Identifying Relevant Cross-Layer Interactions in Cognitive Processes

Christian Facchini and Fabrizio Granelli
Dept. of Information Eng. and Comp. Science (DISI)
University of Trento
Via Sommarive 14, I-38123 Trento, Italy
Email: facchini.granelli@disi.unitn.it

Nelson L. S. da Fonseca
Institute of Computing
State University of Campinas
P.O. Box 6176 – 13.084-971, Campinas, SP, Brazil
Email: nfONSECA@IC.unicamp.br

Abstract—Cognitive networks were recently proposed to cope with the complexity and the dynamics of network management, exploiting reasoning to adapt the behavior of protocols. Among the reasoning formalisms that can be employed, Fuzzy Cognitive Maps seem to be very promising, as they potentially allow the cognitive process to consider cross-layer interactions in the characterization of the performance of a network node. However, when considering a high number of cross-layer interactions, reasoning schemes can be too time consuming and may not provide a suitable solution as environmental conditions change. In order to decrease the demand of reasoning time it is of utmost importance to discover which cross-layer relationships carry relevant information to the cognitive process. This paper discusses how to make such differentiation. Moreover, it proposes a metric to evaluate the influence a cross-layer interaction has on the cognitive process.

Index Terms—cognitive networks, cross-layer interactions, fuzzy cognitive maps

I. INTRODUCTION

Recent advances in communications led to significant improvements in network performance at the price of increased management complexity. In order to cope with such complexity, new networking paradigms have been proposed. In this scenario, “cognitive networking” attempts to embed networks with intelligence to facilitate monitoring, reasoning, and acting towards the achievement of performance goals [1].

Reasoning is undoubtedly the most critical step in network cognitive process since it needs to balance the potential lack of precision of information obtained from sensing relevant activities; moreover, it determines the actions to be taken in the acting step. As a consequence, reasoning has received significant attention in the literature. Although there is a general consensus that reasoning should exploit information from all layers of the protocol stack of a network node [2]–[4], it seems there is no general agreement on the reasoning technique to be employed for that. Neural networks, Bayesian networks, expert systems, multidimensional optimization algorithms, and techniques from the fields of control theory and pattern recognition have been employed for this goal [1], [2]. Moreover, quite often, the techniques are chosen empirically, without proper justification [5]. In addition to that, although it has been acknowledged that reasoning schemes must converge to a solution before the environment significantly changes [1], to the best of the authors’ knowledge, apparently no work focuses on reasoning time.

Since in cognitive processes based on Fuzzy Cognitive Maps (FCMs) it is feasible to use information on cross-layer interactions for reasoning [6], the procedures introduced here are targeted to identify the interactions that are most relevant for the cognitive process. In particular, relevant cross-layer relations can be identified by their causal strength and variability; weak relations and variable causality can be ignored, reducing reasoning time without affecting the effectiveness of the cognitive process. This paper introduces novel methods to classify cross-layer interactions and identify those relevant to reasoning in cognitive processes based on Cognitive Maps. Such identification is the very first step towards reducing reasoning time and consequently improving the performance of cognitive networks.

The remaining sections are organized as follows. Section II discusses how FCMs can be applied to cognitive networks, underlining benefits and drawbacks. Section III delves into the importance of distinguishing relevant from irrelevant cross-layer interactions in FCM-based reasoning. Section IV proposes a procedure for such differentiation and Section V provides an example. Finally, Section VI draws some conclusions.

II. MOTIVATION

Despite the existing consensus in the literature about the need for the holistic consideration of network node operation, examples provided so far seem to pay little attention to the issue. Indeed, some cognitive network architecture proposals stress the importance of involving all layers of the protocol stack to observe requirements and constraints. However, they do not offer implementations of the target scheme [2], [3] or they propose implementations on a limited subset of the available layers [4]. Although it is deemed that cognitive network architectures will make intense use of cross-layering [1] and that cross-layer interactions impact the performance of network nodes [7], so far no attempt has been made to account for the impact of cross-layer interactions on the performance of network nodes [2]. Even more important, to the best of
the authors’ knowledge, no research work has explicitly taken
into consideration cross-layer dependencies in the reasoning
process, which is fundamental to fully enable a truly holistic
approach. Only recently, the authors proposed a reasoning
mechanism based on fuzzy cognitive maps (FCMs) which is
capable of reasoning on cross-layer interactions [6].

Introduced in the ’80s [8] as a tool for causal reasoning,
FCMs can be used to model dynamic systems, since they
emphasize the cause-effect relationships among internal variables.
Mathematically, FCMs can be represented by directed labeled
graphs (and, thus, by adjacency matrices). Nodes in such graph
symbolize causal objects, i.e. general concepts that can entail
or be entailed by other concepts. Node domains are usually
discrete and the most common set is the binary one \{0, 1\}: zero
means a concept is not considered, while one that it is
considered active. The vector of all node values univocally
define the system state.

The labels of the edges express the degree of causality
between two concepts and they can take any value in an
interval of real numbers. The domains of the labels are usually
zero-centered, so that causality can be positive, negative or
non-existent (in which case there is no edge). Since the graph
is directed, the direction of an edge discriminates the cause
(where the edge starts) from the effect (where the edge ends).
Throughout the remainder of the paper, the term “FCM” is
used to refer to both the formalism and the graph/matrix.

It can be objected that other formalisms could be used
in cognitive network nodes; however, FCMs offer some advan-
tages over the other potential reasoning techniques available,
such as neural networks and Markov networks among others.

Differently from the edges in a neural network, those in an
FCM reflect the actual relationships among the variables of the
problem being inspected. Hence, they are particularly suited
to explore the cross-layer relationships affecting the internal
variables of a network node. This represents a clear advantage
of FCMs over neural networks.

In FCMs, the process of finding the effects entailed by
causes, commonly referred to as deductive reasoning (or
simply inference), is computationally inexpensive, which is
not the case with Bayesian and Markov networks. In FCMs,
deductive reasoning is performed by multiplying the system
state by the FCM matrix and by comparing the result of the
multiplication with pre-defined thresholds. This process can
be repeated until it converges to a solution.

Although FCMs are abstracted by directed graph as are
Bayesian networks, they are more powerful, in that they can
deal with causality loops: the same inference mechanism can
be employed regardless of the presence of loops.

Furthermore, multiple FCMs can be exchanged and merged,
mimicking the exchange of opinions peculiar to human beings.
Like human beings, who can develop different opinions about
a single subject, FCMs from different cognitive entities can
be contrasted, and the combination of multiple FCMs can
smooth out inconsistencies, reducing the risk of biased rea-
soning. Interestingly, augmented FCMs can be composed of
non-overlapping FCMs, which means that nodes can merge
their knowledge even if the domains of their knowledge do
not perfectly overlap. This is an advantage when dealing
with uncertain knowledge. Such characteristics make FCMs
preferable in these contexts than other formalisms, such as,
first-order logic.

FCMs also have some drawbacks; the most limiting of them
concerns abductive reasoning. Unlike deductive reasoning,
abductive reasoning, i.e. the process of discovering which
causes lead to a given effect, is an NP-hard problem [9] (the
same complexity occurs when Bayesian networks are used).
Despite of such disadvantage, their advantages make them
particularly interesting as reasoning mechanism in cognitive
network nodes.

III. FUZZY COGNITIVE MAPS IN COGNITIVE PROCESSES

The capability of FCMs of representing systems through the
relationships of their internal mechanisms matches adequately
with the representation of cross-layer interactions in nodes of
cognitive networks.

A preliminary description of the application of FCMs to
cognitive networks was introduced by the authors in [6]. It was
shown that the state of a node is composed of three vectors
of concepts: the vector $q$, related to requirements of quality
specific to the communications which the node is involved
with; the vector $e$, comprising environment-related concepts,
and $a$, the vector of actions a node can take. The fundamental
problem is, thus, to find a particular value of $a, a^*$, so that
the constraints given by $q$ are met, before the environmental
conditions specified by $e$ change.

Solving this kind of problem implies the use of abductive
reasoning; therefore we have to resort to exhaustive search to
find an exact solution. The problem can be somewhat mitigated
by the fact that the search space can be reduced. Since only
$a$ can be modified by a node, and assuming, without loss of
generality, that the node domain is the binary set $\{0, 1\}$, the
search space has $2^{\dim(a)}$ elements, which makes exhaustive
search faster. As a consequence, the inference procedures
converge (either to a fixed point or to a limit cycle) within
$l_c$ steps, where $l$ is the cardinality of the discrete domain onto
which concepts are mapped, and $c$ is the number of concepts
[10]. Clearly, it is beneficial to keep both $l$ and $c$ as small as
possible.

However, reducing the number of levels can be difficult once
the fuzzy cognitive map has been designed. Also, as it is often
the case, many variables are naturally mapped to the smallest
possible domain (the binary domain $\{0, 1\}$), which cannot be
further reduced. As a consequence, it is important to lower the
number of concepts needed for the computation as much as
possible, taking care to avoid impinging the performance of
the reasoning process.

A concept cannot be removed from the computation without
first confirming that all the edges linking it to other concepts
do not impact the cognitive process. The problem, then, is
to find a metric to identify which edges can be discarded,
i.e. which cross-layer interactions are not relevant for the
reasoning process.
IV. IDENTIFICATION OF NON-RELEVANT CROSS-LAYER RELATIONS

This section discusses the relevance of cross-layer relations for the reasoning process and how to identify those which are relevant. First, we present two standalone approaches. An exhaustive search approach is described in IV-A, while a lightweight, yet also approximate, approach is described in IV-B. Our proposal combines the advantages of the two methods presented and is described in IV-C.

A. An Exhaustive Approach: Logic Circuit Minimization

Let us suppose a system state has \( c \) distinct concepts. Thus, the matrix of the associated FCM is a \( c \times c \) matrix. Without loss of generality, let us also suppose that the FCM is a matrix with \( n \) non-zero elements, \( n \leq c^2 - c \) (the elements on the diagonal are set to zero by definition). Then, reasoning will be based at most on \( n \) elements. However, not all the elements may be relevant to the reasoning process.

Combinations of the \( n \) elements result in \( 2^n - 1 \) non-trivial combinations. We can think of each combination as a binary string of length \( n \), in which each element is a logic variable that is set to 1 when it is considered by the reasoning process, or to 0, otherwise. Each combination will lead to a particular performance level, that can be classified by a binary variable as well: the value 1 means that the performance achieved is satisfactory, while the value 0 expresses the opposite. In other words, the reasoning process can be seen as a binary function \( b : \{0, 1\}^n \rightarrow \{0, 1\} \).

As a consequence, a Karnaugh map can be drawn, and, by synthesizing the part of the map where the performance level is deemed acceptable, a formula can be derived, indicating which FCM elements should be taken into account by the cognitive process and which should be left out.

The formula can be composed by two kinds of terms: (i) non-negated logic variables, and (ii) negated logic variables. Both kinds of terms are important to the reasoning process, in that both have some effect on the inference result. However, they are important in different ways. FCM elements that appear as non-negated variables should be considered when running the inference method because they lead to meaningful results, whereas FCM elements that appear as negated variables should be avoided, because their use can mislead the reasoning process and lead to wrong predictions.

All the terms that are not present in the formula should be considered unimportant, i.e., taking them into account does not change the overall result. Hence, only the elements appearing in the final formula are really important to be carried out by the reasoning task, while all the others can be safely ignored. Each element represents a cross-layer relation and is an edge of the FCM. Consequently, deleting some elements can lead to the isolation of some concepts; since these concepts do not have strong causal relationships with other concepts, they can be deleted.

However, we can reduce the number of concepts even further. The final formula will likely contain both OR and AND operations. If we refer to any set of terms connected only by the AND operator as an “AND-expression”, then we can say that expanding the formula will lead to several AND-expressions connected to one another by the OR operator. Each of these AND-expressions is equivalent to any other, since they lead to acceptable performance. Therefore, only the AND-expressions comprising the fewest non-negated elements could, in principle, be considered, and, among them, those involving the smallest number of concepts could be selected as the final formula.

Nevertheless, care should be taken when deciding whether adopting the full equation or the minimum AND-expression. One could be tempted to choose the second option over the first, because such choice is less demanding from the point of view of the computation but should lead to the same performance.

However, it should be noted that the full formula lists all the combinations that lead to acceptable performance levels. As a consequence, bad performance can only be caused by the lack of a concept in the matrix, i.e., concepts that should have been considered in the design phase, but were not. Instead, the minimum AND-expression comprises a subset of all possible combinations and the reasoning process can achieve the same performance if the terms appearing in that AND-expression remain relevant all the time. Thus, poor performance can be experienced, not only in the case concepts have been neglected, but also in cases where some considered terms are no longer relevant.

It is, thus, reasonable to expect that the minimum AND-expression is less reliable. If, on one hand, there is a gain on the complexity of reasoning, on the other hand overhead should be added to perform more frequent recalculations to ensure reliability.

B. A Lightweight Approach: The Discriminating Index

As mentioned in Section II, edge labels in FCMs measure the strength of the cause-effect relation between two variables. As a consequence, the higher the absolute value of an edge label, the greater the cause-effect relationship. Therefore, the modulus of the edge labels, or magnitude, can be used for discriminating the edges that are relevant to the reasoning process from those which are not.

As the values of the edge labels fluctuate, an average value needs to be computed. The modulus can be taken either before or after the averaging operation. In practice, none of the two orders is robust. Let us consider two extreme situations (exemplified in Fig. 1): a high-frequency and a low-frequency polar unitary square waves, symbolizing rapid (in the first case) and slow (in the second case) variations of causality. In a context where the values of the edge labels are slowly changing (Fig. 1b), an edge label that rapidly oscillates between positive and negative values (Fig. 1a) can be a sign of an unreliable causal connection, which can mislead the reasoning process. As a consequence, it would be beneficial to distinguish the two situations.

Averaging before taking the modulus yields a null values for both the causal relations in Fig. 1, and it does not allow
to discriminate between them. Taking the modulus before averaging yields one. Clearly, also in this case, no distinction between the two relations is possible.

This suggests selecting the oscillation between positive and negative values as a second discriminative feature. More specifically, the speed of oscillations can be approximated by the frequency of zero-crossings, i.e., the number of times an edge label value crosses the horizontal axis (null value). This way, causal connections that are characterized by a high variability can then be discarded in favor of others.

However, as the edge labels are set to zero at the outset, it is likely that they experience a transient phase, during which some of them can oscillate around the axis, thus scoring a high number of zero-crossings. Considering the absolute number of zero-crossings would result in the exclusion of such edges from the reasoning process from the beginning. For this reason, it is advisable to normalize the number of zero-crossings of the edge label values with respect to the maximum number of zero-crossings registered.

In summary, the proposal is to build a discriminating metric based on two values, namely, the absolute magnitude and the number of zero-crossings of the edge label values. However, since the aim of this study is to show that it is possible to recognize which cross-layer relations influence reasoning, and not to show which is the best classifier, we have chosen to simply combine these values by means of multiplication, and use the modulus before averaging yields one. Clearly, also in this case, no distinction between the two relations is possible.

The choice of multiplication as the feature extraction operation can be explained by looking at the final equation describing the discriminating index $d_{f_{ij}}(t)$ for a generic FCM element $f_{ij}$ at time $t$:

$$d_{f_{ij}}(t) = \text{avg}_{0 \rightarrow t} |f_{ij}| \cdot \left(1 - \frac{\text{zc}(f_{ij})}{\max_{f_{ij}} \{\text{zc}(f_{ij})\}}\right) \tag{1}$$

where $\text{zc}$ is a function returning the number of zero-crossings of a sequence. As we are interested only in the elements that are characterized by high magnitude and low variability, the use of multiplication allows us to quickly discard all the other combinations (high magnitude and high variability, low magnitude and high variability, low magnitude and low variability). Indeed, the resulting index will be low in all these cases. Of course, more advanced classifiers may be preferred.

Once the index is computed, a classifier needs to be employed, in order to discern the elements that are fundamental for the reasoning process from those that are not. As there is no a priori knowledge about the probability density of the data points, it is advisable to employ a non-parametric classification technique. We have chosen to utilize the Parzen window algorithm [11] to estimate the probability density function fitting the data, based on which we can distinguish the cluster of relevant elements from the cluster of non-relevant elements. The cluster associated to the greatest discriminating index comprises the most important variables. All the other clusters, if any, contain non-relevant variables.

As the network evolves, it may be the case that some FCM elements lose their importance whereas others can have their relevance increased. It is therefore clear that the discriminating index must be frequently updated during the lifetime of the cognitive process.

### C. A Hybrid Method

Both methods illustrated in Sections IV-A and IV-B can identify which cross-layer relationships are relevant to the reasoning process.

The main advantage of the logic circuit minimization approach is that it provides a closed-form solution to the problem. The formula is particularly helpful, since it explicitly specifies which elements reasoning should be based on and which elements can be neglected. The drawback is that, to synthesize the formula, the performance of all possible cases must be recorded, which is both time and resource consuming.

The discriminating index approach identifies the FCM important elements for the reasoning process, too. Although it finds which elements influence the reasoning process, it cannot discriminate positive from negative influences. In other words, the discriminating index alone helps to distinguish the strength of the influence an element has, but not its kind.

This may represent an issue only when negated terms appear in the Boolean equation synthesizing the Karnaugh map. More specifically, if, after expanding the formula, each AND-expression contains at least one negated term, then the use of the discriminating index will inevitably lead to poor performance. However, if at least one AND-expression is exclusively composed of non-negated terms, then the two approaches are equivalent, in the sense that they allow the reasoning process to reach the same solution.

To maintain the positive aspects of both schemes, we propose to merge them, to derive a joint method: the discriminating index is employed to preliminarily reduce the dimensionality of the problem, by discarding irrelevant FCM elements, while the logic circuit minimization is applied to definitively discern the positively influencing elements from the others. The resulting pseudocode is listed as Algorithm 1.

Figure 1 shows extreme-case time evolutions of a cause-effect relation.
Algorithm 1 “Hybrid” approach

1: \texttt{maxzc ← 1}
2: \texttt{FCM}_R ← \texttt{FCM}
3: \texttt{// FCM\_R is the matrix containing the elements important for reasoning}
4: \texttt{if zeroCrossing then}
5: \texttt{update maxzc}
6: \texttt{rel\_v.clear()}
7: \texttt{// rel\_v is the vector containing the relevant elements}
8: \texttt{for i, j = 1 \in \{1, 2, \ldots, c\}, i \neq j do}
9: \texttt{if Classifier(DiscreteIndex(FCM\_[i,j])) == relevant then}
10: \texttt{rel\_v.push(FCM\_[i,j])}
11: \texttt{// the classifier output can be either relevant or not}
12: \texttt{end if}
13: \texttt{end for}
14: \texttt{\hat{n} ← dim(rel\_v)}
15: \texttt{for rel\_v[1] ∈ \{0, 1\} do}
16: \texttt{for rel\_v[2] ∈ \{0, 1\} do}
17: \texttt{for ... do}
18: \texttt{Kmap(rel\_v[1], rel\_v[2], \ldots, rel\_v[\hat{n}]) ← RunFCM(rel\_v[1], rel\_v[2], \ldots, rel\_v[\hat{n}])}
19: \texttt{// Kmap contains the performance for every}
20: \texttt{// combination of the relevant elements}
21: \texttt{end for}
22: \texttt{end for}
23: \texttt{end for}
24: \texttt{end for}
25: \texttt{end for}
26: \texttt{for all FCM\_R[i,j] \\&\& MinimizationAlgorithm(Kmap) do}
27: \texttt{FCM\_R[i,j] ← 0}
28: \texttt{// suppose the MinimizationAlgorithm}
29: \texttt{\hat{n} \leftarrow \text{returns just the list of non-negated variables}}
30: \texttt{end for}
31: \texttt{end if}

V. TEST CASE

The scenario set up to validate the methodology proposed in this paper involves a wireless ad-hoc multihop network based on the IEEE 802.11b standard, which was simulated using the ns-3 simulator. Nodes are positioned in chain topology, as depicted in Fig. 2. In such configuration, the node at one end (S) starts an FTP transfer towards the node at the other end of the chain (R), lasting 100 seconds. To simulate the FTP protocol, we resorted to the use of the ns-3 on/off application, specifying a duty cycle of 100%, and a data rate of 10 Mb/s. TCP Reno was used as the transport protocol with segment size of 1460 bytes. Each node can communicate only with its immediate neighbors and routing is static.

Five input parameters have been selected: number of nodes, bit error rate, physical data rate, the use of RTS/CTS, and

TABLE I: Domains of the input variables

<table>
<thead>
<tr>
<th>Property</th>
<th>Abbr.</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>(n)</td>
<td>({3, \ldots, 9})</td>
</tr>
<tr>
<td>PHY bit error rate</td>
<td>(e)</td>
<td>(0.3 \cdot 10^{-8}, 3 \cdot 10^{-7}, \ldots, 3 \cdot 10^{-5})</td>
</tr>
<tr>
<td>PHY data rate</td>
<td>(d)</td>
<td>({1, 2, 5.5, 11} \text{ Mb/s})</td>
</tr>
<tr>
<td>MAC fragmentation</td>
<td>(f)</td>
<td>{on, off}</td>
</tr>
<tr>
<td>MAC RTS/CTS handshake</td>
<td>(r)</td>
<td>{on, off}</td>
</tr>
</tbody>
</table>

Fig. 2: Test case network topology

Fig. 3: Test case Fuzzy Cognitive Map. \(t, n, e, d, r, \) and \(f\) stand for throughput, number of nodes, error rate, physical data rate, RTS/CTS handshake, and fragmentation, respectively.

Fig. 4: Karnaugh map of the test case (reduced, due to space limitations)

TABLE II: Cross-layer relations characterizing the test case analyzed

<table>
<thead>
<tr>
<th>Cause concept</th>
<th>Effect concept</th>
<th>Abbr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error rate</td>
<td>Throughput</td>
<td>et</td>
</tr>
<tr>
<td>Physical data rate</td>
<td>Throughput</td>
<td>dt</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>Throughput</td>
<td>nt</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>Error rate</td>
<td>ne</td>
</tr>
<tr>
<td>RTS/CTS handshake</td>
<td>Throughput</td>
<td>rt</td>
</tr>
<tr>
<td>Fragmentation</td>
<td>Throughput</td>
<td>ft</td>
</tr>
<tr>
<td>Error rate</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE III: Performance achieved using different subsets of elements to perform the reasoning. Student’s \(t\) distribution with 29 d.o.f., \(p = 0.95\).

<table>
<thead>
<tr>
<th>Elements considered</th>
<th>Rel. impr. [%]</th>
<th>Conf. int.</th>
</tr>
</thead>
<tbody>
<tr>
<td>All 9 elements considered</td>
<td>+4.67</td>
<td>0.29</td>
</tr>
<tr>
<td>(dt + ne \cdot et \cdot \overline{de})</td>
<td>+5.04</td>
<td>0.31</td>
</tr>
<tr>
<td>(dt \cdot nt)</td>
<td>+5.04</td>
<td>0.31</td>
</tr>
<tr>
<td>(dt \cdot ne \cdot et \cdot \overline{de})</td>
<td>+4.34</td>
<td>0.45</td>
</tr>
<tr>
<td>(dt \cdot ne \cdot et \cdot \overline{de})</td>
<td>−5.27</td>
<td>0.48</td>
</tr>
<tr>
<td>(nt + ne \cdot et \cdot \overline{de})</td>
<td>−5.69</td>
<td>0.03</td>
</tr>
</tbody>
</table>
are not worth to be considered or all of them are important and all should be comprised by the reasoning process.

Each combination of the inputs has been simulated 15 times and the results averaged, in order to eliminate potential spurious values. The cognitive engine was used to predict the throughput level, given the inputs.

As can be inferred from the FCM reported in Fig. 3, at the outset there were 9 cross-layer relations that could potentially be used by the reasoning engine. For the sake of clarity, they are listed in Table II, where, besides indicating the name of the two concepts linked by the cross-layer relations, we have introduced an abbreviation, by joining the letters of the concepts involved. For example, according to this notation, \( dt \) denotes the relationship between data rate and throughput.

By applying the procedure used for minimization of logic circuits (described in Section IV-A) we managed to find the combination of cross-layer relationships upon which reasoning should be based. The final formula, that can be derived by means of the Karnaugh map shown in Fig. 4, is:

\[
b = dt \left( nt + et \cdot de \cdot ne \right) \quad (2)
\]

Thus, only 5 out of 9 cross-layer relations can influence the reasoning process. Besides that, one of them, i.e. the relationship between data rate and errors, appears negated, which means it should not be considered by the inference mechanism. It could be objected that, although it is well known that fragmenting packets decreases the error rate, the formula suggests not to consider segmentation. However, it should be noted that this does not mean that fragmentation does not reduce the error rate. Rather, it means that the causal relationship between segmentation and error rate is not meaningful to the reasoning process. In particular, the relation between them is weaker or oscillates more than the relations among the other variables.

Table III shows some of the results that can be achieved by the reasoning process, as a function of the elements considered: as long as the terms of either one of the AND-expressions are included in the reasoning process, performance is positive. When this condition fails, the reasoning process is bound to yield bad results. As illustrated in Fig. 5, the discriminating index approach described in Section IV-B managed to capture the relevance of the terms appearing in Equation 2. As expected, \( de \) appears among the relevant terms, and nothing points out it should not be considered during reasoning.

VI. CONCLUSION

The reasoning process in cognitive networks must converge to a solution before the operating environment changes. Therefore, it is crucial to maintain reasoning times as low as possible, while conserving high reliability of the results.

According to the reasoning formalism presented in [6], to achieve the ultimate objective of reducing reasoning times, it is mandatory to first reduce the number of relationships considered, without affecting the effectiveness of the reasoning process. In this direction, we proposed a method to identify the cross-layer relations that are not relevant to the reasoning process, i.e. those relations that do not exert any influence on the reasoning process. The method is composed of two steps. In the first step it attempts to reduce the dimensionality of the problem, keeping only the relations characterized by a high causal significance and low variability. In the second step it aims at discovering the relevant cross-layer relations, among those that were not discarded during the first phase, by means of exhaustive search.

The proposed scheme is validated through simulations, to underline it represents a possible solution for reducing reasoning time. Future work will deal with the analysis of the complete cognition loop (including action).

REFERENCES