ELSEVIER

Fuzzy Sets and Systems 118 (2001) 181–197

FUZZY sets and systems

www.elsevier.com/locate/fss

# A methodology for fuzzy modeling of engineering systems

Y.M. Ali *, Liangchi Zhang

*Department of Mechanical and Mechatronic Engineering, The University of Sydney, J07 Sydney, NSW 2006, Australia*

Received November 1996; received in revised form June 1998

**Abstract**

This paper describes a systematic approach to the modeling of engineering systems using a fuzzy formulation that is independent of human knowledge. The computer algorithm described here operates on a set of experimental observations of the system and constructs an *optimum* fuzzy model for these observations. The program automatically selects membership functions, deduces inference rules, constructs logical relations, and determines the formulae for conducting union and intersection operations. Membership functions, rules, and logical operations are defined parametrically. Model parameters are optimized so that the model can, at least, re-produce with minimum error the data that were used in obtaining the membership functions and rules. Therefore, model parameters are optimized to minimize error or entropy of the back-inferences of the observations from which the model was constructed. To reach the global minimum and avoid entrapment in a local minimum, a random search is carried out, then followed by a systematic Hooke–Jeeves search optimization algorithm. It has been found that this technique is more successful, compared with other statistical techniques. © 2001 Elsevier Science B.V. All rights reserved.

*Keywords:* Membership functions; Empirical research; Linguistic modeling; Rule-tuning; Engineering; Grinding; Financial analysis

## 1. Introduction

Fuzzy logic has progressed greatly since the first introduction of fuzzy sets by Zadeh [23], and has established itself as an important branch of human thinking and knowledge representation. The degree of progress and sophistication achieved in the mathematical theory of fuzzy sets can be clearly demonstrated by many publications in specialized journals and monographs. However, practical engineering applications have been based on approaches that were developed in the earlier stages. For example, many of the most successful commercial engineering applications of fuzzy logic, i.e. fuzzy controllers, are still based on the work of Mamdani and Assilian in 1975 [11], where symmetric triangular membership functions were used, and logical operations for union and intersection were performed using min. and max. operations, respectively [3]. Decisions on many aspects of fuzzy modeling have been mostly justified by the argument that “they work”.

The absence of a clear procedure for fuzzy modeling has always been emphasized by anti-fuzzy critics [12]. Moreover, by using these simple formulations, fuzzy models cannot achieve the level of accuracy required by many engineering

* Corresponding author.


PII: S0165-0114(98)00272-3

applications. Although low accuracy can be compromised for faster computation in control applications, this may not be acceptable in the modeling of many other engineering systems. Any system theory should be able to provide efficient means for model building as well as decision making (control) of the system involved [15]. In this sense, fuzzy modeling of engineering systems seems to be lagging behind fuzzy control, even though many engineering applications can be fuzzy modeled if a high level of accuracy and reliability can be achieved.

### *1.1. Major difficulties in fuzzy modeling*

To construct a new fuzzy model for a given system, engineers usually face the following questions:

(1) How to define membership functions? How to describe a given variable by linguistic terms? How to define each linguistic term within its universe of discourse and membership function, and how to determine the best shape for each of these functions?
(2) How to obtain the fuzzy rule base? In modeling many engineering problems, usually, nobody has sufficient experience to provide a comprehensive knowledge base for a complex system that cannot be modeled physically, and where experimental observations are insufficient for statistical modeling. Moreover, human experience is debatable and almost impossible to be verified absolutely.
(3) What are the best expressions for performing union and intersection operations? In other words, which particular function of the t-norms and s-norms should be used for a particular inference.
(4) What is the best defuzzification technique for a given problem?
(5) How to reduce computational effort in operating with fuzzy sets, which are normally much slower than operating with crisp numbers?
(6) How to improve computational accuracy of the model? Being fuzzy does not necessarily mean inaccurate. At least, accuracy should be acceptable by the nature of the engineering problem.

Many of the above issues were pointed out by Zimmermann [24] as areas where urgent empirical research is needed. In particular, the assumption that membership functions "are given" and the fact that "excessively voluminous" computational effort is needed for any large complex problem, remain the major challenges for the application of fuzzy theory.

Selection of optimum membership functions and deduction of rules from observed data can be implicitly carried out using artificial neural networks, which can adaptively adjust membership functions and fine-tune rules to achieve better performance. This has resulted in many Neuro-Fuzzy approaches [9,10,19]. This approach, however, does not contribute much to the development of fuzzy logic. Moreover, neural network methods are still premature and suffer from the same disadvantages of fuzzy logic, as mentioned above. For example, the selection of the number of processing nodes, the number of layers, and the interconnections among these nodes and layers, is still an art and lacks systematic procedures [13]. Therefore, it would be more appropriate to address these questions in the framework of fuzzy logic, rather than to transform them to another field.

The subject of "fine-tuning of fuzzy models" has been addressed by many researchers. Gurocak and Lazaro [5] assumed that fuzzy rules and initial shapes of membership functions are given. They used a Hooke–Jeeves optimization algorithm to adjust the location of each membership function in order to minimize inference error. They pointed out that performance of this method depends very much on the initial condition of the rule base to be tuned. Shimojima et al. [17] observed that even if fine tuning is conducted using an adaptive neural network approach, a solution may converge to a local minimum. On the other hand, the problem of selecting the best mathematical representation of logical union and intersection operations is more theoretical and was not given much attention by empirical studies, even though the axiomatic structure of fuzzy sets [4,8] allows for a large class of equally valid t-norms and s-norms to be implemented.

### *1.2. Objectives of the present algorithm*

The purpose of this paper is to tackle most of the above issues all in one pass, by providing a systematic procedure for the construction of fuzzy models. The algorithm presented in this paper can be viewed on one hand as an extension and improvement on the fine-tuning approaches in [5,17]. On the other hand, it can be viewed as a surface-fitting technique, where huge computational power is used to fit experimental data over a very complex hyperspace of very large dimension. It can also be viewed as an explicit formulation of what is otherwise implicit in the adaptation process of a back-propagation neural network. However, the main objectives of the algorithm are:

(1) Automatic generation of fuzzy rules that are not biased by human factors or context-dependent experience.
(2) Provision of clear *physical* meaning of each linguistic term or fuzzy set without any a priori knowledge about the system.
(3) Establishment of clear systematic procedure for constructing a fuzzy model, where trial and error is minimized.

Membership functions, which are not known a priori, are defined parametrically. The exact value of each of the parameters, and the corresponding *optimum* fuzzy rules, are then deduced by the program from given experimental observations. To avoid entrapment in a local minimum, a random search algorithm is used to find a global minimum zone before the Hooke–Jeeves search method attempts to achieve the actual minimum. Similarly, logical operations are also represented parametrically, and an algorithm to search for the optimum parameter is used to select the most compatible mathematical representation of logical operations.

Hence, once a reasonable defuzzification method is employed, the program is able to select and define all membership functions, deduce all relevant rules, and identify the most appropriate logical operations. The selection criterion is to obtain the best accuracy in back-deduction of observation from which the model has been derived. Once such a model is constructed, many hardware and software approaches can be implemented to achieve high computational efficiency. Although the present study uses a super-computer with 4096 parallel processors, this is required only during the model building stage. Implementations of the derived models can be made very fast for practical applications on a personal computer. Current advances in computing speed, development of fuzzy computers, and fast software techniques, e.g. table-lookup, can be used to improve computational efficiency for practical purposes [18].

## 2. The methodology for fuzzy modeling

Finding the state of an engineering system, $\boldsymbol{X}$, is the objective of any model, and can be implicitly expressed as

$$X^i = f(\boldsymbol{X}, \boldsymbol{X}', \boldsymbol{X}'', \ldots), \quad X^i \in \Re,\ i = [1 \ldots n]$$

for $n$ interdependent variables, $X^i$,[1] where $\boldsymbol{X}'$ and $\boldsymbol{X}''$ are the first and second time derivatives of the state vectors, respectively. Generally, function $f$ is very complex for most engineering systems such that it cannot be deducted purely by physical modeling. Moreover, the size of the vector $\boldsymbol{X}, n$, can be excessively large and prohibit thorough experimental investigation and production of useful empirical formulations. On the other hand, even when a physical or statistical model does exist (for a relatively simple problem or after many simplifying assumptions), the nature of the variables in the model and the methods of measuring them may make it difficult to quantify them precisely. Therefore, fuzzy modeling of the system becomes advantageous. In this case, function $f$ is replaced by a relation $R$ that describes the fuzzy rules. For this type of large-scale complex problems, it is unrealistic to expect the existence of a single expert, or a small group of experts, who can provide unbiased valid rules or even a universally acceptable definition of various linguistic terms. Therefore, we will generate all rules and definitions based on experimental observations without reliance on human experience.

When the interaction among input system variables is low, it is possible to represent the above-mentioned system as a multiple-input–multiple-output system correlating a number of independent variables $\boldsymbol{Y} \in \boldsymbol{X}$ to a number of dependent variables $\boldsymbol{Z} \in \boldsymbol{X}$ via a set of $\ell$ fuzzy rules of the form

$$Y_j^1 \cap Y_j^2 \cap \cdots \cap Y_j^m \Rightarrow Z_j^{m+1} \cap Z_j^{m+2} \cap \cdots \cap Z_j^n, \quad (1)$$

where $j = [1 \ldots \ell]$. This model can be further simplified into $(n-m)$ multiple-input–single-output models by transforming Eq. (1) to the form

$$\bigcap_{k=m+1}^{n} (Y_j^1 \cap Y_j^2 \cap \cdots \cap Y_j^m \Rightarrow Z_j^k), \quad (2)$$

[1] Superscripts are used as indices, not exponents, unless otherwise indicated.

or the more compact form of the fuzzy rule base

$$R = \bigcup_{j=1}^{\ell} \bigcap_{k=m+1}^{n} \left( \bigcap_{r=1}^{m} Y_j^r \Rightarrow Z_j^k \right) \tag{3}$$

which is to be used in this study.

### 2.1. Definition of membership functions

A fuzzy variable $X^i$ can be defined by the quadruple [3,24]

$$(x, U, \boldsymbol{T}(x), \boldsymbol{M}(x))^i, \tag{4}$$

where the label '$x$' is a text expression in natural language that expresses the name of the variable. The universe of discourse $U \equiv [U_L, U_U]$ defines the interval of real values that $X^i$ can belong to, where $X^i \in U$. Alternatively, a fuzzy variable can be expressed in terms of its degree of belonging to a set of linguistic terms that is defined by the term set $\boldsymbol{T}(x)$. For practical purposes a minimum of three linguistic terms is required, while using more than ten linguistic terms can be confusing in normal human contexts. It is also recommended that the number of terms should be odd, in order to represent the "middle" or "medium" state. Therefore, a reasonable number of linguistic terms can be 3, 5, 7, or 9. The user of the program can choose any number of linguistic terms from 3 to 9, but five is considered the most reasonable and is used by default (for example: very low, low, medium, high, very high). The transformation from crisp numbers to linguistic terms is conducted by the set of membership functions $\boldsymbol{M}(x)$. The fuzzification of a variable is the mapping of its values from $U$ to $\boldsymbol{T}$ using the membership functions $\boldsymbol{M}$. A certain linguistic term $T_s^i(x)$, $s = [1, n_t]$, is defined by the triple

$$(t, S, P)_s^i, \tag{5}$$

where '$t$' is the text label of the term, $S \equiv [S_L, S_U]$, $S \subseteq U$, is its support subset, and $P$ is the set of parameters defining its membership function $\mu_s^i(x)$. In the present work, membership functions are generically defined as

$$\mu_T(y) = \begin{cases} 0, & y \notin S \\ \frac{1}{2}\left(\frac{y - S_L}{a - S_L}\right)^e, & y \in [S_L, a] \\ 1 - \frac{1}{2}\left(\frac{b - y}{b - a}\right)^e, & y \in [a, b] \\ 1, & y \in [b, c] \\ 1 - \frac{1}{2}\left(\frac{y - c}{d - c}\right)^e, & y \in [c, d] \\ \frac{1}{2}\left(\frac{S_U - y}{S_U - d}\right)^e, & y \in [d, S_U] \end{cases} \tag{6}$$

in terms of a set of parameters $P = \{a, b, c, d, e\}$.

Expression (6) is very general and can describe a large class of shape variations, as shown in Fig. 1, if the ordered set $\{S_L, a, b, c, d, S_U\}$ and the exponent e are given. The form in Eq. (6) will be called a *mold*. By adjusting the seven parameters in $\{S, P\}$ one can *cast* any linguistic term into the corresponding mold.

A linguistic term can have a fully defined membership function, if the seven parameters above are known. Therefore, for a given system involving $n$ variables, we can identify $m$-independent variables, and $(n-m)$-dependent variables, which can be de-coupled and solved as $(n-m)$ multiple-input–single-output systems. Each variable can be described as belonging to a number of linguistic terms, $n_t$. Each term, in turn, is defined using seven parameters. Therefore, the problem of membership functions definition reduces to that of finding the best

$$N = 7 \sum_{i=1}^{n} n_t^i \tag{7}$$

parameters. For example, for a problem involving ten variables where each variable is described by five linguistic terms, a total of 350 parameters (10 variables $\times$ 5 terms $\times$ 7 mold parameters) need to be identified.

### 2.2. Selection of logical operations

Zadeh [23] suggested that the intersection (AND) operation can be expressed in terms of the min.-operator or by the algebraic product, while the union

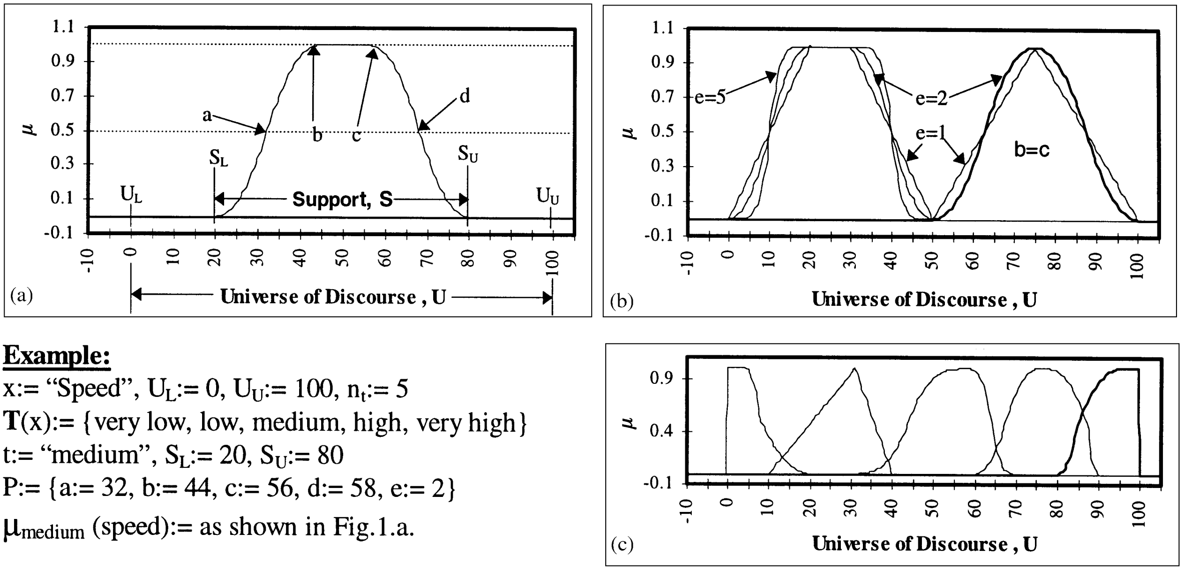



Fig. 1. Definition of membership functions. (a) Generalized membership function (mold). (b) Typical symmetric membership functions. (c) Typical unsymmetric membership functions.

(OR) operation can be expressed by max.-operator or the algebraic sum

$$\mu_{A\cap B} = \min(\mu_A, \mu_B), \quad \mu_{A\cup B} = \max(\mu_A, \mu_B). \tag{8}$$

Bellman and Giertz [2] provided the axiomatic structure for fuzzy set operations, which resulted in a class of t- and s-norms for intersection and union, respectively. Consequently, other authors proposed alternative operations; all of them conform with the axioms and belong to t- and s-norms [4,6,22,24]. As each operation can yield different numeric values of membership, selection of the logical operational forms can affect the performance of the fuzzy model; yet, the most appropriate representation is not known a priori and has to be determined by an empirical approach.

Similar to our membership function representation, logic operations are also described parametrically, and are based on the "fuzzy" operations given by Werners [20]

$$\mu_{A\cap B} = \gamma \min(\mu_A, \mu_B) + \frac{1-\gamma}{2}(\mu_A + \mu_B)$$

and

$$\mu_{A\cup B} = \gamma \max(\mu_A, \mu_B) + \frac{1-\gamma}{2}(\mu_A + \mu_B), \tag{9}$$

where $\gamma$ is a parameter to be determined empirically. If $\gamma = 1$, Eq. (9) becomes identical to Eq. (8); while when $\gamma = 0$, both union and intersection operations become equal, indicating a complete fuzziness. Any operation that belongs to the t- or s-norms exists somewhere between these two extremes. Therefore, selection of the proper value for $\gamma$ corresponds to selection of the best representation of logical operations. A separate optimization algorithm is used to find the value of that parameter. Yet, for all problems solved by the present algorithm, $\gamma$ always converged to 1, i.e. the Zadeh representation. Therefore, Eq. (8) is justified as a good representation of logic operations from an empirical point of view.

### 2.3. *Defuzzification*

Fuzzification of crisp numbers to linguistic terms is clearly defined, but defuzzification from fuzzy terms to crisp numbers is not. The most accurate method is probably the center of area method (COAM), which obtains the center of the area under the $\mu$-cut zone. However, this method is slow, and poses some doubts when overlapping zones exist. An alternative method, which offers the best performance in terms of computational speed, continuity, dis-ambiguity, plausibility, and weight counting [3] is the weighted height method (WHM). In this method, the peak of each term (usually the 1.0-cut) is multiplied by its weight (its $\mu$) and a weighted average is obtained. However, as can be seen from Fig. 2, WHM is insensitive to the particular shape of the membership function, and can become erroneous when the membership function is highly un-symmetric.

A modification to the WHM, called the weighted sections method (WSM), is used here. It uses the value of the nucleus of the $\mu$-cut and multiplies it by the membership value $\mu$ for each term, and then obtains a weighted average of all terms. As can be seen from Fig. 2, this method is more sensitive to un-symmetric membership functions and can produce better results than the WHM without much additional computational effort. It can also be seen that the WSM always produces defuzzification values that are bounded between those obtained by the COAM and WHM. Therefore, the WSM provides a better compromise between computational accuracy and time, and is adopted as the standard defuzzification procedure in the present study.

### 2.4. Optimization criteria

As can be seen from the previous discussion, fuzzy modeling of a system can be viewed as a problem of optimal selection of the $N$ parameters in Eq. (7). The criterion for selection of these parameters has to be defined. There is an obvious requirement that any model should satisfy: *It should be able to reproduce the original data from which it was obtained, and upon which it was designed, with minimum error*. For example, a given database of $M$ experimental observations, each defined by the state vector $\boldsymbol{X} \equiv \boldsymbol{Y} \cap \boldsymbol{Z}$, can be transformed into a fuzzy rule base of the form in Eq. (3). If the membership functions and parameter $\gamma$ used in the transformation are true representations of the physical and linguistic nature of the problem, then for given antecedents $\boldsymbol{Y}$, the rule base $R$ should predict consequents $\boldsymbol{Z}'$ such that:

$$\forall_{\boldsymbol{X}}\colon \phi(\boldsymbol{Z} - \boldsymbol{Z}') = e \to 0, \tag{10}$$

where $e$ is a small positive number that approaches zero in the limit after a large number of optimization iterations. There are many measures of the function $\phi$ that can be used such as

- *Average absolute error*

$$\varepsilon = \frac{1}{M} \sum_{j=1}^{M} \frac{\mathrm{abs}(\boldsymbol{Z}^j - \boldsymbol{Z}'^j)}{\boldsymbol{Z}^j}. \tag{11}$$

- *Deviation of error*

$$\delta = \sqrt{\frac{1}{M} \sum_{j=1}^{M} \left( \frac{(\boldsymbol{Z}^j - \boldsymbol{Z}'^j)}{\boldsymbol{Z}^j} \right)^2}. \tag{12}$$

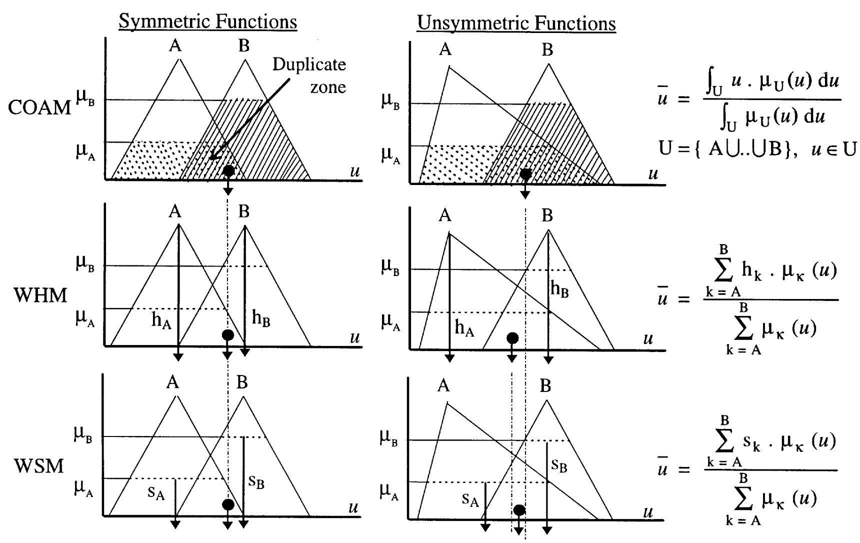



Fig. 2. Comparison among COA, WH, and WS methods of defuzzification.

- *Specific entropy* [9,10]

$$E = \frac{\mu \cap \neg\mu}{\mu \cup \neg\mu}, \tag{13}$$

where entropy, $E$, can be understood as a measure of the degree of uncertainty or fuzziness of a certain fuzzy rule, the degree of inability to justify the soundness of a given argument. Therefore, entropy is zero at the two Boolean states $\mu = 0$ and 1, and is unity at the intermediate, completely fuzzy state, $\mu = 0.5$. Clearly, the more deterministic the rules, the more valid they are. Thus pushing the possibility of a rule away from the 0.5 zone towards one of the $(0, 1)$ limits is recommended. Therefore, if the entropy of every rule is minimized, we should expect the rule base to perform a better system modeling and yield more accurate predictions. A weaker formulation requires that the total entropy of the rule base be minimized; then, for most of the inference situations the rule base will yield acceptable inferences. Only when entropy of the whole rule base is zero, then predictions of the rule base for the fuzzy model will become exact and would perfectly match experimental observations.

For a large number of independent variables, however, the size of the rule base, $R$ in Eq. (3), can be huge. In the absence of complete information about the system, it is reasonable to expect that most of the rules in $R$ will have zero possibility, and partially sound rules will occupy few localized zones of the $R$-space. Therefore, it is better to define a kernel, $\kappa$, that is a subset of $R$ containing only those rules that are partially sound,

$$\kappa = R \cap \neg\emptyset. \tag{14}$$

Therefore, it is possible to define the specific entropy of a rule base as

$$\sigma = \frac{\sum_{\kappa} E}{\mathrm{card}(\kappa)}, \tag{15}$$

where card($\kappa$) equals the number of partially sound rules. Since $\sigma$ is a property of the rule base only and its evaluation does not require performing any inferences, it is much easier and computationally faster to evaluate than $\varepsilon$ or $\delta$. In all problems solved using the present algorithm, we found that minimization of the error measures $\varepsilon$ or $\delta$, always, results in minimization of the specific entropy $\sigma$ of the rule base. Therefore, minimization of specific entropy can also be used as a criterion for the optimization of the fuzzy model.

There are at least two ways for minimizing the specific entropy defined by Eq. (15)

(1) by concentration of rules, or
(2) by diffusion of entropy.

Rules can be made more deterministic, thus reducing total entropy while keeping the number of rules more or less the same. Moreover, increasing the number of sound rules, without significant reduction in the fuzziness of each rule will also result in a lower specific entropy. The first mechanism will result in reducing inference error and improves performance of the system, but the second one may increase the inference error and render the whole system redundant. Our empirical research shows that minimizing specific entropy can sometimes increase error while minimization of error will always result in lower specific entropy. Therefore, minimization of specific entropy of the rule base is a necessary but not sufficient condition for the optimum design of a fuzzy model.

### 2.5. *The full algorithm*

The algorithm described in this paper was developed in two versions. The first version was implemented using Turbo Pascal for Windows and runs on a PC. For a Pentium 166 MHz processor with 8 Mbytes free RAM, this program is capable of optimizing models with up to nine variables, and up to five linguistic terms for each variable. Computational time increases exponentially with the number of variables involved, and increases more or less linearly with increase in the number of linguistic terms and data points in the input database. For the typical engineering problems studied using this program, computational time ranged from 50 to 200 CPU h.

The second version was implemented on a Connection Machine CM5 computer, and was written in C* – a data-parallel dialect of standard C. The CM5 machine used consists of 128 processing nodes. Each node contains four vector processors, and each processor contains 8 Mbytes RAM and eight pipeline processors. Therefore, the machine available was capable of delivering a peak performance of 20 Gflops and 4 Gbyte of RAM. For the typical problems implemented on this program, optimization time was from 5 to 50 CPU h. Clearly, the optimization algorithm is, computationally, very exhaustive. However, it is important to note that this program is usually run only once in the lifetime of a model. Once the optimum design is reached, actual computations can be conducted very efficiently on a PC or a programmable calculator. The improvement in model accuracy is found to justify the initial effort, as will be shown later. The high demand of CPU time and memory stems from the need for frequent re-construction of the rule base. For $n$ variables with $n_t$ linguistic terms for each variable, the rule base, $R$ in Eq. (3), is an $n$-dimensional relation whose size is

$$\mathrm{card}(R) = \prod_{i=1}^{n} n_t^i. \tag{16}$$

For the present hardware configuration, this size should not exceed 128 million rules. This huge number of rules can be reached, in principle, with 17 variables, three linguistic terms each, or with nine variables and up to nine linguistic terms each; other combinations are also possible. Again, the ability to systematically model a 17-variables system can justify the initial overhead of the optimization process.

To sum up the information from the previous sections, a flow chart of the algorithm is shown in Fig. 3 and is described below:

- The number of variables as well as the universe of discourse of each variable are dictated by the physical nature of the problem, but the number of linguistic terms for each variable needs to be chosen. Quite often the number of linguistic terms is known from experience about the system. However, if this is not the case, five or seven linguistic terms would be adequate for most practical cases.
- Each linguistic term is defined by its interval and the five parameters $a$, $b$, $c$, $d$, and $e$, as shown in Eq. (5) and Fig. 1. Therefore, for the complete model, $N$ parameters need to be identified (Eq. (7)). Furthermore, the parameter $\gamma$ for logic operations (Eq. (9)) needs to be selected. The selection criterion is to minimize the errors $\varepsilon$, $\delta$ or specific entropy, $\sigma$, defined by Eqs. (10)–(15). Therefore, model construction becomes that of optimization of the objective function over an $N+1$ space, which can be significantly large because of the large number of variables and linguistic terms involved. However, the problem is still a standard multi-dimensional optimization.
- The program starts by selecting initial values for the $N+1$ parameters, which make the starting point in the space where the global minimum needs to be found. Although an intelligent guess of the initial state may expedite convergence, such a guess is not necessary. The $N+1$ parameters can be supplied by a random number generator. During various stages of execution, the program will check these values to ensure that they make sense. For example, one obvious requirement is that

$$U_\mathrm{L} \leqslant S_\mathrm{L} \leqslant a < b < c < d \leqslant S_\mathrm{U} \leqslant U_\mathrm{U}. \tag{17}$$

  This and other constraints are examined to ensure that the input model complies with the definitions given in Eqs. (1)–(17). When the random numbers do not satisfy the model, the algorithm has various strategies for automatic adjustment of the numbers in order to match the model.
- For a given $N+1$ parameters, the program then constructs the complete model, i.e. defines all membership functions, the fuzzifier algorithm, the logical operators, and the defuzzifier.
- Each crisp value for each variable in an experimental record from the database is fuzzified according to Eqs. (5) and (6) and Fig. 1, and transformed into its equivalent linguistic form. Each possible combination of linguistic statements results in a single rule such as that in Eq. (1) or (2). Since a single crisp number may belong to more than one linguistic term, each data record can be transformed into a large number of rules, with each rule having a partial possibility. The maximum number of rules generated can be very large (Eq. (16)). For example, for $n$ variables, each having partial membership to two linguistic terms, there could be up to $2^n$ rules generated from one data record. However, in practice a much smaller number will be generated.
- The above step is repeated for each record in the database. Rules are then assembled by union operations. The result is the construction of the fuzzy rule base, $R$ in Eq. (3). Clearly, many rules will be deduced from various data records, and many of them will be repetitive. Only a small kernel of $R$ will contain possible rules. The union operation ensures that the possibility of a rule can only increase when this rule is further deduced from many data records.

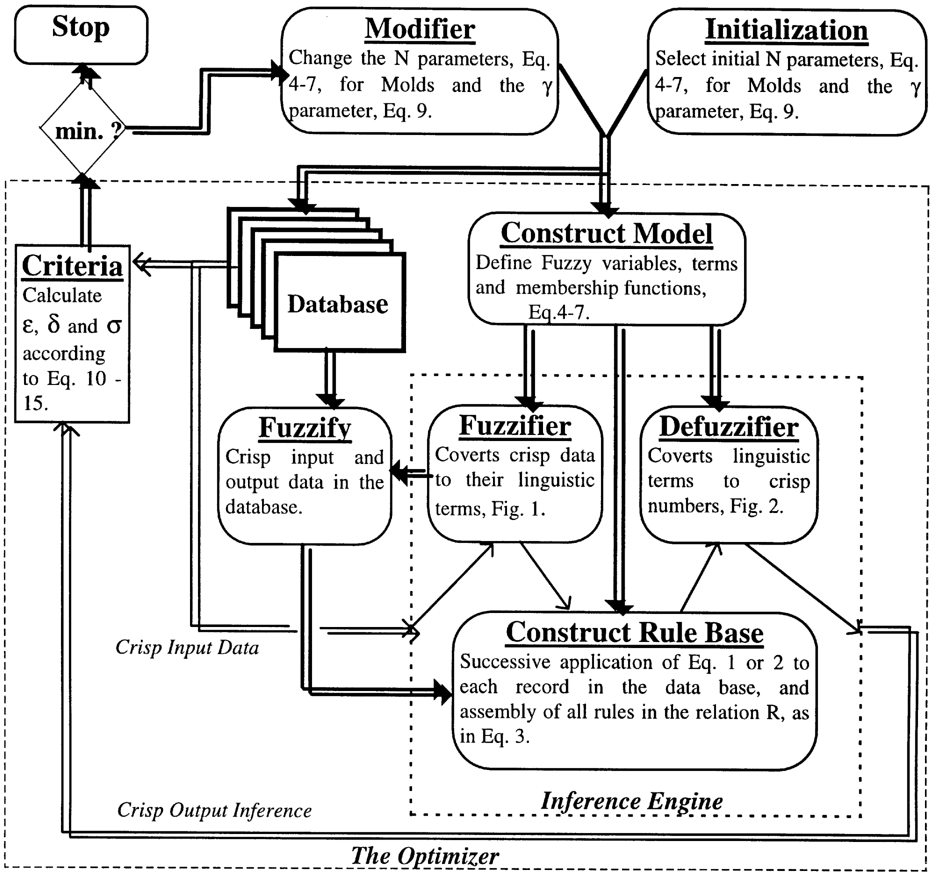



Fig. 3. Flow chart of the optimization algorithm.

- Consequently, all rules can be deduced from experimental observations without reliance on any human expert to provide the rules. Moreover, a complete inference engine has been constructed.
- Using the above inference engine and the input (antecedent) data from the database, re-prediction of the data can be done by fuzzifying them, inferring the consequent (output), and defuzzifying this output to produce the model's prediction. Ideally, the model should predict exactly the data that have been used for its construction. If not, the error in inference should be minimized according to a certain criterion from Eqs. (10)–(15).
- The function of the optimizer is to modify the model by changing the $N+1$ parameters in order to minimize the inference error or entropy. The optimizer keeps adjusting the model until the criterion is satisfied.
- A major problem that was reported by many research workers [5,17] is the entrapment in a local minimum and failure to improve the performance of the model. This problem is solved in the present algorithm where the optimizer performs three optimization tasks that ensure convergence to the global minimum, as shown in Fig. 4:
  (A) Random optimization: Searching randomly in all directions within the $(N+1)$-dimensional space and checking with the objective function. When a lower value is obtained, that point is used as a new base for the random search. This technique is very successful in finding the global minimum "zone" quickly. For a typical problem involving 150-dimensions, the

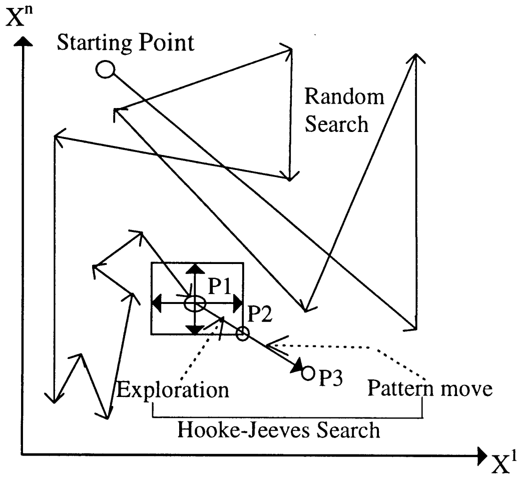



Fig. 4. The optimization algorithm.

random search was able to reduce inference error to 10% of its initial value in less than 2000 steps.

(B) After reaching the global minimum zone, a systematic Hooke–Jeeves search algorithm is used to find the global minimum. This search algorithm is very time consuming and requires many function evaluations at each step. However, when the random search ends somewhere near the global minimum, the H–J algorithm slides towards the global point very quickly and few search moves need to be done.

(C) The search for optimum $\gamma$ value is conducted in parallel with the previous stages, using a golden-sections optimization algorithm.

## 3. Examples

### 3.1. A model for residual stresses induced by grinding

Grinding is one of the most complex manufacturing engineering problems, which involves a large number of variables and physical processes that are non-linear and interdependent. Moreover, many variables have no precise numeric values, e.g. grit size or wheel grade. Therefore, grinding has been for decades a very difficult area for physical modeling or experimental investigations. Quality grinding still depends to a great extent on skilled machine operators who use rules-of-thumb based on many years of trial-and-error experience. However, modern complex surface requirements, such as induced residual stresses, are beyond everyday experience of skilled operators. Then, there is a need to generate fuzzy rules from experimental observation alone. Therefore, grinding is a process that can benefit greatly from fuzzy modeling using the present algorithm.

The process can be modeled by the implicit form

$$S = f(T, W, D),$$

where $T$, table speed, $W$, wheel speed, and $D$, depth of cut, are the major independent variables affecting the output $S$, residual stresses. There exist over twenty other variables that have lower partial effects on $S$ and over 50 other parameters affecting $S$ marginally [1]. Therefore, many variables affect the function $f$, and many of them are difficult to precisely quantify. Attempt to construct a general physical model of this system is not possible, because it is hard to account for all these variables even if precise measuring methods were available. Some empirical modeling may result in a simpler formula such as

$$S = f(H) = k \cdot H^n, \quad \text{where } H = \frac{T \cdot D}{W} \tag{18}$$

which is obviously very coarse, has a limited applicability, and often leads to contradicting results. Moreover, experimental work can be very expensive and time consuming which makes reliable experimental observations very scarce. Experimental data for the present example are based on those reported in [14].

The algorithm described in the previous sections accepts a text file as an input, see Fig. 5. Each line marked with "$**$" defines a variable in terms of its name, the universe of discourse, and the number of linguistic terms used in describing this variable. Each variable is followed by the definition of its linguistic terms, marked with "$*$". Each term is defined by its text label, the supporting subset, and the five parameters defining the shape of its membership function. This simple file format describes to the program the initial $N$ parameters defining the various linguistic terms. It also provides the database from which an initial rule base is constructed. The program keeps searching for a better set of $N+1$ parameters as shown in Fig. 3 and described previously. A comparison between the

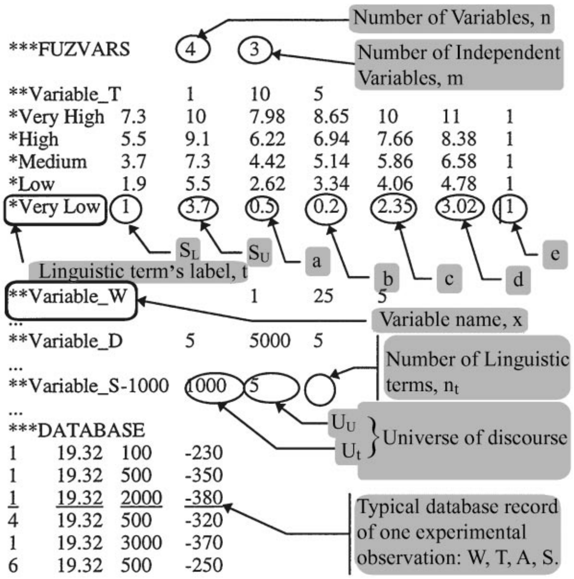



Fig. 5. The input file to the program.

original model provided by the user and the final model provided by the program is shown in Fig. 6.

Table 1 lists the rules generated by the program for both the un-optimized and the optimized models. Table 2 shows a comparison of inference accuracy among the three approaches: least-square data fitting according to Eq. (18), the original un-optimized model given by the user, and the optimized model generated by the program. The superiority of the optimized model is clearly demonstrated by a great reduction in the inference error as well as the specific entropy of the rule base. The product of the model is a true representation of membership functions, for each linguistic term, and the most robust and accurate set of fuzzy rules.

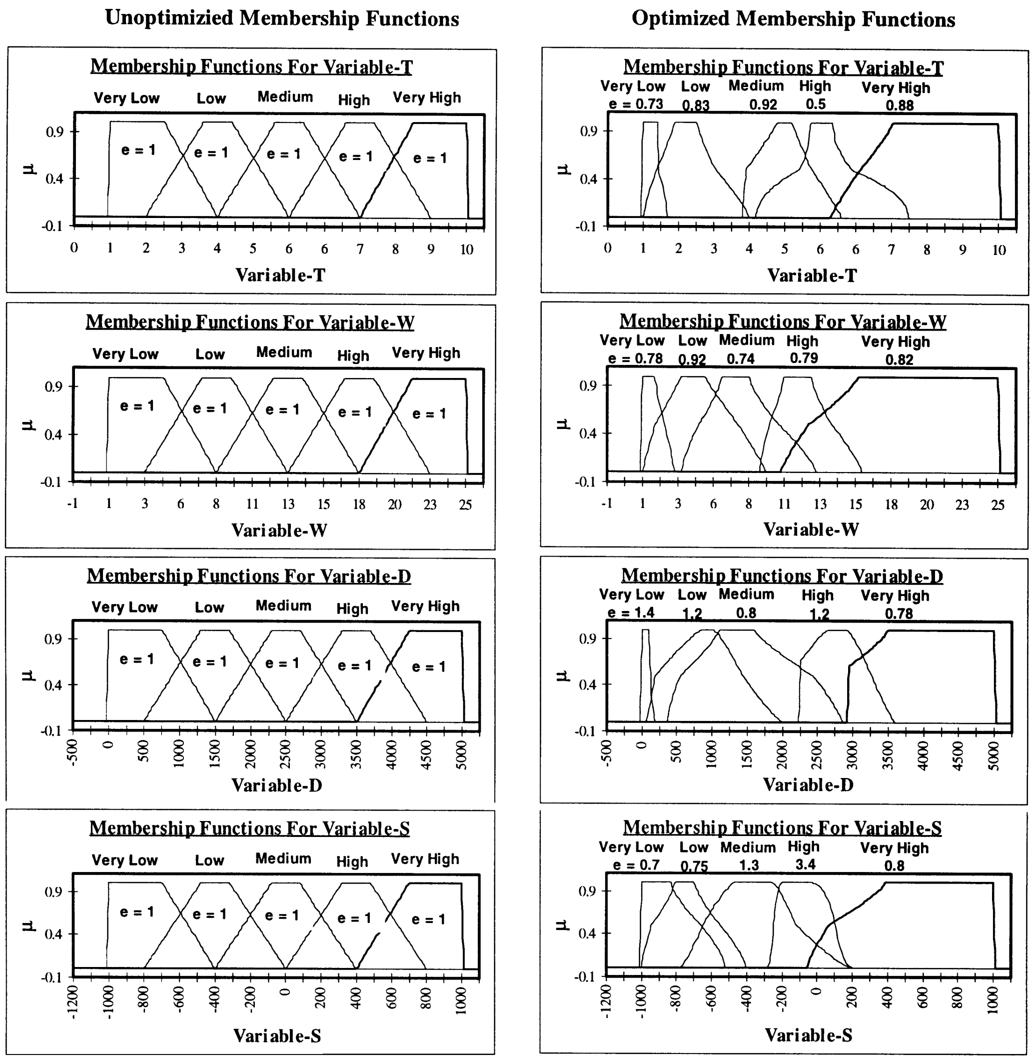
### 3.1.1. Discussion

Considering the membership functions, rules, and results shown in Fig. 6 and Tables 1 and 2, we can make the following observations:

- The common usage of triangular membership functions is not justifiable. Optimum membership functions can be un-symmetric, non-linear, and even discontinuous in shape. Some functions have interesting shapes with characteristics significantly different from the triangular shape. For example, Var_$T$: medium or high, and Var_$D$: low or high, show clear un-symmetric or non-linear discontinuous shapes.
- Multiple overlapping of membership functions is possible. For example (Var_$W = 12$) or (Var_$S = 0$) can be viewed as medium, high or very high.
- The program not only changes the shape of membership functions, but also relocates the position of each linguistic term in order to fit various rules and avoid redundancy in utilization of any term.
- Although these functions seem complex, in practical applications computational techniques, such as table-lookup, are always implemented and the complexity of the membership function has no effect on the on-line performance of the inference engine.
- The optimized model resulted in a reduction in specific entropy which is consistent with the model requirements. Entropy was reduced by both the mechanisms of rule concentration (by reducing the number of active rules from 28 to 24) and entropy dilution (by generating more deterministic rules with higher possibility weights).
- The set of rules after optimization is significantly different from the original one. By redefining membership functions, the program was able to find a completely new set of rules that are more compatible with experimental observations.
- The problem of entrapment in a local minimum has been avoided by performing a quick random search before going to the standard Hooke–Jeeves search algorithm.
- Averages of absolutes and squares of inference error have been dramatically reduced. Although the initial un-optimized model performs better than the curve-fit formula in Eq. (18), the performance after optimization is much superior to both. The inference error obtained by the final fuzzy model is almost negligible compared with the other two.

### 3.2. A fuzzy model for the Australian financial market (ASX-SPI)

The Australian financial market is a small market representing less than 2% of the global market. Performance of the Australian market is measured by the All Ordinates Index which is a weighted average of major companies registered at the Australian Stock Exchange (ASX). One of the most important financial derivatives is the Share Price Index (SPI) which




| | | | |
|---|---|---|---|
| **Average Error, ε** | *9.070%* | **Average Error, ε** | *0.037%* |
| **Average Square Error, δ** | *13.18%* | **Average Square Error, δ** | *0.060%* |
| **Specific Entropy, σ** | *0.447* | **Specific Entropy, σ** | *0.322* |

Fig. 6. Membership functions before and after optimization.

is almost identical to the All Ordinates Index. Therefore, the ability to model the ASX-SPI has both a commercial value on its own as well as an importance as a measure of the performance of the Australian economy.

There are two major approaches to the modeling of a financial product:

(1) Fundamental analysis, which assumes that the price of a financial product is determined by its fundamental value and the expected earnings. For the ASX-SPI factors such as Gross Domestic Product (GDP), productivity, revenues, unemployment, Consumer Price Index (CPI, inflation), interest & exchange rates, etc. are considered most important [7]. Unfortunately, fundamental analysis does not take into account such factors as human fear, greed, and panic which are fuzzy in nature. The ASX-SPI lost more than 40% of its

Table 1
Comparison between rule bases before and after optimization

| | IF Var_T | and Var_W | and Var_D | Then Var_S | Weight |
|---|---|---|---|---|---|
| Un-optimized rules | | | | | |
| 1 | Very low | Very high | High | Medium | 0.094 |
| 2 | Very low | High | High | Medium | 0.094 |
| 3 | Very low | Very high | Medium | Medium | 0.094 |
| 4 | Very low | High | Medium | Medium | 0.094 |
| 5 | Very low | Very high | Low | Medium | 0.062 |
| 6 | Very low | High | Low | Medium | 0.062 |
| 7 | High | Very high | Very low | Medium | 0.347 |
| 8 | Medium | Very high | Very low | Medium | 0.422 |
| 9 | Low | Very high | Very low | Medium | 0.250 |
| 10 | Very low | Very high | Very low | Medium | 0.422 |
| 11 | High | High | Very low | Medium | 0.347 |
| 12 | Medium | High | Very low | Medium | 0.469 |
| 13 | Low | High | Very low | Medium | 0.250 |
| 14 | Very low | High | Very low | Medium | 0.531 |
| 15 | Very low | Very high | High | Low | 0.422 |
| 16 | Very low | High | High | Low | 0.622 |
| 17 | Very low | Very high | Medium | Low | 0.422 |
| 18 | Very low | High | Medium | Low | 0.628 |
| 19 | Very low | Very high | Low | Low | 0.422 |
| 20 | Very low | High | Low | Low | 0.629 |
| 21 | High | Very high | Very low | Low | 0.347 |
| 22 | Medium | Very high | Very low | Low | 0.422 |
| 23 | Low | Very high | Very low | Low | 0.422 |
| 24 | Very low | Very high | Very low | Low | 0.422 |
| 25 | High | High | Very low | Low | 0.347 |
| 26 | Medium | High | Very low | Low | 0.781 |
| 27 | Low | High | Very low | Low | 0.854 |
| 28 | Very low | High | Very low | Low | 0.854 |
| Optimized rules | | | | | |
| 1 | Very high | Very high | Medium | High | 0.161 |
| 2 | High | Very high | Medium | High | 0.361 |
| 3 | Very high | Very high | Low | High | 0.161 |
| 4 | High | Very high | Low | High | 0.485 |
| 5 | Low | Very high | Low | High | 0.024 |
| 6 | Very low | Very high | Low | High | 0.074 |
| 7 | Low | Very high | Very low | High | 0.024 |
| 8 | Very low | Very high | Very low | High | 0.525 |
| 9 | Low | Very high | Very high | Medium | 0.024 |
| 10 | Very low | Very high | Very high | Medium | 0.653 |
| 11 | Low | Very high | High | Medium | 0.024 |
| 12 | Very low | Very high | High | Medium | 0.943 |
| 13 | Very high | Very high | Medium | Medium | 0.161 |
| 14 | High | Very high | Medium | Medium | 0.361 |
| 15 | Medium | Very high | Medium | Medium | 0.361 |
| 16 | Low | Very high | Medium | Medium | 0.024 |
| 17 | Very low | Very high | Medium | Medium | 0.716 |
| 18 | Very high | Very high | Low | Medium | 0.161 |
| 19 | High | Very high | Low | Medium | 0.652 |
| 20 | Medium | Very high | Low | Medium | 0.709 |
| 21 | Low | Very high | Low | Medium | 0.024 |
| 22 | Very low | Very high | Low | Medium | 0.747 |
| 23 | Low | Very high | Very low | Medium | 0.024 |
| 24 | Very low | Very high | Very low | Medium | 0.525 |

Table 2
Comparison between actual experimental observation, prediction of formula produced by curve fitting, the un-optimized fuzzy model given by the user, and the optimized model produced by the program

| | *T* | *W* | *D* | *S* | | | |
|---|---|---|---|---|---|---|---|
| | | | | Actual | Curve fitting | Un-optimized fuzzy | Optimized fuzzy |
| 1 | 1 | 19.3 | 100 | −230 | −262 | −246.6 | −229.99 |
| 2 | 1 | 19.3 | 500 | −350 | −299 | −246.6 | −349.76 |
| 3 | 1 | 19.3 | 2000 | −380 | −330 | −348.1 | −379.51 |
| 4 | 4 | 19.3 | 500 | −320 | −330 | −309.4 | −319.99 |
| 5 | 1 | 19.3 | 3000 | −370 | −340 | −348 | −370.03 |
| 6 | 6 | 19.3 | 500 | −250 | −340 | −250 | −250.02 |
| Rules | | | | | $k=-224$ | 28 | 24 |
| Specific entropy | | | | | $n=-52$ | 0.477 | 0.322 |
| Average error | | | | | 14.8% | 9.07% | 0.037 |
| Average deviation | | | | | 18% | 13.18% | 0.06% |

value during the Black Monday crash of October 1987 for no obvious fundamental reason.

(2) Technical analysis, which assumes that all information about the price of a financial product is stored in its present and previous price histories. Fluctuations in prices are thought to be random in nature and various statistical techniques are used to model the market, e.g. time series analysis, correlation, Monte Carlo method, fractals, non-linear partial differential equations such as Black–Scholes equation, and neural networks, among many other approaches [7,16,21]. All these approaches assume the existence of a probabilistic variable called "market volatility". Combined with other market theories such as efficient market hypothesis and arbitrage [21], volatility ensures that it is impossible, in principle to predict the market with absolute accuracy. There is a certain amount of uncertainty, fuzziness, or minimum entropy that must exist as a fundamental variable in the market.

In this example, we postulate that market volatility is a possibility not a probability, and therefore fuzzy modelling can be better than statistical approaches. To do so, daily historic data for various financial variables and indicators have been collected for the period from 1971 to 1996. Correlation analysis was conducted on these data series to find which variables are more influential on the ASX-SPI. It was found that the ASX-SPI has good correlation with six variables: (1) Standard and Poor's S + P500 index, or the very similar Dow Jones index, as an indicator of the American economy; (2) The Hang Seng index of Hong Kong, as an indicator of the Asian economy; (3) The US treasury bonds or bills, as a measure of interest rates and direction of global capital flow; (4) gold price index; (5) wheat price index; and (6) live cattle price index. The ASX-SPI was found to be not correlated with the Japanese Nikkei, unemployment rates, CPI, interest rates, currency exchange rates, or crude oil prices. This is compatible with the nature of the Australian economy which is linked to the US and Asian markets, and whose main earnings are driven from exports in mining (e.g. gold), agriculture (e.g. wheat) and farming (e.g. live cattle).

To avoid numerical bias, the ASX-SPI as well as the six independent variables above were normalized to a value equal to 100% on 15 April 1994. Average weekly values for the period to 18 January 1996 were used to provide the data base for the program and are shown in Fig. 7. The starting point was to define every financial variable using three linguistic terms as shown in Fig. 8. This construction resulted in an average error of 6.15% which is a very large error that corresponds to either a small market crash or a huge bull market.

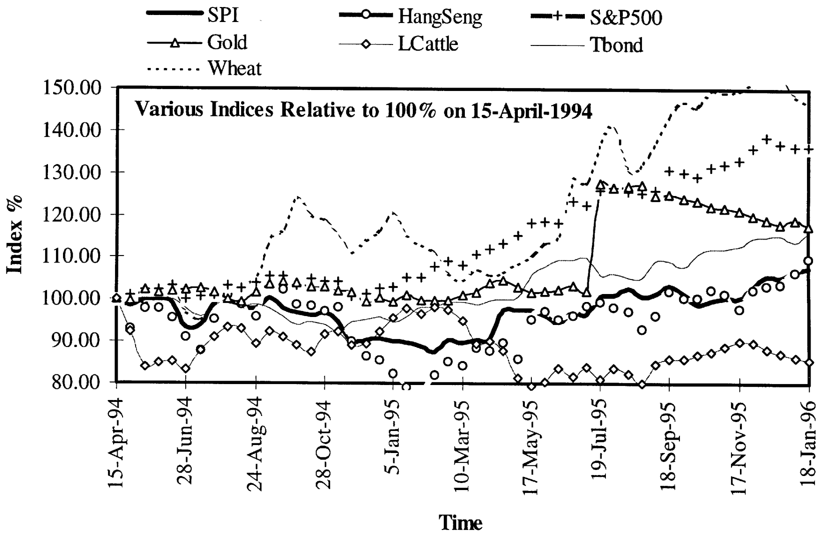
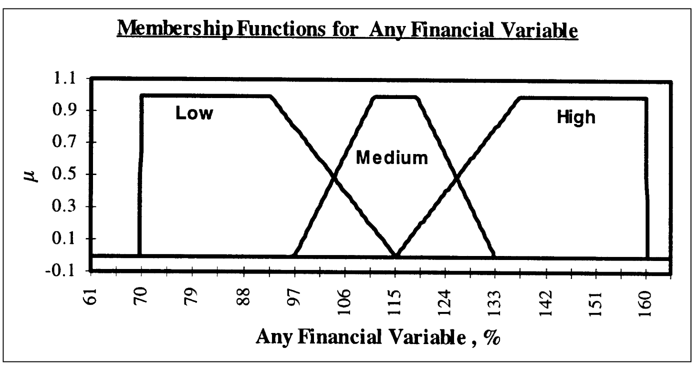



Fig. 7. Historical data for variables affecting the ASX-SPI, normalized to 100% value on April 1994.



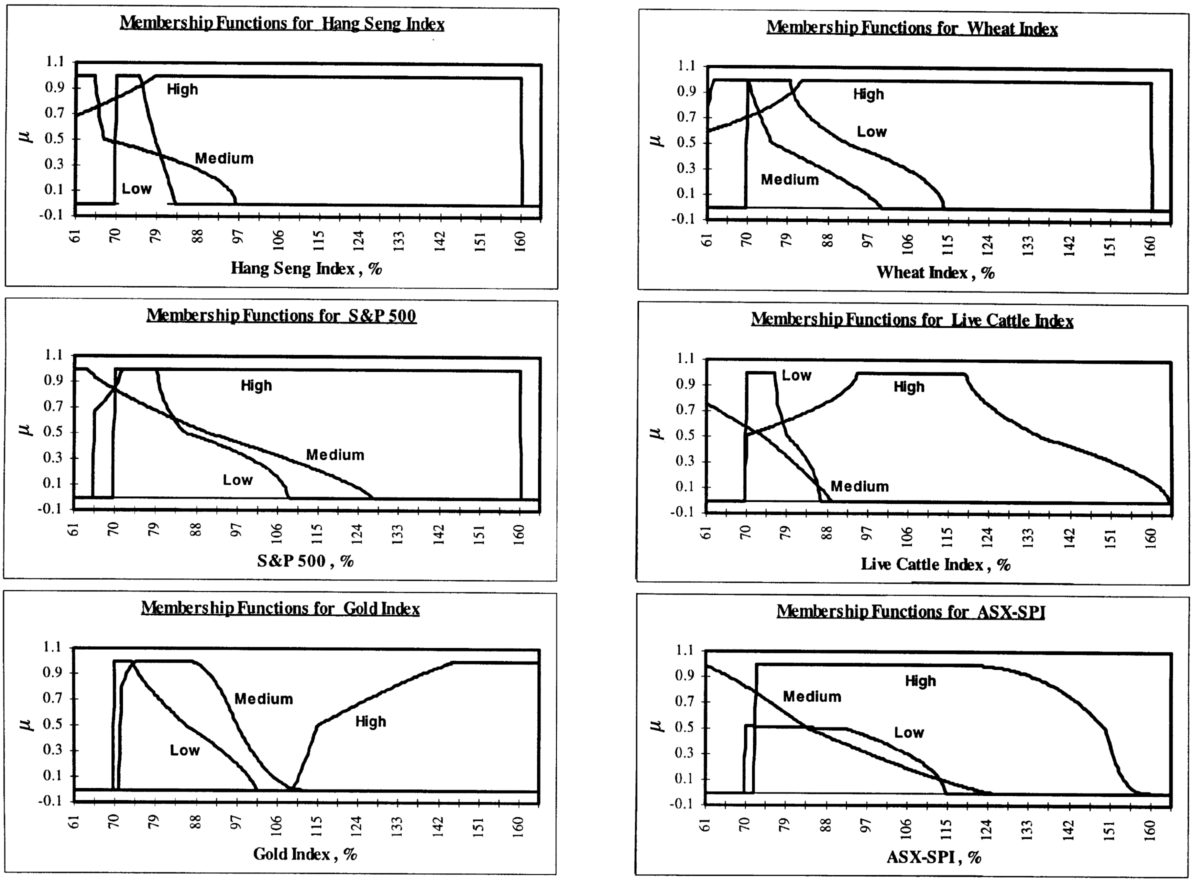

**Before Optimization**

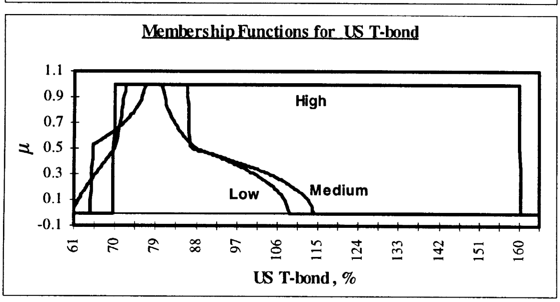
| | |
|---|---|
| **Average Error, ε** | *6.15%* |
| **Average Square Error, δ** | *7.813%* |
| **Specific Entropy, σ** | *0.31715* |
| **Total Number of Rules** | *218* |

Fig. 8. Original definitions of membership functions for the ASX-SPI model.

Daily fluctuations in the ASX-SPI can be around $\pm 1\%$ while a $\pm 2\%$ is considered a nervous trading. A weekly fluctuation of $\pm 2\%$ is not uncommon while a weekly change of more than $\pm 3\%$ represents an obvious trend. Therefore, it seems that for a medium-term market trader (a hedger), $\pm 2\%$ error in modeling the market is just as best as can be achieved, given average market volatility. When, the above model was given to the algorithm presented in this paper, and after 30 CPU hours on CM5, the model in Fig. 9 was obtained.

The membership functions produced by the algorithm look strange in shape and uncommon. Yet, they were able to reduce the average error to only 2.44%. Various observations from the discussion in the previous example apply for this example as well. One difference is that for the present example entropy minimization has been achieved primarily by entropy diffusion and the number of valid rules has increased from 218 to 486 rules. This is expected due to the inherent fuzzy nature of the system.




**After Optimization**

| | |
|---|---|
| **Average Error, ε** | *2.438%* |
| **Average Square Error, δ** | *3.401%* |
| **Specific Entropy, σ** | *0.23288* |
| **Total Number of Rules** | *486* |

Fig. 9. Optimized definitions of membership functions for the ASX-SPI model.

## 4. Conclusions

A methodology for fuzzy modeling of engineering systems has been proposed. The program developed operates on a given set of experimental observations without any other *a priori* information. The algorithm searches for the optimum configuration of membership functions, deduces the corresponding rule base such that a measure of back-inference error is minimized, without entrapment in a local minimum. Using this algorithm, fuzzy modeling of engineering systems can be constructed with a high inference accuracy and without reliance on human experience. We hope that this approach may improve the position of fuzzy modeling methods into a more deterministic form that engineers desire.

The approach is equally valid for designing control systems.

## Acknowledgements

This work is supported by an ARC Large Grant.

## References

[1] Y.M. Ali, L.C. Zhang, Estimation of residual stresses induced by grinding using a fuzzy logic approach, J. Mater. Process. Technol. 63 (1997) 875–880.
[2] R. Bellman, M. Giertz, On the analytic formalism of the theory of fuzzy sets, Inform. Sci. 5 (1973) 149–156.
[3] D. Driankov, H. Hellendoorn, M. Reinfrank, Introduction to Fuzzy Control, Springer-Verlag, Berlin, 1993.
[4] D. Dubois, H. Prade, Fuzzy Sets and Systems: Theory and Applications, Academic Press, New York, 1980.
[5] H.B. Gurocak, A.S. Lazaro, A fine tuning method for fuzzy logic rule bases, Fuzzy Sets and Systems 67 (1994) 147–161.
[6] H. Hamacher, Uber logische Aggregationen nicht-binar expliziter, Entscheidungskriterien, Frankfurt/Main.
[7] S.J. Harris, Trading 101, Wiley, New York, 1996.
[8] A. Kandel, Fuzzy Mathematical Techniques with Applications, Addison-Wesley, Reading, MA, 1986.
[9] B. Kosko, Fuzzy Thinking: The New Science of Fuzzy Logic, Harper Collins, London, 1993.
[10] B. Kosko, Neural Networks And Fuzzy Systems: A Dynamical Systems Approach To Machine Intelligence, Prentice-Hall, Englewood Cliffs, NJ, 1992.
[11] E.H. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, Int. J. Man Mach. Studies 7 (1975) 1–13.
[12] D. McNeill, P. Freiberger, Fuzzy Logic, Simon & Schuster, New York, 1993.
[13] M.M. Nelson, W.T. Illingworth, A Practical Guide to Neural Nets, Addison-Wesley, Reading, MA, 1991.
[14] H. Noguchi, T. Waida, T. Suto, Proc. 1st Int. Abrasive Technology Conf., Seoul, 1993, p. 7.
[15] M. Peschel, Preface, in: S. Bocklisch et al. (Eds.), Fuzzy Sets Applications, Methodological Approaches, and Results, Akademie_verlag, Berlin, 1986.
[16] A.N. Refenes, P. Bolland, Modeling quarterly returns on the FTSE, in: C.H. Chen (Ed.), Fuzzy Logic and Neural Network Handbook, McGraw-Hill, New York, 1996, p. 19.
[17] K. Shimojima, T. Fukuda, Y. Hasegawa, Self-tuning fuzzy modeling with adaptive membership function, rule, and hierarchical structure based on genetic algorithm, Fuzzy Sets and Systems 71 (1995) 295–309.
[18] T. Tilli, High performance software implementations of fuzzy logic algorithms, Fuzzy Sets and Systems 66 (1994) 233–240.
[19] C. Von Altrock, Fuzzy Logic and Neuro-Fuzzy Applications Explained, Prentice-Hall, Englewood Cliffs, NJ, 1995.
[20] B. Werners, Aggregation models in mathematical programming, in: G. Mitra (Ed.), Mathematical Models for Decision Support, Heidelberg, Berlin, 1988.
[21] P. Wilmott, S. Howison, J. Dewynne, The Mathematics of Financial Derivatives, Cambridge University Press, Cambridge, 1995.
[22] R.R. Yager, On a general class of fuzzy connectives, Fuzzy Sets and Systems 4 (1980) 235–242.
[23] L.A. Zadeh, Fuzzy sets, Inform. and Control 8 (1965) 338–353.
[24] H.J. Zimmermann, Fuzzy Set Theory and its Applications, Kluwer Academic, Boston, 1991.