# Reasoning with Words: Steps Towards Applying in Mobile System

Nguyen Van Han[1,2](✉) and Phan Cong Vinh[1]

[1] Faculty of Information Technology, Nguyen Tat Thanh University, 300A Nguyen Tat Thanh Street, Ward 13, District 4, Ho Chi Minh, Vietnam
{nvhan,pcvinh}@ntt.edu.vn

[2] Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava, Czech Republic

**Abstract.** In this paper, we introduce two algorithms for reasoning with words on fuzzy dynamic system. The systems that use linguistic variables which are variables whose values may be expressed in terms of a specific natural or artificial language, for example $\mathbb{L} = \{$*very less true; less true; true; more true; very true; very very true* ...$\}$. In language of hedge algebra ($\mathbb{HA}$), $\mathbb{L}$ set which is generated from $\mathbb{HA}$ is the POSET (partial order set). The algorithms are Static reasoning and Dynamic reasoning. The former traverses the branch of the fuzzy graph whereas the later transform according to the equation of state and create a space of states of the system. Algorithms performed on linguistic variables and applied labeling techniques. And finally, the application of the algorithm on the mobile network model is also investigated.

**Keywords:** Linguistic variable · Fuzzy system · Mobile system

## 1 Introduction

Fuzzy engineering has been studied and applied in artificial intelligence. Fuzzy set and fuzzy logic or "computing with words" (CWW) were introduced by Lotfi A. Zadeh in 1965 as an extension of the classical notion of set [12,18] and was just a tool to knowledge represent and reasoning in intelligent system [13]. As Zadeh indicates [13], human acknowledgment is nothing different from words. In daily activity, we see the real world through words. Many smart devices that established based on CWW such as fuzzy cognitive map, mobile payment system, fuzzy data mining, ... and so on have been studied [1–3,14–16] The rest of the paper is organized as follows: Section 2 recalls some of the main foundation concepts of fuzzy cognitive map and linguistic cognitive map. Section 3 proposes two algorithms for reasoning with words and applies the algorithms for Mobile Payment System Project. Section 4 summaries outlines the corollary as well as future work.


P. Cong Vinh and H. Mahfooz Ul Haque (Eds.): ICTCC 2023, LNICST 586, pp. 133–147, 2024.
https://doi.org/10.1007/978-3-031-59462-5_10

## 2 Preliminary

This section reviews the basic knowledge related to the article that is Fuzzy cognitive map ($\mathbb{FCM}$) and Linguistic cognitive map ($\mathbb{LCM}$).

### 2.1 Reasoning on Fuzzy Cognitive Map

An integration model between fuzzy logic and neural networks is fuzzy cognitive map ($\mathbb{FCM}$) [11] in which, the inference process is mainly on the domain [0, 1].

*Example 1.* The fuzzy knowledge base is described in Table 1 which is formalized into the $\mathbb{FCM}$ graph as shown in Fig. 1.

Graph $\mathbb{FCM}$s operate as a neuro-fuzzy system ($\mathbb{NFS}$) with static and dynamic reasoning.

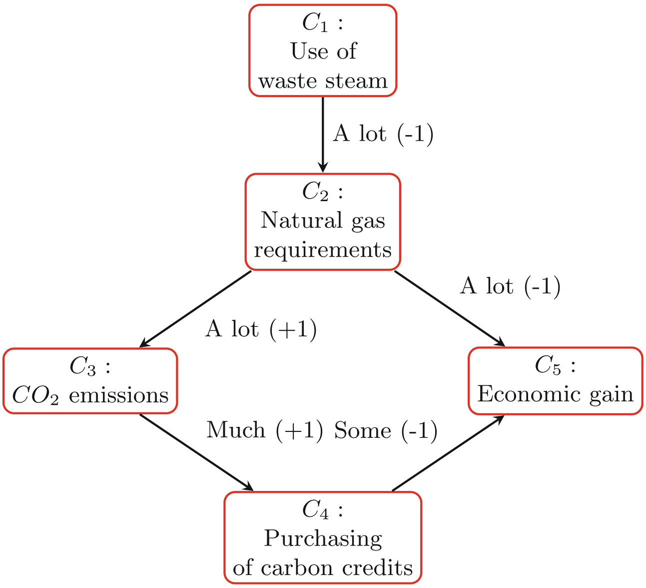


**Fig. 1.** FCM graph

1. *Static reasoning*: This method calculates the branches of the graph $\mathbb{FCM}$.

**Table 1.** Knowledge Base

| | | | |
|---|---|---|---|
| IF | Use of waste steam | THEN | Natural gas requirements |
| IF | Natural gas requirements | THEN | $CO_2$ emissions |
| IF | Natural gas requirements | THEN | Economic gain |
| IF | $CO_2$ emissions | THEN | Purchasing of carbon credits |
| IF | Economic gain | THEN | Purchasing of carbon credits |

*Example 2.* Suppose there is a set of language values $\mathbb{L} = \{a\ lot,\ some,\ none,\ much\}$ with order $\{none < some < much < a\ lot\}$. The Fig. 1 shows that there are 2 paths between vertices $C_1$ and $C_5$, they are $P_1 = (1, 2, 5)$ and $P_2 = (1, 2, 3, 4, 5)$.

$$
\begin{aligned}
P_1(C_1, C_5) &= \mathscr{Min}\{e_{12}, e_{25}\} \qquad = \mathscr{Min}\{a\ lot, a\ lot\} \\
&= a\ lot \\
P_2(C_1, C_5) &= \mathscr{Min}\{e_{12}, e_{23}, e_{3,4}, e_{4,5}\} \qquad = \mathscr{Min}\{a\ lot, a\ lot, much, some\} \\
&= some
\end{aligned}
$$

Total path ways :

$$
\begin{aligned}
T(C_1, C_5) &= \mathscr{Max}\{P_1(C_1, C_5), P_2(C_1, C_5)\} \qquad = \mathscr{Max}\{a\ lot, some\} \\
&= a\ lot
\end{aligned}
$$

2. *Dynamic reasoning*: This method considers the transformation of the state vector according to the Eq. 1 [17]

$$
[C_1 C_2 \ldots C_n]_{new} = [C_1 C_2 \ldots C_n]_{old} \otimes \begin{pmatrix} e_{11} & \ldots & e_{1,n} \\ \vdots & \ddots & \vdots \\ e_{n,1} & \ldots & e_{n,n} \end{pmatrix}. \tag{1}
$$

*Example 3.* The adjacency matrix of the graph Fig. 1 is:

$$
M = (e_{ij})_{5\times5} = \begin{pmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 & -1 \\ 0 & 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.
$$

Let $\mathfrak{C}^0 = [C_1 C_2 C_3 C_4 C_5] = [1\ 0\ 0\ 0\ 0]$ [17] be the initial term. Equation 2 is the recursive equation.

$$
\mathfrak{C}^t = \mathfrak{C}^{t-1} \vee \mathfrak{C}^{t-1} \wedge M \tag{2}
$$

With t=1, 2, 3, 4, we have:

$$\mathfrak{C}^0 = [1\ 0\ 0\ 0\ 0]$$

$$\mathfrak{C}^1 = [1\ 0\ 0\ 0\ 0] \vee [1\ 0\ 0\ 0\ 0] \wedge \begin{pmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 & -1 \\ 0 & 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = [1\ \ -1\ 0\ 0\ 0]$$

$$\mathfrak{C}^2 = [1\ \ -1\ 0\ 0\ 0] \vee [1\ \ -1\ 0\ 0\ 0] \wedge \begin{pmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 & -1 \\ 0 & 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = [1\ \ -1\ \ -1\ 0\ 1]$$

$$\mathfrak{C}^3 = [1\ \ -1\ \ -1\ 0\ 1] \vee [1\ \ -1\ \ -1\ 0\ 1] \wedge \begin{pmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 & -1 \\ 0 & 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = [1\ \ -1\ \ -1\ \ -1\ 1]$$

$$\mathfrak{C}^4 = [1\ \ -1\ \ -1\ \ -1\ 1] \vee [1\ \ -1\ \ -1\ \ -1\ 1] \wedge \begin{pmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 & -1 \\ 0 & 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = [1\ \ -1\ \ -1\ \ -1\ 1]$$

After four interactions using Eq. 2, the system converges to a fixed point since $\mathfrak{C}^4$'s state is a repeat of $\mathfrak{C}^3$'s state; that is, $\mathfrak{C}^4 = \mathfrak{C}^3 = [1\ \ -1\ \ -1\ \ -1\ 1]$. This indicates a hidden pattern.

Associating both static and dynamic reasoning results in our final $\mathbb{NFS}$ structure [17].

*Example 4.* Combining $T(C_1,\ C_5)$= **a lot** from Example 2 and $C_5 \in \mathfrak{C}^4 = C_5 \in \mathfrak{C}^3 = 1$ from Example 3 to infer, then:

**IF** the use of waste steam increases **THEN** economic gain increases **alot**.

The $\mathbb{FCM}$'s structure in Fig. 1 is simple because only three values from the set $\{-1,\ 0,\ 1\}$ were used. This article expands to a set of linguistic values for a more natural representation.

### 2.2 Linguistic Fuzzy Cognitive Maps

Paper stands on $\mathbb{LCM}$ which is extended from fuzzy cognitive map $\mathbb{FCM}$ [11]. The $\mathbb{LCM}$ model, based on linguistic variables, is constructed from linguistic hedge of $\mathbb{HA}$ in [4–7].

**Definition 1.** *A linguistic cognitive map ($\mathbb{LCM}$) is a 4- Tuple:*

$$\mathbb{LCM} = \{C, E, \mathbb{C}, f\} \tag{3}$$

*In which:*

1. $C = \{C_1, C_2, \ldots, C_n\}$ *is the set of concepts to form the nodes of a graph.*
2. $E : (C_i, C_j) \longrightarrow e_{ij} \in \mathbb{L}$; $e_{ij}$ = *directed edges' weight from* $C_i$ *to* $C_j$*. The weighted matrix* $E(N \times N) = \{e_{ij}\}_{N\times N} \in \mathbb{L}^{N\times N}$
3. *The map:* $\mathbb{C} : C_i \longrightarrow C_i(t) \in \mathbb{L}, t \in N$
4. $\mathbb{C}(0) = [C_1(0), C_2(0), \ldots, C_n(0)] \in \mathbb{L}^N$ *is the initial vector, recurring transformation function* $f$ *is defined as:*

$$C_j(t+1) = f(\sum_{i=1}^{N} e_{ij} C(t) \in \mathbb{L} \tag{4}$$

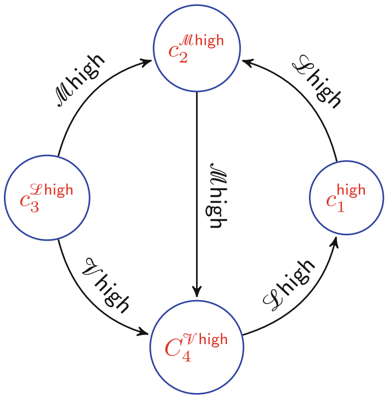


**Fig. 2.** A simple $\mathbb{LCM}$

*Example 5.* Figure 2 shows a simple $\mathbb{LCM}$. Let

$$\mathbb{HA} = <\mathscr{X} = SPEED; c^+ = \mathsf{high}; \mathscr{H} = \{\mathscr{L}, \mathscr{M}, \mathscr{V}\}> \tag{5}$$

be an $\mathbb{HA}$ with order as $\mathscr{L} < \mathscr{M} < \mathscr{V}$ ( $\mathscr{L}$ for less, $\mathscr{M}$ for more and $\mathscr{V}$ for very are hedges ). $C = \{c_1, c_2, c_3, c_4\}$ is the set of 4 concepts with corresponding values $\mathscr{C} = \{\mathsf{high}, \mathscr{M}\mathsf{high}, \mathscr{L}\mathsf{high}, \mathscr{V}\mathsf{high}\}$

## 3 Algorithms

This section presents two reasoning algorithms on the $\mathbb{LCM}$ graph, which are branch-based and state-based reasoning.

### 3.1 Static Reasoning

Branch reasoning algorithm, Algorithm 1, find the weight of the path between any two vertices of the graph $\mathbb{LCM}$.

*Example 6.* The steps to implement Algorithm 1 are detailed step by step as shown in the Fig. 3 - Fig. 6

---

**Algorithm 1.** Branched Inference Algorithm

---

**Input: Graph** $\mathbb{LCM} = (V, E)$
▷ $\mathbb{LCM}$ has vetices: s=$C_1, \ldots, C_n = d$ and edges' weight $e(v_i, v_j) \in \mathbb{L}$
**Ra:** $\mathscr{L}(d)$
▷ $\mathscr{L}(d) \in \mathbb{L}$ is the length of the path from $s$ to $d$
1: **foreach** $C_i \in V$ **do** ▷ initialize labels for vertices
2: $\quad\mathscr{L}(C_i) \leftarrow 0$ ▷ Labels of vertices are assigned by zero $(0 = \mathscr{Min}\{\mathbb{L}\})$
3: **end foreach**
4: $\mathscr{L}(s) = 1$ ▷ The label of the source vertex is assigned 1 $(1 = \mathscr{Max}\{\mathbb{L}\})$
5: $\mathscr{Q} = \emptyset$ ▷ Initially, queue $\mathscr{Q}$ is empty
6: **while** $d \notin \mathscr{Q}$ **do**
7: $\quad u \leftarrow$ vertex in $V - \mathscr{Q}$ and has the largest label $\mathscr{L}(u)$
8: $\quad\mathscr{Q} = \mathscr{Q} \cup \{u\}$ ▷ Put in $\mathscr{Q}$ vertice with the largest label
9: $\quad$**foreach** $v \notin \mathscr{Q}$ **do**
10: $\quad\quad$**if** $\mathscr{L}(u) \wedge e(u, v) \geq \mathscr{L}(v)$ **then**
11: $\quad\quad\quad\mathscr{L}(v) \leftarrow \mathscr{L}(u) \wedge e(u, v)$ ▷ update the label for vertex $v$
12: $\quad\quad$**end if**
13: $\quad$**end foreach**
14: **end while**
15: **return** $\mathscr{L}(d)$ ▷ $\mathscr{L}(d)$ is the length of the path from $s$ to $d$

---

*Property 1.* Let $|V|$ be the size of the vertex set *fuzzy concept*, the complexity of Algorithm 1 is $\mathscr{O}(|V|)^2$.

*Proof.* The complexity of the Algorithm 1 depends on the number of $\leftarrow$ assignment operations in the iteration statements as follows:

| Lines | Maximum number of assignment operations |
|---|---|
| Line 1 to 3 | $\sum_{i=1}^{\lvert V\rvert} 1 = \lvert V\rvert$ |
| Line 6 to 14 | $\sum_{i=1}^{\lvert V\rvert}\sum_{i=1}^{\lvert V\rvert} 1 = (\lvert V\rvert)^2$ |

The total number of assignments of the loops is $(|V|)^2 + (|V|)$ so the complexity of the Algorithm 1 is $\mathscr{O}(|V|)^2$

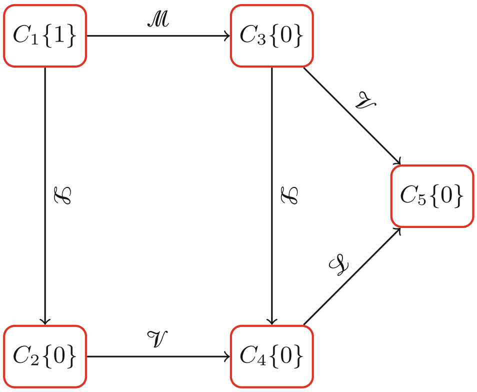


**Fig. 3.** Initialize label $s = C_1 = 1$ other vertices are labeled 0, set $\mathscr{Q}$ is empty

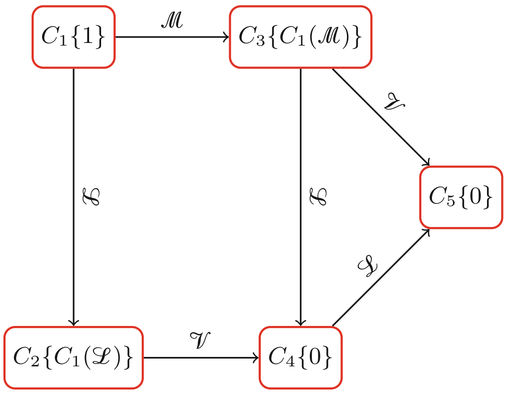


**Fig. 4.** Put $C_1 = 1$ into $\mathscr{Q}$, $\mathscr{Q} = \{C_1\}$, update labels for $C_2$ and $C_3$

### 3.2 Dynamic Reasoning

Different from branch reasoning algorithm, state reasoning algorithm, Algorithm 2, find the convergence vector of the system according to the state equation.

*Example 7.* Consider a graph $\mathbb{LCM}$ with 5 vertices and an adjacency matrix M:

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \mathscr{V}\mathscr{V}\,\mathsf{high} & 0 & \mathscr{V}\mathscr{V}\,\mathsf{low} & 0 & 0 \\ \mathscr{V}\mathscr{V}\,\mathsf{high} & \mathscr{V}\mathscr{V}\,\mathsf{high} & 0 & 0 & 0 \\ 0 & \mathscr{L}\,\mathsf{low} & \mathscr{L}\,\mathsf{low} & 0 & \mathscr{V}\,\mathsf{low} \\ 0 & \mathsf{low} & \mathsf{low} & 0 & 0 \end{bmatrix}$$

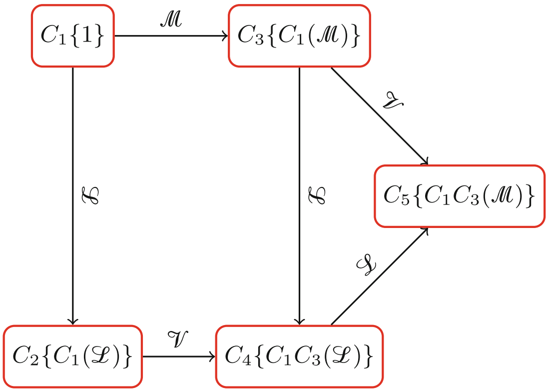


**Fig. 5.** Insert $C_3 = \mathscr{M}$ into $\mathscr{Q}$, $\mathscr{Q} = \{C_1, C_3\}$, update labels for $C_4$ and $C_5$

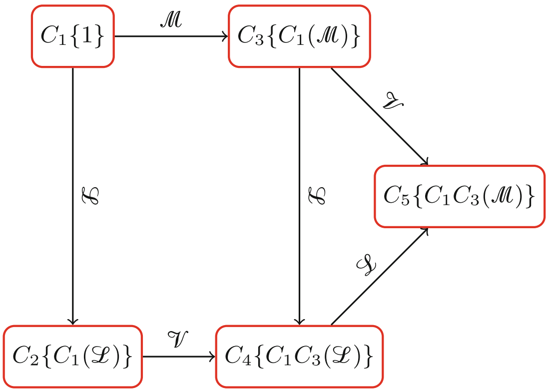


**Fig. 6.** Put $C_5 = \mathscr{M}$ into $\mathscr{Q}$, $\mathscr{Q} = \{C_1, C_3, C_5\}$, end the Algorithm 1

With an initial value of $\mathbb{C}(0) = \{\mathscr{M}\mathsf{high}, \mathscr{L}\mathscr{M}\mathsf{high}, \mathscr{L}\mathsf{high}, \mathscr{M}\mathscr{V}\mathsf{high}, \mathscr{L}\mathscr{M}\mathsf{high}\}$. Applying the Algorithm 2, the steps will show as follows:

$$
\begin{aligned}
\mathbb{C}(1) &= [C_1(1)..C_j(1)..C_5(1)] \quad \text{with } C_j(1) = \bigvee_{i=1}^{5} C_i(0) \wedge e_{ij} \\
&= \bigvee\Big( [\mathscr{M}\mathsf{high}, \mathscr{L}\mathscr{M}\mathsf{high}, \mathscr{L}\mathsf{high}, \mathscr{M}\mathscr{V}\mathsf{high}, \mathscr{L}\mathscr{M}\mathsf{high}] \quad \wedge \\
&\qquad \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \mathscr{V}\mathscr{V}\mathsf{high} & 0 & \mathscr{V}\mathscr{V}\mathsf{low} & 0 & 0 \\ \mathscr{V}\mathscr{V}\mathsf{high} & \mathscr{V}\mathscr{V}\mathsf{high} & 0 & 0 & 0 \\ 0 & \mathscr{L}\mathsf{low} & \mathscr{L}\mathsf{low} & 0 & \mathscr{V}\mathsf{low} \\ 0 & \mathsf{low} & \mathsf{low} & 0 & 0 \end{bmatrix} \\
&\Big) \\
&= [\mathscr{L}\mathscr{M}\mathsf{high}, \mathscr{L}\mathsf{high}, \mathscr{L}\mathsf{low}, 0, \mathscr{V}\mathsf{low}]
\end{aligned}
$$

**Algorithm 2.** State dynamic reasoning

```
Input: Initialization vector ℂ(0), matrix E
Output: Fixed point vector ℂ(fix)
for i ← 1 to n do                          ▷ Initialize values for fuzzy concept
    C_i ← C_i(0)
end for
for i ← 1 to n do                          ▷ Initialize values for edges
    for j ← 1 to n do
        (C_i, C_j) ← e_{i,j}
    end for
end for
𝒯ime ← 0
while ℂ(𝒯ime + 1) ≠ ℂ(𝒯ime) do
    for j ← 1 to n do
        ℳax ← 0                            ▷ Calculate ℳax = ⋁_{i=1}^{n} C_i(𝒯ime) ∧ e_ij
        for i ← 1 to n do
            ℳax ← ℳax ∨ C_i(𝒯ime) ∧ e_ij
        end for
                                           ▷ Calculate C_j(𝒯ime + 1) = ⋁_{i=1}^{n} C_i(𝒯ime) ∧ e_ij
        C_j(𝒯ime + 1) ← ℳax
    end for
    𝒯ime ← 𝒯ime + 1
end while
ℂ(fix) ← ℂ(𝒯ime)
return ℂ(fix)
```

$$
\begin{aligned}
\mathbb{C}(2) = &\ [C_1(2)..C_j(2)..C_5(2)] \quad \text{with } C_j(2) = \bigvee_{i=1}^{5} C_i(1) \wedge e_{ij} \\
= &\ \bigvee\Big( [\mathscr{L}\mathscr{M}\mathsf{high}, \mathscr{L}\mathsf{high}, \mathscr{L}\mathsf{low}, 0, \mathscr{V}\mathsf{low}] \wedge
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
\mathscr{V}\mathscr{V}\mathsf{high} & 0 & \mathscr{V}\mathscr{V}\mathsf{low} & 0 & 0 \\
\mathscr{V}\mathscr{V}\mathsf{high} & \mathscr{V}\mathscr{V}\mathsf{high} & 0 & 0 & 0 \\
0 & \mathscr{L}\mathsf{low} & \mathscr{L}\mathsf{low} & 0 & \mathscr{V}\mathsf{low} \\
0 & \mathsf{low} & \mathsf{low} & 0 & 0
\end{bmatrix} \Big) \\
= &\ [\mathscr{L}\mathsf{high}, \mathscr{L}\mathsf{low}, \mathscr{V}\mathsf{low}, 0, 0]
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{C}(3) = &\ [C_1(3)..C_j(3)..C_5(3)] \quad \text{with } C_j(3) = \bigvee_{i=1}^{5} C_i(2) \wedge e_{ij} \\
= &\ \bigvee\Big( [\mathscr{L}\mathsf{high}, \mathscr{L}\mathsf{low}, \mathscr{V}\mathsf{low}, 0, 0] \wedge
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
\mathscr{V}\mathscr{V}\mathsf{high} & 0 & \mathscr{V}\mathscr{V}\mathsf{low} & 0 & 0 \\
\mathscr{V}\mathscr{V}\mathsf{high} & \mathscr{V}\mathscr{V}\mathsf{high} & 0 & 0 & 0 \\
0 & \mathscr{L}\mathsf{low} & \mathscr{L}\mathsf{low} & 0 & \mathscr{V}\mathsf{low} \\
0 & \mathsf{low} & \mathsf{low} & 0 & 0
\end{bmatrix} \Big) \\
= &\ [\mathscr{L}\mathsf{low}, \mathscr{V}\mathsf{low}, \mathscr{V}\mathscr{V}\mathsf{low}, 0, 0]
\end{aligned}
$$

$$\begin{aligned}
\mathbb{C}(4) &= [C_1(4)..C_j(4)..C_5(4)] \quad \text{with } C_j(4) = \bigvee_{i=1}^{5} C_i(3) \wedge e_{ij} \\
&= \bigvee( \ [\mathscr{L}\mathsf{low}, \mathscr{V}\mathsf{low}, \mathscr{V}\mathscr{V}\mathsf{low}, 0, 0] \wedge \\
&\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \mathscr{V}\mathscr{V}\mathsf{high} & 0 & \mathscr{V}\mathscr{V}\mathsf{low} & 0 & 0 \\ \mathscr{V}\mathscr{V}\mathsf{high} & \mathscr{V}\mathscr{V}\mathsf{high} & 0 & 0 & 0 \\ 0 & \mathscr{L}\mathsf{low} & \mathscr{L}\mathsf{low} & 0 & \mathscr{V}\mathsf{low} \\ 0 & \mathsf{low} & \mathsf{low} & 0 & 0 \end{bmatrix} \\
&) \\
&= [\mathscr{L}\mathsf{low}, \mathscr{V}\mathscr{V}\mathsf{low}, \mathscr{V}\mathscr{V}\mathsf{low}, 0, 0]
\end{aligned}$$

$$\begin{aligned}
\mathbb{C}(5) &= [C_1(5)..C_j(5)..C_5(5)] \quad \text{with } C_j(5) = \bigvee_{i=1}^{5} C_i(4) \wedge e_{ij} \\
&= \bigvee( \ [\mathscr{L}\mathsf{low}, \mathscr{V}\mathscr{V}\mathsf{low}, \mathscr{V}\mathscr{V}\mathsf{low}, 0, 0] \wedge \\
&\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \mathscr{V}\mathscr{V}\mathsf{high} & 0 & \mathscr{V}\mathscr{V}\mathsf{low} & 0 & 0 \\ \mathscr{V}\mathscr{V}\mathsf{high} & \mathscr{V}\mathscr{V}\mathsf{high} & 0 & 0 & 0 \\ 0 & \mathscr{L}\mathsf{low} & \mathscr{L}\mathsf{low} & 0 & \mathscr{V}\mathsf{low} \\ 0 & \mathsf{low} & \mathsf{low} & 0 & 0 \end{bmatrix} \\
&) \\
&= [\mathscr{V}\mathscr{V}\mathsf{low}, \mathscr{V}\mathscr{V}\mathsf{low}, \mathscr{V}\mathscr{V}\mathsf{low}, 0, 0]
\end{aligned}$$

$$\begin{aligned}
\mathbb{C}(6) &= [C_1(6)..C_j(6)..C_5(6)] \quad \text{with } C_j(6) = \bigvee_{i=1}^{5} C_i(5) \wedge e_{ij} \\
&= \bigvee( \ [\mathscr{V}\mathscr{V}\mathsf{low}, \mathscr{V}\mathscr{V}\mathsf{low}, \mathscr{V}\mathscr{V}\mathsf{low}, 0, 0] \wedge \\
&\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \mathscr{V}\mathscr{V}\mathsf{high} & 0 & \mathscr{V}\mathscr{V}\mathsf{low} & 0 & 0 \\ \mathscr{V}\mathscr{V}\mathsf{high} & \mathscr{V}\mathscr{V}\mathsf{high} & 0 & 0 & 0 \\ 0 & \mathscr{L}\mathsf{low} & \mathscr{L}\mathsf{low} & 0 & \mathscr{V}\mathsf{low} \\ 0 & \mathsf{low} & \mathsf{low} & 0 & 0 \end{bmatrix} \\
&) \\
&= [\mathscr{V}\mathscr{V}\mathsf{low}, \mathscr{V}\mathscr{V}\mathsf{low}, \mathscr{V}\mathscr{V}\mathsf{low}, 0, 0]
\end{aligned}$$

Since $\mathbb{C}(6) = \mathbb{C}(5) = [\mathscr{V}\mathscr{V}\mathsf{low}, \mathscr{V}\mathscr{V}\mathsf{low}, \mathscr{V}\mathscr{V}\mathsf{low}, 0, 0]$ the system reaches a fixed point after 6 iteration.

*Property 2.* For a graph $\mathbb{LCM} = (V, E)$ with vertex size $|V|$, the complexity of the algorithm 2 is $\mathscr{O}(|V|)^3$

*Proof.* The complexity of the algorithm depends on the number of $\leftarrow$ assignments in the iteration instructions and is detailed as follows:

| Line | Maximum number of assignment operations |
|---|---|
| Line 1 to 3 | $\sum_{i=1}^{n} 1 = n$ |
| Line 4 to 8 | $\sum_{i=1}^{n} \sum_{j=1}^{n} 1 = n^2$ |
| Line 9 to 19 | $\sum_{\mathscr{T}ime} \sum_{i=1}^{n} \sum_{j=1}^{n} 1 \leq n^3$ |

Since the variable $\mathscr{T}ime \leq n$ the total number of assignment operations $\leftarrow$ in the loops is at most: $n+n^2+n^3$, with $n = |V|$ so the complexity of the algorithm 2 is $\mathscr{O}(n^3) = \mathscr{O}(|V|)^3$

*Property 3.* At any time $t$, the system will reach a fixed point $\mathbb{C}(fix)$ when the vector $\mathbb{C}(t+1)$ depends recursively on the vector $\mathbb{C}(t)$ in the fields the following case:

$$\mathbb{C}(t+1) \geq \mathbb{C}(t) \tag{6}$$
$$\mathbb{C}(t+1) \leq \mathbb{C}(t) \tag{7}$$

*Proof.* Use mathematical induction to prove
Case 1: $\mathbb{C}(t+1) \geq \mathbb{C}(t)$

$$\text{Base step: } With\ t = 0, \text{we have } \mathbb{C}(1) \geq \mathbb{C}(0), so$$
$$\begin{aligned} \mathbb{C}(2) &= \bigvee_{i=0}^{N} C_i(1) \wedge e_{ij} \\ &\geq \bigvee_{i=0}^{N} C_i(0) \wedge e_{ij} \\ &= \mathbb{C}(1) \end{aligned}$$
$$\text{Induction step: Assume (6) is true for } t = k$$
$$\text{Or } : C_j(k) \geq C_j(k-1)$$
$$\begin{aligned} \text{Then } C_j(k+1) &= \bigvee_{i=0}^{N} C_i(k) \wedge e_{ij} \\ &\geq \bigvee_{i=0}^{N} C_i(k-1) \wedge e_{ij} \\ &= C_j(k), \text{therefore} \\ C_J(k+1) \geq C_j(k) \end{aligned}$$

$C_j(k)$ is finite, monotonically increasing, and bounded on so $C_j(k)$ converges.
Case 2: $\mathbb{C}(t+1) \leq \mathbb{C}(t)$ The proof is similar to case 1.

### Mobile Payment System

Mobile Payment System (MPS) is modeled to $\mathbb{FCM}$ [14,16]. Accordingly, the calculation must convert from numbers to words and vice versa. This process increases computational complexity. To reduce the number of mathematical operations in the calculation, the article proposes model of MPS in the form of $\mathbb{LCM}$ by applying domain transformations in the Table 2.

The Figure Fig. 7 indicates a $\mathbb{LCM} = (V, E)$ graph representing the MPS using linguistic variables. In which, $V = \{C_1, \ldots, C_{24}\}$ is the set of fuzzy concepts. The weight of edge $(C_i, C_j)$ is $e_{ij}$ which represents the causal relationship between two adjacent vertices $C_i$ and $C_j$ . By Definition 1, the weight matrix of the $\mathbb{LCM}$ graph in Fig. 7 has the form:

$$E = \begin{cases} e_{ij} \text{ if } (C_i, C_j) \in E \\ 0 \text{ if } (C_i, C_j) \notin E \end{cases} \tag{8}$$

*Example 8.* Apply Table 2, weight of edge $(C_3, C_1)$ in $\mathbb{FCM}$ is $e_{31} = 0.68$ [16] will be changed to linguistic value in $\mathbb{LCM}$ is $e_{31} = \mathscr{LM}\mathsf{high}$.

**Table 2.** Domains conversion

| Range $[-1,\ 1]$ | Positive range $[0,\ 1]$ | Domain of $\mathbb{L}$ | Meaning |
|---|---|---|---|
| $[-1,\ -0.7)$ | $[0,\ 0.15)$ | $\mathscr{VV}\mathsf{low}$ | very very low |
| $[-0.7,\ -0.4)$ | $[0.15,\ 0.3)$ | $\mathscr{LM}\mathsf{low}$ | less more low |
| $[-0.4,\ -0.1)$ | $[0.3,\ 0.45)$ | $\mathscr{LL}\mathsf{low}$ | less less low |
| $[-0.1,\ 0.1)$ | $[0.45,\ 0.55)$ | $\mathscr{W}$ | Neutral |
| $[0.1,\ 0.4)$ | $[0.55,\ 0.7)$ | $\mathscr{VL}\mathsf{high}$ | very less high |
| $[0.4,\ 0.7)$ | $[0.7,\ 0.85)$ | $\mathscr{LM}\mathsf{high}$ | less more high |
| $[0.7,\ 1]$ | $[0.85,\ 1]$ | $\mathscr{VV}\mathsf{high}$ | more more high |

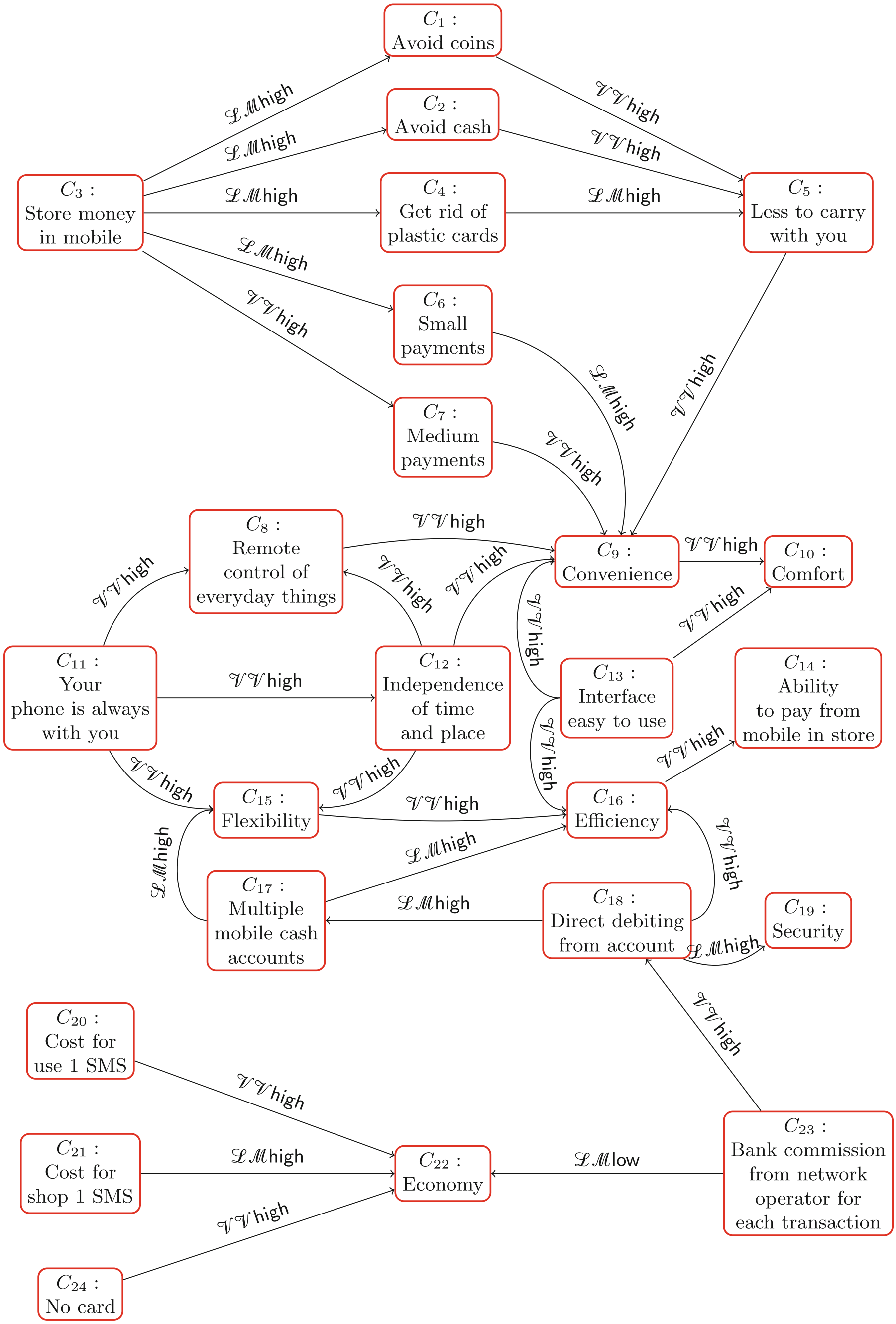


**Fig. 7.** $\mathbb{LCM}$ model for MPS Project

## 4 Conclusion and Forthcoming Study

The paper recommends two algorithms for reasoning with words on linguistic cognitive map.

– Static algorithm allows to find the path value between any two vertices of the graph $\mathbb{LCM}$ on the linguistic value domain
– Dynamic algorithm allows to find the convergence vector in the state space over the linguistic domain
– Research and apply algorithms on Mobile Payment System

In the future, two studies will be:

– Research on modeling and reasoning methods on smart mobile systems based on lingistic variables.
– Prove the correctness and completeness of the algorithms.

## References

1. Caarvalho, J.: On the semantics and the use of fuzzy cognitive maps and dynamic cognitive maps in social sciences. Fuzzy Sets Syst. **214**, 6–19 (2013)
2. Frias, M., Yaima, F., Nápoles, G., Vahoof, K., Bello, R.: Fuzzy cognitive maps reasoning with words: an ordinal approach. In: ISFUROS (2017)
3. Glykas, M.: Fuzzy Cognitive Maps, Advances in Theory, Tools and Applications. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-03220-2
4. Van Han, N., Hao, N.C., Vinh, P.C.: Toward aggregating fuzzy graphs a model theory approach. In: Vinh, P.C., Rakib, A. (eds.) ICCASA/ICTCC -2019. LNICST, vol. 298, pp. 215–222. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34365-1_17
5. Van Han, N., Cong Vinh, P.: Modeling with words based on hedge algebra. In: 7th EAI International Conference, ICCASA 2018 and 4th EAI International Conference, ICTCC 2018, pp. 266:211–217 (2018)
6. Van Han, N., Cong Vinh, P.: Toward modeling and reasoning with words based on hedge algebra. EAI Endorsed Trans. Context-aware Syst. Appl. **5**(15), e5 (2018)
7. Van Han, N., Cong Vinh, P.: Reasoning with words: a hedge algebra linguistic cognitive map approach. Concurrency Comput. Pract. Exp. **33**(2), e5711 (2020)
8. Nguyen Cat Ho and Nguyen Van Long: Fuzziness measure on complete hedge algebras and quantifying semantics of terms in linear hedge algebras. Fuzzy Sets Syst. **158**(4), 452–471 (2007)
9. Cat Ho, N., Thai Son, T., Dinh Khang, T., Xuan Viet, L.: Fuzziness measure, quantified semantic mapping and interpolative method of approximate reasoning in medical expert systems. J. Comp. Sci. Cybern. **18**(3), 237–252 (2002)
10. Cat Ho, N., Wechler, W.: Hedge algebras: an algebraic approach to structure of sets of linguistic truth values. Fuzzy Sets Syst. **35**(3), 281–293 (1990)
11. Kosko, B.: Fuzzy cognitive maps. Int. J. Man Mach. Stud. **24**, 65–75 (1986)
12. Zadeh, L.A.: The concept of a linguistic variable and its applications to approximate reasoning. Inf. Sci. **8**(3), 199–249 (1975)
13. Zadeh, L.A.: Computing with words - Principal Concepts and Ideas. Studies in Fuzziness and Soft Computing. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27473-2

14. Frias, M., Filiberto, Y., Nápoles, G., Falcon, R., Bello, R., Vanhoof, K.: Comparative analysis of symbolic reasoning models for fuzzy cognitive maps. In: Bello, R., Falcon, R., Verdegay, J.L. (eds.) Uncertainty Management with Fuzzy and Rough Sets. SFSC, vol. 377, pp. 127–139. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-10463-4_7
15. Papageorgiou, E.I. (ed.): Fuzzy Cognitive Maps for Applied Sciences and Engineering. ISRL, vol. 54. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-39739-4
16. Rodriguez-Repiso, L., Setchi, R., Salmeron, J.L.: Modelling it projects success with fuzzy cognitive maps. Expert Syst. Appl. **32**(2), 543–559 (2007)
17. Ross, T.J.: Fuzzy Logic With Engineering Applications. 3rd edn. Wiley (2010)
18. Zadeh, L.A., Kacprzyk, J.: Computing with Word in Information Intelligent System 1. Springer, Heidelberg (1999). https://doi.org/10.1007/978-3-7908-1873-4