A Method for Resolving the Consistency Problem Between Rule_Based and Quantitative Knowledge using Fuzzy Simulation

Gyooseok Kim, Paul A. Fishwick
Computer & Information Science and Engineering Department
University of Florida

Abstract

Given a physical system, there are *experts* who have knowledge about how this system operates. In some cases, there exists quantitative knowledge in the form of *deep models* for the identical system. One of the main issues dealing with these different types of knowledge is "how does one address the difference between the two model types, each of which represents a different *level* of knowledge about the system?" We have devised a method that starts with 1) the expert's knowledge about the system, and 2) a quantitative model that can represent all or some of the behavior of the system. This method then adjusts the *knowledge* in either the rule-based system or the quantitative system to achieve some degree of *consistency* between the two representations. Through checking and resolving the inconsistencies, we provide a way to obtain better models in general about systems by exploiting knowledge at all levels, whether qualitative or quantitative.

1 Introduction

Given a physical system, knowledge about the system is often obtained from experts in the form of rules. Although the rule-based model is occasionally associative or shallow in nature, this model can easily capture human heuristic and problem solving knowledge in an efficient way [2, 11, 22]. In some cases, there exists a quantitative model which represents all or part of the behaviors of the physical system. This model provides deeper and more theoretical knowledge when expert system developers want to find solutions for technical problems [11, 7, 22]. Assuming the above two different model types for the identical system, some important questions can arise: *how much do the models differ?* and *how can one resolve the inconsistencies?*. Answering to these types of questions is not easy since there is a big gap between expert rules and quantitative models as shown in Fig 1.

One way of handling the inconsistencies between the expert's level of qualitative knowledge and the lower level of deep knowledge is to form a *knowledge acquisition cycle* as in Fig 2 [11]. Approaches for creating model bases are discussed within the context of computer simulation [9]. For example, the model base represents compiled knowledge about many domains such as the mathematical queuing model for waiting line problems. If a match is found, then shallow rules are generated by means of qualitative or quantitative simulation based on this deep model. Since the number of the shallow rules resulting from the deduction process is usually too big for study and validation against the original expert's rules, induction methods can be employed to obtain a more comprehensive and general set of rules [11].

For domains in which the expert rules contain many linguistic terminologies whose boundaries are not exact, we need a way to encode this vagueness to permit the use of computer

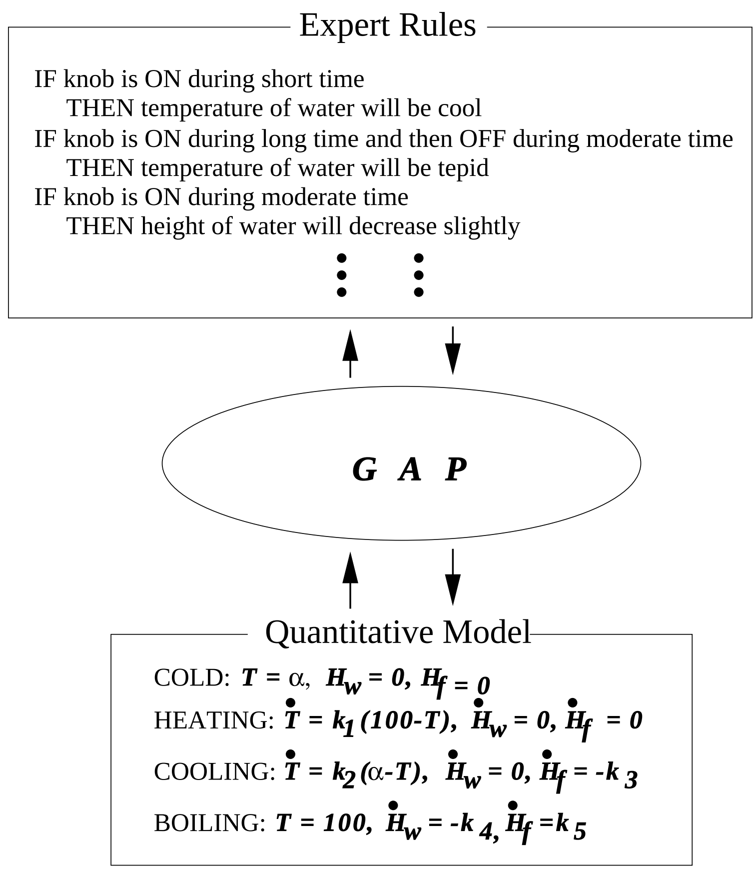


Figure 1: A gap between expert rules and quantitative models

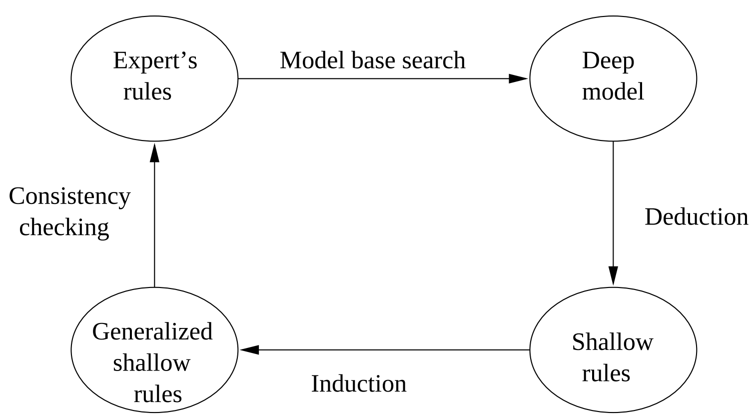


Figure 2: A knowledge acquisition cycle

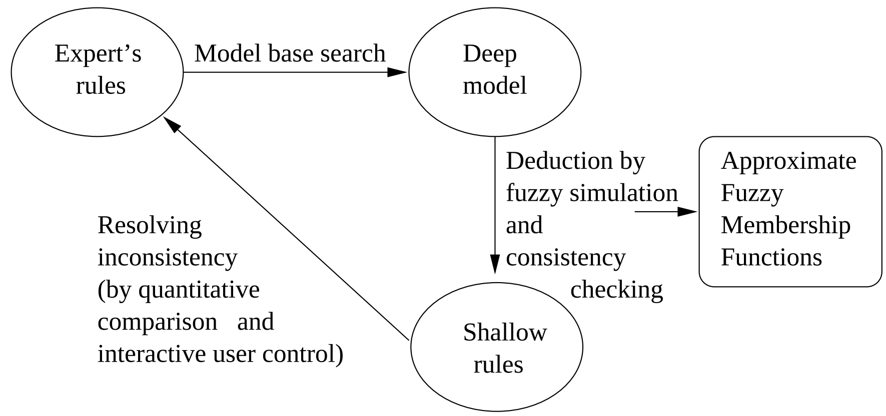


Figure 3: A knowledge acquisition cycle using fuzzy simulation

simulation. For such cases, we can use either *qualitative* or *quantitative* simulation with fuzzy set concepts [1, 18, 8] for the deduction process. However, a well known problem in using the *qualitative* methods is the possibly generating of spurious behaviors of the system during the reasoning process [12, 11]. Moreover, in order to get a compressed and generalized set of fuzzy rules, additional methods such as fuzzy induction or fuzzy system identification methods [15, 24, 17, 23] should be adopted. Consequently, forming the above knowledge acquisition cycle to employ fuzzy set concepts requires a series of difficult tasks.

Our research is concerned with devising a method for resolving the inconsistency between two different levels of knowledge in efficient and systematic manner. To achieve this goal, we have partially developed a method as shown in Fig 3. This method starts with the expert's rules about the system and a quantitative deep model that can represent all or some of the behavior of the system. This method then adjusts the *knowledge* in either the rule-based system or the quantitative system to achieve some degree of *consistency* between the two representations. Here, a *fuzzy simulation* approach [7, 6, 8] is used for directly encoding uncertainty arising from human linguistic vagueness into simulation components as well as for utilizing *quantitative* models for the deduction process. Since this method uses a linguistic mapping process to map simulation inputs and outputs into fuzzy linguistic values that were also used by experts, direct comparison is possible without an additional induction step.

The method consists of two phases: 1) *consistency checking phase*, and 2) *resolving phase*. In the *consistency checking phase*, experts provide various levels of estimates (i.e., central points, intervals, approximate fuzzy sets) for a fuzzy set and then, through fuzzy simulation and incremental optimization over the error surface, fuzzy set boundary vertices are created to *fill in* the expert's knowledge. Currently, an implementation has been made for this first phase where the estimates are presented in the form of *central points*. For quantitative comparison between the two levels of knowledge, quantitative measures have been formulated to gauge the sources and the degree of inconsistency. The final products of this stage are rules derived from quantitative models, approximate fuzzy membership functions for those rules, and the amount of inconsistency against the expert's rules. If the amount of inconsistency exceeds a reasonable range, the *resolving phase* is necessary. In this phase, human intervention is present: either expert rules (including the definitions of fuzzy numbers) or simulation model components are modified to reduce the amount of inconsistency. Even at this point, the quantitative measures mentioned above help them identify and revise the most inconsistent component rapidly and analyze the effectiveness of that modification, thereby allowing the two different levels of knowledge to gradually reach a consensus with high resolution. The knowledge acquisition cycle presented here forms a more potentially organized framework that resolves the inconsistency between two knowledge sources in an efficient and systematic manner.

The primary contribution of this research is that through checking consistency and resolving inconsistency, we provide a way to obtain better models in general about systems by exploiting knowledge at all levels, whether qualitative or quantitative. We provide benefits to expert systems from simulation and benefits to simulation modeling from expert knowledge [7, 6, 8, 13, 16, 3, 20, 21]. In particular, when expert system researchers are studying the acquisition of deep knowledge from an expert or validating the expert's knowledge against quantitatively compiled knowledge, the first type of benefits can be obtained from simulation models [7, 6, 8, 13, 16]. The advantage from the reverse direction is also obtained when simulation model validations are performed during the simulation modeling process with the aid of the expert knowledge [13, 3, 20, 21].

Searching for the appropriate fuzzy membership functions that adequately capture the meaning of the linguistic terms employed in a particular application belongs to the general problem area of *knowledge acquisition* within the underlying framework of fuzzy set theory [10]. Thus, using the proposed method, we deliver a secondary contribution from automatically generating the approximate forms of fuzzy membership functions (see Fig 3) with which expert rules and quantitative models match maximally in spite of the expert's uncertainty about the exact definitions of the fuzzy values in his linguistic rules. Moreover, we interact with the user of the knowledge acquisition tool (to be constructed) to permit manual control over fuzzy set boundaries. Once we obtain those approximate definitions, fuzzy simulations for all different combinations of the fuzzy values, defined by these definitions, give us more detailed rules as a hypothesis of the expert's knowledge.

Fuzzy set theory, which is relevant to this paper and its relation to computer simulation, is discussed in section 2. Then, in sections 3 and 4, we discuss research accomplished to date and preliminary results with an example. Finally, section 5 discusses our future research.

2 Background

2.1 Fuzzy set theory

The theory of *fuzzy sets* can be found in [25, 26, 10, 5]. Fuzzy sets may be viewed as an attempt to deal with a type of imprecision which arises when the boundaries of classes are not sharply defined. A fuzzy set A of a universe of discourse X is characterized by a membership function $\mu_A : X \to [0,1]$ which associates with each element x of X a number $\mu_A(x)$ in the interval [0, 1] which represents the grade of membership of x in A.

Definition 2.1: A fuzzy set A of the universe of discourse X is *convex* if and only if for all x_1, x_2 in X, $\mu_A(\lambda x_1 + (1-\lambda)x_2) \geq Min(\mu_A(x_1), \mu_A(x_2))$,
where $\lambda \in [0,1]$.

Definition 2.2: A fuzzy set A of the universe of discourse X is called a *normal* fuzzy set if $\exists x_i \in X, \mu_A(x_i) = 1$.

Definition 2.3: A *fuzzy number* is a fuzzy subset in the universe of discourse X that is both convex and normal.

To simplify the representation of fuzzy sets, a finite fuzzy subset, A, of X is expressed as $A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \ldots + \mu_A(x_n)/x_n$, or $A = \sum_{i=1}^{n} \mu_A(x_i)/x_i$, where $+$ sign denotes the union rather than an arithmetic sum.

If the fuzzy subset, A, is not finite, A may be represented in the form $A = \int_X \mu_A(x)/x$ in which the integral sign stands for the union of the fuzzy singletons $\mu_A(x)/x$.

Definition 2.4: The complement of A is denoted by $\overline{A}$ and is defined by

$$\overline{A} = \int_X (1 - \mu_A(x))/x. \tag{1}$$

The operation of complementation corresponds to negation.

Definition 2.5: The union of fuzzy sets A and B is denoted by $A \cup B$ and is defined by

$$A \cup B = \int_X (\mu_A(x) \vee \mu_B(x))/x. \tag{2}$$

where $\vee$ is the maximum operator.

Definition 2.6: The intersection of fuzzy set A and B is denoted by $A \cap B$ and is defined by

$$A \cap B = \int_X (\mu_A(x) \wedge \mu_B(x))/x. \tag{3}$$

where $\wedge$ is the minimum operator.

Let A and B represent two fuzzy numbers and let $\star$ denote any of the four basic arithmetic operations. Then we define fuzzy set, $A \star B$ on $\mathcal{R}$, where $\mathcal{R}$ is a set of all real numbers, as

$$\mu_{A\star B}(z) = max_{z=x\star y}(\mu_A(x) \wedge \mu_B(y)), \tag{4}$$

for all $z \in \mathcal{R}$. Thus, for example, if $A, B \subseteq \mathcal{R}$ are two fuzzy numbers with respective membership functions $\mu_A(x)$ and $\mu_B(y)$, then the four basic arithmetic operations, i.e., addition, subtraction, multiplication and division, give for each $x, y, z \in \mathcal{R}$ the following results:

$$\mu_{A+B}(z) = max_{z=x+y}(\mu_A(x) \wedge \mu_B(y)). \tag{5}$$

$$\mu_{A-B}(z) = max_{z=x-y}(\mu_A(x) \wedge \mu_B(y)). \tag{6}$$

$$\mu_{A\times B}(z) = max_{z=x\times y}(\mu_A(x) \wedge \mu_B(y)). \tag{7}$$

$$\mu_{A\div B}(z) = max_{z=x\div y}(\mu_A(x) \wedge \mu_B(y)). \tag{8}$$

2.2 Fuzzy set theory in computer simulation

Probability based methods are useful when most of the uncertainty can be effectively described through the use of large data sets and their associated moments. However, experts often do not think in probability values, but in terms such as *much*, *usually*, *always* and *sometimes*. In domains where estimation or measurement of probabilities is not amenable, fuzzy set theory offers an alternative [14]. Here, we can use any type of fuzzy number, such as an interval-valued fuzzy number, a triangular fuzzy number, a trapezoidal fuzzy number or a general discrete (or continuous) fuzzy number depending on the degree of uncertainty. Owing to the *extension principle* [27] in the fuzzy set theory, nonfuzzy mathematical structures can be made fuzzy. Here is a sample of how this relates to simulation. We can make fuzzy: [8, 26] 1) a state variable value including initial conditions, 2) parameter values, 3) inputs and outputs, 4) model structures, and 5) algorithmic structures.

To simulate mathematical models using the fuzzy set concept, three kinds of fuzzy simulation approaches have been reported: *Qua*litative *Si*mulators (i.e., *Qua.Si* [1]), *Fu*zzy Qualitative *Sim*ulation (i.e., *Fusim* [18]), and three methods (*Monte Carlo, Uncorrelated Uncertainty,* and *Correlated Uncertainty*) of fuzzy simulation introduced by Fishwick [8]. While the first two kinds of fuzzy simulation are useful when there is not enough information to simulate quantitatively, the third kind takes linguistic information from the expert and performs computer simulation *quantitatively* on continuous and discrete event models. Rules or FSA (Finite State Automata) can be extracted from these quantitative models through linguistic mappings, and these results can be validated directly against the expert domain knowledge. The fuzzy simulation method we present is an extension version of the *correlated uncertainty method* [7, 6, 8]. The *correlated uncertainty method* assumes that all errors or uncertainties over time are correlated. In many circumstances, this is the case, since uncertainties specified by a heuristic or a belief are often correlated in that humans are often consistent in their beliefs. For such a process, every vertex in the fuzzy number is issued independently to the simulation function and the outputs of the simulation are mapped into the most closely matched fuzzy linguistic value by using a *distance metric*.

3 A Proposed Method

In this section, we propose a method for resolving the inconsistencies between the expert's rules and the quantitative models. As we discussed, our method consists of two phases: *consistency checking* and *resolving inconsistency*. While the first phase is done through an automatic process, the second phase is performed semi-automatically. An algorithm has been developed for the first phase. Before exploring the algorithm, we must first introduce the input of the algorithm and two important usages of fuzzy simulation that we've devised.

3.1 Format of expert rules as input of proposed method

In what follows, we assume that the format of expert rules is one of the following two types. The input of the proposed method is a collection of the expert's rules below, with conclusions from the *same* fuzzy variable:

- IF (χ is A_1) THEN (Υ is B); CF; CL_{A_1}; CL_B
- IF (χ_1 is A_1) OP (χ_2 is A_2) OP, ..., OP (χ_n is A_n)
  THEN (Υ is B); CF; CL_{A_1}; CL_{A_2}; ...; CL_{A_n}; CL_B

where
$\chi_i, i = 1, 2, \ldots, n$, and Υ are *fuzzy variables* that take real numbers from some universal set

Table 1: Notation

Notation	Usage
$MF_{premise}$	*Membership Function* of fuzzy value in rule *premise.*
MF_{conseq}	*Membership Function* of fuzzy value in rule *consequence.*
$RULE_{simplex}$	Expert's *simplex rule.*
$RULE_{compound}$	Expert's *compound rule.*
CF_{expert}	*Confidence Factor* presented by an *expert.*
CF_{fuzzy}	*Confidence Factor* calculated using *fuzzy simulation.*

X, Y respectively, A_i, $i = 1, 2, \ldots, n$, and B are *fuzzy values* on X, Y respectively, CF is a *confidence factor* in the rule consequence given that the premise conditions are satisfied, OP is a *fuzzy logic* (*or* or *and*) or *fuzzy arithmetic* (+, -, $\times$ or $\div$) operator, and CL_{A_i}, $i = 1, 2, \ldots, n$, and CL_B are expert's *confidence levels* on the fuzzy values in each rule.

The two types of rules above will be called *complex rules* and *compound rules* respectively. The value of CL can be a central point estimate, an interval estimate, an approximate fuzzy number or a complete fuzzy number depending on the expert's confidence level on the linguistic term he used. In this paper, we restrict our discussion within a situation where the values of the CL are central point estimates.

3.2 Two fuzzy simulation methods

In what follows, the notation in Table 1 will be used for simplicity. In the proposed method, the fuzzy simulation approach has two important roles: 1) calculation of CF_{fuzzy} of $RULE_{compound}$, and 2) estimation of MF_{conseq}. We discuss these two roles of the fuzzy simulation in the following two sections.

3.2.1 Calculation of CF_{fuzzy} of $RULE_{compound}$

Since the uncertainty arising from the human reasoning process is easily represented by a rule associated with CF_{expert}, we introduced a way for emulating such a process by showing how fuzzy simulation can derive the confidence factors from quantitative models. By doing this, we benefit from the comparison of the two rules in terms of their CF values. However, since the CF_{expert} involves a subjective opinion, there is no theoretical formulation to derive the CF_{fuzzy} whose value is exactly the same as the CF_{expert}. Our solution is to define an equation in such a way that its results agree with *human intuition* as much as possible. We used a *weighted average method* to create such an intuition.

Let us define the CF_{fuzzy} using the *weighted average method*. Given a $RULE_{compound}$, let its two $MF_{premises}$ be A and B, where A and B are fuzzy subsets of a universe discourse X, and its MF_{conseq} be C, where C is a fuzzy subset of a universe discourse Y. Then we define the CF_{fuzzy} by the following equation:

$$CF_{fuzzy} = \frac{\sum_{j=1}^{n}(\mu_{A\odot B}(x_j) \times \mu_C(y_j))}{\sum_{j=1}^{n} \mu_{A\odot B}(x_j)}, \tag{9}$$

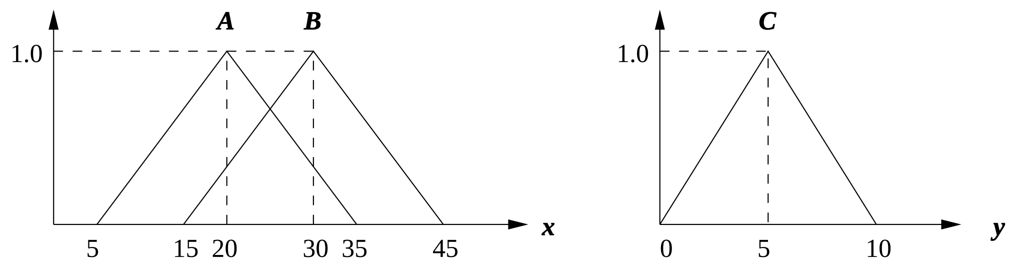


Figure 4: Definitions of fuzzy numbers A, B and C

where $\odot$ denotes a fuzzy logic or arithmetic operator,
$x_j, j = 1, 2, \ldots, n$, denote real values on the fuzzy set resulted from the operation of $A \odot B$,
$y_j, j = 1, 2, \ldots, n$, denote real values on Y obtained from fuzzy simulation using x_j.

Equation (9) can be divided into the following three steps for simplifying its calculation: 1) perform the fuzzy logic/arithmetic operation, 2) simulate using the fuzzy set obtained from the above step, and 3) calculate CF_{fuzzy} using the weighted average method. For example, given a $RULE_{compound}$, IF χ_1 is A OR χ_2 is B THEN Υ is C, with definitions of A, B and C as shown in Fig 4, CF_{fuzzy} for the $RULE_{compound}$ can be calculated by performing the following steps:

1. Perform the fuzzy OR operation for A and B. For each element x in X, the degree of membership of A OR B, $\mu_{\text{A OR B}}(x)$, is obtained by (2). Fig 5(a) shows the result of the operation.

2. Perform the fuzzy simulation on the fuzzy set of Fig 5(a). The result is shown at Fig 5(b).

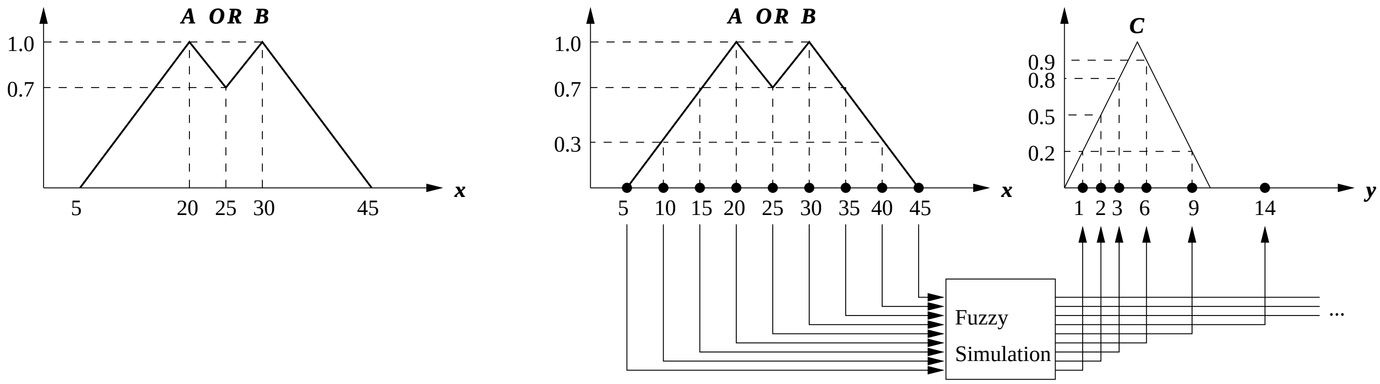
3. Calculate CF_{fuzzy} using the weighted average method.

$$\begin{aligned} CF_{fuzzy} &= \frac{(0.3 \times 0.5) + (0.7 \times 0.8) + (1.0 \times 0.9) + (0.7 \times 0.2)}{0.3 + 0.7 + 1.0 + 0.7 + 1.0 + 0.7 + 0.3} \\ &= 0.37 \end{aligned}$$

The validity of calculating CF_{fuzzy} in this way can be easily shown as in Fig 6. CF_{fuzzy}, using (9), is 1.0 and 0.0 for Fig 6(a) and for Fig 6(b) respectively. The results exactly match our intuition. When the CF falls into some range between the two extreme cases above, we can intuitively say that each member in A supports the conclusion B with a higher confidence, the greater CF we get. Using (9), we also get the results which support such an intuition.

3.2.2 Estimation of MF_{conseq}

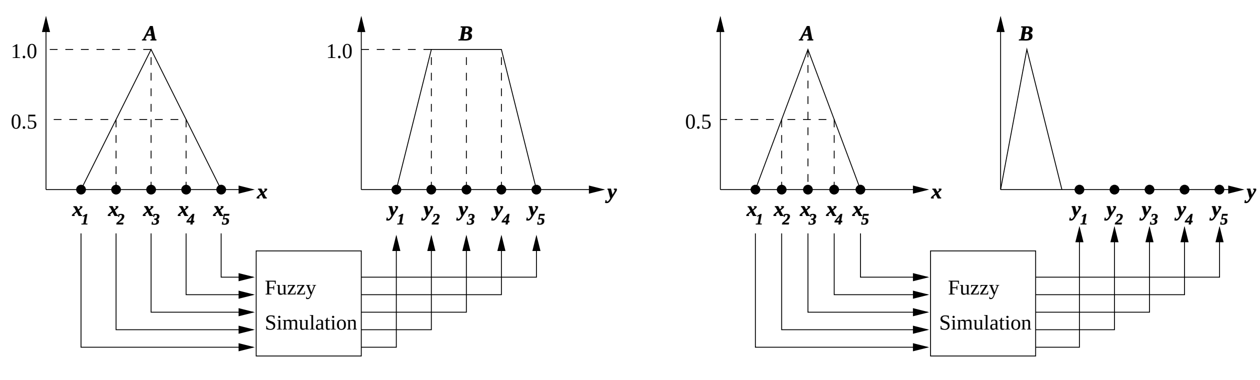
In the previous section, we used a fuzzy simulation to derive the CF_{fuzzy} when all definitions of the linguistic terms in a rule are already known. Conversely, without knowing the definition of the linguistic term, particularly the definition of the linguistic term in the consequence of the rule, we can use the fuzzy simulation to *estimate* its approximate range.



(a) The result of operation ***A OR B***

(b) The result of fuzzy simulation on fuzzy set ***A OR B***

Figure 5: Calculation of CF_{fuzzy} as an example



(a) All members of ***A*** support the conclusion ***B***

(b) None of the members of ***A*** support the conclusion ***B***

Figure 6: All members or none of members support the conclusion

Let's assume B is a symmetric triangular fuzzy number whose members are real numbers y. Knowing its center point c and the width w of B, the degree of membership of any real number, $y_1, y_2, \ldots, y_m$ can be obtained from the equation,

$$\mu_B(y_i) = 1 - \frac{2 \times |y_i - c|}{w}, \tag{10}$$

where $i = 1, 2, \ldots, m$.

Let's assume another fuzzy number A whose members are real numbers x. Given an expert rule, IF χ is A THEN Υ is B, with its CF_{expert}, performing a fuzzy simulation on A and applying the weighted average method to B yields

$$CF_{fuzzy} = \frac{\sum_{i=1}^{n}(\mu_A(x_i) \times \mu_B(y_i))}{\sum_{i=1}^{n} \mu_A(x_i)}, \tag{11}$$

where y_i is a result of the fuzzy simulation on x_i.

However, consider a situation where a fuzzy simulation is executed on A, but the width of B is unknown. Letting the CF_{fuzzy} in (11) be equal to the CF_{expert} of the rule above, and substituting the right-hand side of (10) for the $\mu_B(y_i)$ in (11), we get

$$CF_{expert} = \frac{\sum_{i=1}^{n}(\mu_A(x_i) \times (1 - \frac{2 \times |y_i - c|}{w}))}{\sum_{i=1}^{n} \mu_A(x_i)} \tag{12}$$

From this equation, we can obtain the following equation to estimate the unknown width w of B.

$$w = \frac{2 \times \sum_{i=1}^{n}(\mu_A(x_i) \times |y_i - c|)}{\sum_{i=1}^{n} \mu_A(x_i) - (CF_{expert} \times \sum_{i=1}^{n} \mu_A(x_i))} \tag{13}$$

Equation (13) has an important meaning: if we know an expert rule, its CF_{expert}, its $MF_{premise}$, and the center point of the fuzzy number in the consequence of the rule, then we can estimate the range of the fuzzy number with an aid of fuzzy simulation.

For example, with the rule, IF χ is A THEN Υ is B; $CF_{expert} = 0.5$; $CL_A = 0.03$; $CL_B = 27.0$, and unknown width w of B, suppose that the result of a fuzzy simulation is shown in Fig 7(a). By applying (13), a symmetric triangular membership function for B can be obtained as shown in Fig 7(b).

One constraint of applying (13) is that the CF_{expert} should not be equal to 1.0 (i.e., less than 1.0). Otherwise, the value of the denominator in (13) would be zero. Even though such a case is currently a limitation of the equation, a preliminary approach has been developed.

3.3 Consistency algorithm

An algorithm introduced here has been made for handling the first phase (i.e., *checking consistency phase*) of the proposed method. Once the expert's rules for a physical system have been presented, and a relevant quantitative model has been found during the model base search, we can apply the algorithm presented here for checking consistency between the two models of knowledge. The algorithm generates the approximate definitions of fuzzy linguistic values by increasing the ranges of fuzzy sets from their initial minimal width to *fill in* the expert knowledge. For such a process, two confidence factors, CF_{expert} and CF_{fuzzy},

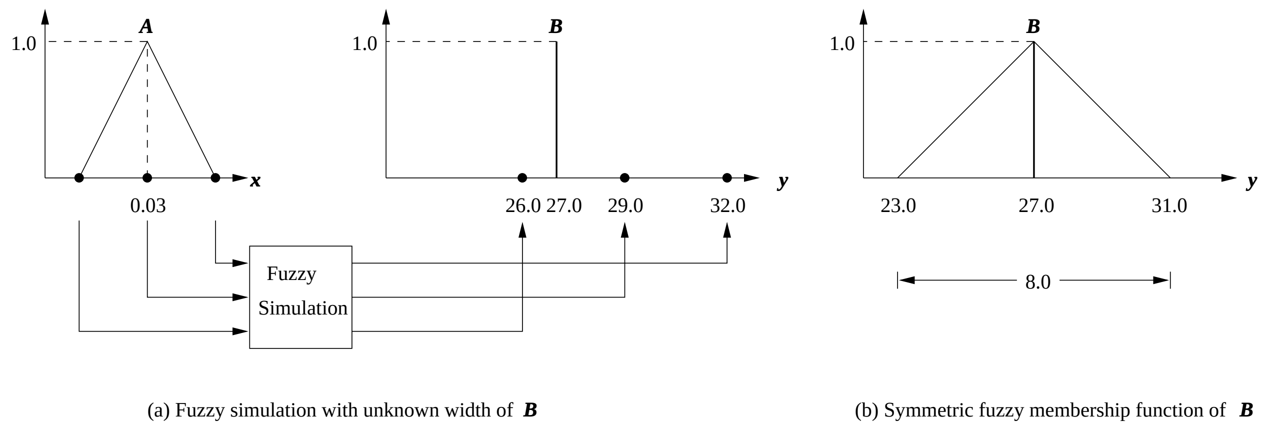


Figure 7: Estimation of unknown width of B using fuzzy simulation

are used to calculate local and global inconsistencies. These serve as the *quantitative closeness measures* between the two different levels of knowledge. Two different methods of fuzzy simulation, discussed in section 3.3.1 and 3.3.2., are involved in this process. When the algorithm reaches a point where tuning membership functions does not improve the amount of closeness any further, the algorithm stops and returns the membership functions that have been tuned so far as an approximate set with which two levels of knowledge *match maximally*. That is, the proposed algorithm uses a gradient_based optimization technique to find the proper sets of fuzzy definitions. If the closeness is out of a reasonable range, human intervention is required for resolving the inconsistencies: either the expert rules or the simulation components which show the inconsistency can be reevaluated, or the definitions of the linguistic values generated by the algorithm can be changed interactively. The algorithm presented here is also useful for this *resolving phase*, since the comparison results are quantitatively calculated and visualized in response to the human interaction. When the *goodness of fit* reaches a reasonable point, another fuzzy simulation with different values of fuzzy variables creates a more detailed level of rules than the level of the expert's rules.

For this algorithm, we employ an iterative improvement method. This algorithm consists of the following three basic steps:

1. Hypothesize membership functions in $RULE_{simplexs}$.

2. Apply hypothetical membership functions to $RULE_{compounds}$.

3. Improve hypothetical membership functions using $RULE_{compounds}$.

Fig 8 shows detailed substeps in each basic step. We explore them in the following three subsections.

3.3.1 Step 1: hypothesize membership functions in $RULE_{simplexs}$

The purpose of this step is to hypothesize each $MF_{premise}$ in $RULE_{simplex}$ and to obtain its corresponding hypothetical MF_{conseq} using (13).

In the first substep, two cases should be handled differently. That is, when the algorithm initially starts, we construct an initial hypothetical $MF_{premise}$ so that its range is $2\Delta d$ with

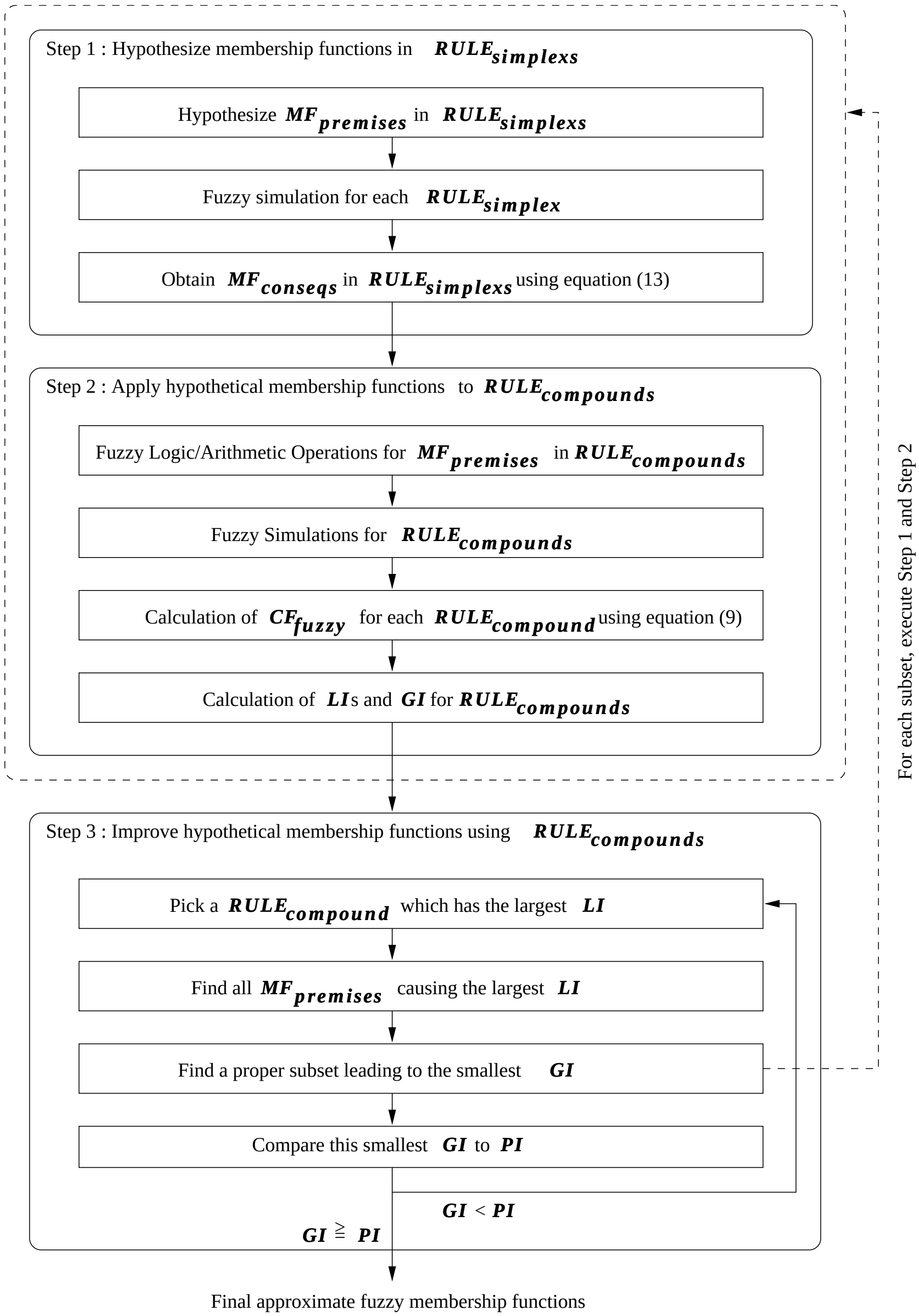


Figure 8: Three basic steps and their substeps of algorithm

the center point, where Δd is a optimal resolution size for simulation execution. Δd can be determined by experts or simulationists. For other case, this substep modifies $MF_{premise}$ by increasing its range by Δd on either sides. After executing the last substep, we obtain a hypothetical pair of $MF_{premise}$ and MF_{conseq} for each $RULE_{simplex}$ which satisfies $CF_{fuzzy} \approx CF_{expert}$.

3.3.2 Step 2 : apply hypothetical membership functions to $RULE_{compounds}$

The obtained $MF_{premises}$ and $MF_{conseqs}$ from the previous step are consistent only for the $RULE_{simplexs}$ in a sense that $CF_{fuzzy} \approx CF_{expert}$ for each $RULE_{simplex}$. Our claim is that if those membership functions are really consistent, then this also should be the case with the all $RULE_{compounds}$. Thus, the purpose of this step is to apply these hypothetical membership functions to the $RULE_{compounds}$ to check their validities.

For each $RULE_{compound}$, we define its *local inconsistency*, LI, as

$$LI = |CF_{fuzzy} - CF_{expert}|. \tag{14}$$

Then, using the LI, we define the *global inconsistency* for all $RULE_{compounds}$, GI, as

$$GI = \sum_{i=1}^{m} LI_i, \tag{15}$$

where $m =$ total number of $RULE_{compounds}$. Searching for the largest LI enables us to identify the most inconsistent $RULE_{compound}$ between two different knowledge sources. Moreover, the GI calculated in this way allows us to measure the total amount of inconsistency.

3.3.3 Step 3 : improve hypothetical membership functions using $RULE_{compounds}$

The purpose of this step is to reduce the GI by picking up a $RULE_{compound}$ which has the largest LI and modifying a proper subset of the $MF_{premises}$ among all subsets of $MF_{premises}$ which caused that LI. We can find the proper subset by searching for a combination of the $MF_{premises}$ which leads the GI to the smallest value among all combinations of the $MF_{premises}$ which caused the largest LI. Notice that we should not regard the $MF_{premises}$ that can reduce the largest LI into the smallest amount as the proper subset. The reason is that if any such $MF_{premise}$ is used also for other rules, then the modification of this definition could make other LIs in those rules worse than before, possibly causing increased GI as a whole. For this reason we introduce the GI instead of the LI as a *performance index* (PI). Therefore, we need to find a subset of $MF_{premises}$ which improves the GI by the greatest amount by executing *step 1* and *step 2* for each subset of $MF_{premises}$. When we eventually reach the smallest GI after incrementally reducing the inconsistencies, we can regard the hypothetical set of the $MF_{premises}$ and the $MF_{conseqs}$ as the final approximate fuzzy set with which the expert's rule-based model matches maximally the quantitative simulation model.

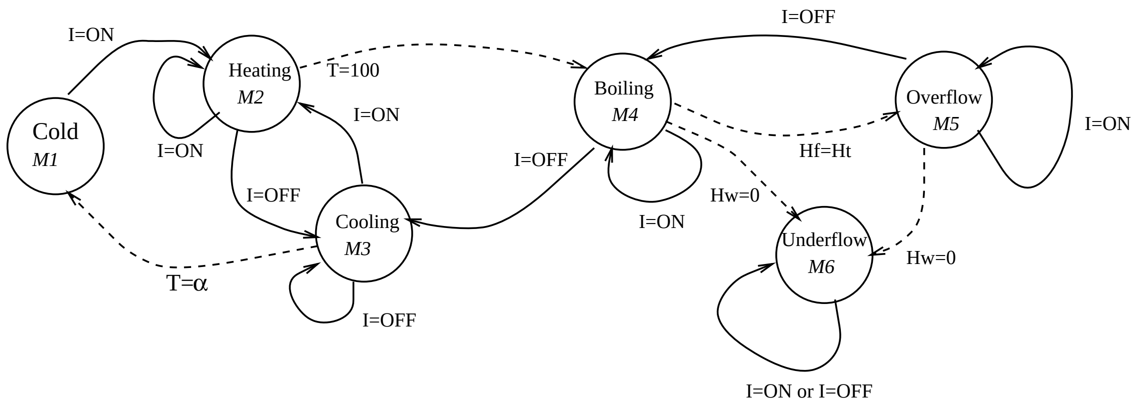


Figure 9: Six state automaton controller for boiling water

4 Example: Boiling Water Model

To illustrate the application of the proposed method, we have chosen *boiling water* [9] as a simple example. All steps in Fig 8 has been implemented using C programming language, and a graphical user interface using Tk/Tcl has been built for illustrating the result of the each step in the algorithm. Consider a pot of boiling water on a stovetop electric element. Initially, the pot is filled to some predetermined level with water. A small amount of detergent is added to simulate the forming activity that occurs naturally when boiling certain foods. This system has one input - the temperature knob. The knob is considered to be in one of two states:*on* or *off*.

The output consists of T (the temperature of the water over time), H_w (height of the water) and H_f (height of the form). The behavior of this model can be represented by six states of FSA as in Fig 9. The low level continuous models for $M_1, \ldots, M_6$ in that figure are defined as shown below by combining Newton's law with the capacitance law [9].

1. (M_1) COLD: $T = \alpha$, $\dot{H}_w = 0$, $\dot{H}_f = 0$.
2. (M_2) HEATING: $\dot{T} = k_1(100 - T)$, $\dot{H}_w = 0$, $\dot{H}_f = 0$.
3. (M_3) COOLING: $\dot{T} = k_2(\alpha - T)$, $\dot{H}_w = 0$, $\dot{H}_f = -k_3$.
4. (M_4) BOILING: $T = 100$, $\dot{H}_w = -k_4$, $\dot{H}_f = k_5$.
5. (M_5) OVERFLOW: same as BOILING with constraint $H_f = H_t$.

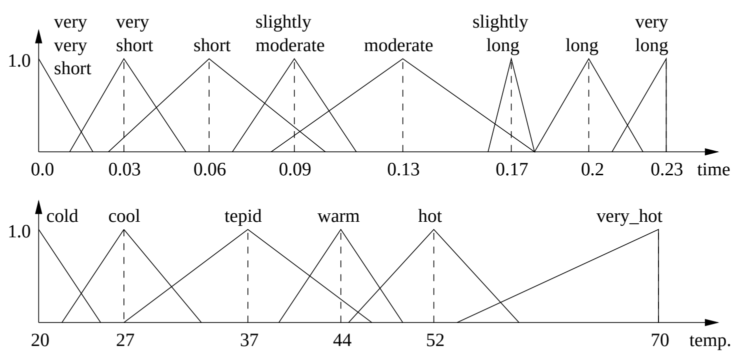
where $k_i, i = 1, \ldots, 5$ are rate constants and α is the ambient temperature of the water.

Among those three types of output (i.e., T, H_w and H_f), let's assume that we are particularly interested in the T (temperature of water) when we turn the knob *on* and *off* over some time period. We used a total of sixteen rules (eight $RULE_{simplexs}$ and eight $RULE_{compounds}$) and their center point estimates as shown in Fig 10(a) as the expert's rules to describe the water temperatures depending on the *on* and *off* position of the knob over time. Using these expert rules as inputs and applying the proposed method to the quantitative simulation model [9] in Fig 9, we obtained sixteen rules and an approximate set of fuzzy membership

PREMISES		CONSEQUENCES	CF_{expert}
KNOB_ON	KNOB_OFF	TEMPERATURE	
very_very_short		cold	0.8
very_short		cool	0.5
short		tepid	0.7
slightly_moderate		warm	0.8
moderate		hot	0.6
slightly_long		very_hot	0.3
long		very_hot	0.6
very_long		very_hot	0.8
slightly_moderate	very_very_short	warm	0.8
slightly_moderate	long	tepid	0.9
moderate	short	hot	0.5
short	very_long	cool	0.4
very_long	very_very_short	very_hot	0.8
moderate	very_long	tepid	0.6
very_short	very_long	cool	0.7
very_very_short	very_long	cold	0.9

PREMISES		CONSEQUENCES	CF_{fuzzy}
KNOB_ON	KNOB_OFF	TEMPERATURE	
very_very_short		cold	0.8
very_short		cool	0.5
short		tepid	0.7
slightly_moderate		warm	0.8
moderate		hot	0.6
slightly_long		very_hot	0.3
long		very_hot	0.6
very_long		very_hot	0.8
slightly_moderate	very_very_short	warm	0.766667
slightly_moderate	long	tepid	0.892000
moderate	short	hot	0.553164
short	very_long	cool	0.410000
very_long	very_very_short	very_hot	0.840909
moderate	very_long	tepid	0.684073
very_short	very_long	cool	0.700000
very_very_short	very_long	cold	0.900000

CENTER POINT ESTIMATE(CL)			
very_very_short	0.0	cold	20.0
very_short	0.03	cool	27.0
short	0.06	tepid	37.0
slightly_moderate	0.09	warm	44.0
moderate	0.13	hot	52.0
slightly_long	0.17	very_hot	70.0
long	0.2		
very_long	0.23		



(a) Expert's rules and center point estimates.

(b) Rules extracted from fuzzy simulations and final approximate fuzzy membership functions

Figure 10: The inputs presented by expert and the outputs generated by the proposed method

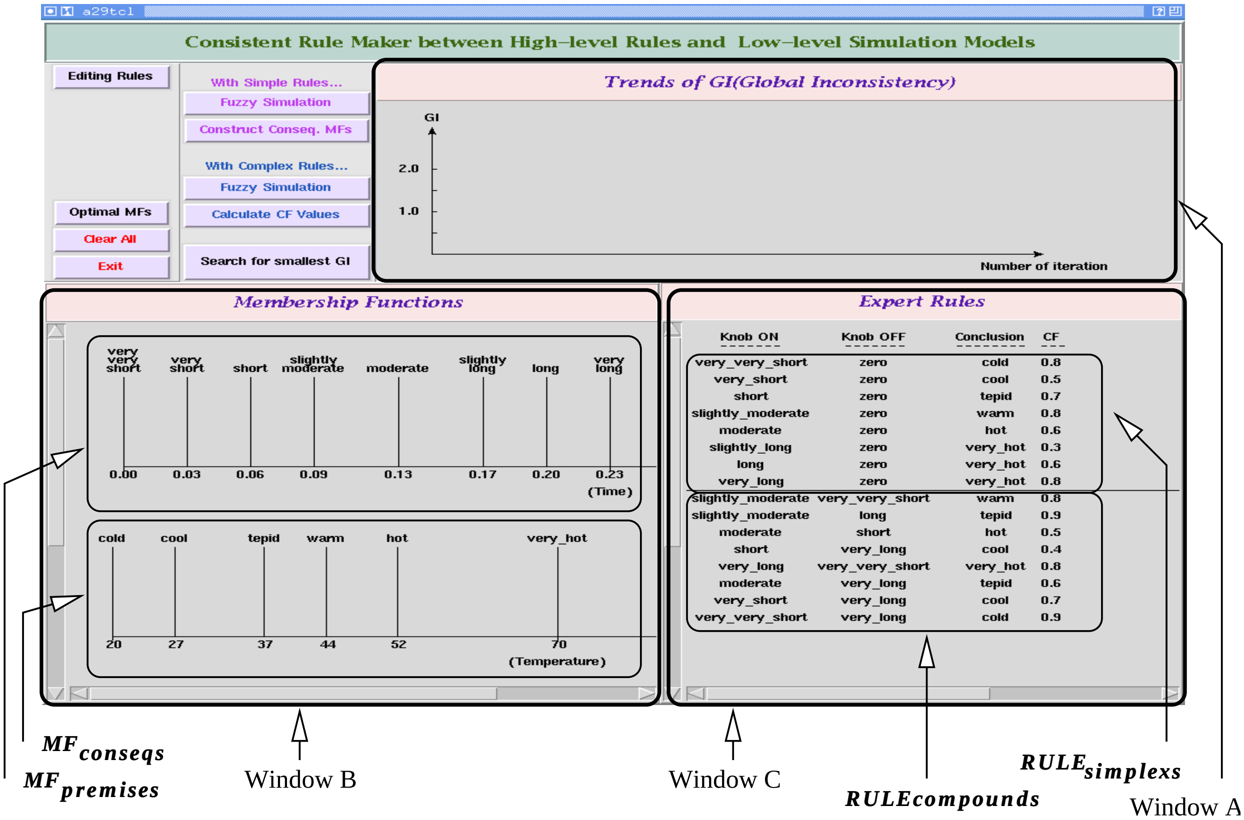


Figure 11: Initial GUI when expert's rules are first processed

functions for those rules as shown in Fig 10(b). In the remained part of this section, we will show how to obtain such results by illustrating all steps depicted in Fig 8.

Fig 11 shows the initial GUI when the expert rules are first processed. The GUI consists of the following three windows:

- Window A for displaying the trends of GI over the number of iterations of algorithm,
- Window B for displaying the membership functions made so far, where the upper and the lower parts represent $MF_{premises}$ and $MF_{conseqs}$ respectively,
- Window C for displaying expert rules, where the upper and the lower parts represent $RULE_{simplexs}$ and $RULE_{compounds}$ respectively.

4.1 Algorithm Execution

1. Hypothesize $MF_{premises}$ in $RULE_{simplexs}$
   The hypothetical $MF_{premises}$ with the resolution size of 0.01 is initially made, and corresponding $MF_{conseqs}$ are obtained as shown in Fig 12 by using equation (13).

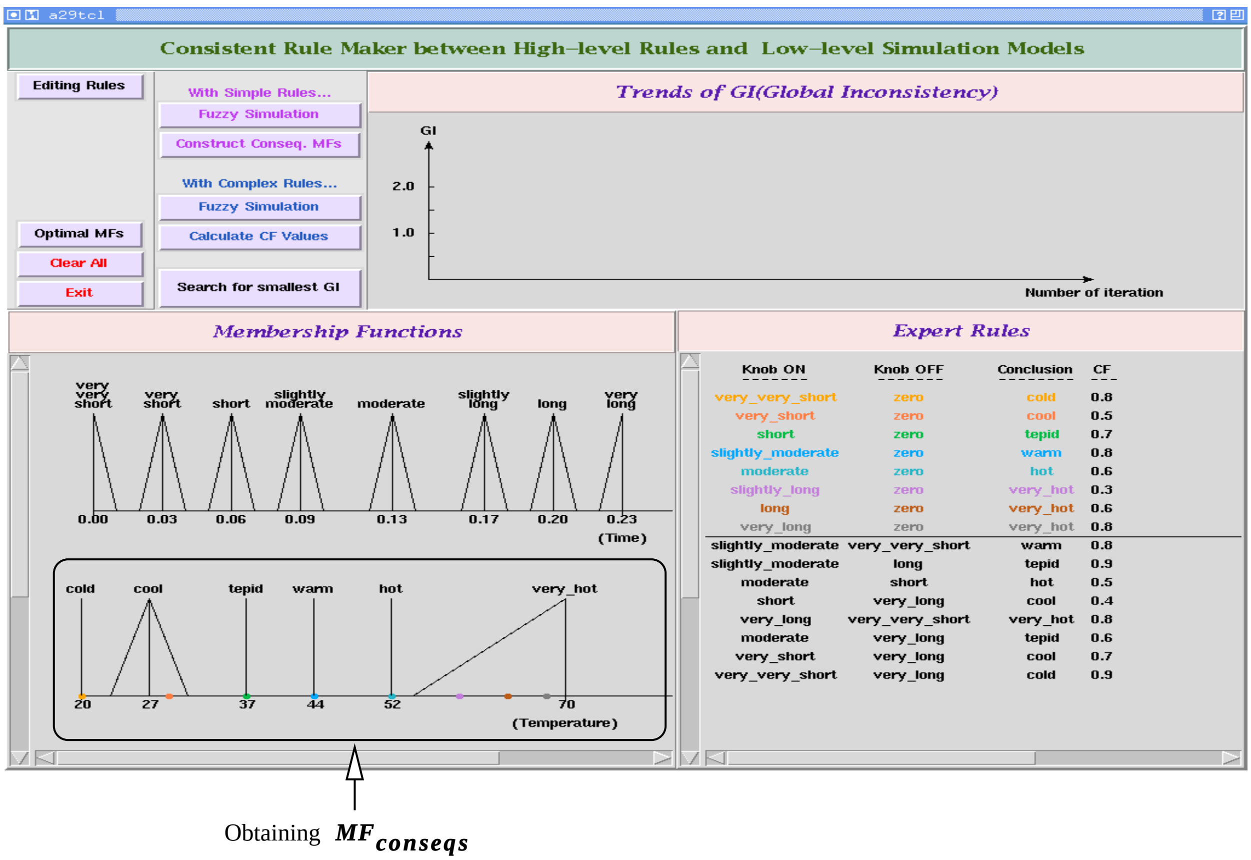


Figure 12: Obtain $MF_{conseqs}$ in $RULE_{simplexs}$

2. Apply hypothetical membership functions to $RULE_{compounds}$
   CF_{fuzzy} for each $RULE_{compound}$ obtained by applying equation (9) and GI are shown at Window C in Fig 13. LI are implicitly calculated.

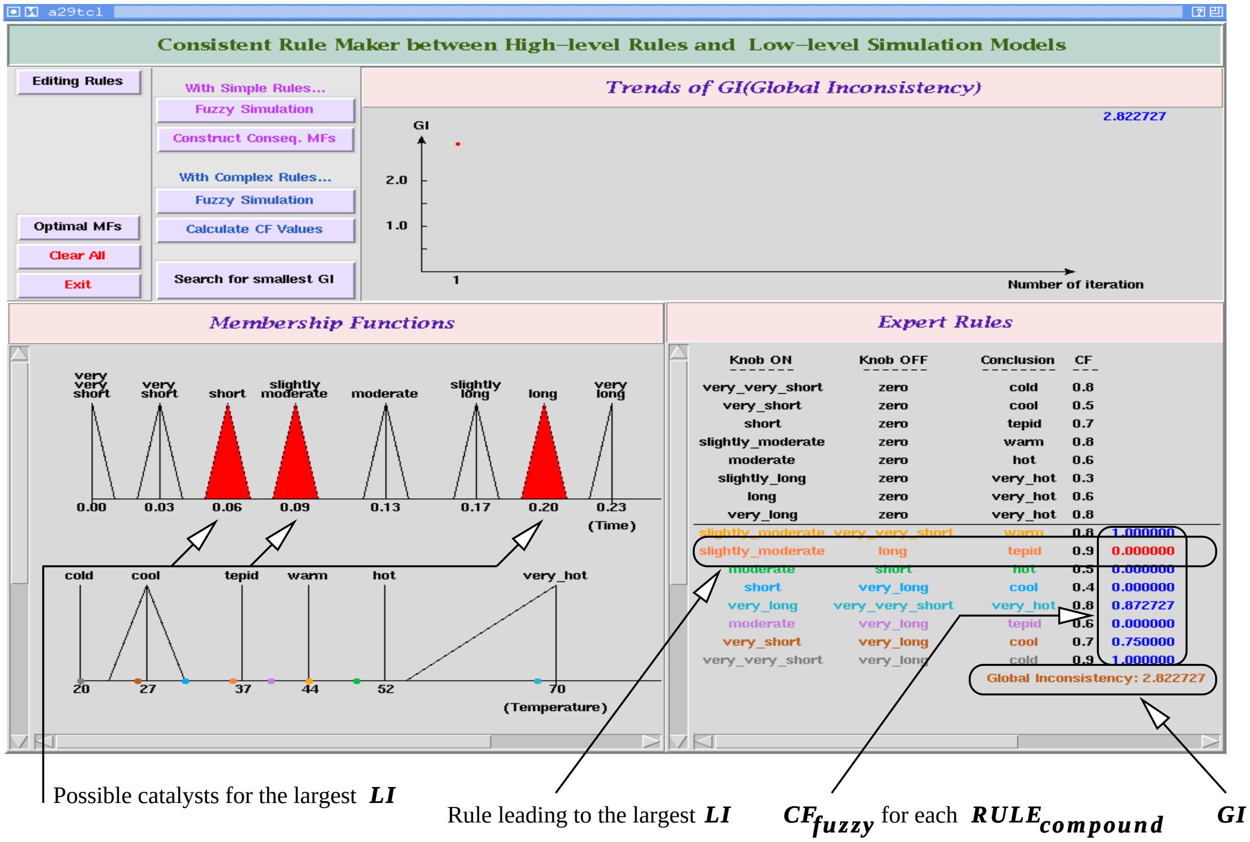


Figure 13: Calculations of CF_{fuzzy}, LIs and GI

3. Improve hypothetical membership functions using $RULE_{compounds}$
   The second rule of $RULE_{compound}$ in Fig 13 was chosen since it has the largest LI, where $LI = 0.9$. The current GI, 2.822727 is saved as an up-to-date PI. Three $MF_{premises}$ (i.e., *slightly_moderate*, *long* and *short*) are identified as the possible catalysts for the largest LI as shown at Window B in Fig 13. Thus, all subsets of these catalysts are {*slightly_moderate*}, {*long*}, {*short*}, {*slightly_moderate, long*}, {*slightly_moderate, short*}, {*long, short*} and {*slightly_moderate, long, short*}. For each subset, *step 1* and *step 2* are executed. Among the seven subsets above, a proper subset was identified as {*slightly_moderate, long, short*} because when we modify these $MF_{premises}$, we get the smallest GI, 1.487727, as shown in Fig 14. Since this GI, 1.487727, is smaller than PI, 2.822727, the stop condition is not met. Thus, the algorithm repeats *step 3*. After ten iterations of *step 3*, we finally met the stop condition as shown in Fig 15. At the *10th* iteration, we got the smallest GI, 0.262242, when the *moderate* $MF_{premise}$ was modified. However, since this GI is not smaller than PI (i.e., GI at the *9th* iteration), 0.229480, the $MF_{premises}$ and $MF_{conseqs}$ shown in Fig 15 are final approximate membership functions which lead to minimal PI, 0.229480.

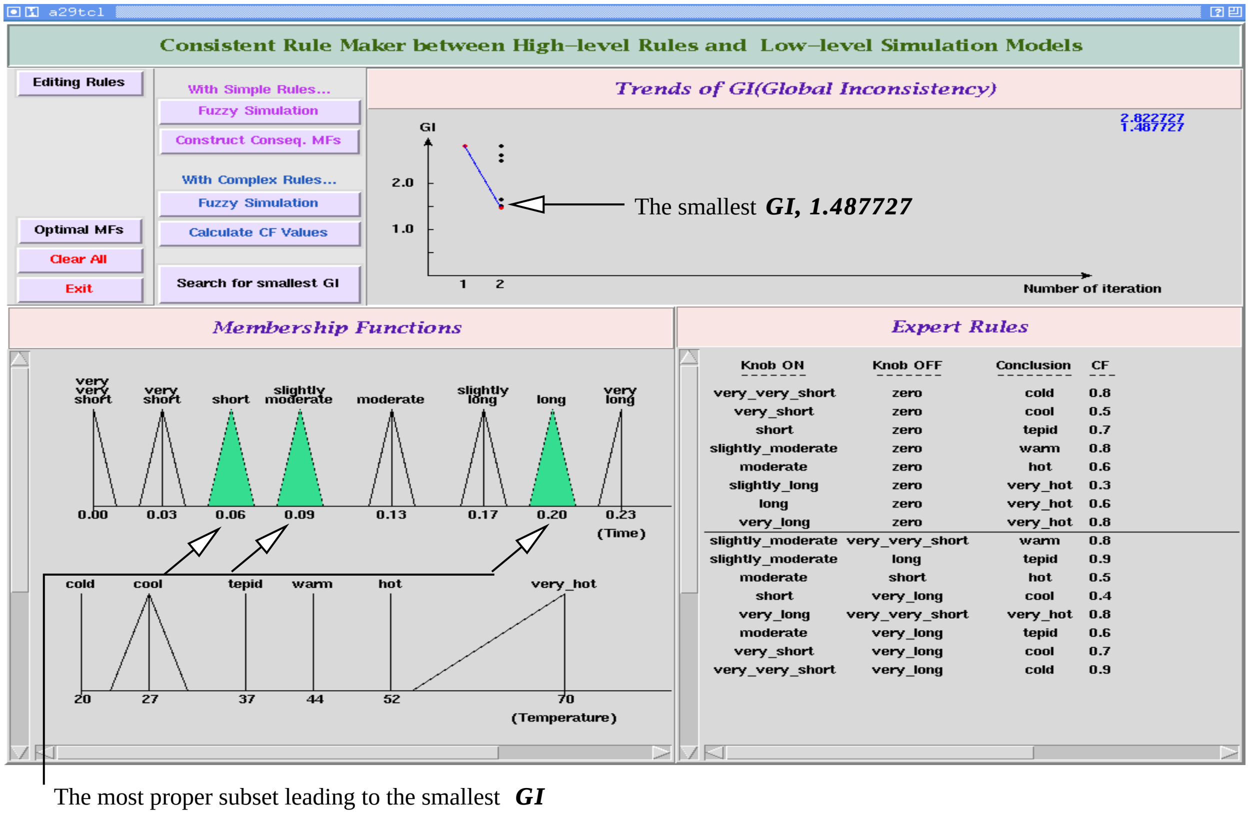


Figure 14: The smallest GI and its proper subset leading to the GI

4.2 Analysis of output

The final rules generated at the *9th* iteration are shown in the Fig 10(b). Notice that, in that figure, each CF_{expert} is closely equal to the corresponding CF_{fuzzy}. Moreover, PI turns out to be 0.229479, which can be regarded as *fairly consistent*. Therefore, the expert's rule-based model and the quantitative simulation model for this particular boiling water problem can be considered *consistent* without processing an additional *resolving inconsistency phase*. If the closeness is out of a predetermined range, human interaction is required for resolving the inconsistencies.

By executing another fuzzy simulation for all different combinations of the fuzzy values defined by the above membership functions, we got more detailed rules ($8 \times 8 = 64$ rules) as a hypothesis of the expert's knowledge. Table 2 shows a part of such knowledge.

5 Discussion and Future Research

The proposed algorithm is an *iterative improvement algorithm* employing the *gradient descent method*, because it executes a loop that continually moves in the direction of decreasing GI. It keeps track of only the current states, and does not look ahead beyond the immediate neighbors of that state. Its solution may be a *local minima* as shown in Fig 16. This local minima problem can be cured if we choose *all paths* whose GIs are better than PI as shown

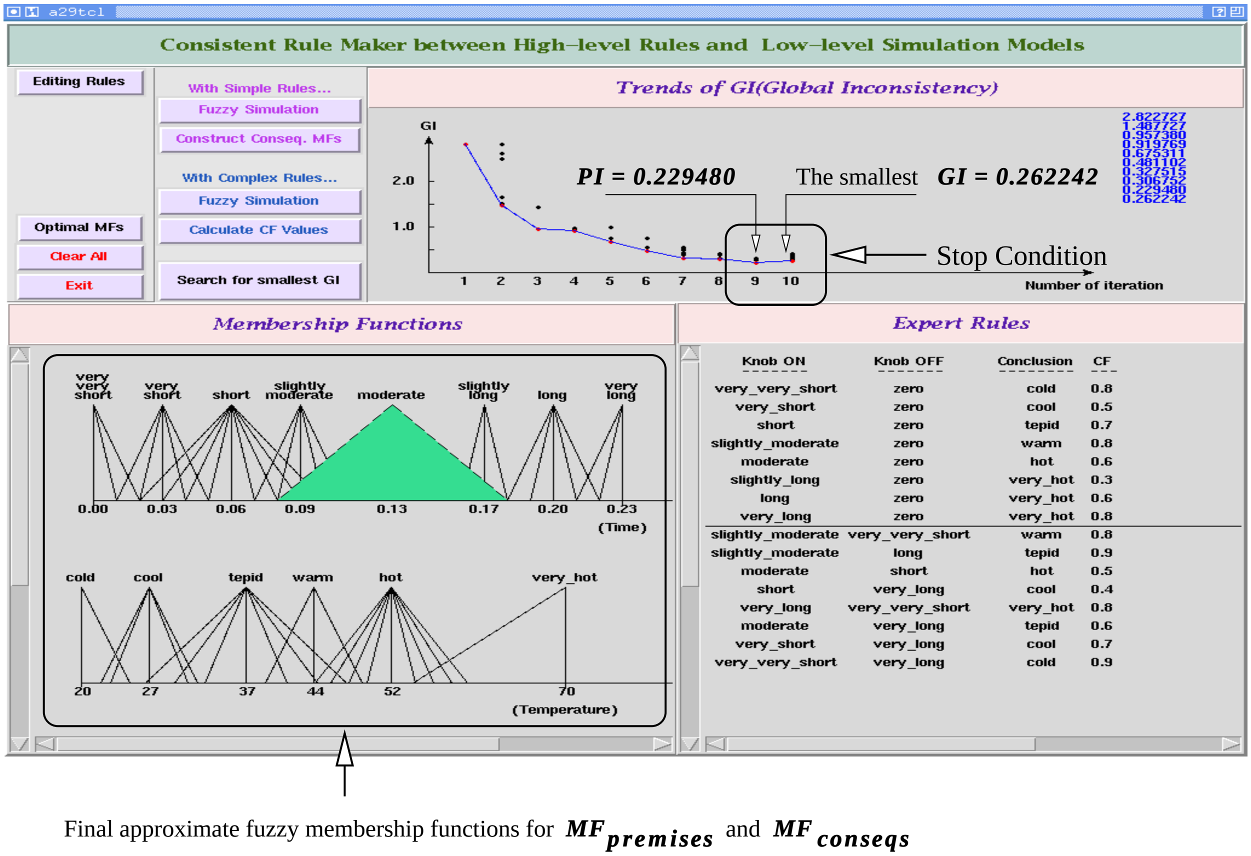


Figure 15: Satisfying stop condition and final membership functions

Table 2: A part of detailed rules extracted from fuzzy simulation

KNOB_ON	KNOB_OFF	TEMP.	CF_{fuzzy}
...	...	...	...
moderate	very_very_short	hot	0.611507
moderate	very_short	hot	0.603797
moderate	short	hot	0.553164
moderate	slightly_moderate	warm	0.403333
moderate	moderate	warm	0.652308
moderate	slightly_long	warm	0.552000
moderate	long	tepid	0.556800
moderate	very_long	tepid	0.684073
...	...	...	...

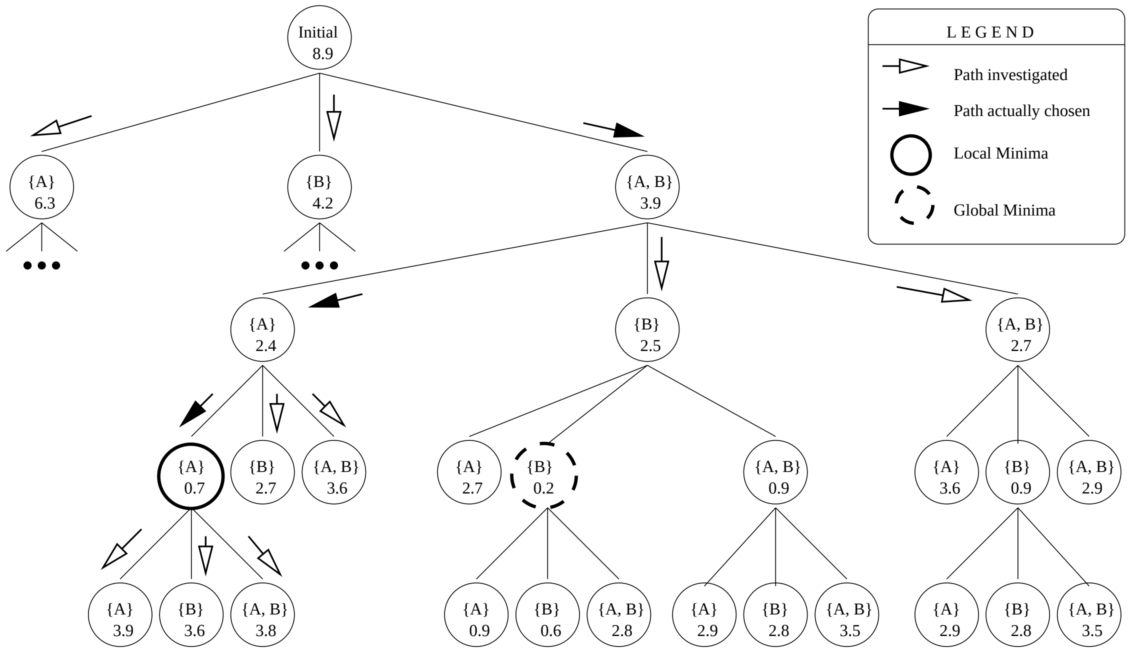


Figure 16: Local minima

in Fig 17, instead of choosing the path which has the best *GI*. Clearly, this solution costs more in terms of simulation time and memory than before, but we can better avoid the local minima problem. Alternatively, we can take a middle position between these extreme strategies. For example, when the problem space is too large to adopt the latter strategy, we can choose two or three best paths at every iteration.

Having developed *the consistency checking phase* with only *central point estimates* for handling the uncertainty arising from linguistic vagueness, we now would like to focus on considering the following issues:

- extending the method to handle the various forms of uncertainty representation.
- developing a phase for resolving inconsistency when the amount of inconsistency exceeds a reasonable range.
- applying the method to more application oriented and realistic rule-based examples.

The following sections discuss the above issues in detail.

5.1 Handling various forms of uncertainty representation

We showed that the proposed algorithm can deal with *central point* estimates. However, in order to handle the various levels of uncertainties arising from linguistic vagueness, we will extend the method to cover the other cases as well. Specifically, we will enhance our method to cover the situations in which experts present various forms of uncertainty about their linguistic terms in the following ways:

- central point estimates

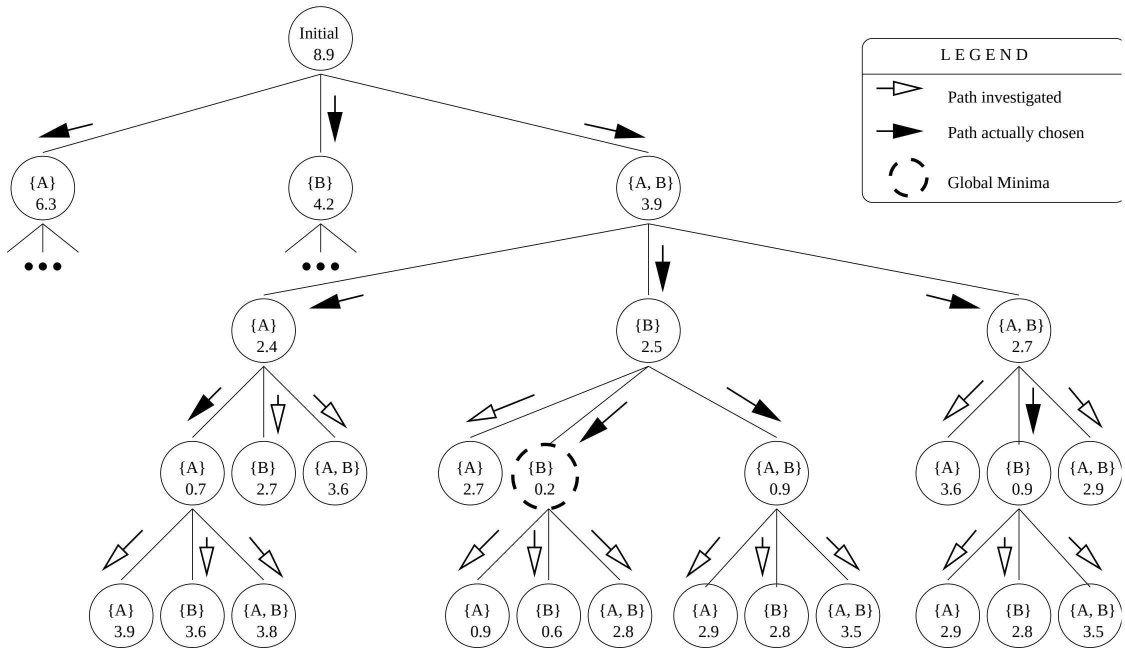


Figure 17: Global minima

- interval estimates
- approximate fuzzy membership functions such as triangular or trapezoid fuzzy numbers
- fuzzy membership functions with their complete definitions

Fig 18 shows the four forms of the uncertainty and the way we discretize such uncertainties into a fuzzy real space. In addition to these forms, we are interested in exploring the following aspects:

- Expert's knowledge may be filled with large gaps. Some rules may have linguistic values without any kind of estimates, and other rules may have different forms of uncertainty mentioned above.
- Other forms of uncertainty such as probability can be introduced into the overall process.

5.2 Developing a phase for resolving inconsistency

The algorithm presented in section 3.3 has been implemented for handling the *consistency checking phase*. When the amount of the inconsistency exceeds a reasonable range, human intervention is required for resolving the inconsistency. Either expert rules (including the definitions of the fuzzy membership functions and the confidence factors of the rules) or simulation model components should be reevaluated or modified. For such a process, we are going to consider the following aspects:

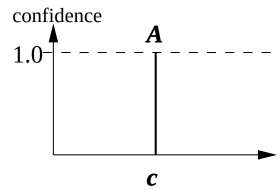
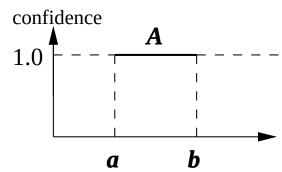
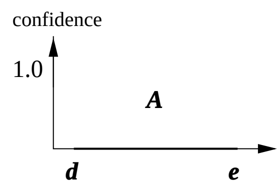
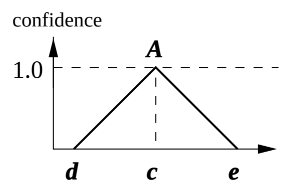
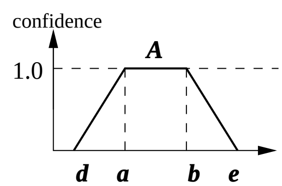
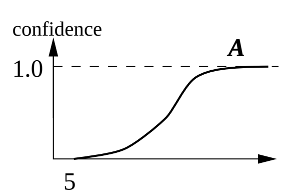
type	form of uncertainty	explanation	Discretization into fuzzy space
1	central point estimate	When experts present the center point ***c*** of ***A***	
2	Interval estimate	When experts present the interval ***[a, b]*** of ***A*** with a full confidence	
		When experts present the possible range ***[d, e]*** of ***A***	
3	Approximate fuzzy number	When experts present both the center point ***c*** and the possible range ***[d, e]*** of ***A***	
		when experts present both the interval ***[a, b]*** of ***A*** with a full confidence and its possible range ***[d, e]***	
4	Complete fuzzy number	When experts present the complete definition of ***A***	

Figure 18: Four forms of uncertainty about the linguistic terms

- The implementation of the *resolving inconsistency phase* needs to be interactive so that experts or users can dynamically change their rules and fuzzy set end_points. Every time this modification happens, the *consistency checking phase* should be reinvoked with some visual aids so that the effect of the modification can be easily recognized. This reflects a theoretical modification of the proposed algorithm as well as an implementation change of the code, since the overall processes now involve *humans in the loop* during the checking consistency and the resolving inconsistency.

- The corresponding quantitative model itself may be considered a source of the inconsistency. Some parameters or even the structure of the model needs to be adjusted. Therefore, some guidelines or facilities for aiding such processes need to be devised.

5.3 Application

The *boiling water* example discussed in section 4 is a simple domain that we selected for the purpose of illustration. More application-oriented and realistic rule-based examples are being studied for good assessment and future development of the proposed method. We are particularly interested in its application within MOOSE (Multimodel Object Oriented Simulation Environment) [19] which is now under construction at the University of Florida. MOOSE is an enabling environment for modeling and simulation based on OOPM (Object Oriented Physical Modeling). OOPM extends object-oriented program design with visualization and a definition of system modeling that reinforces the relation of "model" to "program".

By incorporating the proposed method into MOOSE, we can obtain two types of benefits: one from validating the expert's rules against simulation models, and another from validating the simulation models against the expert's knowledge. To make this point more understandable, we consider the general modeling process depicted in Fig 19 [20, 21, 4]. The conceptual model represents the mathematical, logical or verbal representation of the problem entity developed for a particular study, and the computerized model represents the conceptual model implemented on a computer. The general purpose of the conceptual model validation depicted in this figure is to validate the underlying assumptions and theories. More specifically, the process is concerned with whether this specific model's representation of the problem entity being modeled and its structure, logic and mathematical and causal relationships are *reasonable* for the intended use of the model [20]. One of the primary validation techniques used for this evaluation is *face validation* [21]. Face validation involves having domain experts evaluate the conceptual model to determine if they believe it is correct and reasonable for its purpose. This usually means examing the flowchart or graphical model, or the set of model equations.

The counterpart of the above modeling process in MOOSE can be depicted in Fig 20. MOOSE does not yet employ any form of validation or verification techniques. MOOSE supports many different types of models including FSM (Finite State Machine), FBM (Functional Block Model) and EQN (EQuatioNal Constraint model) for the conceptual modeling process. Then, by translating the conceptual model into C++ code, it constructs the computerized model. Here, the proposed method can come to play. By having the expert's knowledge in the form of rules, and through consistency checking against the results obtained from implementing FSM, FBM and EQN, we can form an *verification arc* between

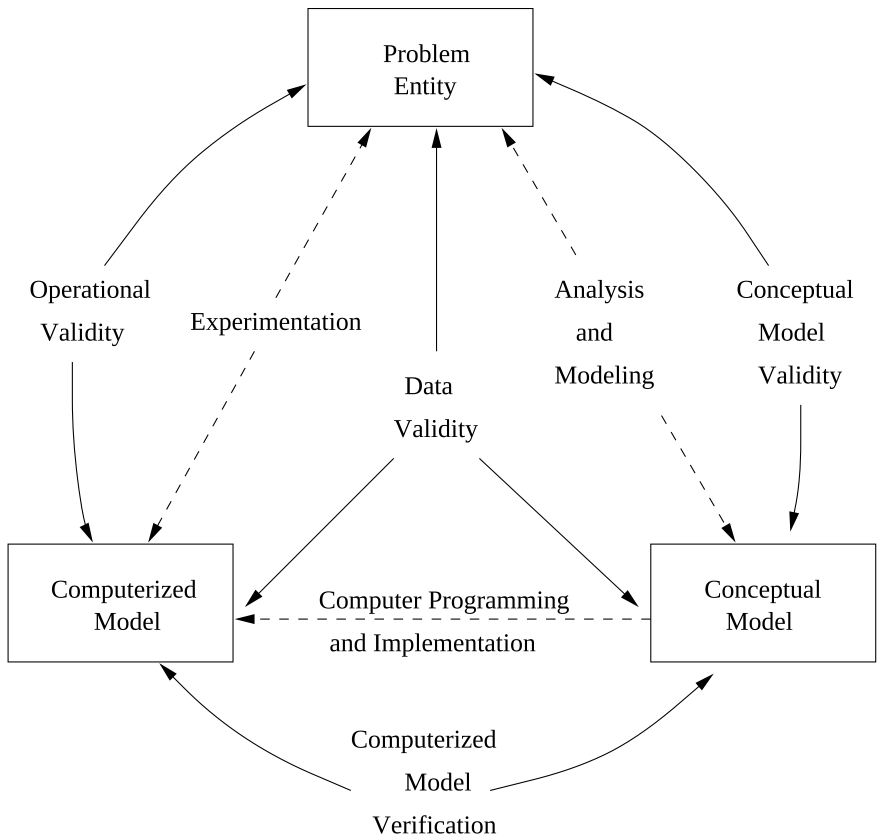


Figure 19: Modeling process and its related validation/verification

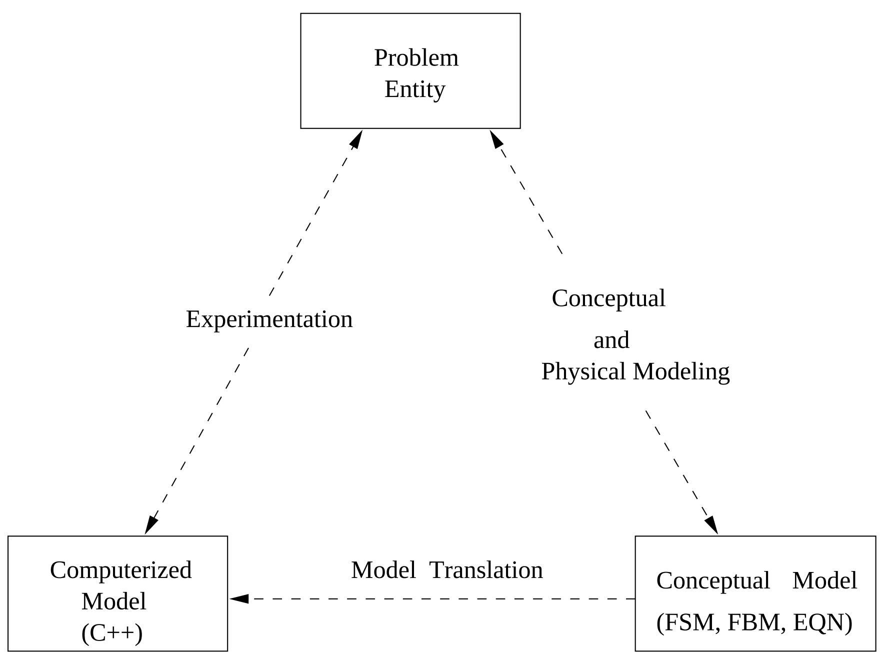


Figure 20: Modeling process in MOOSE

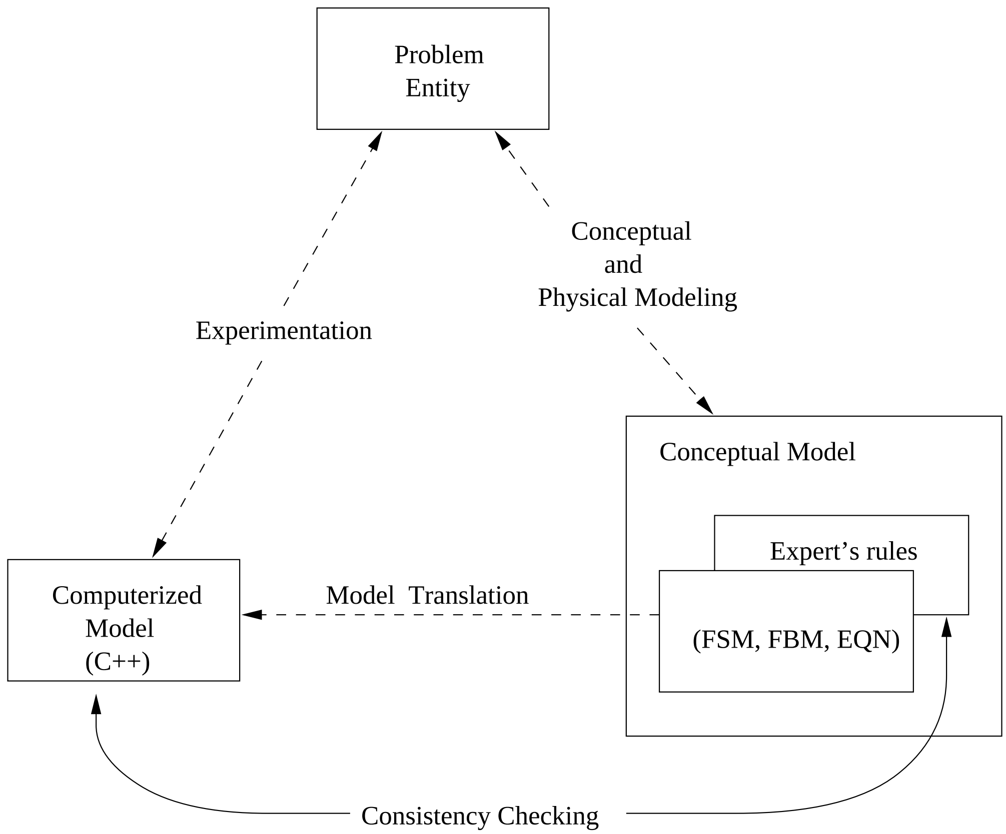


Figure 21: Consistency checking in MOOSE

the conceptual model and computerized model as shown in Fig 21. Since we skipped the conceptual model validation, any inconsistency found can be due to an inadequate conceptual model of MOOSE or the expert's rules, or an improperly programmed or implemented conceptual model on the computer.

6 Conclusion

The motivation for this work lies with the problem of resolving the difference between qualitative and quantitative forms of knowledge about physical systems. The fuzzy simulation method introduced here bridges the gaps between the two different levels of knowledge. We showed how two different extreme levels of knowledge can be directly compared and maintained in a systematic manner. Since the uncertainty arising from the human reasoning process is easily represented by rules associated with confidence factors, we devised a way for emulating such processes by showing how fuzzy simulation can derive the confidence factors from quantitative models. For handling another form of uncertainty arising from linguistic vagueness, we assumed that central point estimates were presented by experts. Although this form of estimation is a very limited form of uncertainty representation, we assert that the presented method serves as a *stepping stone* for developing a more robust method which can capture the other forms of expert's confidence levels in the future. By devising a method of integrated qualitative and quantitative dynamical system knowledge refinement, we hope to provide a way to obtain better models in general about physical systems by exploiting knowledge at all levels, whether qualitative or quantitative.

References

[1] A. Bonarini and G. Bontempi. A Qualitative Simulation Approach for Fuzzy Dynamical Models. *acm Trans. on Modeling and Computer Simulation*, 4(4):285–313, 1994.

[2] A. J. Gonzalez and D. D. Dankel. *The Engineering of Knowledge-based Systems: Theory and Practice*. Prentice Hall, New Jersey, U.S.A., 1993.

[3] L. A. Brita. A Knowledge-Based Approach for the Validation of Simulation Models: The Foundation. *ACM Transactions on Modeling and Computer Simulation*, 6(1):76–98, 1996.

[4] L. G. Brita and F. N. Ozmizrak. A Knowledge-Based Approach for the Validation of Simulation Models: The Foundation. *ACM Transaction on Modeling and Computer Simulation*, 6(1):76–98, 1996.

[5] D. Dubois, H. Prade, and R. R. Yager. Basic Notions in Fuzzy Set Theory. In D. Dubois, H. Prade, and R. R. Yager, editor, *Fuzzy Sets for Intelligent Systems*, pages 27–64. Morgan Kaufmann, 1993.

[6] P. A. Fishwick. Fuzzy Set Methods for Qualitative and Natural Language Oriented Simulation. In O. Balci, R. P. Sadowski, and R. E. Nance, editor, *WSC'90*, volume 9-12 December, pages 513–519, 1990.

[7] P. A. Fishwick. Extracting Rules from Fuzzy Simulation. *Expert Systems With Applications*, 3:317–327, 1991.

[8] P. A. Fishwick. Fuzzy Simulation: Specifying and Identifying Qualitative Models. *Int. J. General System*, 19:295–316, 1991.

[9] P. A. Fishwick. *Simulation Model Design and Execution: Building Digital Worlds*. Prentice Hall, New Jersey, U.S.A., 1995.

[10] G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, New Jersey, U.S.A., 1995.

[11] I. Bratko, I. Mozetič, and N. Lavrač. *KARDIO: A Study in Deep and Qualitative Knowledge for Expert Systems*. MIT Press, London, England, 1989.

[12] B. Kuipers. Qualitative Simulation. *Artificial Intelligence*, 29(3):289–338, 1986.

[13] L. E. Widman and K. A. Loparo. Artificial Intelligence, Simulation, and Modeling: A Critical Survey. In L. E. Widman, K. A. Loparo, and N. R. Nielsen, editor, *Artificial Intelligence, Simulation, and Modeling*, pages 1–44. A Wiley-Interscience, 1989.

[14] L. O. Hall and A. Kandel. The Evolution From Expert Systems to Fuzzy Expert Systems. In A. Kandel, editor, *Fuzzy Expert Systems*, pages 3–21. CRC Press, 1992.

[15] D. Li and F. E. Cellier. Fuzzy Measures in Inductive Reasoning. In O. Balci, R. P. Sadowski and R. E. Nance, editor, *WSC'90*, volume 9-12 December, pages 527–538, 1990.

[16] R. O'Keefe. Simulation and expert systems - a taxonomy and some examples. *Simulation*, 46(1):10–16, 1986.

[17] W. Pedrycz. An Identification Algorithm in Fuzzy Relational Systems. *Fuzzy Sets and Systems*, 13:153–167, 1984.

[18] Q. Shen and R. Leitch. Fuzzy Qualitative Simulation. *IEEE Trans. Systems, Mans, and Cybernetics*, 23(4):1038–1061, 1993.

[19] T. Goktekin R. M. Cubert and P. A. Fishwick. MOOSE: Architecture of an Object-Oriented Multimodeling Simulation System. In , editor, *SPIE AeroSense*, volume April, 1997.

[20] R. G. Sargent. An Exploration of Possibilities for Expert Aids in Model Validation. In M. S. Elzas, T. I. Ören, and B. P. Zeigler, editor, *Modeling and Simulation Methodology in the Artificial Intelligence Era*, pages 279–297. Elsevier Science, 1986.

[21] R. G. Sargent. Simulation Model Verification and Validation. In B. L. Nelson, W. D. Kelton, and G. M. Clark, editor, *WSC'91*, volume 8-11 December, pages 37–47, 1991.

[22] L. Steels. Components of expertise. *AI Magazine*, 11(2):28–49, 1990.

[23] M. Sugeno and K. Tanaka. Successive Identification of A Fuzzy Model and Its Applications to Prediction of A Complex System. *Fuzzt Sets and Systems*, 42:315–334, 1991.

[24] M. Sugeno and T. Yasukawa. A Fuzzy-Logic-Based Approach to Qualitative Modeling. *IEEE Trans. on Fuzzy Systems*, 1(1):7–31, 1993.

[25] L. A. Zadeh. Fuzzy Sets. *Information and Control*, 8:328–353, 1965.

[26] L. A. Zadeh. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Trans. Systems, Mans, and Cybernetics*, SMC-3:28–44, 1973.

[27] H. J. Zimmermann. *Fuzzy Set Theory and Its Applications.* Kluwer-Nijhoff Publishing, MA, U.S.A., 1985.