Evaluation of Concrete Deterioration through Artificial Neural Networks based Systems

Guida Gomes, Henrique Vicente, and José Neves

Abstract—The deterioration of concrete structures is one of the major concerns of our society. Indeed, concrete is a relatively sensitive material, which degrades throughout time. Factors like age, use, periodic maintenance, type of environmental exposure and aggression by biological, chemical, mechanical and physical agents are important to determine the level of degradation of the concrete structures. Logic Programming was used for knowledge representation and reasoning, letting the modeling of the universe of discourse in terms of defective data, information and knowledge. Artificial Neural Networks were used in order to evaluate the deterioration of concrete structures and the degree of confidence that one has on such a happening.

Keywords—Artificial Neuronal Networks, Concrete Degradation, Knowledge Representation and Reasoning, Logic Programming.

I. INTRODUCTION

Despite its ancient origin, concrete is considered a modern material used in the majority of today’s constructions. Concrete is a composite material formed by coarse granular material (the aggregate or filler) embedded in hard matrix of material (the cement or binder) that fills the space among aggregate particles and glues them together [1]. Concrete exhibits high compressive strength but a low tensile one. To avoid this weakness concrete is usually reinforced with materials like steel, which in turn gives rise to other problems like corrosion. Concrete has a very low coefficient of thermal expansion and shrinks as it matures. All concrete structures will crack to some extent, due to shrinkage and tension [2]. Another fundamental limitation of concrete is that it is very sensitive to the conditions in which it is mixed and applied. There are a large number of variables that affect the concrete quality. The lack of attention given to these variables makes concrete more vulnerable and it has been the reason why the service lifespan of many contemporary concrete structures has been unexpectedly short [3]. Concrete is a relatively sensitive material that degrades throughout time, i.e. even if it is well made, sooner or later the defects, which define the deterioration, will appear. For this reason, concrete structures suffer a natural aging caused by the environment (e.g. rain, sun, pollution, wind) and by normal use.

The deterioration of concrete structures can be categorized in different ways (e.g. in terms of damage types, causes, mechanisms of attack, frequency of defects, financial loss or cost of repair [4]). The present study adopts a classification based on the causes of attack. Thus, the causes of attack are grouped in four main families, namely chemical, physical, biological and mechanical factors as illustrated in Fig. 1.

The chemical factors include the chemical reactions causing deterioration of concrete structures like carbonation, chloride attack, effects of acids and sulphates and alkali-aggregate reactions. The physical factors include freezing-thawing cycles, shrinkage and cracking and exposure to temperature extremes such as freezing or fire. The biological factors include the effects of biological agents like microorganisms, fungi, algae and moss. Finally the mechanical factors include abrasion, erosion and cavitation.

Carbonation occurs due to the penetration of atmospheric carbon dioxide into the concrete. In contact with the pore water, the carbon dioxide produces carbonic acid that reacts with calcium hydroxide, creating a slightly less alkaline environment for the reinforcement rods. The passivating film is neutralised and the rods are exposed to aggression by the oxygen.

The aggression by chlorides occurs when concrete is in contact with environments with high chloride content like seawater or de-icing salts or if concrete was prepared using contaminated raw materials. Chlorides attack the passivating film and the rods are exposed to aggression by the oxygen.
Sulphate ions may be present in water and in the ground, and may also be found directly in the aggregates as impurities. The water-soluble sulphates penetrate into concrete pore water and react with aluminates or calcium hydroxide in cement paste. The reaction products expand remarkably, which causes crack propagation and decreases the strength properties of concrete.

Alkali-aggregates reactions may cause considerable expansion and serious deterioration of concrete structures. Reactive siliceous minerals in the aggregate react with alkaline hydroxides originating usually from cement. Alkali-silicate gel is formed in the voids and cracks of the aggregate or on the surface of the aggregate. In contact with water, the gel can expand about 5% to 20% in volume. Internal pressures are generated and eventually cracking can destroy the concrete structure. Thawing and freezing is the most common weather related physical factor. Freeze-thaw damage is generated by repeated hydraulic pressures caused by volume expansion when water turns to ice (volume increases by 9%). In each successive freeze-thaw cycle, the cumulative effect causes expanding deterioration of concrete. The deterioration is visible in the form of cracking, scaling, and general degradation of the surface paste.

The exposure to temperature extremes like fire causes severe damage on concrete structures. The main harm in fire is caused by a combination of the effects of smoke and gases, which are emitted from burning materials, and the effects of flames and high air temperatures. The most serious form of damage to concrete under fire is explosive spalling, which occurs usually during the first 30 minutes after fire starts. The conventional explanation of explosive spalling is that it is caused by the build-up of water vapor pressure in concrete during fire and thermal stresses. During fire concrete undergoes severe microstructural changes that lead to irreversible structural damage. More details about the effect of fire on concrete and concrete structures can be found in [5].

Mechanical abrasion of concrete surfaces occurs when a material is repeatedly struck by particles from a harder body, due to the friction that the harder powder particles exercise on the surface of the material. This type of deterioration can be caused by different agents like the slid of different materials or wheels. Erosion is a particular form of wear due to wind, water or ice that provokes the removal of material from the concrete surface. Cavitation is caused by flowing water when the pressure changes abruptly. The air bubbles (formed in the water flow downstream) collapse and create a strong impact on the concrete surface. If the speed of the water is particularly high damage may be serious.

Solving problems related to degradation of concrete requires a proactive strategy. Thus, the development of models to evaluate the degradation of concrete may be a way to solve or minimize the problem. This work introduces a computational system to evaluate the degradation of concrete centred on logic programming to knowledge representation and reasoning, complemented with a computational framework based on Artificial Neural Networks.

II. KNOWLEDGE REPRESENTATION AND REASONING

Many approaches for knowledge representation and reasoning have been proposed using the Logic Programming (LP) paradigm, namely in the area of Model Theory [6]–[8], and Proof Theory [9], [10]. We follow the proof theoretical approach and an extension to the LP language, to knowledge representation and reasoning. An Extended Logic Program (ELP) is a finite set of clauses in the form:

\[ p \leftarrow p_1, \ldots, p_n, \neg q_1, \ldots, \neg q_m \]  

\[ ? (p_1, \ldots, p_n, \neg q_1, \ldots, \neg q_m) \ (n, m \geq 0) \]

where \( \neg \) is a domain atom denoting falsity, the \( p_i, q_j \), and \( p \) are classical ground literals, i.e., either positive atoms or atoms preceded by the classical negation sign \( \neg \). Under this representation formalism, every program is associated with a set of abducibles [6], [8], given here in the form of exceptions to the extensions of the predicates that make the program. Once again, LP emerged as an attractive formalism for knowledge representation and reasoning tasks, introducing an efficient search mechanism for problem solving.

Due to the growing need to offer user support in decision making processes some studies have been presented [11], [12], related to the qualitative models and qualitative reasoning in Database Theory and in Artificial Intelligence research. With respect to the problem of knowledge representation and reasoning in Logic Programming (LP), a measure of the Quality-of-Information (QoI) of such programs has been object of some work with promising results [13], [14]. The QoI with respect to the extension of a predicate \( i \) will be given by a truth-value in the interval [0,1], i.e., if the information is known (positive) or false (negative) the QoI for the extension of predicate \( i \) is 1. For situations where the information is unknown, the QoI is given by:

\[ QoI_i = \lim_{N \to \infty} \frac{1}{N} = 0 \quad (N > 0) \]  

where \( N \) denotes the cardinality of the set of terms or clauses of the extension of predicate \( i \), that stand for the incompleteness under consideration. For situations where the extension of predicate \( i \) is unknown but can be taken from a set of values, the QoI is given by:

\[ QoI_i = \frac{1}{\text{Card}} \]

where \( \text{Card} \) denotes the cardinality of the abducibles set for \( i \), if the abducibles set is disjoint. If the abducibles set is not disjoint, the QoI is given by:

\[ QoI_i = \frac{1}{c_{i1}^{\text{Card}} + \ldots + c_{i\text{Card}}^{\text{Card}}} \]

where \( c_{i\text{Card}}^{\text{Card}} \) is a card-combination subset, with \( \text{Card} \) elements. The next element of the model to be considered is the relative importance that a predicate assigns to each of its attributes.
under observation, i.e., \( w^K \), which stands for the relevance of attribute \( k \) in the extension of \( \textit{predicate}_i \). It is also assumed that the weights of all the attribute predicates are normalized, i.e.:

\[
\sum_{1 \leq k \leq n} w^K_i = 1, \forall i
\]  

(6)

where \( \forall \) denotes the universal quantifier. It is now possible to define a predicate’s scoring function \( V_i(x) \) so that, for a value \( x = (x_1, \ldots, x_n) \), defined in terms of the attributes of \( \textit{predicate}_i \), one may have:

\[
V_i(x) = \sum_{1 \leq k \leq n} w^K_i \times QoI_i(x)/n
\]  

(7)

It is now possible to engender all the possible scenarios of the universe of discourse, according to the information given in the logic programs that endorse the information depicted in Fig. 3, i.e., in terms of the extensions of the predicates \( \textit{General Information}, \textit{Biological Effects}, \textit{Chemical Effects}, \textit{Mechanical Effects}, \textit{Physical Effects} \) and \( \textit{Environmental Exposure} \).

It is now possible to rewrite the extensions of the predicates referred to above, in terms of a set of possible scenarios according to productions of the type:

\[
\textit{predicate}((x_1, \ldots, x_n)) :: QoI
\]  

(8)

and evaluate the Degree of Confidence (DoC) given by

\[
\text{DoC} = V_i(x_1, \ldots, x_n)/n
\]

which denotes one’s confidence in a particular term of the extension of \( \textit{predicate}_i \). To be more general, let us suppose that the Universe of Discourse is described by the extension of the predicates:

\[
\text{a}_1(\cdots), \text{a}_2(\cdots), \ldots, \text{a}_n(\cdots) \quad (n \geq 0)
\]  

(9)

Therefore, for a given scenario, one may have (where \( \perp \) denotes an argument value of the type unknown; the values of the others arguments stand for themselves):

\[
\begin{align*}
-a_1(x_1, y_1, z_1) & \iff \neg a_1(x_1, y_1, z_1)
\quad a_1([7, 8], 15, \perp) :: 0.75
\quad [5, 10][3, 25][0, 5]
\quad \text{attribute’s domains for } x_1, y_1, z_1
\quad \vdots \\
-a_2(x_2, y_2, z_2) & \iff \neg a_2(x_2, y_2, z_2)
\quad a_2([33, 42], [10, 12], \perp) :: 0.6
\quad [20, 50][6, 14][200, 500]
\quad \text{attribute’s domains for } x_2, y_2, z_2
\end{align*}
\]

(10)

The Degree of Confidence (DoC) is evaluated using the equation

\[
\text{DoC} = \frac{1}{\Delta l}
\]

(11)

as it is illustrated in Fig. 2. Here \( \Delta l \) stands for the length of the arguments’ intervals, once normalized.

Below, one has the expected representation of the universe of discourse, where all the predicates’ arguments are nominal. They speak for one’s confidence that the unknown values of the arguments fit into the correspondent intervals referred to above.
III. A CASE STUDY

Therefore, and in order to exemplify the applicability of our model, we will look at the relational database model, since it provides a basic framework that fits into our expectations [15], and is understood as the genesis of the LP approach to knowledge representation and reasoning.

Consider, for instance, the scenario where a relational database is given in terms of the extensions of the relations or predicates depicted in Fig. 3, which stands for a situation where one has to manage information about concrete structures. Under this scenario some incomplete data is also available. For instance, in relation Biological Effects the presence/absence of microorganisms for case 2 is unknown, while in relation General Information the Age value for example 1 ranges in the interval [50,60].

The Environmental Exposure database (Fig. 3) is populated according to [16]. Thus, 0 (zero) and 1 (one), in column Risk of Corrosion denote, respectively, absence and existence of corrosion risk. Corrosion by Carbonation is classified between 1 (one) and 4 (four). 1 (one) for sub-class XC1, 2 (two) for sub-class XC2, 3 (three) for sub-class XC3 and 4 (four) for sub-class XC4. Corrosion by Chlorides (except seawater) is categorized between 0 (zero) and 3 (three). 0 (zero) means absence of corrosion by chlorides, 1 (one), 2 (two) and (three) stand, respectively, for sub-classes XD1, XD2 and XD3. Similarly, Corrosion by Sea Water Chlorides ranges between 0 (zero) and 3 (three). 0 (zero) means absence of corrosion by seawater chlorides, 1 (one), 2 (two) and (three) denote, respectively, the sub-classes XS1, XS2 and XS3. Impact of Freeze/Thaw Cycles is rated between 1 (one) and 4 (four), respectively for sub-classes XF1, XF2, XF3 and XF4. The last one, Chemical Agents, is classified between 1 (one) and 3 (three), respectively for sub-classes XA1, XA2 and XA3. The value of Environmental Exposure in Concrete Deterioration database is calculated by:

\[
\text{Environmental Exposure} = X_0 \times (X_C + X_D + X_S) + \ + X_F + X_A
\]

where \(X_0\), \(X_C\), \(X_D\), \(X_S\), \(X_F\) and \(X_A\) denote the values of the respective column in Environmental Exposure database. In this way this value is set between \([2,14]\).

The values of the Biological, Chemical, Mechanical and Physical Effects in Concrete Deterioration database are the sum of the respective values, ranging between \([0,3]\) for the two first effects and between \([0,4]\) for the remaining ones. In Biological Effects database the column Animals includes the presence/absence of insects, birds, rodents, termites, worms. The column Other encompasses the presence/absence of seeds, roots, moulds, fungi, moss, algae.

Now, we may consider the extensions of the relations given in Fig. 3 to populate the extension of the concrete predicate, given in the form:

\[
\text{concrete} : \text{Age,} \text{last Inspection,} U_{age}, \text{Biological effects, Chemical effects, Mechanical effects, Physical effects, Environmental Exposure} \rightarrow \{0,1\}
\]

where 0 (zero) and 1 (one) denote, respectively, the truth-values false and true. It is now possible to give the extension of the predicate concrete, in the form:

\[
\begin{align*}
\neg \text{concrete} (\text{Age,} \text{last Inspection,} U_{age}, \text{Biological effects, Chemical effects, Mechanical effects, Physical effects, Environmental Exposure}) \\
\leftarrow \neg \text{concrete} (\text{Age,} \text{last Inspection,} U_{age}, \text{Biological effects, Chemical effects, Mechanical effects, Physical effects, Environmental Exposure})
\end{align*}
\]

\[
\text{concrete} ([50,60], \text{3, 2, 1, 1, 2, 5}) \rightarrow \text{1} \left[0.050][0.25][1.3][0.3][0.3][0.4][0.4][2,14]\right]
\]

where \(x\) is the value of \(X_0\), \(y\) is the value of \(X_C\), \(z\) is the value of \(X_D\), \(w\) is the value of \(X_S\), \(a\) is the value of \(X_F\), \(b\) is the value of \(X_A\) and \(c\) is the value of \(X_{environmental~exposure}\).
Fig. 3 An extension of the relational database model. In column *Usage* of *General Information* database, 1 (one), 2 (two) and 3 (three) stand, respectively, for low, regular and high usage. In column *Freeze/Thaw Cycles* of *Physical Effects* database, 1 (one), 2 (two) and 3 (three) denote, respectively, rare, frequent and very frequent exposure to freeze/thaw cycles. In the remaining columns of *Biological, Chemical, Mechanical and Physical Effects* 0 (zero) denotes absence and 1 (one) denotes presence.

\{
\begin{align*}
\text{not concrete} & \leftarrow \text{concrete} (\text{Age}, L, U, \text{Bio}, \text{Chem}, \text{Mech}, \text{Phy}, \text{Env}) \\
\text{concrete} & \leftarrow [0.33, 0.4], [0.12, 0.12], [0.5, 0.5], [0.33, 0.33], [0.33, 0.33], [0.25, 0.25], [0.5, 0.5], [0.25, 0.25] : 1 \\
& \begin{bmatrix}
[0.1] & [0.1] & [0.1] & [0.1] & [0.1] & [0.1] & [0.1] & [0.1]
\end{bmatrix}
\end{align*}
\}

attribute’s domains

\{
\begin{align*}
\text{not concrete}_D & \leftarrow \text{concrete}_D (\text{Age}, L, U, \text{Bio}, \text{Chem}, \text{Mech}, \text{Phy}, \text{Env}) \\
\text{concrete}_D & \leftarrow (0.998, 1, 1, 1, 1, 1, 1, 1) : 1 \\
& \begin{bmatrix}
[0.33, 0.4], [0.12, 0.12], [0.5, 0.5], [0.33, 0.33], [0.33, 0.33], [0.25, 0.25], [0.5, 0.5], [0.25, 0.25]
\end{bmatrix}
\end{align*}
\}

attribute’s values ranges

attribute’s domains

The logic program referred to above, is now presented in the form:
\[
\text{concrete}_{\text{Doc}} (1, 0, 1, 0, 1, 1, 1, 1) \quad :: \quad 1
\]

\[ \begin{bmatrix}
0.247, 0.247, [0.1], [0.0], [0.1], [0.33, 0.33], [0.25, 0.25], [0.25, 0.25], [0.75, 0.75]
\end{bmatrix}
\]

\[
\begin{bmatrix}
[0.1] & [0.1] & [0.1] & [0.1] & [0.1] & [0.1] & [0.1]
\end{bmatrix}
\]

attribute's values ranges

\[
\begin{bmatrix}
[0.1] & [0.1] & [0.1] & [0.1] & [0.1] & [0.1] & [0.1]
\end{bmatrix}
\]

attribute's domains

where its terms make the training and test sets of the Artificial Neural Network given below (Fig. 4).

IV. ARTIFICIAL NEURAL NETWORKS

In [17]–[19] it is shown how Artificial Neural Networks (ANNs) could be successfully used to model data and capture complex relationships between inputs and outputs. ANNs simulate the structure of the human brain being populated by multiple layers of neurons. As an example, let us consider the third case presented in Fig. 3, where one may have a situation that may lead to concrete degradation, which is given in the form:

\[
\begin{bmatrix}
\text{concrete attributes (Age, Li, U, Bio, Chem, Mech, Phy, Env)}
\end{bmatrix}
\]

\[
\downarrow
\]

\[
\text{concrete (25, [5,8], 3, 2, 0, 2, 3, 4) :: 1}
\]

\[
\begin{bmatrix}
[0.150] & [0.25] & [1.3] & [0.3] & [0.3] & [0.4] & [0.4] & [2,14]
\end{bmatrix}
\]

attribute's domains

\[
\downarrow 1\text{st interaction: transition to continuous intervals}
\]

\[
\text{concrete([25,25], [5,8], [3,3], [2,2], [0,0], [2,2], [3,3], [4,4])}
\]

\[
\begin{bmatrix}
[0.150] & [0.25] & [1.3] & [0.3] & [0.3] & [0.4] & [0.4] & [2,14]
\end{bmatrix}
\]

attribute's domains

\[
\downarrow 2\text{nd interaction: normalization} \quad \frac{Y - Y_{\text{min}}}{Y_{\text{max}} - Y_{\text{min}}}
\]

\[
\text{concrete([0.17,0.17], [0.2,0.32], [1,1], [0.67,0.67], [0,0], [0.5,0.5], [0.75,0.75], [0.17,0.17]) :: 1}
\]

\[
\begin{bmatrix}
[0.1] & [0.1] & [0.1] & [0.1] & [0.1] & [0.1] & [0.1] & [0.1]
\end{bmatrix}
\]

attribute's domains

\[
\downarrow \text{DoC calculation: DoC} = \sqrt{1 - \Delta I^2}
\]

\[
\text{concrete}_{\text{Doc}} (1, 0.993, 1, 1, 1, 1, 1, 1) :: 1
\]

\[
\begin{bmatrix}
0.17, 0.17, [0.2,0.32], [1,1], [0.67,0.67], [0,0], [0.5,0.5], [0.75,0.75], [0.17,0.17]
\end{bmatrix}
\]

attribute's values ranges

\[
\begin{bmatrix}
[0.1] & [0.1] & [0.1] & [0.1] & [0.1] & [0.1] & [0.1]
\end{bmatrix}
\]

attribute's domains
In Fig. 4 it is shown how the normalized values of the interval boundaries and their DoC and QoI values work as inputs to the ANN. The output translates the chance of being necessary to go ahead with an intervention, and concrete\textsubscript{DoC} the confidence that one has on such a happening. In addition, it also contributes to build a database of study cases that may be used to train and test the ANNs.

V. CONCLUSIONS AND FUTURE WORK

To set a timeline to the maintenance of concrete structures is a hard and complex task, which needs to consider many different conditions with intricate relations among them. These characteristics put this problem into the area of problems that may be tackled by AI based methodologies and techniques to problem solving. Despite that, little to no work has been done in that direction. This work presents the founding of a computational framework that uses powerful knowledge representation and reasoning techniques to set the structure of the information and the associate inference mechanisms. This representation is above everything else, very versatile and capable of covering every possible instance by considering incomplete, contradictory, and even unknown data. The main contribution of this work is to be understood in terms of the evaluation of the DoC, and the possibility to address the issue of incomplete information. Indeed, the new paradigm of knowledge representation and reasoning enables the use of the normalized values of the interval boundaries and their DoC values, as inputs to the ANN. The output translates the chance of the deterioration of concrete structures and the degree of confidence that one has on such a happening. Future work may recommend that the same problem must be approached using other computational frameworks like Case Based Reasoning or Particle Swarm, just to name a few.

REFERENCES


