Multi-Valued Cognitive Maps: Calculations with Linguistic Variables without Using Numbers

Dmitry Maximov

Trapeznikov Institute of Control Science Russian Academy of Sciences, 65 Profsoyuznaya str, Moscow

Abstract

A concept of multi-valued cognitive maps is introduced in this paper. The concept expands the fuzzy one. However, all variables and weights are not linearly ordered in the concept, but are only partially-ordered. Such an approach allows us to operate in cognitive maps with partially-ordered linguistic variables directly, without vague fuzzification/defuzzification methods. Hence, we may consider more subtle differences in degrees of experts’ uncertainty, than in the fuzzy case. We prove the convergence of such cognitive maps and give a simple computational example which demonstrates using such a partially-ordered uncertainty degree scale.

Keywords: multi-valued neural networks, multi-valued cognitive maps, fuzzy cognitive maps, linguistic variable lattice

2010 MSC: 68Q85, 68T37

1. Introduction

The fuzzy cognitive map (FCM) concept was introduced by B. Kosko in [1]. FCMs are considered as feedback models of causality, in which fuzzy values are assigned to concepts and causal relationships amongst them. An increase in the value of a concept implies a corresponding positive or negative increase in values of other concepts connected to it, according to the relationships. The concepts are also called nodes, and the relationships are called weights. Thus, we obtain a network similar to a neural network in which all the variables and weights take values in the interval [0, 1].

Email address: jhanjaa@ipu.ru; dmmax@inbox.ru (Dmitry Maximov)

Preprint submitted to Fuzzy Sets and Systems August 11, 2021
Fuzzy cognitive maps have been studied and used in various fields of engineering and hard sciences [2]. Their role is especially important in investigations of the behavior of complex dynamic systems [3], [4], [5], [6]. This is due to the fact that human knowledge uncertainty affects the systems definition and processing [7]. However, fuzzy modelling of uncertainty is rather poor: the theory operates with only linearly-ordered experts’ valuations, which in reality can be unordered: e.g., “yes and no” and “neither yes nor no”.

The main contribution and the novelty of this paper is that we use a lattice (i.e., a partially-ordered set) in cognitive maps as the scale of experts’ valuations (i.e., weights) and as the set of variables (i.e., concept values), instead of a linearly-ordered set. Thus, we may consider more subtle differences in degrees of experts’ uncertainty, than in the fuzzy case. The approach continues the line of investigations in which a system state is estimated, not by numbers, but by various objects (sets, graphs, images, etc.) making up different lattices: [8], [9], [10], [11]. Also, exactly this concept was used in the research dedicated to the related area of multi-valued neural networks: [12], [13], [14]. Similar to these papers, we call such cognitive maps multi-valued ones (MVCM’s).

Thus, all the variables and weights are partially-ordered linguistic ones here, and we do not use numbers in the cognitive maps’ calculations. Nevertheless, such maps converge, and we consider the conditions of convergence (Sec. 3). In Sec. 4 we represent a learning algorithm for weights, applicable when we know the desired range for output values. In Sec. 5 we consider a simple computational model of a hybrid energy system. In Sec. 6 we discuss our experimental results and compare them with the previous ones. We give the necessary definitions used in the text in Sec. 2 and we conclude the paper in Sec. 7.

2. Backgrounds

2.1. Lattices [15]

Definition 2.1. A lattice is a partially-ordered set having, for any two elements, their exact upper bound or join $\lor$ (sup, max) and the exact lower bound or meet $\land$ (inf, min).
bound or meet \(\land (\inf, \min)\).

**Definition 2.2.** The **exact upper bound** of a subset \(X \subseteq P\) of a partially-ordered set \(P\) is the smallest \(P\)-element \(a\), larger than all the elements of \(X\):
\[
\min(a) \in P : a \geq x, \forall x \in P.
\]

**Definition 2.3.** The **exact lower bound** is dually defined as the largest \(P\)-element, smaller than all the elements of \(X\).

**Definition 2.4.** A **complete lattice** is a lattice in which any two subsets have a join and a meet. This means that in a non-empty complete lattice there is the largest “\(\top\)” and the smallest “\(0\)” elements.

If we take such a lattice as a scale of truth values in a multi-valued logic, then the largest element will correspond to complete truth (true), the smallest to complete falsehood (false), and intermediate elements will correspond to partial truth in the same way as the elements of the segment \([0,1]\) evaluate partial truth in fuzzy logic.

In logics, with such a scale of truth values, implication can be determined by multiplying lattice elements, or internally, only from lattice operations.

**Definition 2.5.** Lattice elements, from which all the others are obtained by join and meet operations are called **generators** of the lattice.

**Definition 2.6.** A lattice is called **atomic** if every two of its generators have null meets.

**Definition 2.7.** A **Brouwer lattice** is a lattice that has internal implications.

**Definition 2.8.** In such a lattice, the **implication** \(c = a \Rightarrow b\) is defined as the largest \(c\):
\[
a \land b = a \land c.
\]

Distribution laws for join and meet are satisfied in Brouwer lattices. The converse is true only for finite lattices.
2.2. Residuated Lattices [16]

In non-distributive lattices, the implication cannot be defined. However, we may introduce a multiplication of the lattice elements and use it to define an external implication.

**Definition 2.9.** A *residuated lattice* is an algebra \((L, \lor, \land, \cdot, 1, \rightarrow, \leftarrow)\) satisfying the following conditions:

- \((L, \lor, \land)\) is a lattice;
- \((L, \cdot, 1)\) is a monoid;
- \((\rightarrow, \leftarrow)\) is a pair of residuals of the operation \(\cdot\), that means
  \[
  \forall x, y \in L : x \cdot y \leq z \iff y \leq x \rightarrow z \iff x \leq z \leftarrow y
  \]

In this case, the operation \(\cdot\) is order preserving in each argument and for all \(a, b \in L\) both the sets \(\{y \in L | a \cdot y \leq b\}\) and \(\{x \in L | x \cdot a \leq b\}\) each contains a greatest element \((a \rightarrow b\) and \(b \leftarrow a\) respectively).

The monoid multiplication \(\cdot\) is distributive over \(\lor\):

\[
x \cdot (y \lor z) = (x \cdot y) \lor (x \cdot z).
\]

Also, \(x \cdot 0 = 0 \cdot x = 0\). A special case of residuated lattices is a Heyting algebra, when the monoid multiplication coincides with \(\land\).

In non-commutative monoids, residuals \(\rightarrow\) and \(\leftarrow\) can be understood as having a temporal quality: \(x \cdot y \leq z\) means “\(x\) then \(y\) entails \(z\),” \(y \leq x \rightarrow z\) means “\(y\) estimates the transition had \(x\) then \(z\),” and \(x \leq z \leftarrow y\) means “\(x\) estimates the opportunity if-ever \(y\) then \(z\).” You may think about \(x, y,\) and \(z\) as bet, win, and rich correspondingly (Wikipedia).

**Definition 2.10.** A residuated lattice \(A\) is said to be **integrally closed** if it satisfies the equations \(x \cdot y \leq x \implies y \leq 1\) and \(y \cdot x \leq x \implies y \leq 1\), or equivalently, the equations \(x \rightarrow x = 1\) and \(x \leftarrow x = 1\) [17].

Any upper or lower bounded integrally closed residuated lattice \(L\) is integral, i.e., \(a \leq 1, \forall a \in L\) [17].
Figure 1: An example of a cognitive map

3. Cognitive Map Convergence

A graphical representation of an example of a (multi-valued) cognitive map (MVCM) is depicted in Fig. 1. The expert knowledge on the behaviour of the system is stored in the structure of such a graphical representation of the map. Each concept $C_i$ represents a characteristic of the system under consideration. It may represent goals, events, actions, states, etc. of the system. Each $C_i$ is characterized by an element $A_i$ of a lattice $L$, which represents the value of the concept, and it is obtained from an expert opinion about the real value of the systems’ variable representing this concept. Causality between concepts allows degrees of causality, which also belong to the lattice $L$; thus, the weights $w_{ij}$’s of the connections are the lattice elements and represent the expert uncertainty degrees of the concepts’ mutual influences. The value of $w_{ij}$ indicates how strongly concept $C_i$ influences concept $C_j$. A simple example of such a lattice of experts’ opinions and uncertainty degrees is depicted in Fig. 2.

The equation that calculates the values of concepts of FCM’s with $n$ nodes, can be written in its general form as:

$$A_i^k = f(\sum_{j=1,j\neq i}^{n} w_{ji}A_j^{k-1} + d_{ii}A_i^{k-1}).$$

(1)

Here $A_i^k$ is the value of the concept $C_i$ at discrete time $k$, and $d_{ii}$ is a value of self-feedback to node $i$. All values belong to the interval $[0, 1]$, and the
function \( f \) normalizes its argument up to this interval. Existence and uniqueness of solutions of (1) in FCM’s are proved in [18] for some such trimming functions \( f \)’s.

We use the equation (1) for MVCM’s in the following form:

\[
A_i^k = c_i^{k-1} \cdot f_i^{k-1} \cdot \bigvee_{j=1}^{n} w_{ji} \cdot A_j^{k-1}
\]  

(2)

where all quantities take values in a residuated atomic lattice \( L \), and we use the monoid multiplication and the lattice join, instead of the sum and numeric multiplication. We restrict ourselves to atomic lattices, since we prove the maps’ convergence only in this case. Thus, the example of the simple possible lattice in Fig. 2 is out of our consideration, and we consider a more complicated variant in our modelling example.

We do not need the quantities \( f_i^{k} \)’s and \( c_i^{k} \)’s in (2) to norm the joins, since all joins are inside the lattice. Thus, we use picking up of these quantities’ values in order to provide the map convergence. Although \( f_i \)’s and \( c_i \)’s do not normalize \( wA \), they play the role of adjusting function: for each \( wA \) we get corresponding \( f_i \) and \( c_i \) values.

**Theorem 3.1.** Multi-Valued Cognitive Maps determined by equation (2) where concepts and weights take values in a finite atomic residuated and integrally-
closed lattice \( L \) (hence, \( L \) is integral), converge under a suitable choice of \( c^k_i \)'s and \( f^k_i \)'s.

**Proof 3.1.** **Denotation 1.** We denote the set of generators of a lattice element \( w \) by \( \{w\} \) and the matrix of such sets by \( \{w\}_{ij} \). The matrix elements are the sets of generators of, e.g., the weight matrix elements.

Matrices \( w_{ij} \) and \( \{w\}_{ij} \) are one-to-one correspondent to each other in atomic lattices.

**Denotation 2.** A minus sign will denote the difference operation

\[
\{A\} - \{B\} = (\{A\} \cup \{B\}) \ominus (\{A\} \cap \{B\}),
\]

where \( \ominus \) is the set difference\(^2\).

For convenience, in what follows, we will omit the curly braces, thus, we use \( A_1^i - A_0^i \) instead of \( \{A_1^i\} - \{A_0^i\} \).

The map (2) converges, if

\[
A_i^{k+1} \subseteq A_i^k - A_i^{k-1}, \tag{4}
\]

since the lattice \( L \) is bounded below by \( \emptyset \). The process of calculating \( A_i \) stops when \( A_i^{k+1} = A_i^k = A_i^{k-1} \) for all \( i \). Let us consider the following sequence of inequalities:

\[
\left\{ \begin{array}{c}
A_3^i - A_2^i \subseteq A_1^i - A_0^i; \\
A_3^i \cup A_2^i \ominus A_3^i \cap A_2^i \subseteq A_2^i \cup A_1^i \ominus A_2^i \cap A_1^i; \\
A_3^i \cup A_2^i \ominus A_3^i \cap A_2^i \subseteq A_1^i \cup A_0^i \ominus A_1^i \cap A_0^i;
\end{array} \right. \]

and equalities cannot occur simultaneously.

Hence,

\[
\begin{align*}
A_1^2 & \subseteq A_1^1 \cup A_0^0; \\
A_1^3 & \subseteq A_2^1 \cup A_1^1; \\
& \quad \vdots \\
A_i^k & \subseteq A_i^{k-1} \cup A_i^{k-2}; \tag{5}
\end{align*}
\]

---

\(^2\)Given set \( A \) and set \( B \), the set difference of set \( B \) from set \( A \) is the set of all element in \( A \), but not in \( B \).
\[ A_i^k \supseteq A_i^{k-1} \cap A_i^{k-2}, \quad (6) \]

and equalities cannot occur simultaneously up to the end of the process in order to satisfy (4). Substituting (2) into 5, and passing to the lattice notation, we get:

\[
\begin{align*}
\sum_{j=1}^{n} w_{ji} \cdot (c_j^0 \cdot f_j^0 \cdot \bigvee_{l=1}^{n} w_{lj} \cdot A_l^0) &\leq (c_i^0 \cdot f_i^0 \cdot \bigvee_{j=1}^{n} w_{ji} \cdot A_j^0) \vee A_i^0; \quad (7) \\
\sum_{j=1}^{n} w_{ji} \cdot A_j^2 &\leq \sum_{i=1}^{n} \left( c_i^1 \cdot f_i^1 \cdot \bigvee_{j=1}^{n} w_{ji} \cdot (c_j^0 \cdot f_j^0 \cdot \bigvee_{l=1}^{n} w_{lj} \cdot A_l^0) \vee c_i^0 \cdot f_i^0 \cdot \bigvee_{j=1}^{n} w_{ji} \cdot A_j^0 \right) \vee c_i^1 \cdot f_i^1 \cdot \bigvee_{j=1}^{n} w_{ji} \cdot A_j^0. \quad (8)
\end{align*}
\]

Since all \( c_i \leq 1 \) in the integral lattice, we may define \( f_i^k \)'s as right residuals\(^3\):

\[
\begin{align*}
f_i^1 &= [A_i^1 \vee A_i^0] \leftarrow \left( \bigvee_{j=1}^{n} w_{ji} \cdot A_j^1 \right) = \\
&= [c_i^0 \cdot f_i^0 \cdot \bigvee_{j=1}^{n} w_{ji} \cdot A_j^0] \leftarrow \left( \bigvee_{j=1}^{n} w_{ji} \cdot c_j^0 \cdot f_j^0 \cdot \bigvee_{l=1}^{n} w_{lj} \cdot A_l^0 \right); \quad (9)
\end{align*}
\]

\[
\begin{align*}
f_i^2 &= [A_i^2 \vee A_i^1] \leftarrow \left( \bigvee_{j=1}^{n} w_{ji} \cdot A_j^2 \right) = \\
&= [c_i^1 \cdot f_i^1 \cdot \left( \bigvee_{i=1}^{n} w_{ji} \cdot c_j^0 \cdot f_j^0 \cdot \bigvee_{l=1}^{n} w_{lj} \cdot A_l^0 \right) \vee c_i^0 \cdot f_i^0 \cdot \bigvee_{j=1}^{n} w_{ji} \cdot A_j^0] \leftarrow \\
&\quad \left( \bigvee_{j=1}^{n} w_{ji} \cdot (c_j^1 \cdot f_j^1 \cdot \bigvee_{i=1}^{n} w_{ji} \cdot (c_i^0 \cdot f_i^0 \cdot \bigvee_{m=1}^{n} w_{mj} \cdot A_m^0))) \right). \quad (10)
\end{align*}
\]

\[
\begin{align*}
f_i^k &= [A_i^k \vee A_i^{k-1}] \leftarrow \left( \bigvee_{j=1}^{n} w_{ji} \cdot A_j^k \right) \quad (11)
\end{align*}
\]

\(^3\text{Since, } a \cdot b \leq c \text{ means for a maximal } a: a = c \leftarrow b\)
By recursion, we can obtain all the $f_i^k$’s so as to satisfy the expressions (3).

Also, $f_i^k$’s and $c_i^k$’s must satisfy (6) in order to satisfy (4). However, (6) holds for such chosen $f_i^k$’s. Indeed, it must be $A_i^{k+1} = c_i^k \cdot f_i^k \cdot \bigvee_{i=1}^n w_{ji} A_j^k \supseteq A_i^k \land A_i^{k-1}$.

Let us denote

$$r_i^k = [A_i^k \land A_i^{k-1}] \leftarrow \bigvee_{i=1}^n w_{ji} \cdot A_j^k.$$ (12)

Then, it should be $c_i^k \cdot f_i^k \supseteq r_i^k$ and $c_i^k \cdot f_i^k = r_i^k$ only if $A_i^{k+1} = A_i^k \land A_i^{k-1}$. In this case, $r_i^k \cdot \bigvee_{i=1}^n w_{ji} \cdot A_j^k = A_i^k \land A_i^{k-1}$. Since,

$$f_i^k \cdot \bigvee_{i=1}^n w_{ji} \cdot A_j^k = \{[A_i^k \lor A_i^{k-1}] \leftarrow (\bigvee_{i=1}^n w_{ji} \cdot A_j^k)\} \cdot \bigvee_{i=1}^n w_{ji} \cdot A_j^k \leq [A_i^k \lor A_i^{k-1}]$$ (13)

and

$$r_i^k \cdot \bigvee_{i=1}^n w_{ji} \cdot A_j^k \leq [A_i^k \land A_i^{k-1}],$$ (14)

where such $f_i^k$ and $r_i^k$ are maximal at satisfying the inequalities, we obtain $f_i^k \supseteq r_i^k$, since the lattice is integrally-closed. We may restrict from below $c_i^k \supseteq r_i^k \leftarrow f_i^k$ where it may be $c_i^k = r_i^k \leftarrow f_i^k$ only if $[r_i^k \leftarrow f_i^k] \cdot f_i^k = r_i^k$.

It must be $c_i^k > r_i^k \leftarrow f_i^k$ if $f_i^k \cdot \bigvee_{i=1}^n w_{ji} \cdot A_j^k = A_i^k \lor A_i^{k-1}$ in (3) in order to avoid the simultaneous equality in (3) and (6). However, the simultaneity is not possible up to the end of the process, because, in this case, $r_i^k \leftarrow f_i^k \neq 1$, since

$$f_i^k \cdot \bigvee_{i=1}^n w_{ji} \cdot A_j^k = A_i^k \lor A_i^{k-1} > A_i^k \land A_i^{k-1} \supseteq r_i^k \cdot \bigvee_{i=1}^n w_{ji} \cdot A_j^k.$$ (15)

Hence, $f_i^k > r_i^k$ up to the end of the process. Only at the end of the process, $f_i^k = r_i^k$ and $c_i^k = 1$. Therefore, the process converges if $c_i^k \supseteq r_i^k \leftarrow f_i^k$.

Thus, we can always choose $c_i^k = 1$ in (3), and MVCM’s converge.

However, the decision of (2) is not unique: different sets of generators of initial node values may lead to different final values (unlike metric spaces under certain conditions [18]) due to the fact that the lattice used is not linearly ordered. We may use a learning algorithm if such a situation is undesirable, and we know the required set of possible output values.
4. Learning Weight Values

We are based on ideas of [19] when constructing the required algorithm. However, we do not need two criterions to evaluate the final stage, due to the convergence of the multi-valued cognitive map. Also, the learning algorithm may be applied only to the final node values, not at every step (as in [19]); this is again due to the convergence. Finally, the algorithm is synchronous, unlike [19]. Hence, we do not need additional expert suppositions about the firing sequence.

Let us consider the following expressions:

\[ A^k_i = f^{k-1}_i \cdot \bigvee_{i=1}^{n} [(w^{k-1}_{ji} \oplus \Delta w^{k-1}_{ji}) \cdot (A^{k-1}_j \oplus \Delta A^{k-1}_j)] = \]
\[ = f^{k-1}_i \cdot \bigvee_{i=1}^{n} (w^{k-1}_{ji} \cdot A^{k-1}_j) \oplus \Delta A^k_i; \quad (16) \]

\[ \Delta A^k_i = f^{k-1}_i \cdot \bigvee_{i=1}^{n} (\Delta w^{k-1}_{ji} \cdot A^{k-1}_j) \oplus F(w^{k-2}, A^{k-2}, \Delta w^{k-2}, \Delta A^{k-2}). \quad (17) \]

Here, we have introduced a change in weights and concepts in order to pick up their values such that the output concepts \( A^k_i \) would find themselves in demanding output regions \{doc\}_j^i \]. Such output sets of lattice \( L \) elements should be established by experts for given initial values. The operation \( \oplus \) means the join or difference (see below) depending on what you need: increase \( A^k_i \) or decrease it. The term \( F(w^{k-2}, A^{k-2}, \Delta w^{k-2}, \Delta A^{k-2}) \) in (17) should not be considered, since, such a term is absent at the first step, and all \( A^{k-1}_j \)’s and \( w^{k-1}_{ji} \) are already calculated at the \( k \)’s step.

Let us consider the difference \( gen^k_{r_i} = doc_{r_i} - A^k_i \) of a concept with one of its desired output values. The result is the number of generators in the symmetric difference [3] of generator sets of \( r \)’s desired output concept for \( A^k_i \) and \( A^k_i \) at the \( k \)’s step. The number should be added to, or deducted from, the concept generator number in order for the concept to become equal to the desired value. Hence,

\[ \Delta A^k_i \leq gen^k_{r_i}; \quad (18) \]

\(^4\text{doc}_{ji} \text{ means } j \text{'s possible value of desired output concept set of } A^k_i \)
Thus, we obtain the following expression using (17):

\[ \Delta w_{ji}^{k-1} = (f_i^{k-1} \rightarrow \text{gen}_{ri}) \leftarrow A_j^{k-1}. \] (19)

Then, we should deduct \( \Delta w \) from \( w \) if \( A_i^k > \text{doc}_{ri} \): \( w_{ji}^k = w_{ji}^{k-1} \ominus \Delta w_{ji}^{k-1} \) (let us note that the set difference operation \( \ominus \) is used here, not the symmetric difference \( \ominus \), since we need to decrease exactly the number of generators). Otherwise, we take their join: \( w_{ji}^k = w_{ji}^{k-1} \lor \Delta w_{ji}^{k-1} \).

In the case of incomparable \( A_i^k \) and \( \text{doc}_{ri} \), the concept first increases under the algorithm work, up to it becoming greater than the desired value, and, after that, it decreases as is described above.

If we have negative weight values, the sequence is inverse: we join \( \Delta w \) and \( w \) if \( A_i^k > \text{doc}_{ri} \) and take the difference otherwise.

Such a comparison may be made with all elements of the desired output set, in order to choose the most suitable learned weight matrix. This algorithm describes the weight correction at every step. However, it is not necessary: we may check the condition of hitting the required region at the end of the process of firing the map, since the process converges. Thus, we may calculate (19) only once at an iteration cycle if the condition is not satisfied. Such a calculation can be more suitable (e.g., it changes the initial weights less) if some concepts get the desired values only at the end of the process. In this case, they should not be corrected with the others. Naturally, changing concepts at every step can give another result in this case (see Sec. 6).

5. Modelling Hybrid Energy Systems

We use the problem formulation of a modelling example from [20], without, however, a feedback to natural concepts. Indeed, it is hard to understand how the energy system functioning can influence sun insolation or wind. Thus, we consider the example of a Hybrid Energy System combining wind and photovoltaic subsystems; its cognitive map model is depicted in Fig. 1.

The model includes the following five concepts:

- C1: sun insolation;
- C2: environment temperature;
- C3: wind;

5. Modelling Hybrid Energy Systems

We use the problem formulation of a modelling example from [20], without, however, a feedback to natural concepts. Indeed, it is hard to understand how the energy system functioning can influence sun insolation or wind. Thus, we consider the example of a Hybrid Energy System combining wind and photovoltaic subsystems; its cognitive map model is depicted in Fig. 1.

The model includes the following five concepts:

- C1: sun insolation;
- C2: environment temperature;
- C3: wind;
• C4: PV-subsystem;

• C5: Wind-Turbine-subsystem.

In this model there are two energy source decision concepts (outputs), i.e., the two energy sources are considered: the C4: PV-subsystem and the C5: Wind-Turbine-subsystem. Concepts C1–C3 of nature and technical factors influence the subsystems and determine how each energy source will function in this model. The concept’s initial values can be obtained from experts’ assessments of measurements, which take values in a lattice of linguistic variables. The experts’ assessments of concept influence take values also in this lattice. Detailed information for hybrid renewable energy systems is given in [5], [21]. The case study from the literature was examined in [20], and we consider it here using a multi-valued scale for weights and values in the cognitive map (2) where all $c_i = 1$ due to Theorem 3.1.

We use the bi-lattice $L$ that is built from two lattices the $L_1$ and the $L_2$ (Fig. 3, 4) by direct multiplication $\times$, as the scale of experts’ assessments.

However, we consider the bi-lattice as the lattice where the unique partial order is generated by atoms $ba0, b, bn0, 0c$, and $0d$ Fig. 4. We use two linearly-ordered branches in the lattice $L_1$ in order to regularly obtain the distributive and atomic lattice $L$. Then, we may use the meet $\land$ as the monoid multiplication. In this case, both residuals are equal and coincide with the lattice implication $\Rightarrow$. The nodes $c$ and $d$ in the $L_1$, corresponding to the medium value, may be interpreted as “not high” and “not weak”
Figure 4: The b-c-h path in the uncertainty degree lattice $L = L_2 \times L_1$ which is used in the modelling example. The d-branch values are the same as in the c-branch, by replacing c to d in the names.
uncertainty degrees (this difference is not reflected in the interpretations in Fig. 4). There is not a universal method to build a monoid for a residuated lattice, and one should use some heuristics to determine the multiplication. Thus, we do not consider a general residual construction here and propose to investigate variants to do it in future.

It is possible to use the $L_1$ built from only one linearly-ordered branch. Then, the $L$ would be built by a quasi-direct multiplication to be an atomic lattice: the node 0h must be connected with 0 directly. However, we consider the lattice $L = L_2 \times L_1$ for more variety.

A variant of the lattice $L_2$ was proposed in [22] as the interpretation of N. A. Vasil’ev’s logic ideas. Vasil’ev has suggested three types of statement: positive, negative and indifferent, instead of only positive and negative. He considered also intermediate types as a hesitation between these main ones. Similarly, we consider here three main uncertainty degrees: $ba0$, $b$, $bn0$ (Fig. 3) — some assessment “$b$”, the assessment “$bn0$: Neither $b$ nor 0”, and the estimation “$ba0$: $b$ and 0” — and the same at the levels $c - d$ and $h$ Fig. 4. The estimation, e.g., “born: $b$ or $bn0$” is the join of $b$ and $bn0$ and can be considered as the hesitation between $b$ and “Neither $b$ nor 0”. Similarly, “bora: $b$ or $ba0$” is the join of $b$ and $ba0$ and can be considered as the hesitation between $b$ and “$b$ and 0” and so on. Thus, we obtain many different variants of uncertainty degrees in assessments, available to experts.

The connections between the concepts of the cognitive map are represented in Table 1 and the initial concept values are determined in Table 2. We tried to more or less match the data from [20]. In Case 1, all concept values are concentrated in one $L$ branch (b-c-h). In Cases 2 and 3, one initial concept value belongs to the b-d-h branch of the $L$. 

|     | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ |
|-----|-------|-------|-------|-------|-------|
| $C_1$ | Th born 0 hora b |
| $C_2$ | 0 0 0 Tb 0 |
| $C_3$ | 0 ca0c Th bora hn0h |
| $C_4$ | 0 0 0 0 0 |
| $C_5$ | 0 0 0 0 0 |
Thus, we calculate the map concepts’ values by (2) in the following form:

\[ C_k^i = f_k^{i-1} \land \bigvee_{j=1}^{5} (w_{ji} \land C_j^{k-1}) \] (20)

where coefficients are calculated by (11) in the following form:

\[ f_k^i = \left[ \bigvee_{j=1}^{5} (w_{ji} \land C_j^k) \right] \Rightarrow \left[ C_k^i \lor C_k^{i-1} \right]. \] (21)

The nodes are triggered simultaneously, and their values interact with ones to be updated through this process of interaction in the same iteration step. We take values \( f_i^0 = Th \). Hence, natural concepts the \( C_1 \) and the \( C_3 \) do not change unlike (1) (taken from [20]) where they are changed due to the sigmoid transfer function \( f \). However, such a choice of the \( f_i^0 \) is not necessary: these values may be chosen arbitrarily.

Here, the lattice of weights and values satisfies the conditions of Theorem 3.1 where the lattice top element \( Th \) coincides with the monoid neutral element, and the monoid multiplication coincides with the lattice meet. The calculation results are represented in Tables 3 – 4.

We see that the final state depends on initial values and on distribution of them over \( L_1 \)-branches. The result is also dependant on weight values, as even they are of the same degree of uncertainty. However, the final states are more or less equal in degrees of uncertainty in these cases (see also Sec. 6). Moreover, these final states are stable: if we take them as initial ones, we obtain them at the end. Thus, the model corresponds well to the idea of the stationary state of the system under constant insolation and wind.

Let us note one more time, that these natural factors are not changed by a transfer function in model iterations, unlike [20]. However, we can change them outwardly in order to compare results with the similar ones of [20].
Table 3: Iterations of concept values, Case 1

| k | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| C1 | horn | horn | horn | horn |
| C2 | c | corn | corn | corn |
| C3 | c | c | c | c |
| C4 | c | h | h | h |
| C5 | 0c | c | c | c |

Table 4: Iterations of concept values, Case 2 and Case 3

| k | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| C1 | horn | horn | horn | horn |
| C2 | c | born | born | born |
| C3 | d | d | d | d |
| C4 | h | h | h | h |
| C5 | caorn | d | d | d |

| k | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| C1 | horn | horn | horn | horn |
| C2 | b | born | born | born |
| C3 | d | d | d | d |
| C4 | h | h | h | h |
| C5 | c | d | d | d |
Table 5: Iterations of concept values, Case 3, $C_1 = \text{const}$ and the $C_3$ increases

| k | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| C1 | horn | horn | horn | horn | horn |
| C2 | b | born | born | born | born |
| C3 | d | h | hora | hora | hora |
| C4 | h | h | h | h | h |
| C5 | c | d | h | h | h |

Table 6: Iterations of concept values, Case 3, the $C_1$ decreases and the $C_3$ increases

| k | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| C1 | horn | born | dn0d | dn0d | dn0d | dn0d |
| C2 | b | born | born | r | r | r |
| C3 | d | h | hora | hora | hora | hora |
| C4 | h | h | d | d | d | d |
| C5 | c | d | h | 0h | 0h | 0h |

6. Discussion

6.1. Natural concepts’ changing

We consider only Case 3 here. Let insolation first be a constant — $C_1 = \text{const}$ — and wind $C_3$ increases: Table 5. We see the environment temperature $C_2$ has changed the uncertainty value but remains at the same $L_1$ level. The output of the photovoltaic subsystem $C_4$ as expected, has not changed, and the wind-turbine $C_5$ output has increased.

Let us decrease the insolation value else: Table 6. We see the environment temperature $C_2$ has changed the uncertainty value but less than in the previous case. The output of the photovoltaic subsystem $C_4$ has slightly decreased, and the wind-turbine $C_5$ output has increased its $L_1$ level but less than in the previous case. This result is intuitively clear; it corresponds with our weight definitions, and it is perhaps better than in [20] where the $C_5$ and the $C_2$ may be greater than here (with all the ambiguity in establishing the correspondence between numbers and lattice elements).

6.2. Weights with negative values

However, some negative number values are used in the weight matrix in [20]. The members in the sum (which define the map) with such weights
Table 7: Weights with negative elements in the cognitive map for Hybrid Energy Systems

|      | C₁ | C₂   | C₃   | C₄   | C₅   |
|------|----|------|------|------|------|
| C₁   | Th | born | 0    | hora | b    |
| C₂   | 0  | 0    | 0    | - Tb | 0    |
| C₃   | 0  | - ca0c| Th   | bora | hn0h |
| C₄   | 0  | 0    | 0    | 0    | 0    |
| C₅   | 0  | 0    | 0    | 0    | 0    |

Table 8: Iterations of concept values with weights negative elements, Case 3, \( C₁ = const \) and \( C₃ = const \)

| k   | 0  | 1  | 2  | 3  | 4  |
|-----|----|----|----|----|----|
| C₁  | horn | horn | horn | horn | horn |
| C₂  | b   | born | born | born | born |
| C₃  | d   | d   | d   | d   | d   |
| C₄  | h   | 0h  | 0h  | 0h  | 0h  |
| C₅  | c   | d   | d   | d   | d   |

deducted from the sum. We can also use a similar deduction with the help of (3). We mark such weight matrix members with the minus sign. Sets of generators of the lattice elements, including such weights in the join, will be deducted from the join by (3). Thus, we consider the following weight matrix corresponding to the similar one of (20) and Table 1: Table 7.

Let us consider again first that insolation is constant — \( C₁ = const \)— and the wind \( C₃ = const \) is constant too: Table 8. This is almost Table 4. Only, the \( C₄ \) has slightly decreased its uncertainty value.

Let us increase the wind value: Table 9. Again, this is almost Table 5.

Table 9: Iterations of concept values with the weights’ negative elements, Case 3, \( C₁ = const \) and the concept \( C₃ \) increases

| k   | 0  | 1  | 2  | 3  | 4  |
|-----|----|----|----|----|----|
| C₁  | horn | horn | horn | horn | horn |
| C₂  | b   | born | born | born | born |
| C₃  | d   | h   | hora | hora | hora |
| C₄  | h   | 0h  | 0h  | 0h  | 0h  |
| C₅  | c   | d   | h   | h   | h   |
Table 10: Iterations of concept values with weights’ negative elements, Case 3, the $C_1$ decreases and the $C_3$ increases

| k  | 0 | 1 | 2 | 3 | 4 | 5 |
|----|---|---|---|---|---|---|
| C1 | horn | dorn | dorn | dorn | dorn | dorn |
| C2 | b | born | born | r | r | r |
| C3 | d | h | hora | hora | hora | hora |
| C4 | h | 0h | 0d | 0d | 0d | 0d |
| C5 | c | d | h | 0h | 0h | 0h |

Table 11: Iterations of concept values with weights’ negative elements, Case 3, the $C_1$ decreases and the $C_3$ increases, initial $C_5 = d$

| k  | 0 | 1 | 2 | 3 | 4 | 5 |
|----|---|---|---|---|---|---|
| C1 | horn | dorn | dorn | dorn | dorn | dorn |
| C2 | b | born | born | r | r | r |
| C3 | d | h | hora | hora | hora | hora |
| C4 | h | 0h | 0d | 0d | 0d | 0d |
| C5 | d | d | d | 0d | 0d | 0d |

and the $C_4$ has the same decrease as in the previous case.

Let us decrease else the insolation value: Table 10. This is almost Table 6. Only, the $C_4$ has slightly decreased its uncertainty value. Thus, we see the results do not change qualitatively.

6.3. The dependence on initial values

However, what happens if we change, e.g., the wind initial value? Let it be in Case 3 (Table 2) $C_5 = d$ instead of $C_5 = c$. In the uncertainty sense, such a replacement changes almost nothing intuitively. The results are in Table 11. We see that wind-turbine output even decreases, though, the wind value increases. All other values are the same. Removing minus from the weight matrix elements only replaces the $C_4$ uncertainty value to the same in Table 6. Thus, initial values can influence the modelled system behaviour crucially in general.

The thing is that different sets of generators in initial values may really lead to different final values, since the lattice used is not linearly ordered, and all $C_i$’s lie inside the two initial ones: $\forall k: C_i^k \leq C_i^0 \lor C_i^1$. In our case, $\forall k C_5^k \leq c \lor d = h$ in Table 10 and $C_5^k \leq d$ in Table 11. $C_5^1 = d$ in both the cases,
it does not depend on $C_5^0$. Thus, different initial values may lead to different system stable states. However, such a dependence may be excluded by the learning process of weight elements, if we know the demanding output range (see Sec. 4). We consider these calculations in the following subsection.

Also, such effects of intuitive contradiction can be indirectly related to our lattice determination: we went from the bi-lattice $L_2 \times L_1$ to the lattice $L$ where the elements $0c$ and $0d$ are the same generators as $b, ba0$, and $bn0$. Though, in the lattice $L_1$, such values are more significant than the bottom level (the level of $b, ba0$, and $bn0$ in the $L$). Hence, our real partial order is generated by the number of generators of lattice elements, and it may be different from the intuitive interpretation of $L_2 \times L_1$ partial orders.

6.4. Learning

We use here the algorithm of Sec. 4 which trains weights so that the output concepts would be inside the desired lattice $L$ subsets. We use formula (19) in the following form in our case:

$$\Delta w_{ji}^{k-1} = A_j^{k-1} \Rightarrow (f_i^{k-1} \Rightarrow gen_{ri}),$$

(22)

since, both residuals become the lattice implication when the monoid multiplication is the meet.

We determine desired output concept sets for the $C_4$ and the $C_5$ as:

$$Doc_4 = \{0d, d, da0d, dn0d\}$$
$$Doc_5 = \{0h, h, ha0h, hn0h\}.$$ 

(23) 

(24)

First, we apply the learning algorithm at the end of firing the map (Case 3, initial $C_5 = d$). Then we obtain the weight matrix in Table 12 instead of Table 7.

In this case, we obtain the following iterations of concept values when the weight matrix is in Table 12 for initial $C_5 = c$ and $C_5 = d$, and the comparison is made with the first elements of $Doc_4, Doc_5$: Table 13. We see that final output concept values do not depend on the initial ones now.

If we apply learning at each iteration step, we will obtain different weights: Table 14. We see that matrix elements for $C_4$ calculation are also changed in this case. We have seen, though, that it is not needed in reality: the $C_4$ process converges in the desired region without learning. In Table 15 the output concept results are also obtained with the learning process. We see,
Table 12: Weight matrix obtained from the learning process of the $C_5$ at the end of iterations, Case 3, the $C_1$ decreases and the $C_3$ increases, initial $C_5 = d$.

|   | C1 | C2  | C3  | C4  | C5  |
|---|----|-----|-----|-----|-----|
| C1 | Th | born | 0   | hora | cora |
| C2 | 0  | 0   | 0   | -Tb | 0   |
| C3 | 0  | -ca0c | Th | bora | hno0h |
| C4 | 0  | 0   | 0   | 0   | 0   |
| C5 | 0  | 0   | 0   | 0   | 0   |

Table 13: Iterations of concept values with weights from Table 12. Case 3, the $C_1$ decreases and the $C_3$ increases, initial $C_5 = c$ and $C_5 = d$.

| k  | 0  | 1  | 2  | 3  | 4  | 5  |
|-----|-----|-----|-----|-----|-----|-----|
| C1  | horn | dorn | dn0d | dn0d | dn0d | dn0d |
| C2  | b  | born | born | r  | r  | r  |
| C3  | d  | h   | hora | hora | hora | hora |
| C4  | h  | 0h  | 0d  | 0d  | 0d  | 0d  |
| C5  | c  | h   | h   | 0h  | 0h  | 0h  |

| k  | 0  | 1  | 2  | 3  | 4  | 5  |
|-----|-----|-----|-----|-----|-----|-----|
| C1  | horn | dorn | dn0d | dn0d | dn0d | dn0d |
| C2  | b  | born | born | r  | r  | r  |
| C3  | d  | h   | hora | hora | hora | hora |
| C4  | h  | 0h  | 0d  | 0d  | 0d  | 0d  |
| C5  | d  | h   | h   | 0h  | 0h  | 0h  |
Table 14: Weight matrix obtained from the learning process of the \( C_5 \) at each step of the iterations, Case 3, the \( C_1 \) decreases and the \( C_3 \) increases, initial \( C_5 = d \)

|   | C1 | C2 | C3 | C4 | C5 |
|---|----|----|----|----|----|
| C1 | 0  | 0  | 0  | 0  | 0  |
| C2 | 0  | 0  | 0  | 0  | 0  |
| C3 | 0  | 0  | 0  | 0  | 0  |
| C4 | 0  | 0  | 0  | 0  | 0  |
| C5 | 0  | 0  | 0  | 0  | 0  |

Table 15: Iterations of concept values with weights from Table 14. Case 3, the \( C_1 \) decreases and the \( C_3 \) increases, initial \( C_5 = c \) and \( C_5 = d \).

| k  | 0 | 1 | 2 | 3 | 4 | 5 |
|----|---|---|---|---|---|---|
| C1 | horn | born | dorn | dorn | dorn | dorn | dorn |
| C2 | b | born | born | r | r | r |
| C3 | d | h | hora | hora | hora | hora | hora |
| C4 | h | 0h | 0d | 0d | 0d | 0d |
| C5 | c | d | h | 0h | 0h | 0h |

that the process leads to different final values of the \( C_5 \) for initial ones \( c \) and \( d \), though, both of them are inside the desired output set.

If we use the learned matrix of Table 14 in the map firing, we will obtain the output concept result for the initial \( C_5 = c \), which matches the same one for the initial \( C_5 = d \) and differs from the similar one in Table 15.

We see that the output concepts converge in the desired range in all these variants, and the output values do not depend on whether the initial one is \( c \) or \( d \) (if we calculate them with the learned weight matrix), unlike the previous Subsection.
Table 16: Iterations of concept values without learning with weights from Table 14. Case 3, the $C_1$ decreases and the $C_3$ increases, initial $C_5 = c$ and $C_5 = d$.

| k | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| C1 | horn | dorn | dn0d | dn0d | dn0d | dn0d |
| C2 | b | born | born | r | r | r |
| C3 | d | h | hora | hora | hora | hora |
| C4 | h | 0d | 0d | 0d | 0d | 0d |
| C5 | c | h | h | h | h | h |

| k | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| C1 | horn | dorn | dn0d | dn0d | dn0d | dn0d |
| C2 | b | born | born | r | r | r |
| C3 | d | h | hora | hora | hora | hora |
| C4 | h | 0d | 0d | 0d | 0d | 0d |
| C5 | d | h | h | h | h | h |

6.5. Runtime

Finally, all the calculations of implications in (21) were performed by the quick algorithm of [12] and the resulting timing is depicted in Fig. 5.

7. Conclusion

We have considered the concept of cognitive maps in which all weights and data take values in a partially-ordered set (exactly, in a lattice) of linguistic quantities. Thus, experts get a wider scale for their linguistic assessments than in a fuzzy case. Such maps converge under some limitations on the set of map variable values. We give also the algorithm to learn the map weight matrix in order to select values so that they are in the desired range of the lattice.

We give a detailed consideration of a modelling example versus using a fuzzy cognitive map. We obtain even more realistic results, since, in our approach, immutable or externally modifiable concepts do not change by the map recount. In ordinary fuzzy cognitive maps, such concepts are automatically changed by a transfer (trimming) function.

Thus, it seems the consideration of multi-valued cognitive maps in the paper demonstrates the self-consistency and correspondence to the reality.
of the approach, despite some ambiguity in interpretation of linguistic assessments. Moreover, the approach also provides more opportunities than in fuzzy maps for expert evaluations.

We have left for future investigations the problem of the comparison of different expert opinions. You need to introduce some conception of multi-valued numbers as the lattice subsets in order to do this. Also, we have left for the future the investigation of variants to define a universal residuated construction for the lattice used as a scale of linguistic values of map variables.

References

References

[1] B. Kosko, Fuzzy cognitive maps, Intern. Journal of Man-Machine Studies (24) (1986) 65–75.

[2] Fuzzy cognitive maps advances in theory, methodologies, tools and applications, in: M. Glykas (Ed.), Studies in Fuzziness and Soft Computing, Vol. 247, Springer, Berlin Heidelberg, 2010.

[3] M. Hagiwara, Extended fuzzy cognitive maps, in: Proceedings of IEEE Int. Conference on Fuzzy Systems, 1992, pp. 795–801.
[4] J. A. Dickerson, B. Kosko, Virtual worlds as fuzzy cognitive maps, Presence (3) (1994) 173–189.

[5] H. Gould, J. Tobochnik, D. C. Meredith, et al., An introduction to computer simulation methods: applications to physical systems, Computers in Physics 10 (4) (1996) 349.

[6] J. P. Craiger, D. F. Goodman, R. J. Weiss, A. Butler, Modeling organizational behavior with fuzzy cognitive maps, Intern. Journal of Computational Intelligence and Organisations (1) (1996) 120–123.

[7] J. Faulin, A. A. Juan, S. S. M. Alsina, J. E. R. Marquez, Simulation methods for reliability and availability of complex systems, Springer, Berlin-New-York, 2010.

[8] D. Y. Maximov, Y. S. Legovich, S. Ryvkin, How the structure of system problems influences system behavior, Automation and Remote Control 78 (4) (2017) 689–699.

[9] D. Maximov, S. Ryvkin, Systems smart effects as the consequence of the systems complexity, in: Proc. 17th International Conf. on Smart Technologies (IEEE EUROCON 2017, Ohrid), IEEE, Ohrid, 2017, pp. 576–582.

[10] D. Maximov, S. Ryvkin, Multi-valued logic in graph transformation theory and self-adaptive systems, Annals of Mathematics and Artificial Intelligence 87 (4) (2019) 395–408.

[11] D. Maximov, Control in a group of unmanned aerial vehicles based on multi-valued logic, in: Proceedings of the 12th International Conference 'Management of Large-Scale System Development' (MLSD’2019), IEEE, Providence, 2019, pp. 1–5. doi:10.1109/MLSD.2019.8911092
URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8911092

[12] D. Maximov, Multi-valued neural networks and their use in decision making on the management of a group of unmanned vehicles, in: Proceedings of the 2020 13th International Conference ‘Management of Large-Scale System Development’ (MLSD), IEEE, Providence, 2020, pp. 1–5. doi:10.1109/MLSD49919.2020.9247800
URL https://ieeexplore.ieee.org/document/9247800
[13] D. Maximov, V. I. Goncharenko, Y. S. Legovich, Multi-valued neural networks I: A multi-valued associative memory, Neural Computing and Applications (2021, accepted to publication).

[14] D. Maximov, Multi-valued neural networks II: A robot group control, Advances in system science and applications 20 (4) (2020) 70–82.

[15] G. Birkhoff, Lattice Theory, Providence, Rhode Island, 1967.

[16] K. Blount, C. Tsinakis, The structure of residuated lattices, Int. J. Algebra Comput. (13) (2003) 437–461.

[17] J. Gil-Férez, F. M. Lauridsen, G. Metcalfe, Integrally closed residuated lattices, Stud. Logica (2019) 1–24. URL https://doi.org/10.1007/s11225-019-09888-9

[18] T. L. Kottas, Y. S. Boutalis, M. A. Christodoulou, Fuzzy cognitive networks: Adaptive network estimation and control paradigms, in: M. Glykas (Ed.), Studies in Fuzziness and Soft Computing, Vol. 247, Springer, Berlin Heidelberg, 2010, pp. 89–134.

[19] E. I. Papageorgiou, C. D. Stylios, P. P. Groumpos, Active hebbian learning algorithm to train fuzzy cognitive maps, International Journal of Approximate Reasoning 37 (2004) 219–249. doi:10.1016/j.ijar.2004.01.001

[20] P. P. Groumpos, Why model complex dynamic systems using fuzzy cognitive maps?, Robotics and Automation Engineering Journal 1 (3) (2017) 1–13. doi:10.19080/RAEJ.2017.01.555563

[21] K. K. Damghani, M. T. Taghavifard, R. T. Moghaddam, Decision making under uncertain and risky situations, in: Enterprise Risk Management Symposium Monograph, Society of Actuaries, Schaumburg, Illinois, 2009.

[22] D. Maximov, N.A. Vasil’ev’s logic and the problem of future random events, Axiomathes 28 (2018) 201–217. doi:10.1007/s10516-017-9355-1