\text{max}_Y(\boldsymbol{\eta}_Y)$ (e.g. assume $\boldsymbol{\eta}_Y \leq 1$ for all Y and $\Delta t \leq 1$).

a) Suppose X_i are the states for a network model using as combination functions $\mathbf{c}_i$ Euclidean combination functions $\mathbf{eucl}_{n,k}(V_1, \ldots, V_k)$ with scaling factor $\lambda = k$ or geometric mean combination functions $\mathbf{sgeomean}_{\lambda}(V_1, \ldots, V_k)$ with scaling factor $\lambda = 1$ and

X'_i for the same model using *maximum* combination functions $\mathbf{c}'_i$. If for the initial values it holds $X_i(0) \leq X'_i(0)$ for all i, then for any achieved equilibrium with equilibrium values $\mathbf{X}_i$ and $\mathbf{X}'_i$ it holds $\mathbf{X}_i \leq \mathbf{X}'_i$ for all i.

b) Suppose X_i are the states for a network model using as combination functions $\mathbf{c}_i$ Euclidean combination functions $\mathbf{eucl}_{n,k}(V_1, \ldots, V_k)$ with scaling factor $\lambda = k$ or geometric mean combination functions $\mathbf{sgeomean}_\lambda(V_1, \ldots, V_k)$ with scaling factor $\lambda = 1$ and X'_i for the same model using combination functions *minimum* functions $\mathbf{c}'_i$. Moreover, suppose two simulation traces $X_i(t)$ and $X'_i(t)$ are given with initial values $X'_i(0) \leq X_i(0)$ for all i. If an equilibrium is achieved with equilibrium values $\mathbf{X}_i$ and $\mathbf{X}'_i$, then it holds $\mathbf{X}'_i \leq \mathbf{X}_i$ for all i.

5. Equilibrium analysis under aggregation conditions: scalar-freeness

In this section, equilibrium analysis is addressed for networks satisfying another aggregation condition: the combination functions are assumed scalar-free.

5.1. Functions for aggregation that are scalar-free

It is sometimes believed that for dynamical models the borderline between linear and nonlinear functions is also the borderline between well-analyzable behavior and less well-analyzable behavior. In contrast to this, it has been found that this borderline between well-analyzable behavior and less well-analyzable behavior lies somewhere within the domain of nonlinear functions: between one class (called monotonic scalar-free functions) covering both linear and nonlinear functions and another subclass of the class of nonlinear functions not satisfying these.

More specifically, whether or not combination functions are scalar-free is an important factor determining whether or not by social contagion all members of a well-connected social network converge to the same level of emotion, opinion, information, belief, intention, or any other mental or physical state; e.g. (Treur, 2020a) and (Treur, 2020b), Ch 11 and 12. The class of scalar-free functions includes all linear functions but also includes a number of types of nonlinear functions, such as the weighted euclidean functions and weighted geometric functions. In this section some further analysis is made of scalar-free functions, thereby also using a weakened variant of them called weakly scalar-free functions. The definitions are as follows.

Definition (weakly scalar-free and scalar-free functions)

Consider functions f: $R^k \rightarrow \mathbb{R}$ and θ: $R \rightarrow \mathbb{R}$ for some subset $R \subseteq \mathbb{R}$ which is $\mathbb{R}$ or $\mathbb{R}_{>0}$.

a) A function f: $R^k \rightarrow \mathbb{R}$ is called *weakly scalar-free* for function θ if for all $V_1, \ldots, V_k \in R$ and all $\alpha \in R$ it holds $f(\alpha V_1, \ldots, \alpha V_k) = \theta(\alpha)\ f(V_1, \ldots, V_k)$

b) A function f: $R^k \rightarrow \mathbb{R}$ is called *scalar-free* if for all $V_1, \ldots, V_k \in R$ and all $\alpha \in R$ it holds $f(\alpha V_1, \ldots, \alpha V_k) = \alpha\ f(V_1, \ldots, V_k)$

Examples (weakly scalar-free functions)

There are many examples of weakly scalar-free functions. For example, the following functions $f(V) = V^k$ and $f(V_1, \ldots, V_k) = V_1{}^* \ldots {}^*V_k$ on proper domains are weakly scalar-free with function $\theta(\alpha) = \alpha^k$. The example $f(V_1, V_2, V_3) = w_1\, V_1V_2 + w_2\, V_2V_3 + w_3\, V_3V_1$ is weakly scalar-free with function $\theta(\alpha) = \alpha^2$.

5.2. Properties and comparative equilibrium analysis for scalar-free functions

The following basic properties can easily be verified (see Section 11 for proofs).

Proposition 4 (scalar-free and increasing functions)

a) Suppose f is increasing and weakly scalar-free for function θ. Then for all $V_1, \ldots, V_k \in R$ it holds

$f(1, \ldots, 1)\ \theta(\min(V_1, \ldots, V_k)) \leq f(V_1, \ldots, V_k) \leq f(1, \ldots, 1)\theta(\max(V_1, \ldots, V_k))$
When moreover f is scalar-free, then
$f(1, \ldots, 1)\ \min(V_1, \ldots, V_k) \leq f(V_1, \ldots, V_k) \leq f(1, \ldots, 1)\ \max(V_1, \ldots, V_k)$

b) Any weighted Euclidean or geometric function with positive weights w_i is monotonically increasing and scalar-free.

c) If

$$g(V_1, \ldots, V_k) = \sqrt[n]{w_1V_1^n + \ldots + w_kV_k^n}$$

is a weighted Euclidean function with positive weights w_i, then g is scalar-free and increasing; moreover, it holds

$\sqrt[n]{w_1 + \ldots + w_k}\min(V_1, \ldots, V_k) \leq g(V_1, \ldots, V_k) \leq \sqrt[n]{w_1 + \ldots + w_k}\max(V_1, \ldots, V_k)$

If g is a weighted Euclidean average function (i.e. the sum of the weights w_i is 1), then

$\min(V_1, \ldots, V_k) \leq g(V_1, \ldots, V_k) \leq \max(V_1, \ldots, V_k)$

d) If

$$g(V_1, \ldots, V_k) = V_1^{w_1} \ldots . V_k^{w_k}$$

is a weighted geometric function with positive weights w_i, then g is scalar-free and increasing; moreover, it holds.

$\min(V_1, \ldots, V_k) \leq g(V_1, \ldots, V_k) \leq \max(V_1, \ldots, V_k)$

Note that Proposition 4 places some of the specific types of functions considered in Section 4 (weighted Euclidean and weighted geometric functions) in the wider perspective of scalar-free functions.

Proposition 5 (scalar-free and strictly increasing functions)

a) Any function composition of scalar-free functions is scalar-free
b) Any function composition of strictly increasing functions is strictly increasing
c) All linear functions with positive coefficients are scalar-free and strictly increasing
d) Any scalar-free function f is weakly scalar-free for $\theta = \text{id}$, the identity function.

Theorem 3 (comparison for any scalar-free function to min and max)

Suppose X_i are the states for a network model (with only positive connection weights and at least some nonzero speed factors) using monotonically increasing combination functions $\mathbf{c}_i$ and X'_i for the same model using monotonically increasing combination functions $\mathbf{c'}_i$. Assume $0 < \Delta t \leq 1/\max_Y(\boldsymbol{\eta}_Y)$; e.g. assume $\boldsymbol{\eta}_Y \leq 1$ for all Y and $0 < \Delta t \leq 1$.

a) If for the initial values it holds $X_i(0) \leq X'_i(0)$, the $\mathbf{c}_i$ are monotonically increasing and scalar-free with $\mathbf{c}_i(1, \ldots, 1) = 1$, and $\mathbf{c}'_i$ is the maximum function $\mathbf{c}'_i = \mathbf{max}$ then for any achieved equilibrium with equilibrium values $\mathbf{X}_i$ and $\mathbf{X}'_i$ it holds $\mathbf{X}_i \leq \mathbf{X}'_i$ for all i.

b) If for the initial values it holds $X'_i(0) \leq X_i(0)$, the $\mathbf{c}_i$ are monotonically increasing and scalar-free with $\mathbf{c}_i(1, \ldots, 1) = 1$, and each $\mathbf{c}'_i$ is the minimum function $\mathbf{c}'_i = \mathbf{min}$ then for any achieved equilibrium with equilibrium values $\mathbf{X}_i$ and $\mathbf{X}'_i$ it holds $\mathbf{X}'_i \leq \mathbf{X}_i$ for all i.

Note that Theorem 3 generalizes to the class of scalar-free functions, the comparative equilibrium analysis that was found in Section 4 for specific examples of scalar-free functions: weighted Euclidean and weighted geometric functions.

6. Equilibrium analysis under aggregation conditions: using the strongly connected components

In this section an equilibrium analysis approach is discussed that takes into account how the network is composed of its strongly connected components.

6.1. Introducing stratification for the stronly connected components of a network

As an illustration, consider the example of a mental network model with connectivity depicted in Figure 2. This is a mental network model for how a person is sensing (sensor state ss_s) a stimulus s in the world (word state ws_s), represents this (representation state srs_s), and is triggered to prepare (preparation state ps_a) and perform (execution state es_a) action a, after evaluation of the predicted (predicted effect representation state srs_e) effect e of this action. In simulations it can be seen that as a result of a constant value a of stimulus ws_s all state values are increasing until they reach an equilibrium value a as well. The question then is whether these observations based on one or more simulation

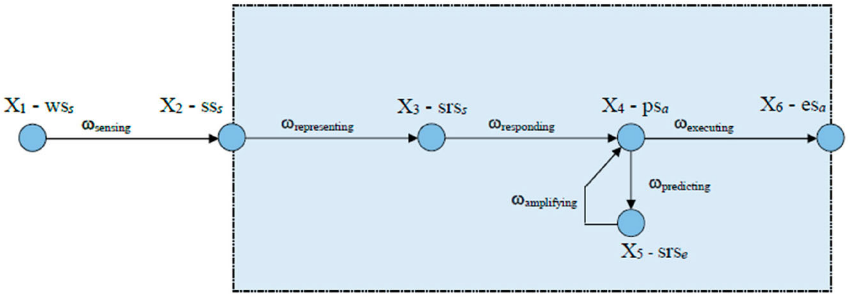


Figure 2. Connectivity of the example network model.

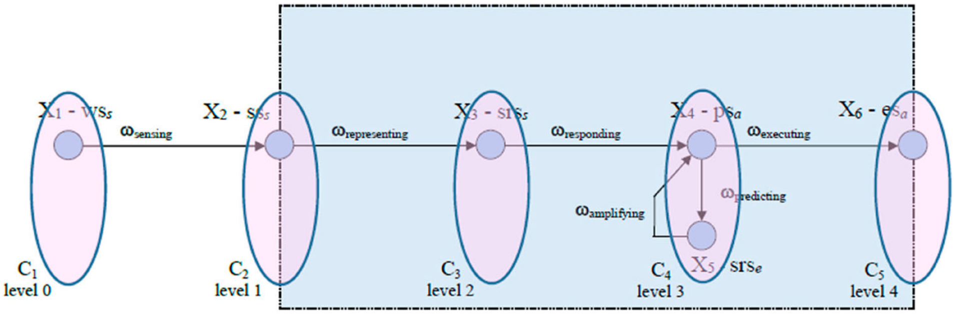


Figure 3. Stratified strongly connected components for the example network model.

experiments are in agreement with a mathematical equilibrium analysis. This will be addressed in two ways.

In the current section a general perspective is followed and theorems are discussed that have been found based on the network's strongly connected components described in (Treur, 2020a). The perspective is based on the notion of *(strongly connected) component* of a network; this is a maximal subnetwork *C* such that every node within *C* can be reached from every other node via a path following the direction of the connections; e.g. (Bloem et al., 2006; Fleischer et al., 2000; Harary et al., 1965; Łacki, 2013; Wijs et al., 2016). From this literature, it is known that these components partition the set of nodes in disjoint subsets and the connections between them induce a socalled *condensation graph* with the components as nodes which is always acyclic. In Figure 3 these components are shown for the example network: C_1 to C_5. In (Treur, 2020a) the notion of *stratification* was introduced for the condensation graph based on this a partition of a network so that each component gets a level (or stratum) assigned; see Section 3 for a more precise definition of this notion of stratification of an acyclic network or graph in general. In this case the levels are 0–4 as indicated in Figure 3.

6.2. Using the stratification to relate equilibrium values for different components

Based on the levels defined by this notion of stratification, a number of general theorems and corollaria have been found and proven and presented in (Treur, 2020a); see also (Treur, 2020b), Ch 12 and 15. For aggregation these are not limited to linear functions and for connectivity no condition at all is demanded; some of these results are the following.

Theorem 4 (relating equilibrium values of states in components at different levels)

If the following aggregation conditions are fulfilled

- The combination functions are normalized, scalar-free and strictly increasing

then in an achieved equilibrium the following hold:

a) In any level 0 component *C*

- All states in *C* have the same equilibrium value *V*

- This V is between the highest and lowest initial value of the states within C

b) If for any level $i > 0$ component C the components C_1, .., C_k are the strongly connected components from which C gets an incoming connection, then
   - The equilibrium values of the states in C are between the highest and lowest equilibrium values of the states in C_1, .., C_k
   - If all states in C_1, .., C_k have the same equilibrium value V, then also all states in C have this same equilibrium value V

Corollary 4 (dependence of all equilibrium values on the values in level 0 components)

If the following aggregation conditions are fulfilled

- The combination functions are normalized, scalar-free and strictly increasing

then in an achieved equilibrium:

a) The equilibrium values of all states in the network
   - are between the highest and lowest equilibrium values of the states in the level 0 components
   - are between the highest and lowest initial values of the states in the level 0 components

b) If all states in all level 0 components C have the same equilibrium value V, then all states of the whole network have that same equilibrium value V

For the special case of a strongly connected network (consisting of one component), this implies:

Corollary 5 (strongly connected networks)

If the following connectivity and aggregation conditions are fulfilled

- The network is strongly connected
- The combination functions are normalized, scalar-free and strictly increasing

then in an achieved equilibrium:

- All states have the same equilibrium value V
- This equilibrium value V is between the highest and lowest initial values of the states

Given that in the example network model there is only one level 0 component with constant value a, by Theorem 4 or Corollary 4 above it can be concluded that all states will have equilibrium value a, as long as the aggregation conditions are fulfilled.

Note that in an acyclic network, each state forms a (singleton) strongly connected component. Applying Theorem 4 and its corollaries to this special case will again provide Theorem 1 and Corollary 1 from Section 3. That shows that the above results generalize the results from Section 3.

7. Application for equilibrium analysis of multilevel organisational learning

In this section a number of the results from the previous sections will be applied to equilibrium analysis for the domain of multilevel organizational learning (Crossan et al., 1999; Kim, 1993; Wiewiora et al., 2019, 2020). In particular, this will be addressed for the type of adaptive computational network models for multilevel organizational learning based on self-modeling networks (Treur, 2020a, 2020b) as addressed in (Canbaloğlu, Treur, & Roelofsma, 2022a, 2022b; Canbaloğlu, Treur, and Wiewiora, 2022a, 2022b); see also (Canbaloğlu, Treur, & Wiewiora, 2022c). The relations for equilibrium values found are confirmed by the simulations that were performed.

7.1 Computational modeling of multilevel organisational learning

In recent work (Canbaloğlu & Treur, 2021a, 2021b; Canbaloğlu, Treur, & Roelofsma, 2022a, 2022b; Canbaloğlu, Treur, & Wiewiora, 2022a, 2022b), it has been found out how multilevel organizational learning processes can be modeled in a systematic and conceptually transparent manner by self-modeling networks. To illustrate how the equilibrium analysis methods from the previous sections can be applied, the model described in (Canbaloğlu, Treur, & Roelofsma, 2022b) is considered in particular. A picture of the overall connectivity of this second-order adaptive network model is shown in Figure 4. Here, mental models (Craik, 1943; Treur & Van Ments, 2022) are used to represent what is learnt. They are relational structures describing (by nodes and connections) blueprints of processes in the world that may occur or that are suggested to be followed in certain circumstances; for example, they can be used to specify medical protocols or workflow. Within an overall model for mental or social processes, mental models are modeled according to different levels, in relation to what is done with them:

- at the base level, the use of mental models by *internal simulation*
- at the first-order self-model level, the *learning* or *adaptation* (e.g. revision or forgetting) of them
- at the second-order self-model level, the *control* of the adaptation.

The focus is here on the learning by the first-order self-model level in the middle plane. This level includes 21 **W**-states representing the connection weights for seven mental models (each with three connections: a→b, b→c, c→d). From these seven mental models, four are from individuals A, B, C, D (left-hand side), two of them are shared mental models from teams T1 and T2 (middle), and one is the shared mental model from the organization O (right-hand side).

In the literature on multilevel organizational learning such as (Crossan et al., 1999; Kim, 1993; Wiewiora et al., 2019, 2020), *feed forward learning* indicates how shared team mental models can be learned from individual mental models and how shared mental models of the organization can be learned from shared team mental models (or in some cases also directly from individual mental models, in particular when there are no teams). This is modeled by the connections from left to right in the middle plane in Figure 4. In addition, in such literature, *feedback learning* indicates how teams can learn their mental models from a shared organization mental model and how individuals can learn their mental models from shared team mental models (or in some cases also directly from a shared organization mental model, in particular when there are no teams). This is modeled by the connections from right to left in the middle plane in Figure 4.

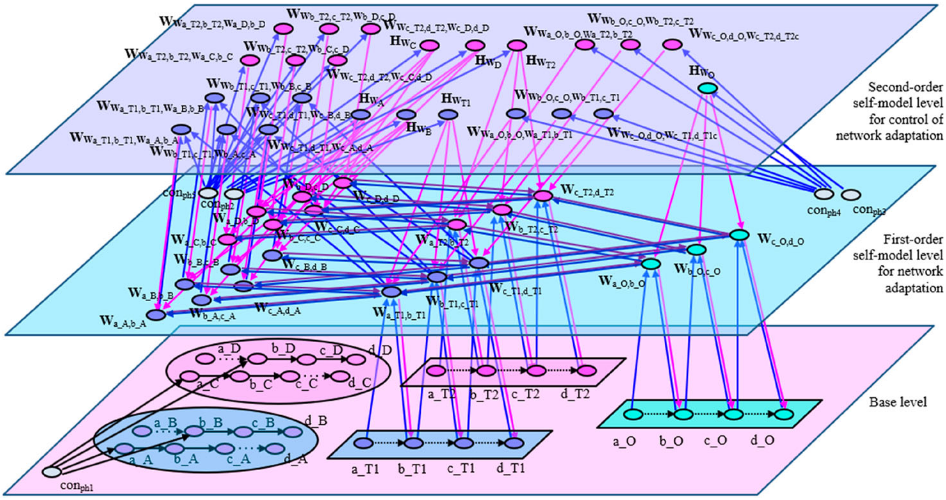

$\mathbf{W}_{W_{a\_T2,b\_T2},W_{a\_D,b\_D}}$
$\mathbf{W}_{W_{b\_T2,c\_T2},W_{b\_D,c\_D}}$
$\mathbf{W}_{W_{c\_T2,d\_T2},W_{c\_D,d\_D}}$
$\mathbf{W}_{W_{a\_T2,b\_T2},W_{a\_C,b\_C}}$
$\mathbf{W}_{W_{b\_T2,c\_T2},W_{b\_C,c\_D}}$
$\mathbf{W}_{W_{c\_T2,d\_T2},W_{c\_C,d\_D}}$
$\mathbf{H}_{W_C}$
$\mathbf{H}_{W_D}$
$\mathbf{H}_{W_{T2}}$
$\mathbf{W}_{W_{b\_O,c\_O},W_{b\_T2,c\_T2}}$
$\mathbf{W}_{W_{a\_O,b\_O},W_{a\_T2,b\_T2}}$
$\mathbf{W}_{W_{c\_O,d\_O},W_{c\_T2,d\_T2c}}$
$\mathbf{H}_{W_O}$
$\mathbf{W}_{W_{b\_T1,c\_T1},W_{b\_B,c\_B}}$
$\mathbf{W}_{W_{a\_T1,b\_T1},W_{a\_B,b\_B}}$
$\mathbf{W}_{W_{c\_T1,d\_T1},W_{c\_B,d\_B}}$
$\mathbf{H}_{W_A}$
$\mathbf{H}_{W_{T1}}$
$\mathbf{H}_{W_B}$
$\mathbf{W}_{W_{b\_O,c\_O},W_{b\_T1,c\_T1}}$
$\mathbf{W}_{W_{a\_T1,b\_T1},W_{a\_A,b\_A}}$
$\mathbf{W}_{W_{b\_T1,c\_T1},W_{b\_A,c\_A}}$
$\mathbf{W}_{W_{c\_T1,d\_T1},W_{c\_A,d\_A}}$
$\mathbf{W}_{W_{a\_O,b\_O},W_{a\_T1,b\_T1}}$
$\mathbf{W}_{W_{c\_O,d\_O},W_{c\_T1,d\_T1c}}$
Second-order self-model level for control of network adaptation
con_{ph1}
con_{ph2}
$\mathbf{W}_{b\_D,c\_D}$
$\mathbf{W}_{c\_D,d\_D}$
$\mathbf{W}_{c\_T2,d\_T2}$
con_{ph4}
con_{ph3}
$\mathbf{W}_{a\_D,b\_D}$
$\mathbf{W}_{c\_C,d\_C}$
$\mathbf{W}_{b\_T2,c\_T2}$
$\mathbf{W}_{a\_C,b\_C}$
$\mathbf{W}_{b\_C,c\_C}$
$\mathbf{W}_{a\_T2,b\_T2}$
$\mathbf{W}_{c\_O,d\_O}$
First-order self-model level for network adaptation
$\mathbf{W}_{b\_B,c\_B}$
$\mathbf{W}_{c\_B,d\_B}$
$\mathbf{W}_{b\_O,c\_O}$
$\mathbf{W}_{a\_O,b\_O}$
$\mathbf{W}_{a\_B,b\_B}$
$\mathbf{W}_{c\_T1,d\_T1}$
$\mathbf{W}_{b\_T1,c\_T1}$
$\mathbf{W}_{b\_A,c\_A}$
$\mathbf{W}_{c\_A,d\_A}$
$\mathbf{W}_{a\_T1,b\_T1}$
$\mathbf{W}_{a\_A,b\_A}$
a_D
b_D
c_D
d_D
a_C
b_C
c_C
d_C
a_T2
b_T2
c_T2
d_T2
Base level
a_B
b_B
c_B
d_B
a_A
b_A
c_A
d_A
a_O
b_O
c_O
d_O
a_T1
b_T1
c_T1
d_T1
con_{ph1}

Figure 4. Example of an adaptive network model for organizational learning.

Such learning processes depend on some forms of control, for example, involving (among others) managers to initiate or approve certain steps or proposed steps; e.g. (Canbaloğlu, Treur, & Wiewiora, 2022b). For the equilibrium analysis addressed here, for the sake of presentation it is assumed that all required control actions are positive: they indicate green lights for all considered learning processes. Next, in subsequent sections it is shown how the equilibrium analysis results presented in the previous Sections 3–6 can be illustrated for this type of organizational learning model.

7.2. Applying equilibrium analysis under connectivity conditions to a network model for multilevel organisational learning

In this section, the equilibrium analysis approach discussed in Section 3 is applied to the example network model for multilevel organizational learning with connectivity depicted in Figure 4. In particular, this means application of Theorem 1 and Corollary 1. However, a similar analysis can also be obtained by applying Theorem 4 and Corollary 4 from Section 6 as they generalise Theorem 1 and Corollary 1 as noted at the end of Section 6. Subsequently, feed forward learning, feedback learning, and a sequential combination of them are addressed.

Feed forward learning

For equilibrium analysis of feed forward learning, the subnetwork depicted in Figure 5 is considered. Here it is assumed that due to the control that is applied the backward connections (from right to left) have weights 0. As this provides an acyclic network, Theorem 1 and Corollary 1 apply. Therefore, the following can be concluded for an equilibrium:

- Expressing equilibrium values of shared team mental models in terms of those of individual mental models

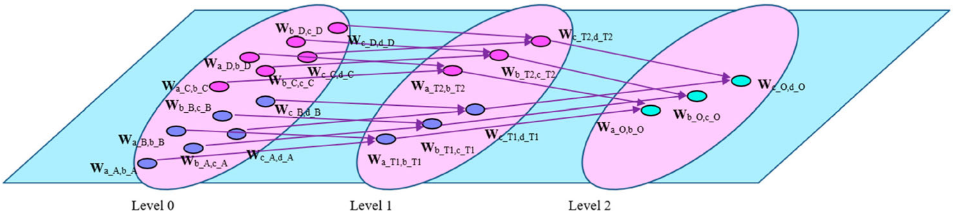


Figure 5. Only feed forward learning. The **W**-states of the four persons A to D form stratification level 0, the **W**-states of the two teams T1 and T2 form level 1, and the **W**-states of O form level 2.

○ The equilibrium value of state $\mathbf{W}_{a\_T1,b\_T1}$ for team T1's shared team mental model is a function (the combination function of $\mathbf{W}_{a\_T1,b\_T1}$) of the equilibrium values of states $\mathbf{W}_{a\_A,b\_A}$ and $\mathbf{W}_{a\_B,b\_B}$ for the individual mental models of A and B; for the case of connection weights 1 this is:

$$\underline{\mathbf{W}}_{a\_T1,b\_T1} = \mathbf{c}_{\mathbf{W}_{a\_T1,b\_T1}}(\underline{\mathbf{W}}_{a\_A,b\_A}, \underline{\mathbf{W}}_{a\_B,b\_B})$$

Here, $\mathbf{c}_{\mathbf{W}_{a\_T1,b\_T1}}(..)$ is the combination function used for aggregation for state $\mathbf{W}_{a\_T1,b\_T1}$. Similarly, this can be done for $\mathbf{W}_{b\_T1,c\_T1}$ and $\mathbf{W}_{c\_T1,d\_T1}$, and for T2 instead of T1.

- Expressing equilibrium values of a shared organization mental model in terms of those of shared team mental models

  ○ The equilibrium value of state $\mathbf{W}_{a\_O,b\_O}$ for the organization O's shared mental model is a function (the combination function of $\mathbf{W}_{a\_O,b\_O}$) of the equilibrium values of states $\mathbf{W}_{a\_T1,b\_T1}$ and $\mathbf{W}_{a\_T2,b\_T2}$ for the team mental models for T1 and T2; for the case of connection weights 1 this is:

$$\underline{\mathbf{W}}\ _{a\_O,b\_O} = \mathbf{c}_{\mathbf{W}_{a\_O,b\_O}}(\underline{\mathbf{W}}\ _{a\_T1,b\_T1}, \underline{\mathbf{W}}\ _{a\_T2,b\_T2})$$

  Here, $\mathbf{c}_{\mathbf{W}a\_O,b\_O}(..)$ is the combination function used for aggregation for state $\mathbf{W}_{a\_O,b\_O}$. Similarly, this can be done for $\mathbf{W}_{b\_O,c\_O}$ and $\mathbf{W}_{c\_O,d\_O}$.

- Expressing equilibrium values of a shared organization mental model in terms of those of individual mental models

  ○ The equilibrium value of state $\mathbf{W}_{a\_O,b\_O}$ for the organization O's shared mental model is a function (the combination function of $\mathbf{W}_{a\_O,b\_O}$ composed with those of $\mathbf{W}_{a\_T1,b\_T1}$ and $\mathbf{W}_{a\_T2,b\_T2}$) of the equilibrium values of states $\mathbf{W}_{a\_A,b\_A}$, $\mathbf{W}_{a\_B,b\_B}$, $\mathbf{W}_{a\_C,b\_C}$, and $\mathbf{W}_{a\_D,b\_D}$ for the individual mental models of A, B, C, and D; for the case of connection weights 1 this is:

$$\begin{aligned}\underline{\mathbf{W}}_{a\_O,b\_O} &= \mathbf{c}_{\mathbf{W}_{a\_O,b\_O}}(\underline{\mathbf{W}}_{a\_T1,b\_T1}, \underline{\mathbf{W}}_{a\_T2,b\_T2})\\ &= \mathbf{c}_{\mathbf{W}_{a\_O,b\_O}}(\mathbf{c}_{\mathbf{W}_{a\_T1,b\_T1}}(\underline{\mathbf{W}}_{a\_A,b\_A}, \underline{\mathbf{W}}_{a\_B,b\_B}), \mathbf{c}_{\mathbf{W}_{a\_T1,b\_T1}}(\underline{\mathbf{W}}_{a\_C,b\_C}, \underline{\mathbf{W}}_{a\_D,b\_D}))\end{aligned}$$

  Here, $\mathbf{c}_{\mathbf{W}a\_O,b\_O}(..)$ is the combination function used for aggregation for state $\mathbf{W}_{a\_O,b\_O}$ and $\mathbf{c}_{\mathbf{W}_{a\_T1,b\_T1}}(..)$ and $\mathbf{c}_{\mathbf{W}_{a\_T2,b\_T2}}(..)$ are those for $\mathbf{W}_{a\_T1,b\_T1}$ and $\mathbf{W}_{a\_T2,b\_T2}$. Similarly, this can be done for $\mathbf{W}_{b\_O,c\_O}$ and $\mathbf{W}_{c\_O,d\_O}$.

For example, if the **max** function is used as combination function for all states and the connections in Figure 5 all have weight 1, then following the network connectivity the equilibrium value $\underline{\mathbf{W}}_{a\_O,b\_O}$ for state $\mathbf{W}_{a\_O,b\_O}$ can be expressed in terms of the equilibrium values $\mathbf{W}_{a\_T1,b\_T1}$, $\mathbf{W}_{a\_T2,b\_T2}$, $\mathbf{W}_{a\_A,b\_A}$, $\mathbf{W}_{a\_B,b\_B}$, … for the teams and individuals as follows:

$$\begin{aligned}\underline{\mathbf{W}}_{a\_O,b\_O} &= \mathbf{max}(\underline{\mathbf{W}}_{a\_T1,b\_T1}, \underline{\mathbf{W}}_{a\_T2,b\_T2})\\ &= \mathbf{max}(\mathbf{max}(\underline{\mathbf{W}}_{a\_A,b\_A}, \underline{\mathbf{W}}_{a\_B,b\_B}), \mathbf{max}(\underline{\mathbf{W}}_{a\_C,b\_C}, \underline{\mathbf{W}}_{a\_D,b\_D}))\\ &= \mathbf{max}(\underline{\mathbf{W}}_{a\_A,b\_A}, \underline{\mathbf{W}}_{a\_B,b\_B}, \underline{\mathbf{W}}_{a\_C,b\_C}, \underline{\mathbf{W}}_{a\_D,b\_D})\end{aligned}$$

In this way, it is predicted that the model will form a shared organization mental model by maximally incorporating the knowledge of each of the individuals.

Feedback learning

For equilibrium analysis of feedback learning, the subnetwork depicted in Figure 6 can be considered. Here it is assumed that due to the control that is applied the forward connections (from left to right) have weights 0. Again, as this provides an acyclic network, Theorem 1 and Corollary 1 apply. Therefore the following can be concluded for any equilibrium:

- Expressing equilibrium values of shared team mental models in terms of those of a shared organization mental model
  - The equilibrium value of state $\mathbf{W}_{a\_T1,b\_T1}$ for team T1's shared team mental model is a function (the combination function of $\mathbf{W}_{a\_T1,b\_T1}$) of the equilibrium value of state $\mathbf{W}_{a\_O,b\_O}$ for the organization O's shared mental model; for the case of connection weights 1 this is:

    $$\underline{\mathbf{W}}_{a\_T1,b\_T1} = \mathbf{c}_{\mathbf{W}_{a\_T1,b\_T1}}(\underline{\mathbf{W}}_{a\_O,b\_O})$$

    Here, $\mathbf{c}_{\mathbf{W}_{a\_T1,b\_T1}}$**(..)** is the combination function used for aggregation for state $\mathbf{W}_{a\_T1,b\_T1}$. Similarly, this can be done for $\mathbf{W}_{b\_T1,c\_T1}$ and $\mathbf{W}_{c\_T1,d\_T1}$, and for T2.
- Expressing equilibrium values of individual mental models in terms of those of shared team mental models
  - The equilibrium value of state $\mathbf{W}_{a\_A,b\_A}$ for the individual mental model of A is a function (the combination function of $\mathbf{W}_{a\_A,b\_A}$) of the equilibrium value of state

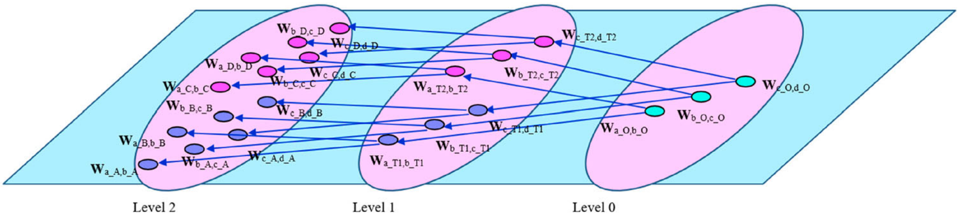


Figure 6. Only feedback learning. Each **W**-state is by itself a strongly connected component as singleton. The **W**-states of O form stratification level 0, the **W**-states of the two teams T1 and T2 form level 1, and the **W**-states of the four persons A to D form level 2.

$\mathbf{W}_{a\_T1,b\_T1}$ for team T1's shared team mental model; for the case of connection weights 1 this is:

$$\underline{\mathbf{W}}_{a\_A,b\_A} = \mathbf{c}_{\mathbf{W}_{a\_A,b\_A}}(\underline{\mathbf{W}}_{a\_T1,b\_T1})$$

Here, $\mathbf{c}_{\mathbf{W}_{a\_A,b\_A}}(..)$ is the combination function used for aggregation for state $\mathbf{W}_{a\_A,b\_A}$. Similarly, this can be done for $\mathbf{W}_{b\_A,c\_A}$ and $\mathbf{W}_{c\_A,d\_A}$, and for persons B, C, and D instead of A.

- Expressing equilibrium values of individual mental models in terms of those of a shared organization mental model
  - The equilibrium value of state $\mathbf{W}_{a\_A,b\_A}$ for the individual mental model of A is a function (the combination function of $\mathbf{W}_{a\_A,b\_A}$ composed with the one of $\mathbf{W}_{a\_T1,b\_T1}$) of the equilibrium value of state $\mathbf{W}_{a\_O,b\_O}$ for the organization O's shared mental model; for the case of connection weights 1 this is:

    $$\underline{\mathbf{W}}_{a\_A,b\_A} = \mathbf{c}_{\mathbf{W}_{a\_A,b\_A}}(\underline{\mathbf{W}}_{a\_T1,b\_T1}) \qquad = \mathbf{c}_{\mathbf{W}_{a\_A,b\_A}}(\mathbf{c}_{\mathbf{W}_{a\_T1,b\_T1}}(\underline{\mathbf{W}}_{a\_O,b\_O}))$$

    Here, $\mathbf{c}_{\mathbf{W}_{a\_A,b\_A}}(..)$ is the combination function used for aggregation for state $\mathbf{W}_{a\_A,b\_A}$ and $\mathbf{c}_{\mathbf{W}_{a\_T1,b\_T1}}(..)$ is the combination function used for aggregation for state $\mathbf{W}_{a\_T1,b\_T1}$. Similarly, this can be done for for $\mathbf{W}_{b\_A,c\_A}$ and $\mathbf{W}_{c\_A,d\_A}$, and for persons B, C, and D instead of A.

As a more specific example, if the **max** function is used as combination function for all states as was done in (Canbaloğlu, Treur, & Roelofsma, 2022b) and the connections in Figure 6 all have weight 1, then the equilibrium value $\underline{\mathbf{W}}_{a\_A,b\_A}$ for state $\mathbf{W}_{a\_A,b\_A}$ can be expressed in terms of the equilibrium values $\underline{\mathbf{W}}_{a\_T1,b\_T1}$ and $\underline{\mathbf{W}}_{a\_O,b\_O}$ for the teams and individuals as follows:

$$\begin{aligned}
\underline{\mathbf{W}}_{a\_A,b\_A} &= \mathbf{max}(\underline{\mathbf{W}}_{a\_T1,b\_T1}) \\
&= \mathbf{max}(\mathbf{max}(\underline{\mathbf{W}}_{a\_O,b\_O})) \\
&= \underline{\mathbf{W}}_{a\_O,b\_O}
\end{aligned}$$

This shows how each person gets a perfect individual mental model based on perfect knowledge transfer from the shared mental model of the organization. This perfection depends on the connection weights 1. If (some of) these connection weights are < 1, less perfect learning can be modeled.

Feedforward learning until equilibrium followed by feedback learning until equilibrium:

Next a scenario is considered where by applying control in a first phase feed forward learning takes place (forward connections weight 1, backward connections weight 0) until an equilibrium is reached and subsequently in a second phase feedback learning (forward connections weight 0, backward connections weight 1) until again an equilibrium is reached. In this case, we can combine the two equilibrium analyses above in a sequential manner. Then nesting of combination functions of 4 levels deep takes place as follows.

$$\begin{aligned}
\underline{\mathbf{W}}_{a\_A,b\_A} &= \mathbf{c}_{\mathbf{W}_{a\_A,b\_A}}(\underline{\mathbf{W}}_{a\_T1,b\_T1}) \\
&= \mathbf{c}_{\mathbf{W}_{a\_A,b\_A}}(\mathbf{c}_{\mathbf{W}_{a\_T1,b\_T1}}(\underline{\mathbf{W}}_{a\_O,b\_O}))
\end{aligned}$$

$$= \mathbf{c}_{\mathbf{W}_{a\_A,b\_A}}(\mathbf{c}_{\mathbf{W}_{a\_T1,b\_T1}}(\mathbf{c}_{\mathbf{W}_{a\_O,b\_O}}(\underline{\mathbf{W}}_{a\_T1,b\_T1}, \underline{\mathbf{W}}_{a\_T2,b\_T2})))$$

$$= \mathbf{c}_{\mathbf{W}_{a\_A,b\_A}}(\mathbf{c}_{\mathbf{W}_{a\_T1,b\_T1}}(\mathbf{c}_{\mathbf{W}_{a\_O,b\_O}}(\mathbf{c}_{\mathbf{W}_{a\_T1,b\_T1}}(\underline{\mathbf{W}}_{a\_A,b\_A}, \underline{\mathbf{W}}_{a\_B,b\_B}), \mathbf{c}_{\mathbf{W}_{a\_T2,b\_T2}}(\underline{\mathbf{W}}_{a\_C,b\_C}, \underline{\mathbf{W}}_{a\_D,b\_D}))))$$

For example, specifically assuming the **max** function for all combination functions:

$$\underline{\mathbf{W}}_{a\_A,b\_A} = \mathbf{max}(\underline{\mathbf{W}}_{a\_T1,b\_T1})$$

$$= \mathbf{max}(\mathbf{max}(\underline{\mathbf{W}}_{a\_O,b\_O}))$$

$$= \mathbf{max}(\mathbf{max}(\mathbf{max}(\underline{\mathbf{W}}_{a\_T1,b\_T1}, \underline{\mathbf{W}}_{a\_T2,b\_T2})))$$

$$= \mathbf{max}(\mathbf{max}(\mathbf{max}(\mathbf{max}(\underline{\mathbf{W}}_{a\_A,b\_A}, \underline{\mathbf{W}}_{a\_B,b\_B}), \mathbf{max}(\underline{\mathbf{W}}_{a\_C,b\_C}, \underline{\mathbf{W}}_{a\_D,b\_D}))))$$

$$= \mathbf{max}(\underline{\mathbf{W}}_{a\_A,b\_A}, \underline{\mathbf{W}}_{a\_B,b\_B}, \underline{\mathbf{W}}_{a\_C,b\_C}, \underline{\mathbf{W}}_{a\_D,b\_D})$$

This predicts that using this type of aggregation, every individual gets a mental model representing equal knowledge, based on the maximal knowledge available among individuals A, B, C, and D.

7.3. Application of equilibrium analysis for comparison relations

Next, it is illustrated how equilibrium analysis based on comparison relations as presented in Sections 4 and 5 can be applied to the considered network model for multilevel organizational learning. In particular, this involves application of Theorem 2 and 3 and Corollaries 2 and 3. For example:

- Theorem 2a) indicates that higher initial values will lead to higher equilibrium values. For example, applied to feed forward learning as considered in Section 7.2, this makes that for monotonically increasing combination functions used for aggregation, higher initial values for all **W**-states will lead to higher equilibrium values for all **W**-states.
- As another example, Corollary 2a) expresses that when **slogistic** is used, then a higher threshold value will lead to lower equilibrium values. Applied to feed forward learning as considered in Section 7.2, this provides the following:
  $\underline{\mathbf{W}}_{a\_O,b\_O} \leq \underline{\mathbf{W}'}_{a\_O,b\_O}$
  when $\underline{\mathbf{W}'}_{a\_O,b\_O}$ is achieved using a lower threshold value
- Theorem 3 expresses, for example, that all normalised monotonically increasing scalar-free functions lead to equilibrium values $\underline{\mathbf{W}}_{a\_O,b\_O}$ between the equilibrium values $\underline{\mathbf{W}''}_{a\_O,b\_O}$ and $\underline{\mathbf{W}'}_{a\_O,b\_O}$ obtained when **min** or **max** functions are used: $\underline{\mathbf{W}''}_{a\_O,b\_O} \leq \underline{\mathbf{W}}_{a\_O,b\_O} \leq \underline{\mathbf{W}'}_{a\_O,b\_O}$

7.4. Application of equilibrium analysis based on strongly connected components

Finally, it is discussed how Theorem 4 and Corollary 4 from Section 6 can be applied. As these can be considered generalisations of the results for acyclic networks in Section 3, they can also be used to obtain what is discussed in Section 7.2 (noticing that in these cases each state of the network forms a strongly connected component). However, here they are applied to the case that feed forward and feedback learning take place in a fully integrated manner. Then the picture with both left-to-right and right-to-left arrows shown in Figure 7 applies. Now not each state forms a strongly connected component but there are only three larger strongly connected components (each consisting of 7 **W**-states), indicated in Figure 7 by different colours:

$C_1 = \{ \mathbf{W}_{a\_Z,b\_Z} \mid Z \in \{A, B, C, D, T1, T2, O\}\}$ (highlighted purple in Figure 7)

$C_2 = \{ \mathbf{W}_{b\_Z,c\_Z} \mid Z \in \{A, B, C, D, T1, T2, O\}\}$ (highlighted green in Figure 7).

$C_3 = \{ \mathbf{W}_{c\_Z,d\_Z} \mid Z \in \{A, B, C, D, T1, T2, O\}\}$ (highlighted yellow in Figure 7)

It can be noted that there are no mutual connections between these three components; therefore all three have stratification level 0. By Theorem 4a) it follows that in an equilibrium, for each of the three components C_i all states in it will have the same equilibrium value:

C_1: $\underline{\mathbf{W}}_{a\_Z,b\_Z} = \underline{\mathbf{W}}_{a\_Z',b\_Z'}$ for all $Z, Z' \in \{A, B, C, D, T1, T2, O\}$

C_2: $\underline{\mathbf{W}}_{b\_Z,c\_Z} = \underline{\mathbf{W}}_{b\_Z',c\_Z'}$ for all $Z, Z' \in \{A, B, C, D, T1, T2, O\}$

C_3: $\underline{\mathbf{W}}_{c\_Z,d\_Z} = \underline{\mathbf{W}}_{c\_Z',d\_Z'}$ for all $Z, Z' \in \{A, B, C, D, T1, T2, O\}$

But note that these common equilibrium values within each of the three components may differ for different components due to the lack of mutual connections between the components.

8. Equilibrium analysis for nonlinear aggregation: two examples

In this section, it is discussed how some specific nonlinear scalar-free functions introduced for aggregation can somehow be transformed by a kind of coordinate transformation into linear functions and how based on that their equilibrium equations can be solved analytically by applying a symbolic linear solver to the transformed equations. These transformations apply instances of the underlying mathematical concept of conjugate relation between functions, a concept that will be analysed in more depth in Section 9.

8.1. Symbolic Solving of nonlinear equilibrium equations for Euclidean functions

It was found that two specific types of scalar-free functions can be related to linear functions by a socalled function conjugate relation. This is not only a theoretical result (which

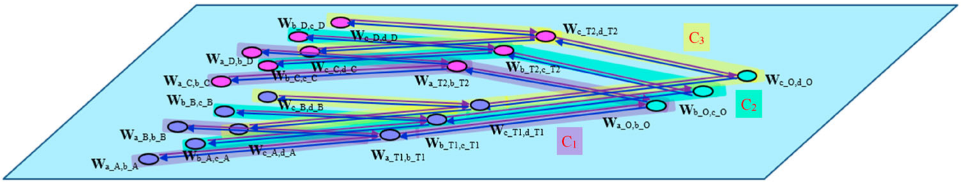


Figure 7. Full integration of feed forward learning and feedback learning. Three strongly connected components C_1 for all connections a→b (purple highlighted), C_2 for all connections b→c (green highlighted), C_3 for all connections c→d (yellow highlighted). These three components have no mutual connections and therefore have mutually independent but internally common equilibrium values.

will be addressed in more depth in Section 9) but can also be used in practice to solve equilibrium equations for them. The idea is to transform the equilibrium equations of non-linear functions into linear equations, solve these linear equations by a symbolic solver and transform the found solutions back. This will be illustrated for two examples, for a euclidean function $\mathbf{eucl}_{\boldsymbol{n},\lambda}$ in the current Section 8.1 and for a geometric function in Section 8.2. The euclidean combination function is defined by:

$$\mathbf{eucl}_{n,\lambda}(V_1, \ldots, V_k) = \sqrt[n]{\frac{V_1^{\boldsymbol{n}} + \ldots + V_k^{\boldsymbol{n}}}{\lambda}}$$

As analysed further in Section 9, this function can be written as a conjugate relation.

$\mathbf{eucl}_{n,\lambda} = \theta^{-1}$ o f o θ where $\theta(X) = X^n$ and $f(V_1, \ldots, V_k) = (V_1 + \ldots + V_k)/\lambda$

and $\theta(V_1, \ldots, V_k) = (\theta(V_1), \ldots, \theta(V_k))$.

An equilibrium equation involving this combination function for a state $Y = X_j$ typically is of the form

$$X_j = \sqrt[n]{\frac{V_1^{\boldsymbol{n}} + \ldots + V_k^{\boldsymbol{n}}}{\boldsymbol{\lambda}}} \text{ where } V_i = \boldsymbol{\omega}_{X_i,X_j} X_i(t) \text{ are single impacts of } X_i \text{ on } X_j$$

This can be rewritten into

$$X_j^n = (\boldsymbol{\omega}_{X_1,X_j}{}^n X_1{}^n + \ldots + \boldsymbol{\omega}_{X_k,X_j}{}^n X_k{}^n) \;/\; \boldsymbol{\lambda}$$

Now take $Y_i = \theta(X_i) = X_i^{\boldsymbol{n}}$ (with inverse relation $X_i = \theta^{-1}(Y_i) = \sqrt[n]{Y_i}$) and rewrite the above equation into a linear equation in Y_i; this obtains

$$Y_j = (\boldsymbol{\omega}_{X_1,X_j}{}^n \; Y_1 \; + \ldots + \boldsymbol{\omega}_{X_k,X_j}{}^n Y_k) \;/\; \boldsymbol{\lambda}$$

$$\boldsymbol{\lambda} Y_j = \boldsymbol{\omega}_{X_1,X_j}{}^n \; Y_1 \; + \ldots + \boldsymbol{\omega}_{X_k,X_j}{}^n \, Y_k$$

According to the criterion in Section 2, for the linear case of a sum function, for the example shown in Figure 3 the equation expressing that state ps$_a$ is stationary at time t is.

$\boldsymbol{\omega}_{\text{responding}}$ $\mathrm{X}_3(t) + \boldsymbol{\omega}_{\text{amplifying}}$ $\mathrm{X}_5(t) = \mathrm{X}_4(t)$

which in a simplified notation is the following equation for the state values X_3, X_4, X_5 for the three states:

$\boldsymbol{\omega}_{\text{responding}}$ $\mathbf{X}_3 + \boldsymbol{\omega}_{\text{amplifying}}$ $\mathbf{X}_5 = \mathbf{X}_4$

Now, consider that for ps$_a$ (which is X$_4$) and the other states a weighted Euclidean combination function $\mathbf{eucl}_{\mathbf{2},\lambda}(V_1,V_2)$ of order 2 is used. Remember from the above analysis that:

$\mathbf{eucl}_{\mathbf{2},\lambda} = \theta^{-1}$ o f o θ where $\theta(X) = X^2$ and $f(V_1,V_2) = (V_1 + V_2)/\lambda$

Note that

$V_1 = \boldsymbol{\omega}_{\text{responding}}$ X_3
$V_2 = \boldsymbol{\omega}_{\text{amplifying}}$ X_5

According to this, putting $Y_i = \theta(V_i) = V_i^2$, the above equation for $\mathbf{X}_4$ becomes

$$\sqrt{}((\omega_{\text{responding}}\ \mathbf{X}_3)^2 + (\omega_{\text{amplifying}}\ \mathbf{X}_5)^2)/\lambda = \mathbf{X}_4$$

$$((\omega_{\text{responding}}\ \mathbf{X}_3)^2 + (\omega_{\text{amplifying}}\ \mathbf{X}_5)^2)/\lambda = \mathbf{X}_4^2$$

$$(\omega_{\text{responding}}{}^2\ Y_3 + \omega_{\text{amplifying}}{}^2\ Y_5)/\lambda = Y_4$$

Using $Y_i = \theta(V_i) = V_i^2$, all equations become

Y_1	ws_s	$Y_1 = Y_1$
Y_2	ss_s	$\omega_{\text{sensing}}{}^2\ Y_1 = Y_2$
Y_3	srs_s	$\omega_{\text{representing}}{}^2\ Y_2 = Y_3$
Y_4	ps_a	$\omega_{\text{responding}}{}^2 Y_3 + \omega_{\text{amplifying}}{}^2\ Y_5 = \lambda\ Y_4$
Y_5	srs_e	$\omega_{\text{predicting}}{}^2\ Y_4 = Y_5$
Y_6	es_a	$\omega_{\text{executing}}{}^2\ Y_5 = Y_6$

This transforms the quadratic equations in the $\mathbf{X}_i$ into linear equations in Y_i. These linear equations can be solved symbolically in an automated manner by a Linear Solver, such as the WIMS solver available online at URL https://wims.univ-cotedazur.fr/wims/en_tool~linear~linsolver.en.html.

In Figure 8 it is shown how this set of equations was entered in this Linear Solver and (in the shaded lower area) what solutions are found. These solutions are (note that a is used as a parameter for an assumed stimulus level represented by $\mathbf{X}_1$) translated back from the Y_i to the solutions in terms of the $\mathbf{X}_i$ as follows:

$$\mathbf{X}_1^2 = a^2$$

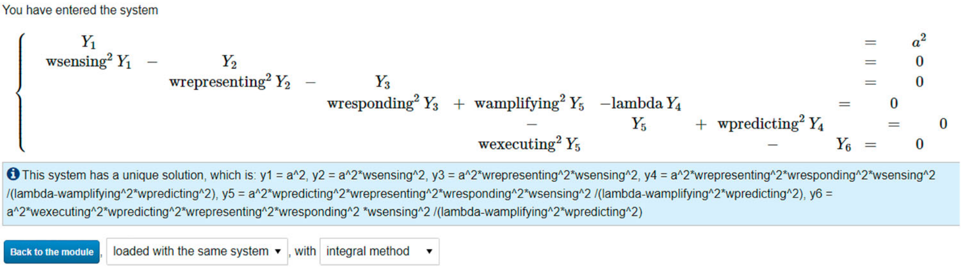
$$\mathbf{X}_2^2 = \omega_{\text{sensing}}{}^2\ a^2$$

$$\mathbf{X}_3^2 = \omega_{\text{representing}}{}^2\ \omega_{\text{sensing}}{}^2\ a^2$$

$$\mathbf{X}_4^2 = \omega_{\text{responding}}{}^2\ \omega_{\text{representing}}{}^2\ \omega_{\text{sensing}}{}^2\ a^2\ /\ (\lambda\ \text{-}\ \omega_{\text{amplifying}}{}^2\ \omega_{\text{predicting}}{}^2)$$

$$\mathbf{X}_5^2 = \omega_{\text{predicting}}{}^2\ \omega_{\text{responding}}{}^2\ \omega_{\text{representing}}{}^2\ \omega_{\text{sensing}}{}^2\ a^2\ /\ (\lambda\ \text{-}\ \omega^2_{\text{amplifying}}\ \omega^2_{\text{predicting}})$$

$$\mathbf{X}_6^2 = \omega_{\text{executing}}{}^2\ \omega_{\text{predicting}}{}^2 \omega_{\text{responding}}{}^2\ \omega_{\text{representing}}{}^2\ \omega_{\text{sensing}}{}^2\ a^2\ /\ (\lambda\ \text{-}\ \omega_{\text{amplifying}}{}^2\ \omega_{\text{predicting}}{}^2)$$



You have entered the system

$$\begin{cases} Y_1 = a^2 \\ \text{wsensing}^2\, Y_1 - Y_2 = 0 \\ \text{wrepresenting}^2\, Y_2 - Y_3 = 0 \\ \text{wresponding}^2\, Y_3 + \text{wamplifying}^2\, Y_5 - \text{lambda}\, Y_4 = 0 \\ -Y_5 + \text{wpredicting}^2\, Y_4 = 0 \\ \text{wexecuting}^2\, Y_5 - Y_6 = 0 \end{cases}$$

This system has a unique solution, which is: y1 = a^2, y2 = a^2*wsensing^2, y3 = a^2*wrepresenting^2*wsensing^2, y4 = a^2*wrepresenting^2*wresponding^2*wsensing^2 /(lambda-wamplifying^2*wpredicting^2), y5 = a^2*wpredicting^2*wrepresenting^2*wresponding^2*wsensing^2 /(lambda-wamplifying^2*wpredicting^2), y6 = a^2*wexecuting^2*wpredicting^2*wrepresenting^2*wresponding^2 *wsensing^2 /(lambda-wamplifying^2*wpredicting^2)

Figure 8. Using the WIMS Linear Solver to solve the nonlinear equilibrium equations for weighted Euclidean functions used for aggregation within the example network.

Therefore, the solutions are (assuming all are nonnegative):

$$\mathbf{X}_1 = a$$

$$\mathbf{X}_2 = \boldsymbol{\omega}_{\text{sensing}}\, a$$

$$\mathbf{X}_3 = \boldsymbol{\omega}_{\text{representing}}\, \boldsymbol{\omega}_{\text{sensing}}\, a$$

$$\mathbf{X}_4 = \boldsymbol{\omega}_{\text{responding}}\, \boldsymbol{\omega}_{\text{representing}}\, \boldsymbol{\omega}_{\text{sensing}}\, a \,/\, \surd(\lambda - \boldsymbol{\omega}_{\text{amplifying}}{}^2\, \boldsymbol{\omega}_{\text{predicting}}{}^2)$$

$$\mathbf{X}_5 = \boldsymbol{\omega}_{\text{predicting}}\, \boldsymbol{\omega}_{\text{responding}}\, \boldsymbol{\omega}_{\text{representing}}\, \boldsymbol{\omega}_{\text{sensing}}\, a \,/\, \surd(\lambda - \boldsymbol{\omega}^2{}_{\text{amplifying}}\, \boldsymbol{\omega}^2{}_{\text{predicting}})$$

$$\mathbf{X}_6 = \boldsymbol{\omega}_{\text{executing}}\, \boldsymbol{\omega}_{\text{predicting}}\, \boldsymbol{\omega}_{\text{responding}}\, \boldsymbol{\omega}_{\text{representing}}\, \boldsymbol{\omega}_{\text{sensing}}\, a \,/\, \surd(\lambda - \boldsymbol{\omega}_{\text{amplifying}}{}^2\, \boldsymbol{\omega}_{\text{predicting}}{}^2)$$

This provides explicit predictions for the equilibrium values that are reached. In particular, for all connection weights 1 except $\boldsymbol{\omega}_{\text{responding}}$ and $\boldsymbol{\omega}_{\text{amplifying}}$ which are 0.5 and $\boldsymbol{\lambda} = 0.5$ (which a guarantees normalisation), the predicted values are $\mathbf{X}_i = a$ for all i, which is confirmed from a practical perspective by example simulations performed.

8.2. Solving nonlinear equations for geometric functions

In this section the scaled geometric mean combination function **sgeomean**$_\lambda$ is addressed. For this, the following conjugate relation holds:

sgeomean$_\lambda$ = c θ^{-1} of o θ where c = $1/\sqrt[k]{\lambda}$ holds for θ = log and

$f(V_1, \ldots, V_k) = (V_1 + \ldots + V_k)/k$ and $\theta(V_1, \ldots, V_k) = (\theta(V_1), \ldots, \theta(V_k))$.

An equilibrium equation involving this combination function for state $Y = X_j$ typically is of the form

$X_j = \sqrt[k]{\frac{V_1 \ldots V_k}{\lambda}}$ where $V_i = \boldsymbol{\omega}_{X_i,X_j} X_i(t)$ are single impacts

This can be rewritten into the following set of equilibrium equations (assuming all arguments positive):

$$\mathbf{X}_j^k = (\boldsymbol{\omega}_{X_1,X_j}\mathbf{X}_1 \ldots \boldsymbol{\omega}_{X_k,X_j}\mathbf{X}_{\text{k}})/\boldsymbol{\lambda}$$

$$\boldsymbol{\lambda}\mathbf{X}_j^k = \boldsymbol{\omega}_{X_1,X_j}\mathbf{X}_1 \ldots \boldsymbol{\omega}_{X_k,X_j}\mathbf{X}_k$$

$$\log(\boldsymbol{\lambda}\mathbf{X}_j^k) = \log(\boldsymbol{\omega}_{X_1,X_j}\mathbf{X}_1 \ldots \boldsymbol{\omega}_{X_k,X_j}\mathbf{X}_k)$$

$$\log(\boldsymbol{\lambda}) + k\log(\mathbf{X}_j) = \log(\boldsymbol{\omega}_{X_1,X_j}) + \log(\mathbf{X}_1) + \ldots + \log(\boldsymbol{\omega}_{X_k,X_j}) + \log(\mathbf{X}_k)$$

Take $Y_i = \log(\mathbf{X}_i)$ (with inverse relation $\mathbf{X}_j = \exp(\mathbf{Y}_j)$) and rewrite into a linear equation in variables Y_i:

$$\log(\lambda) + k\,Y_j = \log(\boldsymbol{\omega}_{X_1,X_j}) + Y_1 + \ldots + \log(\boldsymbol{\omega}_{X_k,X_j}) + Y_k$$

$$Y_1 + \ldots + Y_k - kY_j = \log(\lambda) - (\log(\boldsymbol{\omega}_{X_1,X_j}) + \ldots + \log(\boldsymbol{\omega}_{X_k,X_j}))$$

$$Y_1 + \ldots + Y_k - kY_j = \log(\lambda/(\boldsymbol{\omega}_{X_1,X_j} \ldots \boldsymbol{\omega}_{X_k,X_j}))$$

As an illustration, assume in the example of Figure 3 for ps$_a$ (which is X_4) the combination function **sgeomean**$_\lambda$ is used, with $k = 2$ and $\boldsymbol{\lambda} = 0.5$ and for the other states X_2, X_3, X_5, X_6 the function $\mathbf{sgeomean}_{\lambda}(V_1)$ is used, with $\lambda = 1$, which is the identity function. Then the equation for $\mathbf{X}_4$ becomes.

$$Y_3 \ + Y_5 \ \ - 2Y_4 \ = \ \log\left(\boldsymbol{\lambda}/(\boldsymbol{\omega}_{\text{responding}}\boldsymbol{\omega}_{\text{amplifying}})\right)$$

Using $Y_i = \log(\mathbf{X}_i)$ for all states all equations are transformed into the following set of linear equations in the variables Y_i:

Y_1 ws_s $Y_1 = Y_1$

Y_2 ss_s $\log(\boldsymbol{\omega}_{\text{sensing}}) + Y_1 = Y_2$

Y_3 srs_s $\log(\boldsymbol{\omega}_{\text{representing}}) + Y_2 = Y_3$

Y_4 ps_a $Y_3 \ + Y_5 \ \ - 2Y_4 \ = \ \log\left(\boldsymbol{\lambda}/(\boldsymbol{\omega}_{\text{responding}}\boldsymbol{\omega}_{\text{amplifying}})\right)$

Y_5 srs_e $\log(\boldsymbol{\omega}_{\text{predicting}}) + Y_4 = Y_5$

Y_6 es_a $\log(\boldsymbol{\omega}_{\text{executing}}) + Y_5 = Y_6$

Again applying the Linear Solver to them, as shown in Figure 9, provides the following solutions (See the lower, shaded part of Figure 9):

$b = \log(\boldsymbol{\lambda}/(\boldsymbol{\omega}_{responding}{}^{*}\boldsymbol{\omega}_{amplifying})\ )$

$\log(\mathbf{X}_1) = \log(a)$

$\log(\mathbf{X}_2) = \log(\boldsymbol{\omega}_{\text{sensing}}) + \log(a)$

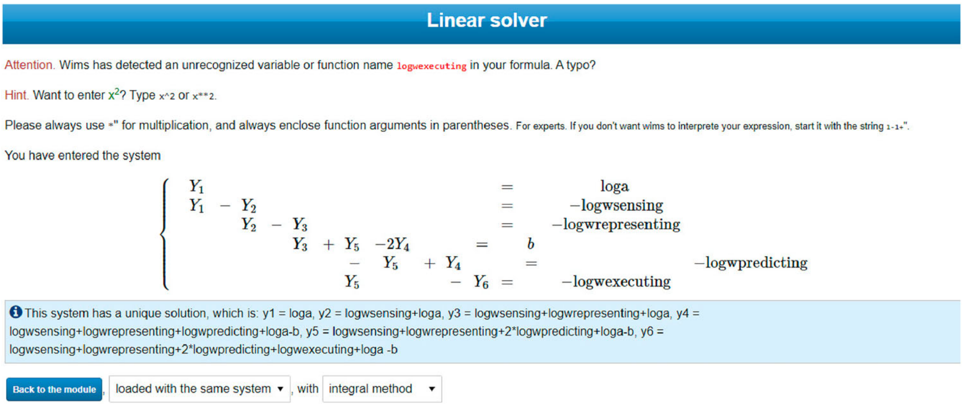
$\log(\mathbf{X}_3) = \log(\boldsymbol{\omega}_{\text{representing}}) + \log(\boldsymbol{\omega}_{\text{sensing}}) + \log(a)$

$\log(\mathbf{X}_4) = \log(\boldsymbol{\omega}_{\text{responding}}) + \log(\boldsymbol{\omega}_{\text{representing}}) + \log(\boldsymbol{\omega}_{\text{sensing}}) + \log(a)$ - b

$\log(\mathbf{X}_5) = 2\log(\boldsymbol{\omega}_{\text{predicting}}) + \log(\boldsymbol{\omega}_{\text{representing}}) + \log(\boldsymbol{\omega}_{\text{sensing}}) + \log(a)$ - b

$\log(\mathbf{X}_6) = \log(\boldsymbol{\omega}_{\text{executing}}) + 2\log(\boldsymbol{\omega}_{\text{predicting}}) \ + \log(\boldsymbol{\omega}_{\text{representing}}) + \log(\boldsymbol{\omega}_{\text{sensing}}) + \log(a) - b$

Therefore, the solutions are:



Linear solver

Attention. Wims has detected an unrecognized variable or function name `logwexecuting` in your formula. A typo?

Hint. Want to enter x^2? Type `x^2` or `x**2`.

Please always use *" for multiplication, and always enclose function arguments in parentheses. For experts. If you don't want wims to interprete your expression, start it with the string `1-1+`".

You have entered the system

$$\left\{\begin{array}{llllllllll} Y_1 & & & & & & & = & \text{loga} & \\ Y_1 & - Y_2 & & & & & & = & -\text{logwsensing} & \\ & Y_2 & - Y_3 & & & & & = & -\text{logwrepresenting} & \\ & & Y_3 & + Y_5 & -2Y_4 & & = & b & & \\ & & & - & Y_5 & + Y_4 & & = & & -\text{logwpredicting} \\ & & & Y_5 & & & - Y_6 & = & -\text{logwexecuting} & \end{array}\right.$$

ℹ This system has a unique solution, which is: y1 = loga, y2 = logwsensing+loga, y3 = logwsensing+logwrepresenting+loga, y4 = logwsensing+logwrepresenting+logwpredicting+loga-b, y5 = logwsensing+logwrepresenting+2*logwpredicting+loga-b, y6 = logwsensing+logwrepresenting+2*logwpredicting+logwexecuting+loga -b

Figure 9. Using the WIMS Linear Solver to solve the nonlinear equilibrium equations for weighted geometric functions used for aggregation within the example network.

$\mathbf{X}_1 = a$

$\mathbf{X}_2 = \boldsymbol{\omega}_{\text{sensing}}\ a$

$\mathbf{X}_3 = \boldsymbol{\omega}_{\text{representing}}\boldsymbol{\omega}_{\text{sensing}}\ a$

$\mathbf{X}_4 = \boldsymbol{\omega}_{\text{responding}}\boldsymbol{\omega}_{\text{representing}}\boldsymbol{\omega}_{\text{sensing}}\ a/b$
$= \boldsymbol{\omega}_{\text{responding}}\boldsymbol{\omega}_{\text{representing}}\boldsymbol{\omega}_{\text{sensing}}\boldsymbol{\omega}_{\text{responding}}\boldsymbol{\omega}_{\text{amplifying}}\ a/\boldsymbol{\lambda}$

$\mathbf{X}_5 = \boldsymbol{\omega}_{\text{predicting}}{}^2\ \boldsymbol{\omega}_{\text{representing}}\boldsymbol{\omega}_{\text{sensing}}\ a/b$
$= \boldsymbol{\omega}_{\text{predicting}}{}^2\ \boldsymbol{\omega}_{\text{representing}}\boldsymbol{\omega}_{\text{sensing}}\ \boldsymbol{\omega}_{\text{responding}}\boldsymbol{\omega}_{\text{amplifying}}\ a/\boldsymbol{\lambda}$

$\mathbf{X}_6 = \boldsymbol{\omega}_{\text{executing}}\boldsymbol{\omega}_{\text{predicting}}{}^2\ \boldsymbol{\omega}_{\text{representing}}\boldsymbol{\omega}_{\text{sensing}}\ a/b$
$= \boldsymbol{\omega}_{\text{executing}}\boldsymbol{\omega}_{\text{predicting}}{}^2\ \boldsymbol{\omega}_{\text{representing}}\boldsymbol{\omega}_{\text{sensing}}\ \boldsymbol{\omega}_{\text{responding}}\boldsymbol{\omega}_{\text{amplifying}}\ a/\boldsymbol{\lambda}$

Substituting 0.5 for $\boldsymbol{\omega}_{\text{responding}}$, $\boldsymbol{\omega}_{\text{amplifying}}$, and $\boldsymbol{\lambda}$ and 1 for the other connection weights, again provides $\mathbf{X}_i = a$ for all i, which again is confirmed from a practical perspective by example simulations.

9. On conjugate relations used for function transformation

In Section 8 it was found how some specific types of nonlinear functions (weighted Euclidean and geometric functions) used for aggregation can be transformed by some kind of coordinate transformation into linear functions and how that can be used to obtain equilibrium analysis for those nonlinear functions. In the setting of linear algebra, coordinate transformations are often formulated as $B = \theta^{-1}A\theta$, where A is the matrix of some linear map under the original coordinate system, B the one for a new coordinate system and θ (which in this setting is also a matrix) defines the transformation from the original to the new coordinate system; e.g. (Anton, 1987; Nering, 1970).

In a more general setting of algebra within mathematics, such a relation $B = \theta^{-1}A\theta$ between A and B is often called a conjugate relation; e.g. (Dummit & Foote, 2004). In Section 8, it was found that in a different setting of (k-ary) functions on the real numbers such conjugate relations also occur in a useful manner in the ways in which some types of nonlinear functions can be related to linear functions. So, this concept of conjugate relation at least for some cases appears to provide a bridge to cross the border between linear and nonlinear functions for aggregation. Therefore it deserves investigation in some more depth in the current section.

9.1. Additive, multiplicative, log-like, and exp-like functions and their characterisation

In this section a few basic types of functions (additive, multiplicative, log-like, exp-like) that are relevant for what has been found in previous sections are discussed in more detail. They are the types of functions θ that can be useful in particular to define conjugate relations between functions used for aggregation in networks. Again, proofs can be found in the Appendix section.

Below, the subset $R \subseteq \mathbb{R}$ used as domain for the considered functions θ in principle will be $\mathbb{R}$ or an interval within $\mathbb{R}$ of the form $\mathbb{R}_{>0} = (0, \infty)$, although in some cases also other intervals may be considered. Note that the symbol o is used to denote function composition (g o f is read for functions f and g as 'g over f' or 'g on f'). Sometimes it is left out: gf

means g o f. The domain of a function f is denoted by Dom(f) and the range f(Dom(f)) by Range(f).

Definition (additive, multiplicative, log-like, exp-like)

a) A function θ: $R \rightarrow \mathbb{R}$ is called *additive* if $\theta(\alpha+\beta) = \theta(\alpha) + \theta(\beta)$ for all $\alpha, \beta \in R$.
b) A function θ: $R \rightarrow \mathbb{R}$ is called *multiplicative* if $\theta(\alpha\beta) = \theta(\alpha)\theta(\beta)$ for all $\alpha, \beta \in R$.
c) A function θ: $R \rightarrow \mathbb{R}$ is called *log-like* if $\theta(\alpha\beta) = \theta(\alpha)+\theta(\beta)$ for all $\alpha, \beta \in R$.
d) A function θ: $R \rightarrow \mathbb{R}$ is called *exp-like* if $\theta(\alpha+\beta) = \theta(\alpha)\theta(\beta)$ for all $\alpha, \beta \in R$.
e) The standard (natural, based on the number e) exponential and logarithmic functions will be denoted by exp and log, respectively.

Note that multiplicative and log-like functions are typically used for domains R that are closed under multiplication and division such as $R = \mathbb{R}_{>0}$, whereas additive and exp-like functions are typically used for domains R that are closed under addition and subtraction such as $R = \mathbb{R}$. The following proposition shows some properties for these functions.

Proposition 6 (relating additive, multiplicative, log-like, and exp-like functions)
Let θ: $R \rightarrow S$ be any function for a finite or infinite interval R in $\mathbb{R}$, then it holds:

a) If θ is multiplicative and $S \subseteq \mathbb{R}_{>0}$, then log o θ is log-like.
b) If $R \subseteq \mathbb{R}_{>0}$ and θ is log-like, then θ o exp is additive.
c) If θ is exp-like, then log o θ is additive.
d) If θ is multiplicative and $S \subseteq \mathbb{R}_{>0}$, then log o θ o exp is additive.
e) For any multiplicative function such that $\theta(\alpha) = 0$ for some $\alpha \neq 0$, it holds that $\theta(\alpha) = 0$ for all α. For any nonzero multiplicative function θ it holds $\theta(1) = 1$ and $\theta(\alpha^{-1}) = \theta(\alpha)^{-1}$ for all α.
f) If a multiplicative θ is injective on Dom(θ), then it has an inverse θ^{-1} with Dom(θ^{-1}) = Range(θ) and Range(θ^{-1}) = Dom(θ); this inverse θ^{-1} is also multiplicative.

The following theorem provides characterisations of these different types of functions.

Theorem 5 (characterisation of additive, multiplicative, log-like and exp-like)
Let v: $R \rightarrow \mathbb{R}$ be continuous. Then the following hold.

a) Assume $R \subseteq \mathbb{R}$ is closed under addition and subtraction with $1 \in R$, then it holds
θ is additive $\Leftrightarrow$ for some c$\in \mathbb{R}$ for all X it holds $\theta(X) = \mathrm{c}\, X$.
b) Assume $R \subseteq \mathbb{R}_{>0}$ is closed under multiplication and division with $\mathrm{e} \in R$, then it holds
θ is multiplicative $\Leftrightarrow$ for some c$\in \mathbb{R}$ for all X it holds $\theta(X) = X^{\mathrm{c}.}$
c) Assume $R \subseteq \mathbb{R}_{>0}$ is closed under multiplication and division with $\mathrm{e} \in R$, then it holds
θ is log-like $\Leftrightarrow$ for some c$\in \mathbb{R}$ for all X it holds $\theta(X) = \mathrm{c}\log(X)$
d) Assume $R = \mathbb{R}$ is closed under addition and subtraction with $1 \in R$, then it holds
θ is exp-like $\Leftrightarrow$ for some c$\in \mathbb{R}$ for all X it holds $\theta(X) = \exp(\mathrm{c}X)$

9.2. Relating weakly scalar-free functions and scalar-free functions

In this section, it is explored in some depth how weakly scalar-free functions and scalar-free functions can be related to each other.

Definition (cartesian product function)

For functions $\theta_1, .., \theta_k$: $R \rightarrow \mathbb{R}$ their *cartesian product function* $\mathrm{X}_{i=1}{}^{k}\theta_i$: $R^k \rightarrow \mathbb{R}^k$ is defined by $\mathrm{X}_{i=1}{}^{k}\theta_i\ (V_1, \ \dots, V_k) = (\theta_1(V_1), \ \dots, \theta_k(V_k))$

When all θ_i are equal to one θ, this cartesian product function $\mathrm{X}_{i=1}{}^{k}\theta_i$ is also denoted by $\mathrm{X}^k\theta$, and then is also called a *cartesian power function* of θ; moreover, in that case $\mathrm{X}_{i=1}{}^{k}\theta_i$ is often denoted simply by θ, by considering the function θ as being extended to multiple variables by $\theta(V_1, \ \dots, V_k) = (\theta(V_1), \ \dots, \theta(V_k))$.

The following theorem describes some properties of scalar-free and weakly scalar-free functions. Again, proofs can be found in the Appendix section.

Theorem 6 (relating weakly scalar-free and scalar-free functions)
Consider functions f: $R^k \rightarrow \mathbb{R}$ and θ: $R \rightarrow \mathbb{R}$ for some subset $R \subseteq \mathbb{R}$ which is $\mathbb{R}$ or $\mathbb{R}_{>0}$.

a) If a nonzero function f is weakly scalar-free for function θ, then θ is multiplicative. If, moreover, f is (strictly) monotonically increasing and has at least one positive value, then θ is also (strictly) monotonically increasing. Therefore for the strict monotonically increasing case, θ is injective and has an inverse θ^{-1} on Range(θ), which is also multiplicative.
b) Any nonzero multiplicative function θ is weakly scalar-free for itself.
c) For any weakly scalar-free function f for θ the following are equivalent:
   (i) Range(f) $\subseteq$ Range(θ)
   (ii) For all $V_1, \ \dots, V_k$ an $\alpha \in R$ exists such that $f(\alpha V_1, \ \dots, \alpha V_k) = 1$
d) For each weakly scalar-free function f: $R^k \rightarrow \mathbb{R}$ for any injective θ, the function g: Range$(\theta)^k \rightarrow \mathbb{R}$ defined by $g = f\ \theta^{-1}$ is scalar-free. If, moreover, Range(f) $\subseteq$ Range (θ), then also the function h: $R^k \rightarrow \mathbb{R}$ defined by $h\ = \theta^{-1}f$ is scalar-free. For strictly increasing f and θ, these functions g,h are strictly increasing too.
e) For each set of strictly increasing and weakly scalar-free functions f_i: $R^k \rightarrow \mathbb{R}_{\geq 0}$ for the same strictly increasing θ, for any linear combination f of the f_i with positive coefficients, the function g: $R^k \rightarrow \mathbb{R}$ defined by $g = f\ \theta^{-1}$ is strictly increasing and scalar-free. If, moreover, Range(f) $\subseteq$ Range(θ), then also the function h: $R^k \rightarrow \mathbb{R}$ defined by $h\ = \theta^{-1}f$ is strictly increasing and scalar-free.
f) If f: $R^k \rightarrow \mathbb{R}$ is scalar-free, θ: $R \rightarrow R$ is multiplicative and $g = f \circ \theta$: $R^k \rightarrow \mathbb{R}$, then g is weakly scalar-free for θ. This holds in particular if f is linear.

Examples (from weakly scalar-free to scalar-free functions)

From Theorem 6d) it follows that the function.

$$g(V_1, V_2, V_3) = \theta^{-1}f(V_1, V_2, V_3) = \sqrt{}(w_1\ V_1{*}V_2 + w_2\ V_2{*}V_3 + w_3\ V_3{*}V_1)$$

is scalar-free. Also, by Theorem 6e)

$$h(V_1, V_2, V_3) = [\ V_1 + V_2 + V_3 + \sqrt{}(w_1\ V_1{*}V_2 + w_2\ V_2{*}V_3 + w_3\ V_3{*}V_1)\ ]/\lambda$$

is scalar-free.

9.3. Using conjugate relations to obtain and analyse scalar-free functions

Next, it is analysed how from given scalar-free functions other scalar-free functions can be obtained by applying some transformation. The type of transformation applied can be interpreted as a form of scale transformation or coordinate transformation. It is done by generating conjugates of scalar-free functions defined as follows.

Definition (function conjugates)

Let subsets $R, S \subseteq \mathbb{R}$ be given. The function g: $S^k \to S$ is a *(function) conjugate* of f: $R^k \to R$ by θ if θ: $S \to R$ is a bijective function and $g = \theta^{-1} \text{ o } f \text{ o } \theta$.

Proposition 7 (function conjugate operator)

Let subsets $R, S \subseteq \mathbb{R}$ be given, and functions g: $S^k \to S$, f: $R^k \to R$, and bijective θ: $S \to R$. Then the following hold:

a) Then the following are equivalent:
   (i) g is a function conjugate of f by θ
   (ii) The following commutation rules hold: $\theta\, g = f\, \theta$ and $\theta^{-1} f = g\, \theta^{-1}$

b) If a) (i) and (ii) hold, then for any g such an f is unique and can be denoted by $f = S_\theta(g)$ for a *function conjugate operator* S_θ; similarly, $g = S_{\theta^{-1}}$ (f) for function conjugate operator $S_{\theta^{-1}}$, so it holds: $\theta\, g = S_\theta(g)\, \theta$ and $\theta^{-1} f = S_{\theta^{-1}}(f)\, \theta^{-1}$

   These operators S_θ and $S_{\theta^{-1}}$ are each other's inverse and they preserve function addition and composition: for all f, g, f_1, f_2, g_1 and g_2 of proper types it holds:

$$S_{\theta^{-1}} S_\theta(g) = g$$

$$S_\theta S_{\theta^{-1}}(f) = f$$

$$S_\theta(g_1 + g_2) = S_\theta(g_1) + S_\theta(g_2)$$

$$S_\theta(g_1 \text{ o } g_2) = S_\theta(g_1) \text{ o } S_\theta(g_2)$$

$$S_{\theta^{-1}}(f_1 + f_2) = S_{\theta^{-1}}(f_1) + S_{\theta^{-1}}(f_2)$$

$$S_{\theta^{-1}}(f_1 \text{ o } f_2) = S_{\theta^{-1}}(f_1) \text{ o } S_{\theta^{-1}}(f_2)$$

Moreover, when conjugate operators S_{θ_1} and S_{θ_2} for θ_1 and θ_2 are applied in turn, it holds

$$S_{\theta_1} S_{\theta_2}(g) = S_{\theta_1 \theta_2}(g) \text{ for all } g$$

In this section, it is established that specific nonlinear functions are scalar-free. First, in a more general setting in Theorem 7 this will be addressed for weighted euclidean functions. Moreover, it is analysed how weighted euclidean functions can be related to linear functions: it turns out that they can be interpreted as conjugates of linear functions via some multiplicative function θ. This is explained by the following:

Theorem 7 (from scalar-free functions to scalar-free conjugates by multiplicative θ)

a. For any scalar-free function f: $R^k \to R$ with $R = \mathbb{R}_{\geq 0}$, all of its conjugates θ^{-1} ofo θ by a multiplicative θ: $R \to R$ are also scalar-free.

b. More specifically, for any scalar-free function f, for any positive real number n the function g defined by

$$g(V_1, \ldots, V_k) = \sqrt[n]{f(V_1^n, \ \ldots, \ V_k^n)}$$

is a conjugate θ^{-1} ofo θ of f by the multiplicative function θ: $X \rightarrow X^n$ and therefore is also scalar-free.

c. All weighted Euclidean functions are conjugates of linear functions by a multiplicative function θ and therefore are scalar-free. In particular, this holds for all functions $\mathbf{eucl}_{n,\lambda}$.

Next, in a more general setting in Theorem 8 it is established that weighted geometric functions are scalar-free and how they can be related to linear functions. Again, it turns out that they can be considered conjugates of linear functions, this time not via a multiplicative function but via a log-like function θ. This is explained by the following:

Theorem 8 (from linear to scalar-free conjugates by log-like θ)

a) For any normalised linear function all of its conjugates θ^{-1} ofo θ by a log-like θ are scalar-free.

b) More specifically, for any normalised linear function f, the function g defined by

$$g(V_1, \ \ldots, V_k) = \exp(f(\log(V_1), \ \ldots, \ \log(V_k))$$

is a conjugate θ^{-1} of o θ of a linear function by the standard log-like function $\theta = \log$ and therefore is scalar-free.

c) All weighted geometric functions are conjugates of a normalised linear function by a log-like function θ and therefore are scalar-free. In particular, this also holds for all functions $\mathbf{sgeomean}_\lambda$.

Note that in Theorem 8c) the scaled geometric mean function $\mathbf{sgeomean}_1$ is scalar-free as it is a special case of a weighted geometric function with all weights 1 and therefore $\mathbf{sgeomean}_1 = \theta^{-1}$ of o θ with f linear and normalised, whereas $\mathbf{sgeomean}_\lambda$ for $\lambda \neq 1$ is not a weighted geometric function and conjugate itself but it is a constant factor $c = 1/\sqrt[k]{\lambda}$ times the weighted geometric function, so $\mathbf{sgeomean}_\lambda = c\,\theta^{-1}$ of o θ; therefore it follows that $\mathbf{sgeomean}_\lambda$ is scalar-free too (which also can be verified from the formula for $\mathbf{sgeomean}_\lambda$).

10. Discussion

In this paper, equilibrium analysis was addressed for network models. A main application focus was on organizational learning (Crossan et al., 1999; Kim, 1993; Wiewiora et al., 2019; Wiewiora et al., 2020) and in particular computational network models for it (Canbaloğlu, Treur, & Roelofsma, 2022a; Canbaloğlu, Treur, & Wiewiora, 2022a, 2022b, 2022c).

In the paper, it was shown how, in contrast to often held beliefs, certain classes of nonlinear functions used for aggregation in network models enable analysis of the emerging dynamics like linear functions do. One of the methods used is the one from (Treur, 2020a), describing equilibrium analysis based on stratification for a network's strongly connected components (Bloem et al., 2006; Fleischer et al., 2000; Harary et al., 1965; Łacki, 2013; Wijs et al., 2016). The presented work adopts elements from (Treur, 2020a), but also includes a number of new concepts and methods introduced here especially for this type of network analysis, such as weakly scalar-free functions, conjugate relations for transformations of functions and the use of a linear solver to solve nonlinear equations via function transformation. These new concepts and methods enable to get more insight in some of the types of nonlinear functions for which analysis is well-feasible.

Nevertheless, still more work is needed, as no complete classification of all possible types of nonlinear functions that are scalar-free has been obtained yet; this still stands as a remaining challenge. Note that from scalar-free functions in a combinatorial manner new scalar-free functions can be generated easily, using (1) linear combinations of them, (2) function compositions of them, and (3) conjugates of them. By iteratively combining these three methods, scalar-free functions can be built of arbitrarily high complexity. This shows that there is a very large space of such nonlinear functions, which all still are well-suitable for analysis.

Note that the focus of this paper was on behaviour of adaptive network models where equilibria occur. This is applicable, for example, in situations where some forms of organization or structuring take place such as in well-organised organizational learning. Of course, also other situations exist where, for example, influential context factors change all the time in a random manner and their influences are not well-organised. This will in general lead to types of (chaotic) behaviour where equilibria do not occur. The disclaimer here is that the current paper was not meant to address such situations involving chaotic behaviour.

11. Appendix: Proofs

In this section the proofs for all theorems, propositions and corollaria from Sections 3–6 and Section 9 are pointed out.

Theorem 1 (dependency of equilibrium values for acyclic networks)

Suppose a network is acyclic and all states with incoming connections from other states have nonzero speed factors. Then the following hold.

a) In any equilibrium for each state Y of any stratification level, the equilibrium value Y depends by some mathematical function on the equilbrium values $\mathbf{X}$ of states X of level 0.

b) More specifically, in any equilibrium for any state Y of stratification level $i > 0$, its equilibrium value $\mathbf{Y}$ can be determined from equilibrium values $\mathbf{X}_j$ of states X_j at lower levels $< i$ by:

$$\mathbf{Y} = \mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}\mathbf{X}_1, \; \ldots, \; \boldsymbol{\omega}_{X_k,Y}\mathbf{X}_k)$$

Proof

b) For a given state Y, let X_1 to X_k be the states from which Y gets incoming connections in the network. Due to the definition of stratification, the X_1 to X_k will have levels $< i$.

Then b) follows from the formulated stationary point and equilibrium criteria for network models based on the canonical difference equation (1).

a) This follows from b) by induction over the stratification levels. By applying Theorem 1b) iteratively according to the stratification levels, in a straightforward manner for each state *Y* of the network, a mathematical expression can be obtained showing exactly how its equilibrium value depends on the equilibrium values of states of level 0.

Corollary 1 Suppose a network is acyclic and all states with incoming connections from other states have nonzero speed factors. Then the mathematical expression in Theorem 1a) defines a mathematical function for **Y** in terms of the equilibrium values **X** of some states *X* of level 0 with as parameters connectivity and aggregation characteristics $\boldsymbol{\omega}_{Z_1,Z_2}$ and $\mathbf{c}_Z(..)$ of the network relating to states *Z*, Z_1, Z_2 on the paths from the involved level 0 states *X* to state *Y*. This mathematical function essentially is based on an iterated composition of combination functions of the states on the paths to *Y* in the network, nested according to the (inverse) branching structure of these paths to *Y*.

Proof

This follows from a more detailed inspection of the proof of Theorem 1a). This is illustrated in Section 3 for the example depicted in Figure 1 and in Section 7 for the examples depicted in Figure 5 and Figure 6.

Theorem 2 (preservation of comparison relations over time and for equilibria)

Suppose X_i are the states of a network model (with only positive connection weights and at least some nonzero speed factors) and all are using monotonically increasing combination functions $\mathbf{c}_i$. Assume $0 < \Delta t \leq 1/\max_Y(\boldsymbol{\eta}_Y)$; e.g. assume $\boldsymbol{\eta}_Y \leq 1$ for all *Y* and $0 < \Delta t \leq 1$. Then the following hold.

a) Suppose two simulation traces $X_i(t)$ and $X'_i(t)$ are given with initial values $X_i(0) \leq X'_i(0)$. Then it holds $X_i(t) \leq X'_i(t)$ for all *t* and *i* and for any achieved equilibrium, for the equilibrium values $\mathbf{X}_i$ and $\mathbf{X}'_i$ of X_i and X'_i it holds $\mathbf{X}_i \leq \mathbf{X}'_i$ for all *i*.
b) Moreover, suppose X'_i are again the states of the same network model but this time using monotonically increasing combination functions $\mathbf{c}'_i$. Then the following hold:
   (i) If $\mathbf{c}_i \leq \mathbf{c}'_i$ for all *i* and for the initial values it holds $X_i(0) \leq X'_i(0)$ for all *i*, then it holds $X_i(t) \leq X'_i(t)$ for all *t* and *i*.
   (ii) If $\mathbf{c}_i \leq \mathbf{c}'_i$ for all *i* and for the initial values it holds $X_i(0) \leq X'_i(0)$ for all *i*, then for any achieved equilibrium for all *i* for the equilibrium values $\mathbf{X}_i$ and $\mathbf{X}'_i$ of X_i and X'_i it holds $\mathbf{X}_i \leq \mathbf{X}'_i$.

Proof

a) This follows from b) and c) when $\mathbf{c}'_i = \mathbf{c}_\mathrm{i}$ is chosen.
b) Note that from $\Delta t \leq 1/\max_Y(\boldsymbol{\eta}_Y)$ it follows $(1 - \boldsymbol{\eta}_Y \Delta t) \geq 0$ for all *Y*.

The proof goes by induction over the Δt-steps for $\Delta t \leq 1/\max_Y(\boldsymbol{\eta}_Y)$ using (1), where X_1 to X_k are the states from which Y gets its incoming connections:

$$
\begin{aligned}
Y(t+\Delta t) &= Y(t)\ +\boldsymbol{\eta}_Y[\mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t),\ \ldots,\ \boldsymbol{\omega}_{X_k,Y}X_k(t)) -\ Y(t)]\Delta t \\
&= Y(t) - \boldsymbol{\eta}_Y Y(t)\Delta t + \boldsymbol{\eta}_Y \mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t),\ \ldots,\ \boldsymbol{\omega}_{X_k,Y}X_k(t))\Delta t \\
&= (1-\boldsymbol{\eta}_Y\Delta t)Y(t) + \boldsymbol{\eta}_Y \mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t),\ \ldots,\ \boldsymbol{\omega}_{X_k,Y}X_k(t))\Delta t \\
&\leq (1-\boldsymbol{\eta}_Y\Delta t)Y'(t) + \boldsymbol{\eta}_Y \mathbf{c}'_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t),\ \ldots,\ \boldsymbol{\omega}_{X_k,Y}X_k(t))\Delta t \\
&\leq (1-\boldsymbol{\eta}_Y\Delta t)Y'(t) + \boldsymbol{\eta}_Y \mathbf{c}'{}_Y(\boldsymbol{\omega}_{X_1,Y}X'_1(t),\ \ldots,\ \boldsymbol{\omega}_{X_k,Y}X'_k(t))\Delta t \\
&= Y'(t) - \boldsymbol{\eta}_Y Y'(t)\Delta t + \boldsymbol{\eta}_Y \mathbf{c}'_Y(\boldsymbol{\omega}_{X_1,Y}X'_1(t),\ \ldots,\ \boldsymbol{\omega}_{X_k,Y}X'_k(t))\Delta t \\
&= Y'(t)\ +\boldsymbol{\eta}_Y[\mathbf{c}'_Y(\boldsymbol{\omega}_{X_1,Y}X'_1(t),\ \ldots,\ \boldsymbol{\omega}_{X_k,Y}X'_k(t)) -\ Y'(t)]\Delta t \\
&= Y'(t+\Delta t)
\end{aligned}
$$

c) This follows from a) by

$$\mathbf{X}_i = \lim_{n\to\infty} X_i(n\Delta t) \leq \lim_{n\to\infty} X'_i(n\Delta t) = \mathbf{X}'_i$$

Proposition 1 (comparison for logistic functions)

a) Suppose $\boldsymbol{\tau}'<\boldsymbol{\tau}$ and $\boldsymbol{\sigma}>0$. Then for any $V_1,\ \ldots, V_k \geq 0$ it holds

$$
\begin{aligned}
&0 \leq \mathbf{alogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(V_1,\ \ldots, V_k) < \mathbf{slogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(V_1,\ \ldots, V_k) \\
&< \mathbf{slogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}'}(V_1,\ \ldots, V_k) < 1
\end{aligned}
$$

b) Moreover, for any $\boldsymbol{\sigma}>0$, and $V_1,\ \ldots, V_k \geq 0$ it holds

$$
\begin{aligned}
&\lim_{\tau\to\infty} \mathbf{alogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(V_1,\ \ldots, V_k) = \lim_{\tau\to\infty} \mathbf{slogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(V_1,\ \ldots, V_k) = 0 \\
&\lim_{\tau\to-\infty} \mathbf{slogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(V_1,\ \ldots, V_k) = 1
\end{aligned}
$$

Proof

a) First, consider **slogistic**. Then for its (partial) derivative to τ it holds

$$\partial\frac{1}{1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k-\boldsymbol{\tau})}}/\partial\tau = -\frac{1}{[1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k-\boldsymbol{\tau})}]^2}\quad \partial(1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})})/\partial t$$

$$= -\frac{1}{[1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}]^2}\ \partial\,\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}/\partial\tau$$

$$= -\frac{1}{[1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}]^2}\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}\partial\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}/\partial\tau$$

$$= -\frac{1}{[1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}]^2}\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}\sigma$$

$$= -\frac{\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}}{[1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}]^2}$$

$$< 0$$

This proves that for **τ'<τ** it holds

$$\mathbf{slogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(V_1,\ \ldots, V_k) < \mathbf{slogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}'}(V_1,\ \ldots, V_k)$$

Finally, for the same values of σ and τ, the functions **slogistic** and **alogistic** can be compared as follows:

$$\frac{1}{1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}} < 1$$

$$\frac{1}{1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}} < \frac{\mathrm{e}^{\boldsymbol{\sigma\tau}}}{1+\mathrm{e}^{\boldsymbol{\sigma\tau}}} + \frac{1}{1+\mathrm{e}^{\boldsymbol{\sigma\tau}}}$$

$$\frac{1}{1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}} - \frac{1}{1+\mathrm{e}^{\boldsymbol{\sigma\tau}}} < \frac{\mathrm{e}^{\boldsymbol{\sigma\tau}}}{1+\mathrm{e}^{\boldsymbol{\sigma\tau}}}$$

$$e^{-\boldsymbol{\sigma\tau}}\left[\frac{1}{1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}} - \frac{1}{1+\mathrm{e}^{\boldsymbol{\sigma\tau}}}\right] < \frac{1}{1+\mathrm{e}^{\boldsymbol{\sigma\tau}}}$$

$$e^{-\boldsymbol{\sigma\tau}}\left[\frac{1}{1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}} - \frac{1}{1+\mathrm{e}^{\boldsymbol{\sigma\tau}}}\right] - \frac{1}{1+\mathrm{e}^{\boldsymbol{\sigma\tau}}} < 0$$

$$\left[\frac{1}{1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}} - \frac{1}{1+\mathrm{e}^{\boldsymbol{\sigma\tau}}}\right] + \mathrm{e}^{-\boldsymbol{\sigma\tau}}\left[\frac{1}{1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}} - \frac{1}{1+\mathrm{e}^{\boldsymbol{\sigma\tau}}}\right]$$
$$< \frac{1}{1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}}$$

$$\left[\frac{1}{1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}} - \frac{1}{1+\mathrm{e}^{\boldsymbol{\sigma\tau}}}\right](1+e^{-\boldsymbol{\sigma\tau}}) < \frac{1}{1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\ldots+V_k^{-\boldsymbol{\tau}})}}$$

Therefore, it holds

$$\mathbf{alogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(V_1,\ \ldots, V_k) < \mathbf{slogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(V_1,\ \ldots, V_k)$$

b) The first part follows from:

$$\begin{aligned}
0 &< \lim_{\tau\to\infty} \mathbf{alogistic}_{\sigma,\tau}(V_1, \;\ldots, V_k) \\
&\le \lim_{\tau\to\infty} \mathbf{slogistic}_{\sigma,\tau}(V_1, \;\ldots, V_k) \\
&= \lim_{\tau\to\infty} \frac{1}{1+e^{-\sigma(V_1+\ldots+V_k^{-\tau})}} \\
&= \lim_{\tau\to\infty} \frac{1}{1+e^{\sigma\tau}e^{-\sigma(V_1+\ldots+V_k)}} \\
&= 0
\end{aligned}$$

The second part follows from

$$\begin{aligned}
\lim_{\tau\to-\infty} \mathbf{slogistic}_{\sigma,\tau}(V_1, \;\ldots, V_k) &= \lim_{\tau\to-\infty} \frac{1}{1+e^{-\sigma(V_1+\ldots+V_k^{-\tau})}} \\
&= \lim_{\tau\to-\infty} \frac{1}{1+e^{\sigma\tau}\,e^{-\sigma(V_1+\ldots+V_k)}} \\
&= 1
\end{aligned}$$

Proposition 2 (comparison between euclidean and min and max functions)

a) Suppose the scaling factor is set at $\lambda = k$, then for any $V_1, \;\ldots, V_k \ge 0$ it holds

$$\mathbf{min}(V_1, \;\ldots, V_k) \le \mathbf{eucl}_{n,k}(V_1, \;\ldots, V_k) \;\le \mathbf{max}(V_1, \;\ldots, V_k)$$

and

$$\lim_{n\to\infty} \mathbf{eucl}_{n,k}(V_1, \;\ldots, V_k) = \mathbf{max}(V_1, \;\ldots, V_k)$$

b) More in general, for any $\lambda>0$ it holds

$$\mathbf{min}(V_1, \;\ldots, V_k).\sqrt[n]{\frac{k}{\lambda}} \le \mathbf{eucl}_{n,}(V_1, \;\ldots, V_k) \;\le \mathbf{max}(V_1, \;\ldots, V_k).\sqrt[n]{\frac{k}{\lambda}}$$

and

$$\lim_{n\to\infty} \mathbf{eucl}_{n,\lambda}(V_1, \;\ldots, V_k) = \mathbf{max}(V_1, \;\ldots, V_k)$$

Proof:

$$\mathbf{eucl}_{n,\lambda}(V_1, \;\ldots, V_k) \;\ge \mathbf{eucl}_{n,\lambda}(\mathbf{max}(V_1, \;\ldots, V_k), \;\ldots., \;\mathbf{max}(V_1, \;\ldots, V_k))$$

$$= \mathbf{max}(V_1, \;\ldots, V_k)\sqrt[n]{\frac{k}{\lambda}}$$

$$\mathbf{eucl}_{n,\lambda}(V_1, \;\ldots, V_k) \ge \;\mathbf{eucl}_{n,\lambda}(\mathbf{min}(V_1, \;\ldots, V_k), \;\ldots., \;\mathbf{min}(V_1, \;\ldots, V_k))$$

$$= \mathbf{min}(V_1, \;\ldots, V_k).\sqrt[n]{\frac{k}{\lambda}}$$

For

$$\lim_{n\to\infty} \mathbf{eucl}_{n,k}(V_1, \ \ldots, V_k) = \mathbf{max}(V_1, \ \ldots, V_k)$$

see (Treur, 2020b), Ch 11, Section 11.6.2 and Ch 15, Section 15.6.

Proposition 3 (comparison between geometric mean and min and max functions)

a) Suppose the scaling factor is set at $\lambda = 1$, then for any $V_1, \ \ldots, V_k \geq 0$ it holds

$$\mathbf{min}(V_1, \ \ldots, V_k) \leq \mathbf{sgeomean}_\lambda(V_1, \ \ldots, V_k) \leq \mathbf{max}(V_1, \ \ldots, V_k)$$

b) More in general, for any λ>0 and any $V_1, \ \ldots, V_k \geq 0$ it holds

$$\mathbf{min}(V_1, \ \ldots, V_k).\sqrt[k]{\frac{1}{\lambda}} \leq \ \mathbf{sgeomean}_\lambda(V_1, \ \ldots, V_k) \ \leq \mathbf{max}(V_1, \ \ldots, V_k).\sqrt[k]{\frac{1}{\lambda}}$$

Proof:

$$\mathbf{sgeomean}_\lambda(V_1, \ \ldots, V_k) \ \leq \mathbf{sgeomean}_\lambda(\mathbf{max}(V_1, \ \ldots, V_k), \ \ldots., \ \mathbf{max}(V_1, \ \ldots, V_k))$$

$$= \mathbf{max}(V_1, \ \ldots, V_k).\sqrt[k]{\frac{1}{\lambda}}$$

$$\mathbf{sgeomean}_\lambda(V_1, \ \ldots, V_k) \ \geq \mathbf{sgeomean}_\lambda(\mathbf{min}(V_1, \ \ldots, V_k), \ \ldots., \mathbf{min}(V_1, \ \ldots, V_k))$$

$$= \mathbf{min}(V_1, \ \ldots, V_k).\sqrt[k]{\frac{1}{\lambda}}$$

Corollary 2 (comparison relations for equilibrium values: logistic and sum functions)

Assume only positive connection weights and at least some nonzero speed factors and $\Delta t \leq 1/\max_Y(\boldsymbol{\eta}_Y)$ (e.g. assume $\boldsymbol{\eta}_Y \leq 1$ for all Y and $\Delta t \leq 1$).

a. Suppose X_i are the states for a network model using *advanced* logistic combination functions $\mathbf{c}_i = \mathbf{alogistic}$ and X'_i for the same network model using *simple logistic* combination functions $\mathbf{c}'_i = \mathbf{slogistic}$ with the same parameters $\boldsymbol{\sigma}_i$ and $\boldsymbol{\tau}_i$ for each state X_i. Moreover, suppose two simulation traces $X_i(t)$ and $X'_i(t)$ are given with initial values $X_i(0) \leq X'_i(0)$ for all i, then for any achieved equilibrium with equilibrium values $\mathbf{X}_i$ and $\mathbf{X}'_i$ it holds $\mathbf{X}_i \leq \mathbf{X}'_i$ for all i.
b. Suppose X_i are the states for a network model using *simple* logistic combination functions $\mathbf{c}_i = \mathbf{slogistic}$ with parameters $\boldsymbol{\sigma}_i$ and $\boldsymbol{\tau}_i$ and X'_i for the same network model using *simple logistic* combination functions $\mathbf{c}'_i = \mathbf{slogistic}$ with the parameters $\boldsymbol{\sigma}_i$ and $\boldsymbol{\tau}'_i$ for each state X_i such that $\boldsymbol{\tau}'_i \leq \boldsymbol{\tau}_i$. Moreover, suppose two simulation traces $X_i(t)$ and $X'_i(t)$ are given with initial values $X'_i(0) \leq X_i(0)$ for all i, then for any achieved equilibrium with equilibrium values $\mathbf{X}_i$ and $\mathbf{X}'_i$ it holds $\mathbf{X}_i \leq \mathbf{X}'_i$ for all i.
c. Suppose X_i are the states for a network model using *advanced* logistic combination functions $\mathbf{c}_i = \mathbf{alogistic}$ with parameters $\boldsymbol{\sigma}_i$ and $\boldsymbol{\tau}_i$ and X'_i for the same network model using scaled sum combination functions $\mathbf{c}'_i = \mathbf{ssum}$ with the parameters $\boldsymbol{\lambda}'_i$ for each state X_i. Moreover, suppose two simulation traces $X_i(t)$ and $X'_i(t)$ are given with initial values $X'_i(0) \leq X_i(0)$ for all i.

If $0 \leq \boldsymbol{\lambda}'_i \leq \boldsymbol{\tau}_i$ or $\boldsymbol{\lambda}'_i \leq 2\,\mathbf{min}(\boldsymbol{\sigma}_i, \boldsymbol{\tau}_i)$, and an equilibrium is achieved with equilibrium values $\mathbf{X}_i$ and $\mathbf{X}'_i$ then it holds $\mathbf{X}_i \leq \mathbf{X}'_i$ for all i.

Proof

This follows from Theorem 2 and Proposition 1a).

Corollary 3 (comparison relations for equilibrium values: Euclidean, geometric, minimum and maximum functions)

Assume only positive connection weights and at least some nonzero speed factors and $\Delta t \leq 1/\max_Y(\boldsymbol{\eta}_Y)$ (e.g. assume $\boldsymbol{\eta}_Y \leq 1$ for all Y and $\Delta t \leq 1$).

a) Suppose X_i are the states for a network model using as combination functions $\mathbf{c}_i$ Euclidean combination functions $\mathbf{eucl}_{n,k}(V_1, \ldots, V_k)$ with scaling factor $\boldsymbol{\lambda} = k$ or geometric mean combination functions $\mathbf{sgeomean}_1(V_1, \ldots, V_k)$ with scaling factor $\boldsymbol{\lambda} = 1$ and X'_i for the same model using *maximum* combination functions $\mathbf{c}'_i$. If for the initial values it holds $X_i(0) \leq X'_i(0)$ for all i, then for any achieved equilibrium with equilibrium values $\mathbf{X}_i$ and $\mathbf{X}'_i$ it holds $\mathbf{X}_i \leq \mathbf{X}'_i$ for all i.

b) Suppose X_i are the states for a network model using as combination functions $\mathbf{c}_i$ Euclidean combination functions $\mathbf{eucl}_{n,k}(V_1, \ldots, V_k)$ with scaling factor $\boldsymbol{\lambda} = k$ or geometric mean combination functions $\mathbf{sgeomean}_1(V_1, \ldots, V_k)$ with scaling factor $\boldsymbol{\lambda} = 1$ and X'_i for the same model using combination functions *minimum* functions $\mathbf{c}'_i$. Moreover, suppose two simulation traces $X_i(t)$ and $X'_i(t)$ are given with initial values $X'_i(0) \leq X_i(0)$ for all i. If an equilibrium is achieved with equilibrium values $\mathbf{X}_i$ and $\mathbf{X}'_i$, then it holds $\mathbf{X}'_i \leq \mathbf{X}_i$ for all i.

Proof

This follows from Theorem 2 and Propositions 2a) and 3a)

Proposition 4 (weakly scalar-free and increasing functions)

a) Suppose f is increasing and weakly scalar-free for function θ. Then for all $V_1, \ldots, V_k \in R$ it holds

$f(1, \ldots, 1)\ \theta(\min(V_1, \ldots, V_k)) \leq f(V_1, \ldots, V_k) \leq f(1, \ldots, 1)\theta(\max(V_1, \ldots, V_k))$

When moreover f is scalar-free, then

$f(1, \ldots, 1)\ \min(V_1, \ldots, V_k) \leq f(V_1, \ldots, V_k) \leq f(1, \ldots, 1)\ \max(V_1, \ldots, V_k)$

b) Any weighted Euclidean or geometric function with positive weights w_i is monotonically increasing and scalar-free.

c) If

$$g(V_1, \ldots, V_k) = \sqrt[n]{w_1 V_1^n + \ldots + \ w_k V_k^n}$$

is a weighted Euclidean function with positive weights w_i, then g is scalar-free and increasing; moreover, it holds.

$$\sqrt[n]{w_1 + \ldots + w_k}\min(V_1, \ldots, V_k) \leq g(V_1, \ldots, V_k) \leq \sqrt[n]{w_1 + \ldots + w_k}\max(V_1, \ldots, V_k)$$

If g is a weighted Euclidean average function (i.e. the sum of the weights w_i is 1), then

$$\min(V_1,\ \ldots,V_k) \leq g(V_1,\ \ldots,V_k) \leq \max(V_1,\ \ldots,V_k)$$

d) If

$$g(V_1,\ \ldots,V_k) = V_1^{w_1} \ldots .V_k^{w_k}$$

is a weighted geometric function with positive weights w_i, then g is scalar-free and increasing; moreover, it holds.

$$\min(V_1,\ \ldots,V_k) \leq g(V_1,\ \ldots,V_k) \leq \max(V_1,\ \ldots,V_k)$$

Proof

a) Since for each i it holds

$\min(V_1, \ldots , V_k) \leq V_i \leq \max(V_1, \ldots , V_k)$

by monotonic increasing and weakly scalar-free, respectively, it follows.

$$f(V_1, \ldots , V_k) \leq f(\max(V_1, \ldots , V_k), \ldots , \max(V_1, \ldots , V_k))$$
$$= \theta(\max(V_1, \ldots , V_k))\ f(1, \ldots , 1)$$
$$f(V_1, \ldots , V_k) \geq f(\min(V_1, \ldots , V_k), \ldots , \min(V_1, \ldots , V_k))$$
$$= \theta(\min(V_1, \ldots , V_k))\ f(1, \ldots , 1)$$

b) is easy to verify
c) and d) follow from a) and b)

Proposition 5 (scalar-free and strictly increasing functions)

a) Any function composition of scalar-free functions is scalar-free
b) Any function composition of strictly increasing functions is strictly increasing
c) All linear functions with positive coefficients are scalar-free and strictly increasing
d) Any scalar-free function f is weakly scalar-free for $\theta = \text{id}$, the identity function.

Theorem 3 (comparison for any scalar-free function to min and max)

Suppose X_i are the states for a network model (with only positive connection weights and at least some nonzero speed factors) using monotonically increasing combination functions $\mathbf{c}_i$ and X'_i for the same model using monotonically increasing combination functions $\mathbf{c'}_i$. Assume $0 < \Delta t \leq 1/\max_Y(\boldsymbol{\eta}_Y)$; e.g. assume $\boldsymbol{\eta}_Y \leq 1$ for all Y and $0 < \Delta t \leq 1$.

a) If for the initial values it holds $X_i(0) \leq X'_i(0)$, the $\mathbf{c}_i$ are monotonically increasing and scalar-free with $\mathbf{c}_i(1, \ldots , 1) = 1$, and $\mathbf{c}'_i$ is the maximum function $\mathbf{c}'_i = \mathbf{max}$ then for any achieved equilibrium with equilibrium values $\mathbf{X}_i$ and $\mathbf{X}'_i$ it holds $\mathbf{X}_i \leq \mathbf{X}'_i$ for all i.
b) If for the initial values it holds $X'_i(0) \leq X_i(0)$, the $\mathbf{c}_i$ are monotonically increasing and scalar-free with $\mathbf{c}_i(1, \ldots , 1) = 1$, and each $\mathbf{c}'_i$ is the minimum function $\mathbf{c}'_i = \mathbf{min}$ then for any achieved equilibrium with equilibrium values $\mathbf{X}_i$ and $\mathbf{X}'_i$ it holds $\mathbf{X}'_i \leq \mathbf{X}_i$ for all i.

Proof

This follows from Theorem 2b) and Proposition 4a).

Theorem 4 (relating equilibrium values of states in components at different levels)
If the following aggregation conditions are fulfilled.

- The combination functions are normalised, scalar-free and strictly increasing then in an achieved equilibrium:

a) In any level 0 component C
   - All states in C have the same equilibrium value V
   - This V is between the highest and lowest initial value of the states within C

b) If for any level $i > 0$ component C the components C_1, .., C_k are the strongly connected components from which C gets an incoming connection, then
   - The equilibrium values of the states in C are between the highest and lowest equilibrium values of the states in C_1, .., C_k
   - If all states in C_1, .., C_k have the same equilibrium value V, then also all states in C have this same equilibrium value V

Corollary 4 (dependence of all equilibrium values on the values in level 0 components)
If the following aggregation conditions are fulfilled.

- The combination functions are normalised, scalar-free and strictly increasing then in an achieved equilibrium:

c) The equilibrium values of all states in the network
   - are between the highest and lowest equilibrium values of the states in the level 0 components
   - are between the highest and lowest initial values of the states in the level 0 components

d) If all states in all level 0 components C have the same equilibrium value V, then all states of the whole network have that same equilibrium value V

For the special case of a strongly connected network (consisting of one component), this implies:

Corollary 5 (strongly connected networks)
If the following connectivity and aggregation conditions are fulfilled.

- The network is strongly connected
- The combination functions are normalised, scalar-free and strictly increasing then in an achieved equilibrium:
- All states have the same equilibrium value V
- This equilibrium value V is between the highest and lowest initial values of the states

Proposition 6 (relating additive, multiplicative, log-like, and exp-like functions)
Let θ: $R \to \mathbb{R}$ be any function for a finite or infinite interval R in $\mathbb{R}$, then the following hold:

a) If θ is multiplicative, then log o θ is log-like
b) If θ is log-like, then θ o exp is additive
c) If θ is exp-like, then log o θ is additive
d) If θ is multiplicative, then log o θ o exp is additive
e) For any multiplicative function such that $\theta(\alpha) = 0$ for some $\alpha \neq 0$, it holds that $\theta(\alpha) = 0$ for all α. For any nonzero multiplicative function θ it holds $\theta(1) = 1$ and $\theta(\alpha^{-1}) = \theta(\alpha)^{-1}$ for all α.
f) If θ is multiplicative then $\theta(1) = 1$ and $\theta(\alpha^{-1}) = \theta(\alpha)^{-1}$. If a multiplicative θ is injective on Dom(θ), then it has an inverse θ^{-1} with Dom(θ^{-1}) = Range(θ) and Range(θ^{-1}) = Dom (θ); this inverse θ^{-1} is also multiplicative.

Proof

a) This follows from

$$\log(\theta(\alpha\beta)) = \log(\theta(\alpha)\theta(\beta)) = \log(\theta(\alpha)) + \log(\theta(\beta))$$

b) This follows from

$$\theta(\exp(\alpha+\beta)) = \theta(\exp(\alpha)\exp(\beta)) = \theta(\exp(\alpha)) + \theta(\exp(\beta))$$

c) This follows from

$$\log(\theta(\alpha+\beta)) = \log(\theta(\alpha)\theta(\beta)) = \log(\theta(\alpha)) + \log(\theta(\beta))$$

d) This immediately follows from a) and b).
e) Suppose $\theta(\alpha) = 0$ for some $\alpha \neq 0$, then for any β it holds
$\theta(\beta) = \theta(\alpha\beta\alpha^{-1}) = \theta(\alpha)\theta(\beta\alpha^{-1}) = 0.$
Next, for any nonzero θ it holds $\theta(1) = \theta(1^2) = \theta(1)^2$; as it cannot be 0 from this it follows that $\theta(1) = 1$. The last part follows from $\theta(\alpha)\ \theta(\alpha^{-1}) = \theta(\alpha\ \alpha^{-1}) = \theta(1) = 1$.
f) Choose any $\alpha', \beta' \in$ Range(θ), then $\alpha' = \theta(\alpha)$ and $\beta' = \theta(\beta)$ for some $\alpha, \beta \in$ Dom(θ)
Then this follows from
$\theta^{-1}(\alpha'\beta') = \theta^{-1}(\theta(\alpha)\theta(\beta)) = \theta^{-1}(\theta(\alpha\beta)) = \alpha\beta = \theta^{-1}(\alpha')\theta^{-1}(\beta')$

Theorem 5 (characterisation of additive, multiplicative, log-like and exp-like functions) Let θ: $R \to \mathbb{R}$ be continuous. Then the following hold.

a) Assume $R \subseteq \mathbb{R}$ is closed under addition and subtraction with $1 \in R$, then it holds
θ is additive $\Leftrightarrow$ for some c$\in \mathbb{R}$ for all X it holds $\theta(X) = c\ X$.
b) Assume $R \subseteq \mathbb{R}_{>0}$ is closed under multiplication and division with e $\in R$, then it holds
θ is multiplicative $\Leftrightarrow$ for some c$\in \mathbb{R}$ for all X it holds $\theta(X) = X^{c.}$
c) Assume $R \subseteq \mathbb{R}_{>0}$ is closed under multiplication and division with e $\in R$, then it holds
θ is log-like $\Leftrightarrow$ for some c$\in \mathbb{R}$ for all X it holds $\theta(X) = c \log(X)$
d) Assume $R = \mathbb{R}$ is closed under addition and subtraction with $1 \in R$, then it holds
θ is exp-like $\Leftrightarrow$ for some c$\in \mathbb{R}$ for all X it holds $\theta(X) = \exp(cX)$

Proof

Note that all implications from right to left are easy to verify. The opposite implications can be found as follows.

a) Note that $0 = 1–1 \in R$ and $\theta(0) = 0$ as from additivity it follows
$\theta(0) = \theta(0 + 0) = 2\theta(0)$

Therefore for any $c\in \mathbb{R}$ it holds $\theta(X) = cX$ for $X = 0$. Now, first for positive rational numbers $X = p/q \in R$ with $p, q \in \mathbb{N}$ with $p, q > 0$, from additivity it follows

$$q\theta(X) = \theta(qp/q) = \theta(p) = p\ \theta(1)$$

and therefore

$$\theta(X) = cX$$

where $c = \theta(1)$. Moreover, for any negative rational number $X = - p/q\in R$ with $p, q > 0$ it holds

$$\theta(-p/q) + \theta(p/q) = \theta(0) = 0$$

and therefore

$$\theta(X) = \theta(-p/q) = -\theta(p/q) = -cp/q = c\ -p/q = c\ X$$

This proves that $\theta(X) = cX$ for all rational numbers X.
Next, as any real number X is the limit of a sequence r_n, $n\in\mathbb{N}$ of rational numbers and both θ and the function $X \rightarrow cX$ are continuous it holds.

$$\theta(X) = \theta(\lim_{n\to\infty} r_n) = \lim_{n\to\infty} \theta(r_n) = \lim_{n\to\infty} c\, r_n = c \lim_{n\to\infty} r_n = cX$$

b) Note that $R' = \log(R)$ is closed under addition and subtraction and $1 = \log(e)\in R'$. By Proposition 1d) the function log o θ o exp on R' is additive. Therefore, by a) it follows that there is a $c\in\mathbb{R}$ such that for any $X\in R$ for $Y = \log(X)$ it holds
$\log \text{ o } \theta \text{ o } \exp\ (Y) = cY$

From this it follows

$$\exp(\log \text{ o } \theta \text{ o } \exp\ (Y)) = \exp(cY)$$

$$\theta \text{ o } \exp\ (Y) = \exp(cY)$$

$$\theta \text{ o } \exp\ (Y) = \exp(Y)^c$$

$$\theta\ (X) = X^c$$

c) Note that $R' = \log(R)$ is closed under addition and subtraction and $1 = \log(e)\in R'$. By Proposition 1b) the function θ o exp is additive on R'. Therefore, by a) it follows that there is a $c\in \mathbb{R}$ such that for any $X\in R$ for $Y = \log(X)$ it holds
$\theta \text{ o } \exp(Y) = cY$
$\theta \text{ o } \exp(\log(X)) = c\ \log(X)$
$\theta\ (X) = c\ \log(X)$

d) By Proposition 1c) the function log o θ is additive. Therefore, by a) it follows that there is a c$\in \mathbb{R}$ such that for all $X\in R$ it holds
log o θ $(X) = cX$
exp(log o θ $(X)) = \exp(cX)$
θ $(X) = \exp(cX)$

Theorem 6 (relating weakly scalar-free and scalar-free functions)

Consider functions f: $R^k \rightarrow \mathbb{R}$ and θ: $R \rightarrow \mathbb{R}$ for some subset $R\subseteq\mathbb{R}$ which is $\mathbb{R}$ or $\mathbb{R}_{>0}$.

a) If a nonzero function f is weakly scalar-free for function θ, then θ is multiplicative.
b) If, moreover, f is (strictly) monotonically increasing and has at least one positive value, then θ is also (strictly) monotonically increasing. Therefore for the strict monotonically increasing case, θ is injective and has an inverse θ^{-1} on Range(θ), which is also multiplicative.
c) Any nonzero multiplicative function θ is weakly scalar-free for itself.
d) (i) For any weakly scalar-free function f for θ the following are equivalent:
(ii) Range(f) $\subseteq$ Range(θ)
(iii) For all $V_1, \ldots, V_k$ an α exists such that $f(\alpha V_1, \ldots, \alpha V_k) = 1$
e) For each weakly scalar-free function f: $R^k \rightarrow \mathbb{R}$ for any injective θ, the function g: Range$(\theta)^k \rightarrow \mathbb{R}$ defined by $g = f\,\theta^{-1}$ is scalar-free. If, moreover, Range(f) $\subseteq$ Range(θ), then also the function h: $R^k \rightarrow \mathbb{R}$ defined by $h = \theta^{-1} f$ is scalar-free. For strictly increasing f and θ, these functions g,h are strictly increasing too.
f) For each set of strictly increasing and weakly scalar-free functions f_i: $R^k \rightarrow \mathbb{R}_{\geq 0}$ for the same strictly increasing θ, for any linear combination f of the f_i with positive coefficients, the function g: $R^k \rightarrow \mathbb{R}$ defined by $g = f\,\theta^{-1}$ is strictly increasing and scalar-free. If, moreover, Range(f) $\subseteq$ Range(θ), then also the function h: $R^k \rightarrow \mathbb{R}$ defined by $h = \theta^{-1} f$ is strictly increasing and scalar-free.
g) If f: $R^k \rightarrow \mathbb{R}$ is scalar-free, θ: $R \rightarrow R$ is multiplicative and $g = f$ o θ: $R^k \rightarrow \mathbb{R}$, then g is weakly scalar-free for θ. This holds in particular if f is linear.

Proof

a) Suppose $f(V_1, \ldots, V_k) \neq 0$, then from
$\theta(\alpha\beta)\ f(V_1, \ldots, V_k) = f(\alpha\beta V_1, \ldots, \alpha\beta V_k) = \theta(\alpha)\ f(\beta V_1, \ldots, \beta V_k) = \theta(\alpha)\theta(\beta)\ f(V_1, \ldots, V_k)$

it follows that θ is multiplicative.
Suppose, moreover, f is (strictly) monotonically increasing and positive for at least one point $f(V_1, \ldots, V_k) > 0$ and $\alpha \leq \beta$ then from
$\theta(\alpha)\ f(V_1, \ldots, V_k) = f(\alpha V_1, \ldots, \alpha V_k) \leq f(\beta V_1, \ldots, \beta V_k) = \theta(\beta)\ f(V_1, \ldots, V_k)$
and follows that $\theta(\alpha) \leq \theta(\beta)$; it works similarly for the strict condition.
b) This follows from $\theta(\alpha\beta) = \theta(\alpha)\theta(\beta)$
c) (i) ⇒ (ii) suppose Range(f) $\subseteq$ Range(θ), then for any $V_1, \ldots, V_k$ it holds
$f(V_1, \ldots, V_k) \in$ Range(θ)
$f(V_1, \ldots, V_k) = \theta(\beta)$ for some $\beta \in$ Dom(θ)

Then

$$\theta(\beta)^{-1}\ f(V_1, \ldots, V_k) = 1$$

Now pick $\alpha = \beta^{-1}$, then it follows

$$f(\alpha V_1, \ldots, \alpha\ V_k) = \theta(\alpha)\ f(V_1, \ldots, V_k) = \theta(\beta^{-1})\ f(V_1, \ldots, V_k) = \theta(\beta)^{-1}\ f(V_1, \ldots, V_k) = 1$$

(ii) ⇒ (i) Suppose for any given $V_1, \ldots, V_k$ an α exists such that $f(\alpha V_1, \ldots, \alpha V_k) = 1$, then:

$$f(V_1, \ldots, V_k) = f(\alpha^{-1}\alpha V_1, \ldots, \alpha^{-1}\alpha\ V_k) = \theta(\alpha^{-1})\ f(\alpha V_1, \ldots, \alpha\ V_k) = \theta(\alpha^{-1}) \in Range(\theta)$$

d) For g the first part follows from
$g(\alpha V_1, \ldots, \alpha V_k) = f(\theta^{-1}(\alpha V_1), \ldots, \theta^{-1}(\alpha V_k))$
$= f(\theta^{-1}(\alpha)\theta^{-1}(V_1), \ldots, \theta^{-1}(\alpha)\theta^{-1}(V_k))$
$= \theta\theta^{-1}(\alpha)\ f(\theta^{-1}(V_1), \ldots, \theta^{-1}(V_k))$
$= \alpha\ f(\theta^{-1}(V_1), \ldots, \theta^{-1}(V_k))$
$= \alpha\ g(V_1, \ldots, V_k)$

and for h from

$$\begin{aligned} h(\alpha V_1, \ldots, \alpha V_k) = \theta^{-1}(f(\alpha V_1, \ldots, \alpha V_k)) \\ &= \theta^{-1}(\theta(\alpha)\ f(V_1, \ldots, V_k)) \\ &= \theta^{-1}\theta(\alpha)\ \theta^{-1}(f(V_1, \ldots, V_k)) \\ &= \alpha\ \theta^{-1}(f(V_1, \ldots, V_k)) \\ &= \alpha\ h(V_1, \ldots, V_k) \end{aligned}$$

The second part follows from a).

e) This follows from d) and a).

f) This follows from

$$\begin{aligned} g(\alpha V_1, \ldots, \alpha V_k) = f(\theta(\alpha V_1), \ldots, \theta(\alpha V_k)) \\ &= f(\theta(\alpha)\theta(V_1), \ldots, \theta(\alpha)\theta(V_k)) \\ &= \theta(\alpha)f(\theta(V_1), \ldots, \theta(V_k)) \\ &= \theta(\alpha)\ g(V_1, \ldots, V_k) \end{aligned}$$

Proposition 7 (function conjugate operator)

Let subsets $R, S \subseteq \mathbb{R}$ be given, and functions g: $S^k \to S$, f: $R^k \to R$, and bijective θ: $S \to R$. Then the following hold:

a) Then the following are equivalent:
   (i) g is a function conjugate of f by θ
   (ii) The following commutation rules for θ, f and g hold:$\theta\ g = f\ \theta\theta^{-1}f = g\ \theta^{-1.}$

b) If a)(i) and (ii) hold, then for any g such an f is unique and can be denoted by $f = S_\theta(g)$ for a *function conjugate operator* S_θ; similarly, $g = S_{\theta^{-1}}(f)$ for function conjugate operator $S_{\theta^{-1}}$, so it holds:$\theta\ g = S_\theta(g)\ \theta\theta^{-1}f = S_{\theta^{-1}}\ (f)\ \theta^{-1.}$

These operators S_θ and $S_{\theta^{-1}}$ are each other's inverse and they preserve function addition and composition: for all f, g, f_1, f_2, g_1 and g_2 it holds.

$$S_{\theta^{-1}}S_\theta(g) = g.$$

$$S_\theta S_{\theta^{-1}}(f) = f.$$

$$S_\theta(g_1 + g_2) = S_\theta(g_1) + S_\theta(g_2)$$

$$S_\theta(g_1 \text{ o } g_2) = S_\theta(g_1) \text{ o } S_\theta(g_2)$$

$$S_{\theta^{-1}}(f_1 + f_2) = S_{\theta^{-1}}(f_1) + S_{\theta^{-1}}(f_2)$$

$$S_{\theta^{-1}}(f_1 \text{ o } f_2) = S_{\theta^{-1}}(f_1) \text{ o } S_{\theta^{-1}}(f_2)$$

Moreover, when conjugate operators S_{θ_1} and S_{θ_2} for θ_1 and θ_2 are applied in turn, it holds.

$$S_{\theta_1\theta_2}\ (g) = S_{\theta_1}S_{\theta_2}(g)\ S_{\theta_1}(\theta_2)\ \theta_1\ \theta_2^{-1}\ \theta_1^{-1} \text{ for all g}$$

If in addition, $\theta_1\theta_2\ = \theta_2\theta_1$ then $S_{\theta_1}S_{\theta_2} = S_{\theta_1\theta_2}$:

$$S_{\theta_1}S_{\theta_2}(g) = S_{\theta_1\theta_2}(g) \text{ for all g}$$

Proof

a) (i) ⇒ (ii) This follows from

$$\theta\ g = \theta\ \theta^{-1} \text{ o } f \text{ o } \theta = f \text{ o } \theta$$

and

$$\theta^{-1}\ f = \theta^{-1}\ f\ \theta\ \theta^{-1} = g\ \theta^{-1.}$$

(ii) ⇒ (i) This follows from

$$g = \theta^{-1}\ \theta\ g = \theta^{-1} \text{ o } f \text{ o } \theta$$

b) First, suppose $\theta\, g = f_1\ \theta = f_2\ \theta$, then from θ bijective it follows $f_1 = f_2$. Then an operator S_θ exists and it holds

$$\theta\ g = S_\theta(g)\ \theta$$

$$\theta^{-1}f = S_{\theta^{-1}}(f)\ \theta^{-1}$$

Furthermore, consider.

$$\theta\ (g_1 + g_2) = S_\theta\ (g_1 + g_2)\ \theta$$

$$\theta\ (g_1 + g_2) = \theta\ g_1 + \theta\ g_2 = S_\theta(g_1)\ \theta + S_\theta(g_2)\ \theta = (S_\theta(g_1) + S_\theta(g_2))\ \theta$$

Then,

$$S_\theta\ (g_1 + g_2)\ \theta = (S_\theta(g_1) + S_\theta(g_2))\ \theta$$

$$S_\theta\ (g_1 + g_2) = S_\theta(g_1) + S_\theta(g_2)$$

Also,

$$\theta\ (g_1 \text{ o } g_2) = S_\theta(g_1 \text{ o } g_2)\ \theta$$

$(\theta\, g_1)\ o\ g_2 = S_\theta(g_1)\theta\, g_2 = S_\theta(g_1)\ S_\theta(g_2)\ \theta$

Here $k_1k_2 = k$; therefore,

$S_\theta(g_1\ o\ g_2) = S_\theta(g_1)\ o\ S_\theta(g_2)$

And the same applies to θ^{-1}.

When conjugate operators S_{θ_1} and S_{θ_2} for θ_1 and θ_2 are applied in turn, it holds

$\theta_1\theta_2\, g = S_{\theta_1\theta_2}\ (g)\ \theta_1\theta_2 = S_{\theta_1\theta_2}\ (g)\ \theta_1\ \theta_2$

$\theta_2\theta_1\ g = \theta_2\ S_{\theta_1}(g)\ \theta_1 = S_{\theta_2}(S_{\theta_1}(g))\ \theta_1\ \theta_2$

Therefore

$S_{\theta_1\theta_2}\ (g) = S_{\theta_1}S_{\theta_2}(g)$

By applying this to $\theta_1 = \theta$ and $\theta_2 = \theta^{\ -1}$ it follows that

$S_{\theta^{-1}}S_\theta(g) = S_{\theta^{-1}\theta}(g) = S_{id}(g) = g$

$S_\theta S_{\theta^{-1}}(f) = S_{\theta\theta^{-1}}(f) = S_{id}(f) = f$

Theorem 7 (from scalar-free functions to scalar-free conjugates by multiplicative θ)

a) For any scalar-free function f: $R^k \to R$ with $R = \mathbb{R}_{\geq 0}$, all of its conjugates by a multiplicative θ: $R \to R$ are also scalar-free.

b) More specifically, for any scalar-free function f, for any positive real number n the function g defined by

$$g(V_1,\ \ldots,\ V_k) = \sqrt[n]{f(V_1^n,\ \ldots,\ V_k^n)}$$

is a conjugate of f by the multiplicative function $X \to X^n$ and therefore is also scalar-free.

a) All weighted euclidean functions are conjugates of linear functions by a multiplicative function θ and therefore are scalar-free. In particular, this holds for all functions $\mathbf{eucl}_{n,\lambda}$**(..)**.

Proof

a) If $f(V_1,\ \ldots,\ V_k)$ scalar-free and θ multiplicative and $g = \theta^{\ -1}$ o f o θ, i.e.
$g(V_1, \ldots, V_k) = \theta^{\ -1}(f(\theta(V_1), \ldots, \theta(V_k))$

then:

$$\begin{aligned}g(\alpha V_1, \ldots, \alpha V_k) &= \theta^{\ -1}(f(\theta(\alpha V_1), \ldots, \theta(\alpha V_k))\\ &= \theta^{\ -1}(f(\theta(\alpha)\theta(V_1), \ldots, \theta(\alpha)\theta(V_k))\\ &= \theta^{\ -1}(\theta(\alpha)f(\theta(V_1), \ldots, \theta(V_k))\\ &= \theta^{\ -1}(\theta(\alpha))\theta^{-1}(f(\theta(V_1), \ldots, \theta(V_k)))\\ &= \alpha\ g(V_1, \ldots, V_k)\end{aligned}$$

Therefore g is scalar-free.

b) Substitute $\theta(X) = X^n$ and $\theta^{-1}(X) = X^{(1/n)}$, then

$$g(V_1, \ldots, V_k) = \theta^{-1}(f(\theta(V_1), \ldots, \theta(V_k)) = \sqrt[n]{f(V_1^n, \ldots, V_k^n)}$$

Then by a) this function is scalar-free.

c) When starting with a linear function in b), you get a weighted Euclidean function.

Theorem 8 (from linear to scalar-free conjugates by log-like θ)

a) For any normalised linear function all of its conjugates by a log-like θ are scalar-free
b) More specifically, for any normalised linear function *f*, the function *g* defined by

$$g(V_1, \ldots, V_k) = \exp(f(\log(V_1), \ldots, \log(V_k)))$$

is a conjugate of a linear function by the standard log-like function $\theta = \log$ and therefore is scalar-free.

c) All weighted geometric mean functions are conjugates of a normalised linear function by a log-like function θ and therefore are scalar-free. In particular, this also holds for all functions **sgeomean**$_\lambda$**(..)**.

Proof

a) If $f(V_1, \ldots, V_k)$ linear and normalised and θ log-like and $g = \theta^{-1} \text{ o } f \text{ o } \theta$, i.e.
$g(V_1, \ldots, V_k) = \theta^{-1}(f(\theta(V_1), \ldots, \theta(V_k))$

then g is scalar-free:

$$\begin{aligned} g(\alpha V_1, \ldots, \alpha V_k) &= \theta^{-1}(f(\theta(\alpha V_1), \ldots, \theta(\alpha V_k)) \\ &= \theta^{-1}(f(\theta(\alpha)+\theta(V_1), \ldots, \theta(\alpha)+\theta(V_k)) \\ &= \theta^{-1}(f(\theta(\alpha), \ldots, \theta(\alpha)) + f(\theta(V_1), \ldots, \theta(V_k))) \\ &= \theta^{-1}(f(\theta(\alpha), \ldots, \theta(\alpha))) * \theta^{-1}(f(\theta(V_1), \ldots, \theta(V_k))) \\ &= \theta^{-1}(\theta(\alpha)) * g(V_1), \ldots, V_k) \\ &= \alpha\, g(V_1), \ldots, V_k) \end{aligned}$$

b) If

$f(V_1, \ldots, V_k) = w_1 V_1 + \ldots + w_k V_k$

then

$$\begin{aligned} g(V_1, \ldots, V_k) &= \exp(w_1 \log(V_1) + \ldots + w_k \log(V_k)) \\ &= V_1^{w_1} \ldots . V_k^{w_k} \end{aligned}$$

c) This immediately follows from b). As **sgeomean**$_\lambda$**(..)** is the weighted geometric function **sgeomean**$_1$**(..)** times a constant factor, it is also scalar-free.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Gülay Canbaloğlu combines her studies in Computer Engineering, Sociology and AI at the prestigious private Koç University in Istanbul with her work as Ph.D. candidate for the Safety and Security Science group at Delft University of Technology under supervision of Prof. Dr. Jan Treur and Dr. Peter Roelofsma. Her research focus is on computational modelling of organisational learning with special application to learning of just safety culture in health care organisations. Part of her research is conducted in collaboration with Dr. Anna Wiewiora from the School of Management of Queensland University of Technology in Brisbane. In her recent work she reached an important achievement by successfully using self-modeling networks to obtain suitable computational models of complex multilevel organisational learning processes. A joint Springer Nature book about this is in preparation and will come out by the end of 2022 or beginning of 2023.

Jan Treur has been a full professsor of AI since 1990 and is a well-recognized expert on the area of multidisciplinary human-like AI-modeling. He has published over 700 well-cited papers about cognitive, affective, and social modeling and AI systems making use of such models. He has also supervised more than 40 Ph.D. students in these areas and from 2016 on written and edited three books on (adaptive) network-oriented AI-modeling and its application in various other disciplines. Current research addresses modeling of higher-order adaptive processes by self-modeling network models with a specific focus on mental processes based on internal mental models and their use by internal simulation, their learning or formation (including organisational learning), and the control over them. An application focus is on the development and use of shared mental models supporting the road toward a just safety culture in organisations such as hospitals. A joint Springer Nature book about computational modeling of multilevel organisational learning is in preparation and will come out by the end of 2022 or beginning of 2023.

References

Abraham, W. C., & Bear, M. F. (1996). Metaplasticity: The plasticity of synaptic plasticity. *Trends in Neurosciences*, *19*(4), 126–130. https://doi.org/10.1016/S0166-2236(96)80018-X

Anton, H. (1987). *Elementary Linear Algebra (5th ed.)*. Wiley.

Bloem, R., Gabow, H. N., & Somenzi, F. (2006). An algorithm for strongly connected component analysis in n log n symbolic steps. *Formal Methods in System Design*, *28*(1), 37–56. https://doi.org/10.1007/s10703-006-4341-z

Canbaloğlu, G., & Treur, J. (2021a). Context-sensitive mental model aggregation in a second-order adaptive network model for organisational learning. In *Proceedings of the 10th International Conference on Complex Networks and their Applications. Studies in Computational Intelligence*, vol 1015, pp 411–423. Springer Nature.

Canbaloğlu, G., & Treur, J. (2021b). Using boolean functions of context factors for adaptive mental model aggregation in organisational learning. In *Proc. of the 12th International Conference on Brain-Inspired Cognitive Architectures, BICA'21. Studies in Computational Intelligence*, vol 1032, pp 54–68. Springer Nature.

Canbaloğlu, G., Treur, J., & Roelofsma, P. H. M. P. (2022a). Computational modeling of organisational learning by self-modeling networks. *Cognitive Systems Research*, *73*, 51–64. https://doi.org/10.1016/j.cogsys.2021.12.003

Canbaloğlu, G., Treur, J., & Roelofsma, P. H. M. P. (2022b). An adaptive self-modeling network model for multilevel organisational learning. In *Proceedings of the 7th International Congress on Information and Communication technology, ICICT'22* (Lecture Notes in Networks and Systems). Springer Nature.

Canbaloğlu, G., Treur, J., & Wiewiora, A. (2022a). Computational modeling of multilevel organisational learning: From conceptual to computational mechanisms. Proceedings of Computational intelligence: Automate your world. The second International Conference on Information technology, InCITe'22. Lecture Notes in Electrical Engineering, to appear. Springer Nature.

Canbaloğlu, G., Treur, J., & Wiewiora, A. (2022b). Computational modeling of the role of leadership style for its context-sensitive control over multilevel organisational learning. Proceedings of the 7th International Congress on Information and Communication technology, ICICT'22. Lecture Notes in Networks and Systems, Springer Nature.

Canbaloğlu, G., Treur, J., & Wiewiora, A. (Eds.). (2022c). *Computational modeling of multilevel organisational learning and its control using self-modeling network models*. Springer Nature.

Craik, K. J. W. (1943). *The nature of explanation*. University Press.

Crossan, M. M., Lane, H. W., & White, R. E. (1999). An organizational learning framework: From intuition to institution. *The Academy of Management Review*, *24*(3), 522–537. https://doi.org/10.2307/259140

Dummit, D. S., & Foote, R. M. (2004). *Abstract Algebra* (3rd ed.). John Whily & Sons, Inc.

Fleischer, L. K., Hendrickson, B., & Pınar, A. (2000). On identifying strongly connected components in parallel. In J. Rolim (Ed.), *Parallel and distributed processing. IPDPS 2000* (Lecture Notes in Computer Science, Vol. 1800, pp. 505–511). Springer.

Harary, F., Norman, R. Z., & Cartwright, D. (1965). *Structural models: An introduction to the theory of directed graphs*. Wiley.

Kim, D. H. (1993). The link between individual and organisational learning. *Sloan Management Review,* Fall 1993. Reprinted In D. A. Klein (Ed.), *The strategic management of intellectual capital* (pp. 37–50). Routledge-Butterworth-Heinemann.

Łacki, J. (2013). Improved deterministic algorithms for decremental reachability and strongly connected components. *ACM Transactions on Algorithms*, *9*(3), 27. https://doi.org/10.1145/2483699.2483707

Nering, E. D. (1970). *Linear algebra and matrix theory* (2nd ed.). Wiley.

Treur, J. (2016). Verification of temporal-causal network models by mathematical analysis. *Vietnam Journal of Computer Science*, *3*(4), 207–221. https://doi.org/10.1007/s40595-016-0067-z

Treur, J. (2018). Relating emerging network behaviour to network structure. *Proc. Of the 7th International Conference on Complex Networks and their Applications, Complexnetworks'18, Vol. 1. Studies in Computational Intelligence* (Vol. 812, pp. 619–634). Springer Publishers.

Treur, J. (2020a). Analysis of a network's asymptotic behavior via its structure involving its strongly connected components. *Network Science*, *8*(S1), S82–S109. https://doi.org/10.1017/nws.2019.24

Treur, J. (2020b). *Network-oriented modeling for adaptive networks: Designing higher-order adaptive biological, mental and social network models*. Springer Nature Publishers.

Treur, J., & Van Ments, L. (2022). *Mental models and their dynamics, adaptation, and control: A self-modeling network modeling approach*. Springer Nature.

Van Ments, L., & Treur, J. (2021). Reflections on dynamics, adaptation and control: A cognitive architecture for mental models. *Cognitive Systems Research*, *70*, 1–9. https://doi.org/10.1016/j.cogsys.2021.06.004

Wiewiora, A., Chang, A., & Smidt, M. (2020). Individual, project and organizational learning flows within a global project-based organization: Exploring what, how and who. *International Journal of Project Management*, *38*(4), 201–214. https://doi.org/10.1016/j.ijproman.2020.03.005

Wiewiora, A., Smidt, M., & Chang, A. (2019). The 'how' of multilevel learning dynamics: A systematic literature review exploring How mechanisms bridge learning between individuals, teams/projects and the organization. *European Management Review*, *16*(1), 93–115. https://doi.org/10.1111/emre.12179

Wijs, A., Katoen, J. P., & Bošnacki, D. (2016). Efficient GPU algorithms for parallel decomposition of graphs into strongly connected and maximal end components. *Formal Methods in System Design*, *48*(3), 274–300. https://doi.org/10.1007/s10703-016-0246-7