Adaptive Fuzzy Interpolation with Prioritized Component Candidates

Shen, Qiang; Yang, Longzhi

Published in:
2011 IEEE International Conference on Fuzzy Systems (FUZZ)

DOI:
10.1109/FUZZY.2011.6007463

Publication date:
2011

Citation for published version (APA):
Shen, Q., & Yang, L. (2011). Adaptive Fuzzy Interpolation with Prioritized Component Candidates. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ)* (pp. 428-435). IEEE Press. https://doi.org/10.1109/FUZZY.2011.6007463

2011 IEEE International Conference on Fuzzy Systems
June 27-30, 2011, Taipei, Taiwan

Adaptive Fuzzy Interpolation with Prioritized Component Candidates

Longzhi Yang and Qiang Shen
Department of Computer Science
Aberystwyth University
Aberystwyth, SY23 3DB, UK
Email: {lly07,qqs}@aber.ac.uk

***Abstract*—Adaptive fuzzy interpolation strengthens the potential of fuzzy interpolative reasoning. It first identifies all possible sets of faulty fuzzy reasoning components, termed the candidates, each of which may have led to all the contradictory interpolations. It then tries to modify one selected candidate in an effort to remove all the contradictions and thus restore interpolative consistency. This approach assumes that all the candidates are equally likely to be the real culprit. However, this may not be the case in real situations as certain identified reasoning components may be more liable to resulting in inconsistencies than others. This paper extends the adaptive approach by prioritizing all the generated candidates. This is achieved by exploiting the certainty degrees of fuzzy reasoning components and hence of derived propositions. From this, the candidate with the highest priority is modified first. This extension helps to quickly spot the real culprit and thus considerably improves the approach in terms of efficiency.**

***Index Terms*—Adaptive fuzzy interpolation, assumption-based truth maintenance systems, reliability-based general diagnostic engine.**

I. INTRODUCTION

Fuzzy rule interpolation enhances the robustness of fuzzy reasoning. When given observations have no overlap with any antecedent values, no rule can be fired in classical inference. However, interpolative reasoning through a sparse rule base may still obtain certain conclusions and thus improve the applicability of fuzzy models. Also, with the help of fuzzy interpolation, the complexity of a rule base can be reduced by omitting those fuzzy rules which may be approximated from their neighboring ones. A number of important interpolation approaches have been presented in the literature, including [1], [2], [3], [4], [7], [8], [9], [10], [11], [12], [13], [14]. In particular, the scale and move transformation-based approach can handle both interpolation and extrapolation, which involve multiple fuzzy rules, with each rule consisting of multiple antecedents. This approach also guarantees the uniqueness as well as normality and convexity of the resulting interpolated fuzzy sets. Yet, it is possible that more than one object value of a single variable may be derived in fuzzy interpolation. This implies that certain inconsistencies may have resulted.

To address this problem, *adaptive fuzzy interpolation* has recently been proposed [15]. This approach is capable of efficiently detecting inconsistencies, locating possible fault candidates and modifying the candidates in an effort to remove all the inconsistencies. It works by viewing the interpolative inference procedures as artificial system components, and then utilizing an assumption-based truth maintenance system (ATMS) [5] to record the dependencies between an interpolated value and its proceeding interpolation components. From this, the classical algorithm of general diagnostic engine (GDE) [6] is employed to manipulate the sets of dependent components of contradictions to hypothesize all possible candidates of defective rules.

The adaptive approach of [15] assumes that all the component candidates are equally reliable, but this may not be true in reality. This is because: i) two derived object values for a common variable which have led to a contradiction may not be equally reliable (moreover, one may be correct and the other wrong); and ii) all the interpolation components that jointly entail one of the two contradictory object values may not be equally reliable. Given information on such differences, all the generated component candidates may be prioritized. This is achieved by firstly extending the traditional ATMS not only to record dependencies, but also to record how reliable these dependencies are. Then GDE is modified in an effort to prioritize all the component candidates. The prioritization of candidates will inevitably help the modification procedure to quickly arrive at a consistent solution and thus save computational effort.

The rest of this paper is structured as follows. Sec. II reviews the adaptive fuzzy interpolation approach. Sec. III presents the extension that shows how component candidates can be efficiently prioritized. Sec. IV reconsiders the example given in [15] to illustrate how this extension can improve the original approach. Sec. V concludes the paper and points out important future research directions.

II. ADAPTIVE FUZZY INTERPOLATION

Adaptive interpolative reasoning [15] provides a way to ensure that inference results remain consistent throughout the fuzzy interpolative process. The degree of consistency is typically expressed as the degree of matching. In particular, the matching degree between two fuzzy sets A_{ij} and A_{ik}, denoted as $M(A_{ij}, A_{ik})$, in the domain D_{x_i} of variable x_i is defined as follows, which is used to evaluate consistency efficiently (though distance-based measure may also be used for this):

$$M(A_{ij}, A_{ik}) = \sup_{x \in D_{x_i}} [\min(\mu_{A_{ij}}(x), \mu_{A_{ik}}(x))]. \tag{1}$$

Then, the degree β of a contradiction with respect to two propositions $P\ (x_i\ is\ A_{ij})$ and $P'(x_i\ is\ A_{ik})$ is defined by:

$$\beta = 1 - M(A_{ij}, A_{ik}). \tag{2}$$

A predefined threshold β_0 $(0 \leq \beta_0 \leq 1)$ is adopted in order to determine those values assigned to a common variable with an unacceptable contradictory degree. A contradiction is called a β_0-*contradiction* if the corresponding degree of contradiction $\beta > \beta_0$.

In implementing fuzzy interpolation, each pair of neighboring rules is defined as a *fuzzy reasoning component*. Such a component takes an input, an observation or a previously interpolated result, including that obtained by extrapolation (which are all hereafter referred to as an observation for simplicity), and produces another (the consequent of the interpolated rule) as output. The process of adaptive interpolation is summarized in Fig. 1. Firstly, the interpolator carries out interpolation and passes the interpolated results to the ATMS for dependency-recording. Then, the ATMS relays any β_0-*contradictions* as well as their dependent fuzzy reasoning components to the GDE which diagnoses the problem and generates all possible component candidates. After that, a modification process takes place to correct a certain candidate to restore consistency. A brief description of each of these key methods is given below.

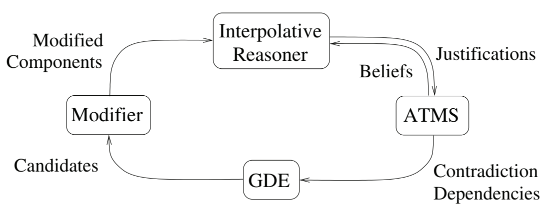

Fig. 1. Adaptive interpolative reasoning

A. Truth maintenance

ATMS is utilized to record the dependency of the interpolated results and that of contradictions, upon the fuzzy reasoning components from which they are inferred. Thus, propositions, contradictions and fuzzy interpolative reasoning components are all represented as ATMS nodes. In addition to the so-called datum field, which trivially denotes the actual meaning of the node, an ATMS node has two other fields: justification and label.

1) Justification: Briefly, a justification describes how a node is derivable from other nodes. Any ATMS node with an inferred proposition is represented by an ATMS justification as:

$$O, R_i R_j \Rightarrow C, \tag{3}$$

where $R_i R_j$ stands for the fuzzy reasoning component containing the two neighboring rules R_i and R_j $(i \neq j)$ that have been used to infer the outcome C from the observation O.

According to the definition of contradiction above, any two propositions $P\ (x_i\ is\ A_{ij})$ and $P'(x_i\ is\ A_{ik})$ concerning the same variable x_i are contradictory to a certain degree β. When β is not higher than β_0, the contradictory degree is acceptable and the two considered propositions are treated as being consistent in ATMS. Otherwise, a β_0-$contradiction$ is deduced, which can be represented as:

$$P, P' \Rightarrow_{\beta_0} \perp. \tag{4}$$

2) Label and label-updating: A label is a set of environments, each of which supports the associated node. In particular, an environment is a minimal set of fuzzy reasoning components that jointly entail the supported node, thereby describing how the node ultimately depends on those fuzzy reasoning components. An environment is said to be β_0-*inconsistent* if β_0-contradiction is derivable propositionally by the environment and a given justification. An environment is said to be $(1-\beta_0)$-*consistent* if it is not β_0-inconsistent.

The label of each node is guaranteed to be $(1 - \beta_0)$-consistent, sound, minimal and complete by the label updating algorithm, except that the label of the special "false" node is guaranteed to be β_0-inconsistent rather than $(1 - \beta_0)$-consistent. In particular, the label of the special "false" node gathers all minimal β_0-inconsistent environments. Its corresponding label-updating process is given as follows. Whenever a β_0-contradiction is detected, each environment in its label is added into the label of "false" node and all such environments and their supersets are removed from the label of every other node. Also, any such environment which is a superset of another is removed from the label of the node "false".

B. Candidate generation

GDE [6] generates minimal candidates by manipulating the label of the specific "false" node. A candidate is a particular set of nodes or fuzzy reasoning components which may be responsible for the whole set of current contradictions. Because a β_0-inconsistent environment indicates that at least one of its elements is faulty, a candidate must have a non-empty intersection with each β_0-inconsistent environment. Thus, each candidate is constructed by taking one fuzzy reasoning component from each environment in the label of "false" node. Supersets removal then ensures such generated candidates to be minimal. In light of this, a successful correction of any single candidate will remove all the contradictions.

C. Candidate modification

Consistency can be restored by successfully correcting any single candidate because each single candidate explains the entire set of current contradictions. Given a set of candidates, the modification procedure is shown in Fig. 2.

For convenience, in the rest of this paper, A^*_{ij} is used to denote the modified consequence of a culprit interpolated rule whose consequent value is A_{ij}, and $A^{*\prime}_{ij}$ and λ^*_{ij} are used to denote the corresponding modified intermediate rule consequence and the *relative placement factor* [15] of A^*_{ij}, respectively. Also, for simplicity, only rules with a single antecedent are considered here, through extension to multi-antecedent rules is straightforward. Suppose that the neighboring rules $A_{11} \Rightarrow A_{21}$ and $A_{1n} \Rightarrow A_{2n}$ are the two rules used

CONSISTENCYRESTORING($\mathbb{Q}$)

$\mathbb{Q}$, the candidate set sorted in descending cardinality, each element of which is a set of fuzzy reasoning components (f); MODIFY(f), the modification procedure for single fuzzy reasoning component (f). Return **true** when modification succeeds and **false** otherwise.

```
success ← false
do
    C ← Dequeue(Q)
    foreach f ∈ C
        success ← MODIFY(f)
        if (success ==false)
            break
until ((success ==true) or (Q == ∅))
return success
```

Fig. 2. The CONSISTENCYRESTORING procedure

for interpolation by a defective fuzzy reasoning component, that $A_{12}, A_{13}, ..., A_{1(n-1)}$ are observations or previously interpolated results located in between A_{11} and A_{1n}, and that A_{1j} $(2 \leq j \leq n-1)$ is the middle most observation. The modification procedure for single fuzzy reasoning component can then be summarized as follows:

1. Find the rule $(A_{1j} \Rightarrow A_{2j})$ whose antecedent is located in the middle most of the neighborhood of the antecedents of any two rules that may be used for interpolation, with respect to their representative values. Assume that the *relative placement factor* of its consequence λ_{2j} is modified to λ_{2j}^*.

2. Calculate the *correction rate pair* according to the *relative placement factor* modification of rule $A_{1j} \Rightarrow A_{2j}$:

$$\begin{cases} c^- = \frac{\lambda_{2j}^*}{\lambda_{2j}} \\ c^+ = \frac{1-\lambda_{2j}^*}{1-\lambda_{2j}}. \end{cases} \tag{5}$$

3. Calculate the modified *relative placement factors* of consequences of all other interpolated rules which are generated from the same defective fuzzy reasoning component as per the *correction rate pair* computed above, where $i \in \{2, 3, ..., j-1\}$ and $k \in \{j+1, j+2, ..., n-1\}$:

$$\begin{cases} \lambda_{2i}^* = \lambda_{2i} \cdot c^- \\ 1 - \lambda_{2k}^* = (1 - \lambda_{2k}) \cdot c^+. \end{cases} \tag{6}$$

4. Calculate the modified consequences of all interpolated rules which are generated from the same defective fuzzy reasoning component in accordance with their modified *relative placement factors*:

$$\begin{cases} {A_{2x}^*}' = (1 - \lambda_{2x}^*)A_{21} + \lambda_{2x}^* A_{2n} \\ T({A_{1x}}', A_{1x}) = T({A_{2x}^*}', A_{2x}^*), \end{cases} \tag{7}$$

where $x \in \{2, 3, ..., n-1\}$, and $T(A', A)$ represents the scale and move transformations [7], [8] from fuzzy set A' to A.

5. Restrict the modified consequence so that it is consistent with the context. Suppose that m object values $A_{i1}, A_{i2}, ..., A_{im}$ are obtained for variable x_i. If they are $(1 - \beta_0)$-consistent, they must satisfy:

$$\bigcap_{j=1}^{m} (A_{ij})_{\beta_0} \neq \varnothing, \tag{8}$$

where $(A_{ij})_{\beta_0}$ denotes the β_0-cut of fuzzy set A_{ij}.

6. Restrict the propagations of all modified consequences so that they are consistent with the context. For simplicity, let function $I(A_{ij}, R_l R_r) = A_{kj}$ denote the standard interpolation from the antecedent fuzzy set A_{ij} to the consequent value A_{kj}, through fuzzy reasoning component $R_l R_r$. Suppose that m object values $A_{i1}, A_{i2}, ..., A_{im}$ of variable x_i, located between the antecedent values of rules R_l and R_r, are modified, that the corresponding modified object values of variable x_k are A_{kj}^*, $j \in \{1, 2, ..., m\}$, and that n object values A_{kl}, $l \in \{1, 2, ..., n\}$, of variable x_k are already derived previously. If the modified consequences A_{kj}^* are all $(1-\beta_0)$-consistent, then they must satisfy:

$$\begin{cases} A_{kl}^* = I(A_{ij}^*, R_l R_r) \\ \left(\bigcap_{j=1}^{m} (A_{kj}^*)_{\beta_0} \right) \bigcap \left(\bigcap_{l=1}^{n} (A_{kl})_{\beta_0} \right) \neq \varnothing. \end{cases} \tag{9}$$

7. Solve all these simultaneous equations. The result is the modified solution which ensures inconsistency-free.

III. THE EXTENSION

The approach described above assumes that each candidate is equally likely to be the real culprit and naturally the candidate with the smallest cardinality is always tentatively modified first. However, this may not be true in real-world problems and, as argued earlier, differences exist amongst such candidates. In order to prioritize all the generated candidates, fuzzy reasoning components and derived propositions need to be ranked first in accordance with their certainty or reliability degrees.

A. Certainty degrees of fuzzy reasoning components

In this work, a fuzzy reasoning component composed by a pair of neighboring rules is used to represent the fuzzy interpolation mechanism which uses these two rules. In particular, the approach is based on the use of the scale and move transformation-based fuzzy interpolation techniques. Essentially, it is a fuzzy extension of classical linear interpolation. Thus, it is expected that such an artificial component will be more appropriate to represent those pairs of neighboring rules where relative distance between the two antecedents is closer to the corresponding relative distance between the two consequences than otherwise. In other words, the fuzzy reasoning components which are defined on such pairs of neighboring rules are more certain to derive correct interpolated results than the others, under the linearity assumption.

Suppose that fuzzy reasoning component $R_i R_j$ is composed by the following two rules:

$$\begin{array}{l} R_i: \text{ If } x_1 \text{ is } A_{1i}, \text{ then } x_2 \text{ is } A_{2i}; \\ R_j: \text{ If } x_1 \text{ is } A_{1j}, \text{ then } x_2 \text{ is } A_{2j}. \end{array} \tag{10}$$

Inspired by the above observation, the certainty degree $c_{R_iR_j}$ of fuzzy reasoning component R_iR_j can therefore be defined by:

$$c_{R_iR_j} = 1 - \left| \frac{d(A_{1i}, A_{1j})}{max_1 - min_1} - \frac{d(A_{2i}, A_{2j})}{max_2 - min_2} \right|, \qquad (11)$$

where $d(A_{ki}, A_{kj})$, $k \in \{1, 2\}$, is the distance between fuzzy sets A_{ki} and A_{kj} (given a certain distance metric); max_k and min_k are the maximum and minimum of the domain values of variable x_k, respectively. Obviously, $c_{R_iR_j} \in [0, 1]$.

B. Certainty degree of an interpolated result

Given an observation or previously inferred result O and a fuzzy reasoning component R_iR_j which is composed by two rules R_i and R_j that flank the given observation O, a logical consequence C can be generated by the scale and move transformation-based fuzzy interpolation approach. The certainty degree c_C of the conclusion C is then calculated by:

$$c_C = c_O \otimes c_{R_iR_j}, \qquad (12)$$

where the composition operator $\otimes$ can be minimum, algebraic product or any other T-norm operator. If O is an observation, it is regarded fully certain in this research and hence, c_O is assumed to be 1. Otherwise, c_O is calculated from a previous interpolation.

In general, the certainty degree of an interpolated result should also depend on the certainty degree (reliability) of the two rules used for interpolation. However, in this research, as with its preceding work, all given rules are presumed to be fixed and true. That is, the certainty degree of any rule in the rule base is 1. Therefore, the contribution of any rule's certainty degree can be ignored in the calculation of the certainty degree of an interpolated result.

Note that two different applications of interpolation procedures may result in the same interpolated result C, but with two different certainty degrees associated, say c_C and $c_{C'}$. Then the overall certainty degree of C, denoted as c is updated such that

$$c = c_C \oplus c_{C'}, \qquad (13)$$

where $\oplus$ is the maximum operator, or any other S-norm operator.

C. Reliability-based ATMS

Having introduced the certainty degrees of fuzzy reasoning components and those of the interpolated results, ATMS needs to be extended accordingly in order to represent such information within the ATMS network. For this purpose, an extra filed, termed certainty degree, is introduced to each ATMS node, which measures to what extent the corresponding node can be (logically disjunctively) derived from all the label environments according to the entire set of current justifications. Each extended ATMS node is therefore of the following form:

$$\langle N, c, \{E_1, E_2, ..., E_n\}, \{J_1, J_2, ..., J_m\} \rangle, \qquad (14)$$

where N is the underlying meaning of the node, c is the certainty degree of N, $\{E_1, E_2, ..., E_n\}$ is the ATMS label and $J_1, J_2, ..., J_m$ are the justifications. In addition, each label environment E_i $(1 \leq i \leq n)$ is of the form:

$$E_i = \{N_{11}, N_{12}, ..., N_{1k_i}\}, \qquad (15)$$

where N_{ij}, $j \in \{1, 2, ..., k_i\}$, are those ATMS nodes which jointly support N.

1) Justification: A justification within this extended ATMS conveys two types of information: i) how the present node is derived from other nodes; and ii) to what extent the node is derivable from the other nodes. The first type of information is the same as that conveyed by the classical ATMS. For all nodes but the special "false" node, any justification is of the form of Eq. 3. Thus, the second type of information can be calculated with respect to Eq. 12.

Note that, for the special "false" node, the justifications are of the form of Eq. 4. However, in this research, no direct use of such justifications is required. Hence, any calculation involving the second type of information for this node is omitted.

2) Label and label-updating: In the extended ATMS, the set of fuzzy reasoning components that compose a label environment not only jointly entails their supported node, but also determines how certain the supported node can be logically derived from that environment. From this, the overall certainty degree of the supported node can be determined by the entire set of label environments. Without losing generality, suppose that node N can be derived by n different environments and that the kth $(1 \leq k \leq n)$ environment consists of m_k fuzzy reasoning components $F_{k1}, F_{k2}, ..., F_{km_k}$. Then the node N can be derived through the kth environment to the degree:

$$c_{N_k} = c_{F_{k1}} \otimes c_{F_{k2}} \otimes ... \otimes c_{F_{Km_k}}. \qquad (16)$$

This is simply the generalized case of Eq. 12 by multiple applications of Eq. 12. By applying Eq. 13, the overall certainty degree c_N of the node N can be calculated such that

$$\begin{aligned} c_N &= c_{F_1} \oplus c_{F_2} \oplus ... \oplus c_{F_n} \\ &= (c_{F_{11}} \otimes c_{F_{12}} \otimes ... \otimes c_{F_{1m_1}}) \\ &\quad \oplus (c_{F_{21}} \otimes c_{F_{22}} \otimes ... \otimes c_{F_{2m_2}}) \\ &\quad \oplus ... \\ &\quad \oplus (c_{F_{n1}} \otimes c_{F_{n2}} \otimes ... \otimes c_{F_{nm_n}}). \end{aligned} \qquad (17)$$

In general, to update the overall certainty degree of a given node, the label-updating algorithm needs to be applied. Hence, when a new justification is present, the extended label-updating algorithm not only updates the label of the node in question in exactly the same way as that of the classical ATMS, but also updates the overall certainty degree of this node according to Eq. 17.

D. Reliability-based GDE

In this work, a β_0-contradiction appears when two object values are derived for a common variable and the contradictory degree of the two object values is greater than a given threshold β_0. A contradiction indicates that at least one of the two derived object values is faulty. In earlier work such as [15], due to lack of information, both derived object values are supposed to be equally faulty. With additional information on certainty degrees, and with the assistance of reliability-based ATMS, any two object values for a common variable can be distinguished. In effect, given a β_0-contradiction, the real defective fuzzy reasoning components which have caused this contradiction are more likely within the label environments of the less certain object value. Note that there may be more than one defective fuzzy reasoning component for a certain contradiction because there may be multiple environments each supporting the same contradiction. Amongst each label environment of one of the two object values which has caused the contradiction, the fuzzy reasoning component with the smallest certainty degree is intuitively regarded as the most likely to be the real culprit.

In the same way, all the components in each label environment of the special "false" node can be ranked. For this, a ranking value r is attached to every component of each of such label environments. The ranking value of each of those components which are in a label environment of the less certain object value, is set to its certainty degree. The ranking value of any other component in the label environments of the special "false" node is then set to its original certainty degree plus 1. Note that the range of a certainty degree is $[0, 1]$. Thus, the integer part of a ranking value indicates whether the environment that contains the given component entails a more certain object value, whilst the decimal part of the ranking value simply denotes the certainty degree of that component.

Example 3.1: Assume that a contradiction is derived from propositions P_1 and P_2, that the label of P_1 is $\{\{F_1, F_2\}, \{F_3\}\}$, and that the label of P_2 is $\{\{F_4\}\}$. Then the label of the contradiction is $\{\{F_1, F_2, F_4\}, \{F_3, F_4\}\}$ as illustrated in Fig. 3.

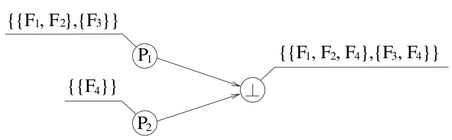

Fig. 3. The contradiction and its dependencies

Suppose that the certainty degrees of F_i, $i \in \{1, 2, 3, 4\}$, are: $c_{F_1} = 0.80$, $c_{F_2} = 0.40$, $c_{F_3} = 0.70$, $c_{F_4} = 0.50$. The certainty degrees of propositions P_1 and P_2 can be calculated with respect to Eq. 17. If the maximum and algebraic product operators are used to implement $\oplus$ and $\otimes$, respectively, then:

$$\begin{aligned} c_{P_1} &= \max\{c_{F_1} \times c_{F_2}, c_{F_3}\} \\ &= \max\{0.80 \times 0.40, 0.70\} \\ &= 0.70. \end{aligned}$$

Similarly, c_{P_2} can be calculated and the result is: $c_{P_2} = 0.50$. For label environment $\{F_1, F_2, F_4\}$ of the "false" node, because $c_{P_1} > c_{P_2}$ and F_4 is an element of the label environment of node P_2, the ranking value of F_4 is set to itself, i.e. $r_{F_4} = 0.50$, and the ranking values of F_1 and F_2 are set to their original certainty degrees plus 1, i.e. $r_{F_1} = 1.80$ and $r_{F_2} = 1.40$. In the same way, the ranking values of components in the label environment $\{F_3, F_4\}$ are set as: $r_{F_3} = 1.70$ and $r_{F_4} = 0.50$.

Without losing generality, suppose that $E_\perp$ is one of the label environments of the "false" node and it is deduced by two contradictory propositions P and P', represented by ATMS nodes N_P and $N_{P'}$ respectively. Then there must exist environments $E = \{F_1, F_2, ..., F_m\}$ and $E' = \{F'_1, F'_2, ..., F'_n\}$ which entail nodes N_P and $N_{P'}$ respectively such that $E \cup E' = E_\perp$. Suppose that the certainty degrees of propositions P and P' are c and c', respectively. Note that c is not necessarily equal to the certainty degree of which the environment E entails node N_P. The same holds for c' and $N_{p'}$. This is because: (1) an object value may be supported by more than one environment (or derived by more than one inference path), and (2) the certainty degree of the object value indicates the overall extent to which this object value may be supported by all its environments (or derived by all possible inference paths). Of course, certain elements in E may be identical to some in E'. In this case, the ranking value of each of those elements will take the less value (i.e. the one with higher priority). The procedure of attaching a ranking value to each element of $E_\perp$ is outlined in Fig. 4.

ComponentRanking(E,E',c,c')

```
E⊥ = E ∪ E'
foreach F ∈ E⊥
    if (c ≤ c' && F ∈ E)
        r_F = c_F
    else
        if (c ≥ c' && F ∈ E')
            r_F = c_F
        else
            r_F = c_F + 1
```

Fig. 4. The ComponentRanking procedure

Each label environment of the special "false" node entails a contradiction and each candidate is formed by taking one element from each environment of the special "false" node. Therefore, if all the duplications are deliberately kept, all the candidates have the same cardinality which is equal to the number of label environments in the special "false" node. From this, all candidates can be prioritized according to the ranking values of their components. This is achieved by two steps of sorting as outlined in Fig. 5. In this figure, $\mathbb{S}$ is the generated candidate set and each candidate C is itself a set of fuzzy reasoning components. Sort(C) can be any standard sorting procedure, which sorts the fuzzy reasoning components in a given candidate C in an ascending order with respect to their ranking values. StableSort($\mathbb{S}, i$) is any stable sorting algorithm which ranks all the candidates in the candidate set $\mathbb{S}$ in an ascending order by the ranking value of the ith reasoning component. Since each candidate is a set of components, any duplication needs to be removed. Also, in order to generate only a minimal candidate set, all those candidates which are a superset of one other candidate are removed. Of course, such removals do not affect the overall ranking of the remaining candidates.

Example 3.2: Continue Ex. 3.1. Suppose that the contradiction in question is the only one in the system. Therefore,

TABLE I
THE CANDIDATES SORTING PROGRESS

Candidates with duplications	Every candidate sorted	Stable-sorted (2nd elements)	Stable-sorted (1st elements)
$C_1 = [(F_1, 1.80), (F_3, 1.70)]$	$C_1 = [(F_3, 1.70), (F_1, 1.80)]$	$C_6 = [(F_4, 0.50), (F_4, 0.50)]$	$C_6 = [(F_4, 0.50), (F_4, 0.50)]$
$C_2 = [(F_1, 1.80), (F_4, 0.50)]$	$C_2 = [(F_4, 0.50), (F_1, 1.80)]$	$C_4 = [(F_4, 0.50), (F_2, 1.40)]$	$C_4 = [(F_4, 0.50), (F_2, 1.40)]$
$C_3 = [(F_2, 1.40), (F_3, 1.70)]$	$C_3 = [(F_2, 1.40), (F_3, 1.70)]$	$C_3 = [(F_2, 1.40), (F_3, 1.70)]$	$C_5 = [(F_4, 0.50), (F_3, 1.70)]$
$C_4 = [(F_2, 1.40), (F_4, 0.50)]$	$C_4 = [(F_4, 0.50), (F_2, 1.40)]$	$C_5 = [(F_4, 0.50), (F_3, 1.70)]$	$C_2 = [(F_4, 0.50), (F_1, 1.80)]$
$C_5 = [(F_4, 0.50), (F_3, 1.70)]$	$C_5 = [(F_4, 0.50), (F_3, 1.70)]$	$C_1 = [(F_3, 1.70), (F_1, 1.80)]$	$C_3 = [(F_2, 1.40), (F_3, 1.70)]$
$C_6 = [(F_4, 0.50), (F_4, 0.50)]$	$C_6 = [(F_4, 0.50), (F_4, 0.50)]$	$C_2 = [(F_4, 0.50), (F_1, 1.80)]$	$C_1 = [(F_3, 1.70), (F_1, 1.80)]$

CANDIDATESORTING($\mathbb{S}$)

foreach $C \in \mathbb{S}$
    SORT (C)
foreach $i = |C| : 1$
    STABLESORT($\mathbb{S}, i$)

Fig. 5. The CANDIDATESORTING procedure

the label of this contradiction is the same as that of the special "false" node. By deliberately keeping all the duplications, six candidates are generated as illustrated in the first column of Tab. I. The second column of this table is the result of sorting all the components of each candidate. Because the cardinality of these candidates are 2, the stable sorting subroutine will be invoked twice. The third column is the result of the first calling which is carried out by comparing the second elements of these candidates. The last column is the result of the second calling (and also the final result) achieved by comparing the first elements of all the candidates based on the previous sorting result. After removing all element duplications in each candidate and all the candidate duplications, there are three candidates remaining, which (in ascending order of priority) are: $C_6 = [(F_4, 0.50)]$, $C_3 = [(F_2, 1.40), (F_3, 1.70)]$, $C_1 = [(F_3, 1.70), (F_1, 1.80)]$.

Having generated the set of prioritized candidates, the next step is to correct a single candidate in order to remove all the contradictions and thus restore system consistency. By the ranking procedure, the higher priority a candidate has, the more likely the candidate is to be the real culprit. Therefore, the tentative modification always starts from the candidate with the highest priority. If the tentative modification succeeds, the algorithm terminates. Otherwise, the candidate will be removed from the candidate set and the candidate with the next highest priority in the current set will be tried, and so on.

IV. AN ILLUSTRATIVE EXAMPLE

To illustrate the potential of this extended adaptive approach, the problem given in [15] is reconsidered. The rule base is given as follows:
R_1: If x_1 is A_{11}, then x_2 is A_{21};
R_2: If x_1 is A_{12}, then x_2 is A_{22};
R_3: If x_2 is A_{23}, then x_3 is A_{31};
R_4: If x_2 is A_{24}, then x_3 is A_{32};
R_5: If x_2 is A_{25}, then x_4 is A_{41};
R_6: If x_2 is A_{26}, then x_4 is A_{42};
R_7: If x_3 is A_{33}, then x_5 is A_{51};
R_8: If x_3 is A_{34}, then x_5 is A_{52};
R_9: If x_4 is A_{43}, then x_5 is A_{53};
R_{10}: If x_4 is A_{44}, then x_5 isA_{54}.

Given $\beta_0 = 0.5$ and three observations, $x_1 = A_{13} = (7.0, 8.0, 9.0)$, $x_1 = A_{14} = (7.6, 8.6, 9.6)$ and $x_4 = A_{45} = (12.0, 13.0, 14.0)$, the interpolation procedures are illustrated in Fig. 6 and the original observations as well as interpolated results by scale and move transformation-based interpolation are presented in Fig. 7. Note that in this example, to keep the illustration simple, it has been assumed that the variable x, takes two different object values A_{13} and A_{14}. In practice, this may be caused by the fact that the observations are taken by different agents or that one is a real observation and the other may be produced by other inference mechanism.

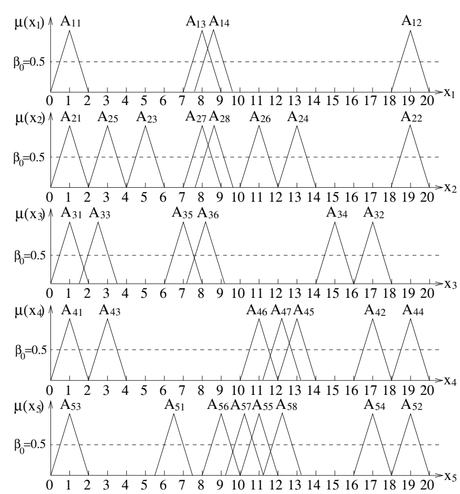

Fig. 7. Fuzzy sets used in the example

A. Certainty degree calculation for reasoning components

The certainty degree of each fuzzy reasoning component can be calculated by applying the approach obtained in Eq. 11. In particular, the distance between two fuzzy sets in this example is defined as the distance between their representative values. For instance, the certainty degree of fuzzy reasoning component R_1R_2 is calculated as follows:

$$\begin{aligned} c_{R_1R_2} &= 1 - \left| \frac{d(A_{11}, A_{12})}{max_1 - min_1} - \frac{d(A_{21}, A_{22})}{max_2 - min_2} \right| \\ &= 1 - \left| \frac{\text{Rep}(A_{12}) - \text{Rep}(A_{11})}{max_1 - min_1} - \frac{\text{Rep}(A_{22}) - \text{Rep}(A_{21})}{max_2 - min_2} \right| \\ &= 1 - \left| \frac{19-1}{20-0} - \frac{19-1}{20-0} \right| \\ &= 1.00, \end{aligned}$$

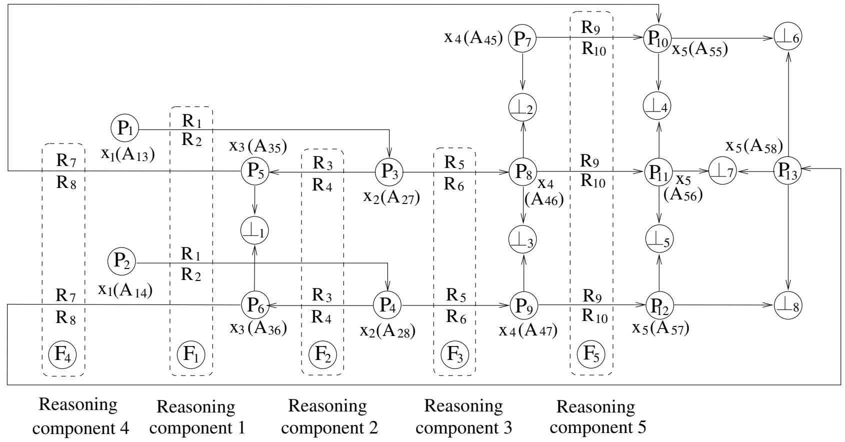

Fig. 6. Discrepancy records in ATMS

where $\mathrm{Rep}(A_{ij})$ denotes the representative value of fuzzy set A_{ij}, $i, j = 1, 2$. Similarly, the certainty degrees of all other fuzzy reasoning components can be calculated. The detailed calculation process is omitted here, but the results are: $c_{R_3R_4} = 0.60$, $c_{R_5R_6} = 0.60$, $c_{R_7R_8} = 1.00$, $c_{R_9R_{10}} = 1.00$.

B. Dependency recording by ATMS

In Fig. 6, an arrowed line which is flanked by two rules R_i and R_{i+1} represents a fuzzy reasoning component, which is denoted as R_iR_{i+1}, $i \in \{1, 3, 5, 7, 9\}$, where R_i and R_{i+1} are the neighboring rules used for interpolation. ATMS nodes and contradictions are represented by circles. Particularly, each $F_j,\ j \in \{1, 2, ..., 5\}$, is a node denoting a fuzzy reasoning component; each of $P_k,\ k \in \{1, 2, ..., 13\}$, is a node denoting a proposition; and each $\perp_l,\ l \in \{1, 2, ..., 8\}$, denotes a β_0-contradiction. These ATMS nodes and contradictions are listed as follows, with all justifications omitted:

$F_1 : \langle R_1R_2, 1.00, \{\{R_1R_2\}\}\rangle$;
$F_2 : \langle R_3R_4, 0.60, \{\{R_3R_4\}\}\rangle$;
$F_3 : \langle R_5R_6, 0.60, \{\{R_5R_6\}\}\rangle$;
$F_4 : \langle R_7R_8, 1.00, \{\{R_7R_8\}\}\rangle$;
$F_5 : \langle R_9R_{10}, 1.00, \{\{R_9R_{10}\}\}\rangle$;
$P_1 : \langle x_1 = A_{13}, 1.00, \{\{\}\}\rangle$;
$P_2 : \langle x_1 = A_{14}, 1.00, \{\{\}\}\rangle$;
$P_3 : \langle x_2 = A_{27}, 1.00, \{\{R_1R_2\}\}\rangle$;
$P_4 : \langle x_2 = A_{28}, 1.00, \{\{R_1R_2\}\}\rangle$;
$P_5 : \langle x_3 = A_{35}, 0.60, \{\{R_1R_2, R_3R_4\}\}\rangle$;
$P_6 : \langle x_3 = A_{36}, 0.60, \{\{R_1R_2, R_3R_4\}\}\rangle$;
$P_7 : \langle x_4 = A_{45}, 1.00, \{\{\}\}\rangle$;
$P_8 : \langle x_4 = A_{46}, 0.60, \{\{R_1R_2, R_5R_6\}\}\rangle$;
$P_9 : \langle x_4 = A_{47}, 0.60, \{\{R_1R_2, R_5R_6\}\}\rangle$;
$P_{10} : \langle x_5 = A_{55}, 1.00, \{\{R_1R_2, R_3R_4, R_7R_8\}, \{R_9R_{10}\}\}\rangle$;
$P_{11} : \langle x_5 = A_{56}, 0.60, \{\{R_1R_2, R_5R_6, R_9R_{10}\}\}\rangle$;
$P_{12} : \langle x_5 = A_{57}, 0.60, \{\{R_1R_2, R_5R_6, R_9R_{10}\}\}\rangle$;
$P_{13} : \langle x_5 = A_{58}, 0.60, \{\{R_1R_2, R_3R_4, R_7R_8\}\}\rangle$;
$\perp_1 : \langle \perp, \{\{R_1R_2, R_3R_4\}\}\rangle$;
$\perp_2 : \langle \perp, \{\{R_1R_2, R_5R_6\}\}\rangle$;
$\perp_3 : \langle \perp, \{\{R_1R_2, R_5R_6\}\}\rangle$;
$\perp_4 : \langle \perp, \{\{R_1R_2, R_5R_6, R_9R_{10}\}\}\rangle$;
$\perp_5 : \langle \perp, \{\{R_1R_2, R_5R_6, R_9R_{10}\}\}\rangle$;
$\perp_6 : \langle \perp, \{\{R_1R_2, R_3R_4, R_7R_8\}\}\rangle$;
$\perp_7 : \langle \perp, \{\{R_1R_2, R_3R_4, R_5R_6, R_7R_8, R_9R_{10}\}\}\rangle$;
$\perp_8 : \langle \perp, \{\{R_1R_2, R_3R_4, R_5R_6, R_7R_8, R_9R_{10}\}\}\rangle$.

Note that a certainty degree is attached to each fuzzy reasoning component and proposition, which is calculated and then recorded within the ATMS network, except the certainty degrees of any contradictions which are irrelevant to the present work and hence omitted. Also, as stated previously, the certainty degree of each observation is assumed to be 1 in this research. The certainty degree of any derived node can be calculated by Eq. 17. Within this illustrative example, maximum and algebraic product are used to implement $\oplus$ and $\otimes$ of Eq. 17, respectively. For example, the certainty degree of proposition P_{10} is computed by:

$$\begin{aligned} c_{P_{10}} &= \max\{c_{R_1R_2} \times c_{R_3R_4} \times c_{R_7R_8}, c_{R_9R_{10}}\} \\ &= \max\{1.00 \times 0.60 \times 1.00, 1.00\} \\ &= 1.00. \end{aligned}$$

A specific ATMS node "false", denoted by $P_\perp$, which collectively represents all the contradictions listed above from $\perp_1$ to $\perp_8$, is given as follows:
$P_\perp : \langle \perp, \{\{R_1R_2, R_3R_4\}, \{R_1R_2, R_5R_6\}\}\rangle$.

C. Candidate generation and prioritization

It is recorded in the ATMS that the label environment $\{R_1R_2, R_3R_4\}$ entails contradiction $\perp_1$, which is derived from propositions P_5 and P_6, while P_5 and P_6 are inferred through the same path. A ranking value is computed and attached to each component of this label using the procedure of Fig. 4. Similarly, each component in label environment $\{R_1R_2, R_5R_6\}$ is also attached by a ranking value by the same approach. The result is given as follows:

$$\{(R_1R_2, 1.00), (R_3R_4, 0.60)\};$$
$$\{(R_1R_2, 1.00), (R_5R_6, 0.60)\}.$$

From this, by deliberately keeping all the duplications, four component candidates are generated as follows:

$$
\begin{aligned}
C_1 &= [(R_1R_2, 1.00), (R_1R_2, 1.00)];\\
C_2 &= [(R_1R_2, 1.00), (R_5R_6, 0.60)];\\
C_3 &= [(R_1R_2, 1.00), (R_3R_4, 0.60)];\\
C_4 &= [(R_3R_4, 0.60), (R_5R_6, 0.60)].
\end{aligned}
$$

Because these four candidates have the same cardinality size, they can be prioritized following the algorithm of Fig. 4. The prioritized result is: C_4, C_2, C_3, C_1, in a descending order. After removing the duplicate candidate component R_1R_2 in component C_1, and then removing supersets C_2 and C_3 (of C_1), the final minimal candidate set consists of two candidates C_1' and C_4':

$$
\begin{aligned}
C_1' &= [(R_1R_2, 1.00)];\\
C_4' &= [(R_3R_4, 0.60), (R_5R_6, 0.60)],
\end{aligned}
$$

where C_4' is of a higher priority.

D. Candidate modification

The candidate with the highest priority, i.e. candidate C_4' in this example, needs to be modified first according to the consistency restoring algorithm given in Fig. 2. Following the modification procedure for single candidates as outlined in Sec. II, a set of simultaneous equations and inequations can be calculated. One of the solutions resulting from solving these equations and inequations simultaneously is illustrated in Fig. 8. It is clear from this figure that there is no β_0-contradiction any more and thus consistency has been restored. This means that the original inconsistent interpolation process has been corrected with consistent interpolated results throughout. Note that the other candidate, C_1' cannot lead to any consistent solution although it has a smaller cardinality size, which shows the potential of the proposed work.

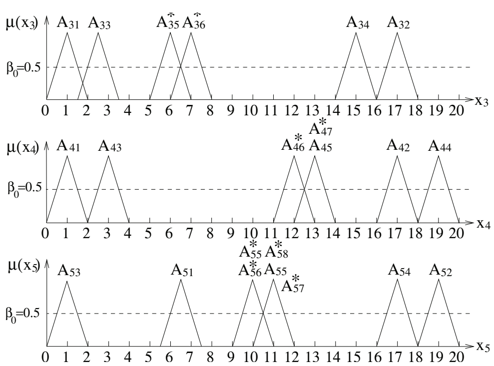

Fig. 8. The solution of the example

V. Conclusions

This paper has extended the recent work on adaptive fuzzy interpolation. It has introduced the concept of the certainty degree of a fuzzy reasoning component and that of the certainty degree of a derived proposition. From this, the classical ATMS has been extended not only to record the dependencies of a node but also to record to what degree the node can be derived from their supporting environments. This is followed by a modified GDE, which is able to prioritize all the generated minimal candidates by exploiting certainty degrees. The actual modification procedure, the same as the one described in [15], takes place, carrying out tentative modification to each minimal candidate of the highest priority until a solution is found. Whilst the working of this method has been illustrated with a practically significant example, real-world applications remain as further work.

Other improvements may enhance the potential of the proposed approach. Currently, the extended approach is only applicable to fuzzy interpolation with two single antecedent rules. It is worthwhile to generalize the approach for fuzzy interpolation with multiple antecedent rules and fuzzy extrapolation. Also, all the rules given in the initial rule base are assumed to be true and fixed. However, this may not be the case in certain real-world situations, despite that it is a common assumption made in the literature of interpolative reasoning. Thus, it is interesting to consider extending the proposed work to allow rules in the given rule base to become themselves diagnosable and modifiable. The initial investigation into such issue has been reported in [16].

References

[1] P. Baranyi, L. T. Kóczy, and T. D. Gedeon, "A generalized concept for fuzzy rule interpolation," *IEEE T. Fuzzy Syst.*, vol. 12, no. 6, pp. 820–837, 2004.

[2] Y.-C. Chang, S.-M. Chen, and C.-J. Liau, "Fuzzy interpolative reasoning for sparse fuzzy-rule-based systems based on the areas of fuzzy sets," *IEEE T. Fuzzy Syst.*, vol. 16, no. 5, pp. 1285–1301, Oct. 2008.

[3] S.-M. Chen, Y.-K. Ko, Y.-C. Chang, and J.-S. Pan, "Weighted fuzzy interpolative reasoning based on weighted increment transformation and weighted ratio transformation techniques," *IEEE T. Fuzzy Syst.*, vol. 17, no. 6, pp. 1412 –1427, 2009.

[4] S.-M. Chen and Y.-K. Ko, "Fuzzy interpolative reasoning for sparse fuzzy rule-based systems based on α-cuts and transformations techniques," *IEEE T. Fuzzy Syst.*, vol. 16, no. 6, pp. 1626–1648, Dec. 2008.

[5] J. de Kleer, "An assumption-based TMS," *Artif. Intell.*, vol. 28, no. 2, pp. 127–162, 1986.

[6] J. de Kleer and B. C. Williams, "Diagnosing multiple faults," *Artif. Intell.*, vol. 32, no. 1, pp. 97–130, 1987.

[7] Z. Huang and Q. Shen, "Fuzzy interpolative reasoning via scale and move transformations," *IEEE T. Fuzzy Syst.*, vol. 14, no. 2, pp. 340–359, 2006.

[8] ——, "Fuzzy interpolation and extrapolation: a practical approach," *IEEE T. Fuzzy Syst.*, vol. 16, no. 1, pp. 13–28, 2008.

[9] L. T. Kóczy and K. Hirota, "Approximate reasoning by linear rule interpolation and general approximation," *Int. J. Approx. Reason.*, vol. 9, no. 3, pp. 197–225, 1993.

[10] ——, "Size reduction by interpolation in fuzzy rule bases," *IEEE T. Syst., Man Cybern.*, vol. 27, no. 1, pp. 14–25, 1997.

[11] D. Tikk and P. Baranyi, "Comprehensive analysis of a new fuzzy rule interpolation method," *IEEE T. Fuzzy Syst.*, vol. 8, no. 3, pp. 281–296, Jun 2000.

[12] K. W. Wong, D. Tikk, T. Gedeon, and L. Kóczy, "Fuzzy rule interpolation for multidimensional input spaces with applications: a case study," *IEEE T. Fuzzy Syst.*, vol. 13, no. 6, pp. 809–819, Dec. 2005.

[13] Y. Yam and L. Kóczy, "Representing membership functions as points in high-dimensional spaces for fuzzy interpolation and extrapolation," *IEEE T. Fuzzy Syst.*, vol. 8, no. 6, pp. 761–772, Dec 2000.

[14] Y. Yam, M. L. Wong, and P. Baranyi, "Interpolation with function space representation of membership functions," *IEEE T. Fuzzy Syst.*, vol. 14, no. 3, pp. 398–411, June 2006.

[15] L. Yang and Q. Shen, "Towards adaptive interpolative reasoning," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2009, pp. 542–549.

[16] ——, "Adaptive fuzzy interpolaion with uncertain observations and rule base," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2011.