y in Artificial Intelligence. Amsterdam:
J. E. (1978). A Manual for Encoding
nlo Park, California: SRI International.
stification and the psychology of inductive
f Experts. Bloomington, Indiana: Indiana
mmetric neural control systems in human
215.
. Contingent Weighting in Judgment and
niversity, Stanford, California.
ory: Foundations, Development, Applica-
heory of Games and Economic Behavior.
cision Analysis and Behavioral Research.
Encoding subjective probabilities: A
ment Science, 29, 151–173.
s. Presentation to the Utah Academy of
ce and expectation: Some preliminary
SON. Jr. Eds. Minnesota Studies in the
polis; University of Minnesota Press.
cientific Research. Hillsdale, New Jersey:
al Systems Thinking. New York: Wiley.
s underlying mathematical models. In J.
eric Computing. Amsterdam: Elsevier.
le of calculi in uncertain reasoning. In
certainty in Artificial Intelligence, Seattle,
A Graphical Decision Support System.
lic Policy, Carnegie-Mellon University.
ng Computers and Cognition. Reading,
r comparing uncertain inference systems
EMMER, Eds., Uncertainty in Artificial
is a weak predictor of performance.
ertainty in Artificial Intelligence, Seattle,
DRICK, R. (1987). The role of tuning
3rd AAAI Workshop on Uncertainty in

Int. J. Man–Machine Studies (1990) **32**, 187–201

Fuzzy windows and classification systems

J. C. SANTAMARINA† AND J. L. CHAMEAU‡

†Department of Civil and Environmental Engineering, Polytechnic University, Brooklyn, NY 11201, USA and ‡School of Civil Engineering, Purdue University, West Lafayette, IN 47907, USA

(Received 15 June 1987)

A fuzzy sets based structure for classification systems is proposed in this paper. The basic idea is to use "windows" to represent the constraints on the possible values variables may take. The formalism is very simple, however, this simplicity makes it attractive in the development of knowledge based systems. The most salient features of this structure include the possibility of developing composite solutions, searching for lacunae, and creating a case based representation of knowledge with an avenue for modeling learning.

Introduction

Production systems have been used to solve classification problems which consist of selecting an alternative from a set of possibilities. The selection is usually based on the most relevant parameters, leaving others aside, and thus creating uncertainty in the result. In addition, there may be uncertainty in the specification of the parameters relevant to a given alternative during system development, as well as uncertainty in the input information corresponding to a particular case when running the system. Incorporating all these uncertainties in a classical architecture, such as a production system, is at least cumbersome and can lead to a system which is difficult to calibrate.

A fuzzy sets based structure for classification systems is proposed in this paper. It consists of a computer simulation of physical classification in which measurements are performed and results are compared to the ranges in value of alternative classes. Matching a measurement to the acceptable range for a parameter within an alternative is like forcing the former through a filter or "window": the similarity between input and output is an index of the compatibility between the measurement and the acceptable range, or standard.

This paper first presents the basic characteristics of the "fuzzy window" approach; the simple applications used for illustration come from the civil engineering field, however, classification problems abound in most domains. Emphasis is then placed on the potential that the technique may have to address some issues relevant to knowledge systems, such as search for lacunae and development of composite solutions.

0020-7373/90/020187 + 16$03.00/0

Windows

A "segment" is defined by its extremes and the line joining them. Mapping the possible range of variation of any variable onto a segment is a simple representation of such variable. For example, the permeability of soils (noted k) can be represented by the segment ranging from extremely low ($k = 10^{-9}$ cm sec^{-1}) to extremely high ($k = 10^{2}$ cm sec^{-1}) on a logarithmic scale:

variable: permeability
range: 10^{-9} to 10^{2}
Segment: X. Low|--------------------------------|X. High

A "window" is a subset of a segment, and represents the range of values that are of particular interest. In most situations the boundaries of the windows cannot be defined with certainty. Indeed, the belief in values within the window may gradually change from non-acceptable to acceptable, or from 0·0 to 1·0 if a numerical scale is used. This transition may be due either to a real transition of interest, or to the vagueness in the information available to define the window. Then, the representation of knowledge is "fuzzy". For example, the permeability of the soil type

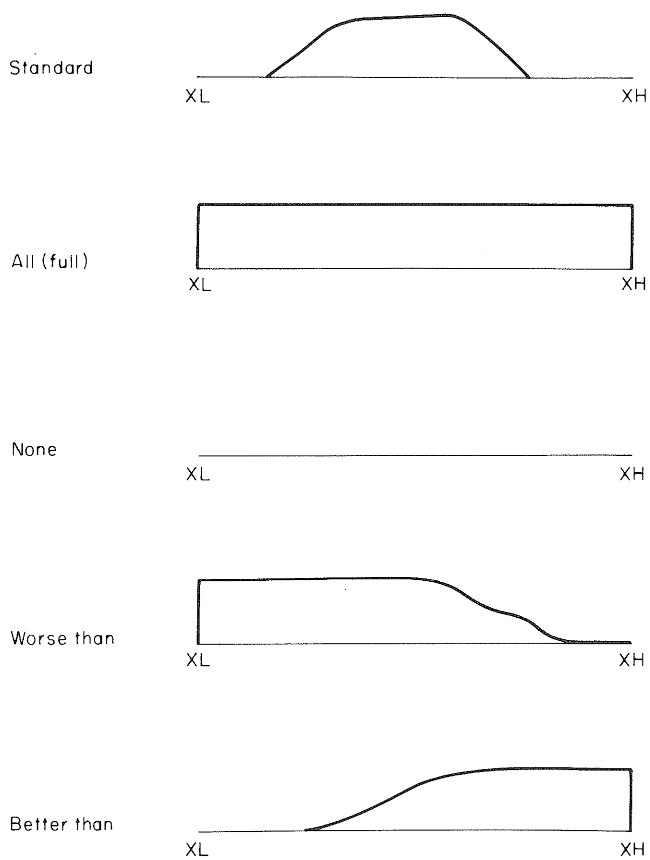
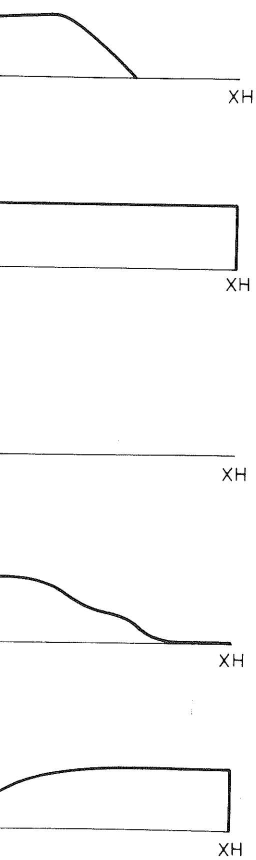


FIG. 1. Alternative forms of windows. Typical examples. (Note: XL and XH indicate the lower and upper bounds of the segment, respectively).

the line joining them. Mapping the
a segment is a simple representation
eability of soils (noted k) can be
xtremely low ($k = 10^{-9}$ cm sec^{-1}) to
mic scale:

High

epresents the range of values that are
oundaries of the windows cannot be
ues within the window may gradually
from 0·0 to 1·0 if a numerical scale is
real transition of interest, or to the
efine the window. Then, the repre-
ole, the permeability of the soil type



(Note: XL and XH indicate the lower and
, respectively).

"sand" could be represented as follows (note: null values in the window are omitted for simplicity):

variable: permeability
segment: (10^{-6} 10^{-5} 10^{-4} 10^{-3} 10^{-1} 10^{0} 10^{1})
interest: permeability of a sand
fuzzy window: (0·2 0·6 1·0 1·0 1·0 0·6 0·1)

The fuzzy set representing the permeability of a sand is called a "fuzzy window" herein to emphasize the concept of a constraint on the possible values a variable may take; it is equivalent to looking through a window with a varying level of transparency. Alternative forms of windows include monotonic and unimodal shapes (Fig. 1); a "full" window is identical to the segment and represents a dimension that makes any input acceptable (this is non-informative for classification purposes). The null window "none" makes any input unacceptable. The idea of a window to represent information is obviously very simple, however, this simplicity makes it efficient in the development of decision and knowledge based systems.

Discrimination scheme using windows

The use of fuzzy windows in decision making will be illustrated through the soil permeability example: One may decide whether a soil is a sand based upon some information on its permeability. The following two steps are proposed to achieve this: (1) "filter" the fuzzy set that characterizes the permeability of the given soil (input) through the window for the permeability of a sand as described above; and (2) compare the "filtered" output with the input to determine the quality of the match.

Intersection was selected to perform the filtering process because it permits efficient implementation while satisfying some basic conditions: (1) if the input is fully contained by the window, then the output is the same as the input; (2) if the input contains the window, then the output is the window; and (3) in all cases the output is a subset of the window and the input. Other operations could have been selected including compensatory operators (Dubois & Prade, 1981; Zimmermann, 1985), however, for the applications that were considered by the authors, the intersection operation gave satisfactory results. The following example illustrates the filtering process given the estimated permeability of a soil sample A:

interest: permeability of a sand
fuzzy window: (0·2 0·6 1·0 1·0 1·0 1·0 0·6 0·1)
(standard)
case: permeability of sample A
fuzzy window: (0·6 1·0 0·3 0 0 0 0 0)
(input)
question: is sample A a sand?
match: (0·2 0·6 0·3 0 0 0 0 0)
(output)

There are several alternatives to assess the quality of the match between input and output. A simple approach, that may suffice in some cases, is to take the maximum membership value of the filtered output as a measure of the match between the input and the standard window. For the example above, this value is 0·6, indicating a low possibility for sample A to be a sand.

Using the maximum membership value is simple, however, it usually does not provide a very high level of discrimination between different alternatives. The approach followed herein is to determine the "resemblance" between input and output by using the ratio of fuzzy cardinalities of the output and the input (this is equivalent to the fuzzy-similarity operation). The cardinality of a set defined in discrete form is the summation of its membership values, and thus the ratio of cardinalities, FCR, can be graphically conceived as the ratio of the surface areas of output and input. For the previous example:

$$\text{FCR(sample } A) = 1{\cdot}1/1{\cdot}9 = 0{\cdot}58$$

In a classification system, such FCR values will be calculated for different attributes and alternatives.

Classification systems

Many decision processes involve several variables or dimensions. Then, the acceptability of an alternative is a function of how well it matches the characteristics of a problem with respect to these variables or dimensions. Classification type knowledge-based systems select the alternative that best suits a given case. Most of these systems are based on the production system architecture, searching the space of alternatives with "IF-THEN" rules. An alternative to this approach is proposed here for decision/classification problems that can be idealized as indicated in Fig. 2: each possible solution is represented with a "stack of windows", one window being specified for each relevant dimension. Then, decision making can be modeled as a process that "filters" the characteristics of a problem through the stack of each alternative (Fig. 2). The acceptability of an alternative is a function of the quality of individual matches; the best solution is that which fits best the characteristics of the problem (input).

An expert in a given domain does not need to check the acceptability of every alternative solution for every dimension (attribute) considered. Doing so would be inefficient, and many problems would become practically unsolvable. A better approximation to the expert's approach is to consider one potential solution, e.g. alternative X, and test its acceptability under a relevant dimension. If it is fully acceptable, then alternative X is tested under another dimension. If it is not fully acceptable, a new choice is made, e.g. alternative Y, and the process is repeated; Alternative X is then kept "on hold" for potential reconsideration in case all other solutions prove to be less acceptable.

This "intelligent" classification process is modeled with a heuristic best-first search procedure in a system named FUZWIN (FUZzy WINdows) written in LISP. The windows for the dimensions (attributes) of each alternative are organized in "stacks" (Fig. 3). Each stack is assigned an initial acceptability value of 1·0, indicating that if nothing is known about a problem, all solutions are equally likely to be acceptable. When search begins, FUZWIN considers the stack of the first

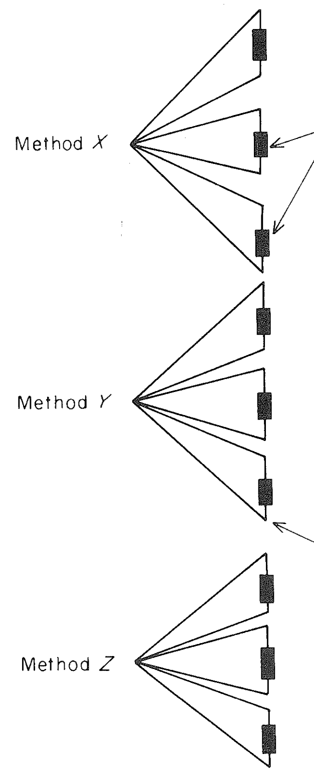


FIG. 2. Representation o

alternative until its acceptability fall
happens, it re-orders the stacks plac
simplicity, the re-ordering of stacks i
representative of the search path).
tested under all dimensions while ke
represented by such stack is selected

Every time a window is encount
match by calculating the fuzzy-simil
window; then it takes the minimum
and the quality of the match, and
process is illustrated in Fig. 3b. D
most critical dimension in each stac
stack to its current value); this featu

When preparing a knowledge ba
stacks of windows), the order of win
the optimum solution. However,
windows should go higher in the st
stacks in the database is not very im
(windows) that are considered will

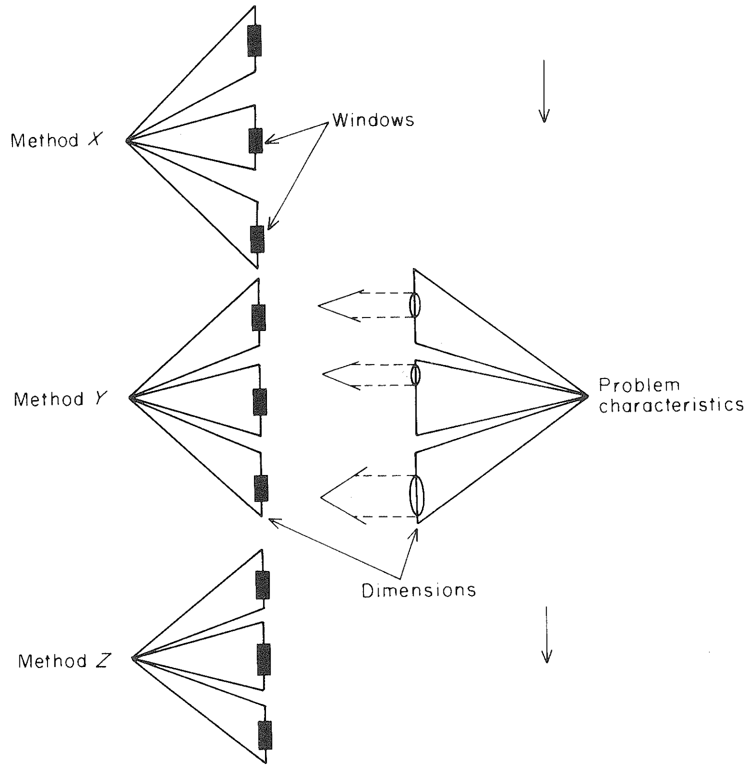


FIG. 2. Representation of a classification system using windows.

alternative until its acceptability falls below that of any other (Fig. 3*a*). When this happens, it re-orders the stacks placing the one with highest acceptability first (for simplicity, the re-ordering of stacks is not shown in Fig. 3*a*; instead the dotted line is representative of the search path). The process continues until one stack has been tested under all dimensions while keeping the highest acceptability. The alternative represented by such stack is selected as the most appropriate decision.

Every time a window is encountered FUZWIN determines the quality of the match by calculating the fuzzy-similarity between the input and the output of the window; then it takes the minimum between the previous acceptability of the stack and the quality of the match, and assigns this value as the new acceptability. The process is illustrated in Fig. 3*b*. During the search, FUZWIN keeps track of the most critical dimension in each stack (the one that lowered the acceptability of the stack to its current value); this feature is important to implement explanations.

When preparing a knowledge base (defined by alternatives and their associated stacks of windows), the order of windows in the stacks is irrelevant for the choice of the optimum solution. However, in terms of efficiency, the most constraining windows should go higher in the stack to minimize search. The initial order of the stacks in the database is not very important as in most cases the first few dimensions (windows) that are considered will reorder the whole database. Nevertheless, two

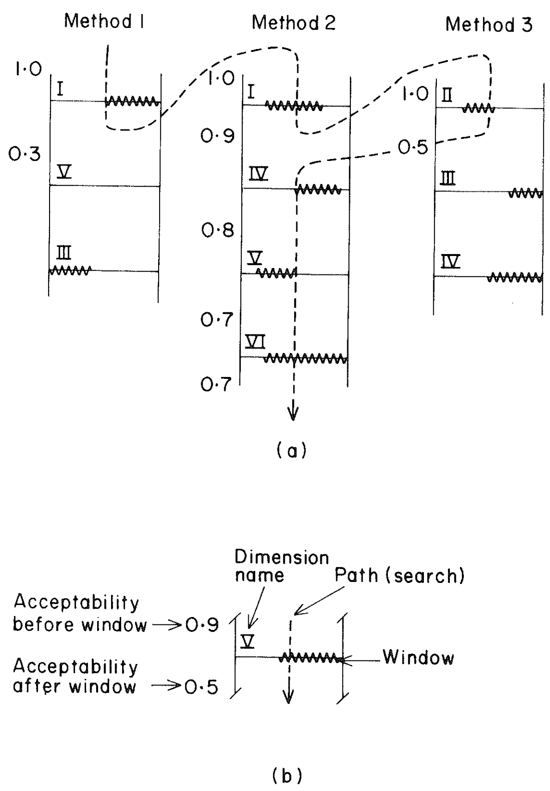


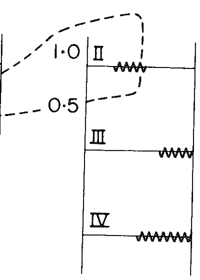
FIG. 3. Fuzwin search procedure. (*a*) Search path, and (*b*) filtering.

approaches are suggested: (1) place the stacks for the alternatives most commonly used first; or (2) place the stacks with the least constraining windows first. The most important factor in the system's performance is the implementation of an efficient sorting algorithm for the reordering of the stacks in the database.

Example: excavation methods

The selection of excavation methods for tunnels is dictated by several relevant attributes like: possible capital investment, required advance rate, length of the tunnel, acceptability of vibrations, variability in rock, need to access the face of the tunnel, importance of contour control and damage, rock type, available man power, and certainty of execution. There are no clearly defined boundaries on these dimensions to indicate when one method should be used.

For simplicity, let us consider only two excavation methods, full-face tunnel boring machines and drill-and-blast, and the first six dimensions listed above. The two stacks of windows forming the database were developed by the authors and are shown in Appendix I (this database is called *DB*). Also shown are the support values for each dimension (database *DIMENSIONS*), the introductory statements to the use of the system (*GENERAL*), and the list of explanations justifying each dimension (*WHY*). It is interesting to note that three of the attributes use



rch path, and (*b*) filtering.

or the alternatives most commonly
onstraining windows first. The most
the implementation of an efficient
s in the database.

nels is dictated by several relevant
uired advance rate, length of the
rock, need to access the face of the
e, rock type, available man power,
arly defined boundaries on these
be used.
avation methods, full-face tunnel
t six dimensions listed above. The
e developed by the authors and are
DB*). Also shown are the support
ONS*), the introductory statements
e list of explanations justifying each
that three of the attributes use

TABLE 1
Sample run using FUZWIN—excavation methods

→(instructions)
(SELECTION OF EXCAVATION METHOD FOR TUNNELS: the importance of a proper selection of an excavation method lies in a more successful and economical execution of the job)

→(search)
HOW IS: capital-investment
FORMAT: (1000 10000 100000 1000000$)
⇒why

I am considering full-face-TBM, which has a current acceptability of 1·0. The analysis of capital-investment is important because tunnel boring machines require quite high initial captial investment and can make the choice of this solution not feasible giving preference to blasting

HOW IS: capital-investment
FORMAT: (1000 10000 100000 1000000$)
⇒(1 1 1 0·5)

HOW IS: tunnel-length
....

THE SELECTED METHOD IS........: full-face-TBM
WITH AN ACCEPTABILITY OF ....: 0·36
MOST CRITICAL CONDITION ......: need-to-access-face

→(search)
THE SELECTED METHOD IS........: drill-and-blast
WITH AN ACCEPTABILITY OF ....: 0·12
MOST CRITICAL CONDITION ......: required-advance-rate

linguistic variables; in this case, linguistic terms are transformed to numerical values through a translator built in the system (Santamarina & Chameau, 1987).

Excerpts of a sample run are shown in Table 1 where user's input follows the "→" or "⇒" signs. The example is obviously contrived since only two methods were considered; however, it illustrates the basic steps followed by FUZWIN: (1) request necessary information (e.g. HOW IS: capital-investment); (2) select solution with highest acceptability (full-face-TBM in this case), indicating the most critical attribute that guided the selection (need-to-access-face); and (3) allow the user to search for less acceptable solutions.

Fuzzy windows and artificial intelligence

The FUZWIN system has been used to develop experimental knowledge based systems in civil engineering, including IMPROVE, a system to help the selection of methods to improve the characteristics of soils e.g. to increase their structural capabilities (Chameau & Santamarina, 1989). This gave the authors the opportunity to evaluate the potential role of windows in Artificial Intelligence (AI) applications by implementing some of the general features of classical expert systems into FUZWIN. Furthermore, experience to date with FUZWIN indicates that concepts not easily modeled in classical knowledge systems may be supported by the window approach.

The goals of this section are twofold: (1) show that features of classical systems can be incorporated in a FUZWIN type system; and (2) show that the fuzzy window approach to represent knowledge may prove very useful to a number of AI concepts. Emphasis is given to this latter goal.

1. *Knowledge Acquisition—Consensus.* Strategies for knowledge acquisition like "Characteristics and Decisions" and "Solution–Exemplification", in combination with the "Interval Estimation" method to develop membership functions (Chameau & Santamarina, 1987) seems most appropriate for the type of problems FUZWIN is meant to solve. Using this method, a procedure "(create)" was added to FUZWIN to facilitate knowledge elicitation.

It is often suggested that information should be elicited from only one expert, to avoid the difficulties that may result from contradicting information and decision processes. Still, confidence may be higher for systems built with more than one expert. In this case, the fuzzy window mode of knowledge representation can prove very flexible and powerful in facilitating consensus. Minor differences in the knowledge from different sources can be resolved for instance by averaging the elicited windows which results in an extension of the transition zone, from 0·0 to 1·0 membership value. Hence, the worse the agreement the higher the fuzziness of the window. Significant differences (windows may be considered "significantly different" when they differ in more than 10 to 20% of the enclosed region) can be either resolved by re-analysing the dimension with the experts, or by judiciously using other information. For example, major differences may correspond to instances when one of the sources acknowledges limited experience for the given conditions.

2. *Explanations.* Explanation capabilities are an attractive feature of knowledge systems. The window structure easily supports explanations: if a user asks the system "why" a parameter is requested, the system provides the stack that it is considering, its current acceptability, and the importance of the particular dimension.

When the best solution is found, the system provides its acceptability and the most critical dimension involved in the selection. The user may request other solutions to be investigated, and may also check an alternative that was not chosen. In this case FUZWIN responds with the most critical dimension in that stack.

3. *Combining Alternatives.* The solution to a problem may require a combination of two or more of the alternatives in the knowledge base. The ability to generate a combined stack of windows for a composite method, and to test its potentials for the given case is a major advantage of the FUZWIN type of knowledge representation. This is more difficult in classical expert systems, since they are essentially restricted to those composite methods that were included in the knowledge base at the time of its development.

FUZWIN permits the dynamic generation of composite solutions. Four steps are required: (1) selecting the alternatives to be combined; (2) generating the combined stack; (3) filtering the problem data; and (4) obtaining the response. Steps 3 and 4 are processed with the same procedure as used for the general case.

Experiments with the generation of combined stacks (step 2) showed several difficulties. The approach consists of "propagating the constraints", i.e. the fuzzy

windows defining each of the method
given dimension, one may be intuitiv
windows as the combined window. H
by minimization, then the acceptab
most the minimum acceptability of
purpose of composite solutions. In
maximization, and there are still oth
as weighted average of windows, m
authors' experiments showed that th
alternative and dimension depende
simplest cases.

The "(combine)" operation imple
two alternatives, which can then be
system asks the user to chose the
relevant dimensions for these altern
are available. It then proceeds to
composite stack and responds with

The process, although cumberso
distinct feature of allowing for the r
with other knowledge representati
logical AND-OR operators is imple
less reaching and useful solution th
for the composite alternative dev
database of methods; then, the n
alternative will be considered amo
generate composite windows; for
alternatives to be combined accord
that are not.

4. *Validation and Lacunae.* Dev
in knowledge. Experience with pro
partial forms of knowledge repre
existing knowledge in the domain,
the contrary, windows are particula
problem: comparing the windows
the immediate observation of the
any of the methods in the databa
union of all windows in the same
indicate gaps. Figure 4 demonstrat

The command "(lacunae)" in F
the conjunction of any two. If on
dimensions are involved, the alter
responding "unknown" to all othe
a parametric study with these latte
(those that have 1·0 in all the ra
making, however they are relev
incorporated in the knowledge ba

The possibility of searching f

windows defining each of the methods, to the stack of the composite solution. For a given dimension, one may be intuitively guided to choose the most restrictive of the windows as the combined window. However, if all constraints were to be combined by minimization, then the acceptability of the composite alternative would be at most the minimum acceptability of the individual alternatives, contradicting the purpose of composite solutions. Indeed some combinations may be done with maximization, and there are still other circumstances where an other operator, such as weighted average of windows, may seem more appropriate. Unfortunately, the authors' experiments showed that the choice of the combination operator is not only alternative and dimension dependent but may also be case specific in all but the simplest cases.

The "(combine)" operation implemented in FUZWIN allows the user to combine two alternatives, which can then be recombined with a third one, and so on. The system asks the user to chose the combination operator as it lists each of the relevant dimensions for these alternatives: union, intersection and average operators are available. It then proceeds to filter the data for a given problem through the composite stack and responds with the acceptability of the composite solution.

The process, although cumbersome because of the required interaction, has the distinct feature of allowing for the real time combination of solutions, a difficult task with other knowledge representation methods. A pseudo-combination based on logical AND-OR operators is implemented in some expert systems, however, it is a less reaching and useful solution than the approach just described. Finally, the stack for the composite alternative developed during a session may be added to the database of methods; then, the next time the system is run the new composite alternative will be considered among all others. Other approaches could be used to generate composite windows; for instance the system could automatically select alternatives to be combined according to the constraints that are fulfilled and those that are not.

4. *Validation and Lacunae.* Developing expert systems may help discover lacunae in knowledge. Experience with productions (IF-THEN rules) indicates that they are partial forms of knowledge representation that do not provide a global view of existing knowledge in the domain, thus reducing the possibility of finding gaps. On the contrary, windows are particularly useful to discover unexplored dimensions in a problem: comparing the windows of all alternatives for a given dimension leads to the immediate observation of the gaps in that dimension, i.e. areas not covered by any of the methods in the database. Numerically, this is equivalent to finding the union of all windows in the same dimension: zones where belief departs from unity indicate gaps. Figure 4 demonstrates this concept.

The command "(lacunae)" in FUZWIN checks for gaps in a given dimension or the conjunction of any two. If one wishes to test for lacunae when more than two dimensions are involved, the alternative is to use the standard "(search)" procedure responding "unknown" to all other dimensions but the ones of interest, and running a parametric study with these latter dimensions. Note that non-constraining windows (those that have 1·0 in all the range of the dimension) are not needed in decision making, however they are relevant in the detection of lacunae and should be incorporated in the knowledge base for that purpose.

The possibility of searching for lacunae with respect to several dimensions

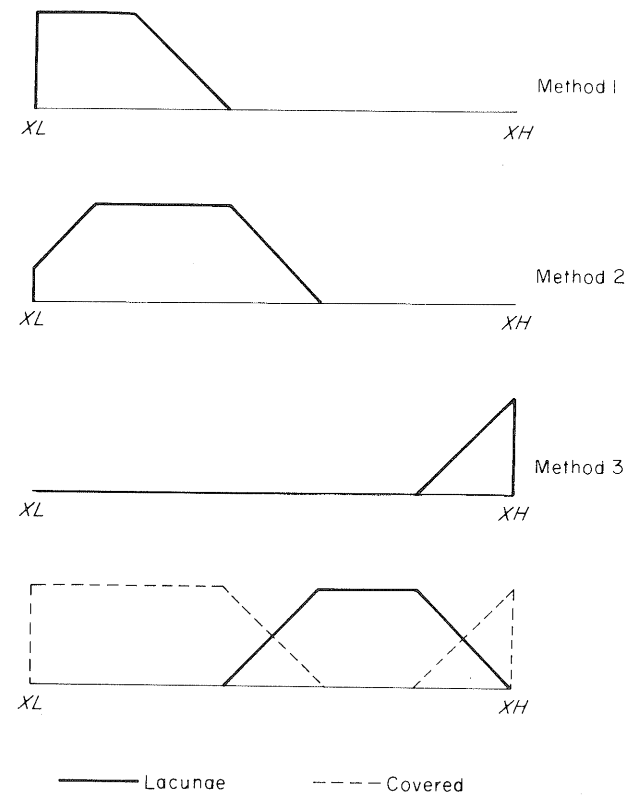


FIG. 4. Lacunae and Fuzzy windows. (Note: XL and XH indicate lower and upper bounds of the segment, respectively).

(attributes) is a very important feature of the window approach. Existing knowledge is often adequate (i.e. "complete") if one is concerned with only one attribute relevant to a decision, however gaps in knowledge tend to increase significantly if several relevant attributes are to be fulfilled at the same time. For example, in the soil improvement problem that the authors investigated (Santamarina, 1987; Chameau & Santamarina, 1989), they are methods of soil improvement to cover almost any desirable depth (i.e. complete coverage with respect to depth attribute). However, gaps start to appear if several attributes, e.g. depth, time of execution, location of layer to be treated, etc. are considered at the same time.

Experiments with FUZWIN in different domains showed that the most important consequence of the search for lacunae is system validation: at each stage of prototype development, a list of gaps can be generated and scrutinized to see if they correspond to errors or to missing information in the knowledge base.

5. *Evaluation Strategies.* Individuals use different evaluation strategies, depending on the perception of the problem and interests, task characteristics, features of the particular case, etc. The lack of a unique methodology is a major difficulty in the development of expert systems, and their validation.

Wallsten (1980) classified the evaluation strategies in two main categories; their meaning in the context of windows follows: (a) wholistic strategy: each stack includes

all the most relevant dimensions and t
(b) dimensionwise strategy: the most
and all stacks are evaluated with re
subdivided in three subgroups: (1) thı

Most knowledge systems availabl
Classification systems based on fuzzy
flexible to implement several evalua
FUZWIN corresponds to the holi
dimensionwise strategy by respondin
one that is considered critical. In addit
Both models have been tested in a v
in the following observations: (1) th
tends to favor the choice of a stack; (
more efficiently in discarding unac
selecting a unique evaluation functio
one evaluation function in some decis

Lastly, it is important to note that
evaluation function; in fact the six f
used with the same search algorithm.

6. *Relaxation.* Studies in group de
McDaniel Johnson, 1980) show that
presentation of ideas, disagreement,
third stage consists of making positio
members of the group. If an initi
constraint to the set of possibilities,
tool to model relaxation. (Note: cons
is achieved by means of utility functi

Relaxing fuzzy constraints is equiv
all levels of the scale when μ is le
studied: pure dilation and translatio
model for the effect of hedges (Zade
of all membership values, i.e. an c
linear formulation was chosen to rep

μ_r =

where μ_r is the relaxed membership

A procedure "(relax)" that "syste
of alternatives was added to FUZV
$c = 0{\cdot}2$. Initial experiments with this
selected alternatives have obviousl
database is used; (b) relaxation does
and (c) it is possible that relaxation
systematic and is only applied to cer

Because of the previous observa
implemented to "selectively relax" t
perception. The procedure called "(
dimensions that characterize the cas

all the most relevant dimensions and the selected strategy is that which rates highest; (b) dimensionwise strategy: the most salient dimension of the problem is selected and all stacks are evaluated with respect to it. Each of these groups is further subdivided in three subgroups: (1) threshold, (2) addition, and (3) average.

Most knowledge systems available today include a fixed evaluation model. Classification systems based on fuzzy windows and best first search are particularly flexible to implement several evaluation strategies. The standard model built in FUZWIN corresponds to the holistic type, however, the user can force a dimensionwise strategy by responding "unknown" to all dimensions except for the one that is considered critical. In addition, a wholistic-additive model is also available. Both models have been tested in a variety of domains and circumstances, resulting in the following observations: (1) the wholistic-additive function is forgiving and tends to favor the choice of a stack; (2) the wholistic-thresholding function performs more efficiently in discarding unacceptable alternatives; and (3) difficulties in selecting a unique evaluation function may indicate that individuals use more than one evaluation function in some decision processes.

Lastly, it is important to note that heuristic search is not impaired by changing the evaluation function; in fact the six forms suggested by Wallsten (1980) can all be used with the same search algorithm.

6. *Relaxation.* Studies in group decision making and group creative thinking (e.g. McDaniel Johnson, 1980) show that there are four stages in a successful process: presentation of ideas, disagreement, relaxation of initial positions, and results. The third stage consists of making positions more flexible to better match those of other members of the group. If an initial position statement can be expressed as a constraint to the set of possibilities, then the window approach may be an excellent tool to model relaxation. (Note: constraint relaxation in mathematical programming is achieved by means of utility functions and relaxation bounds on the variables).

Relaxing fuzzy constraints is equivalent to increasing the membership value, μ, at all levels of the scale when μ is less than 1·0. Two criteria for relaxation were studied: pure dilation and translation-dilation. The first one is based on Zadeh's model for the effect of hedges (Zadeh, 1972), and consists of taking the square root of all membership values, i.e. an original value $\mu = 0{\cdot}5$ is relaxed to $\mu = 0{\cdot}7$. A linear formulation was chosen to represent the translation-dilation model:

$$\mu_r = c + (1 - c)\mu_0$$

where μ_r is the relaxed membership value, and μ_0 is the original one.

A procedure "(relax)" that "systematically relaxes" all constraints in the database of alternatives was added to FUZWIN, using the translation-dilation model with $c = 0{\cdot}2$. Initial experiments with this procedure led to the following observations: (a) selected alternatives have obviously higher acceptability values when the relaxed database is used; (b) relaxation does not affect the ranking of the selected solutions; and (c) it is possible that relaxation in human decision making is selective instead of systematic and is only applied to certain dimensions of the problem.

Because of the previous observations, an improved procedure was successfully implemented to "selectively relax" the data inputed by the user according to her/his perception. The procedure called "(relax-case)" allows the user to relax any of the dimensions that characterize the case under consideration either by: (1) shifting the window to the lower side; (2) shifting the window to the higher side; or (3) widening the window.

7. *Case Based Representation*. Most experts usually do not reason by rules but on the basis of recalling previous cases. For example, a master chess player recalls a large number of settings that permit her/him to recognize potential moves and their effects. Therefore, being able to model information with a case based type of knowledge representation may significantly improve the performance of knowledge based systems.

The possibility of using stacks of windows to model cases became apparent when the authors realized that stored cases are defined by stacks of windows similar to those used in defining the alternatives: every time the user answers a question posed by the system, the name of the dimension and the fuzzy set representing the user's answer is saved in a memory called *CASE*. When the session is over, *CASE* contains a list of dimensions and their windows, i.e. a stack of windows, representative of the characteristics of the case. This stack can be saved in a permanent database of case histories together with the solution which the system selected for that particular case. Thus, after sufficient testing and use of the system, this database of case histories can be searched to select the stored case history that best matches the conditions of a new problem, and suggest the solution given for the case history as a potential alternative. The selection can again be made with the best first search approach used before.

Experimental work in trying to develop the repository of cases for the soil improvement system showed that:

(a) Not all dimensions characterizing a case need to be stored. Determining whether a dimension should be stored depends on how constraining it is, and how important it is to describe the case.
(b) There is a transformation in the memorization of some of the dimensions from "specific but fuzzy" (unimodal) to one of the form "at least as" or "at most as" (monotonic).
(c) Relevant, but not critical constraints can be stored with some degree of relaxation that facilitate the recollection of the case.

It is noted that the same three observations may not be unreasonable for human expertise as well.

8. *Learning*. The compilation of case histories, and their subsequent use in decision-making are a form of learning by experience. The three steps involved in the learning experience are: (1) use a knowledge system to make decisions based on fuzzy windows; (2) store the case histories used for validating the system, and their corresponding feedback from the experts; and (3) allow the system to make decisions based on "its previous experience".

The reverse learning process could also be implemented with FUZWIN. In this case let us assume that a large database of cases is available (i.e. a "training set"), and that one would like the system to automatically develop the database of constraints. The idea here is to start with a generic product or "adaptable mold" consisting of a stack of full windows, i.e. all 1·0's, that is modified by adding information (constraints) until its description satisfactorily matches the phenomenon

as represented by the database of
proceduralization, and can also be aut

Limitations of windows

Windows are one dimensional kno
they interact in a conjunctive mode,
space. This representation transfor
reasoning model of three levels, OR
for example, given a two dimensional
two windows each (A_x, A_y) and (B_x,
circle (Fig. 5)? If the windows are uni
the projections of the circle on the ax
be classified as belonging to the disc.
by the normalized intensity of the p
center of the circle would be fully a
other point in the disc such as "N"
circle would have $\mu = 0{\cdot}0$.

This representation can create prob
are correlated. For example, in th
windows, geotextiles could be used to
would control deformations with ve
geotextiles cannot be used for found
adequately control deformations is
However, there are several alternativ
could be handled with a production
stacks (e.g. one for structures and o
the one-dimensional representation c

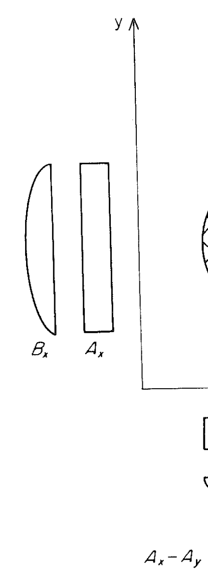


FIG. 5. Limitatio

ow to the higher side; or (3) widening

usually do not reason by rules but on
nple, a master chess player recalls a
recognize potential moves and their
rmation with a case based type of
prove the performance of knowledge

model cases became apparent when
ned by stacks of windows similar to
ne the user answers a question posed
the fuzzy set representing the user's
When the session is over, *CASE*
indows, i.e. a stack of windows,
ase. This stack can be saved in a
with the solution which the system
ficient testing and use of the system,
to select the stored case history that
and suggest the solution given for the
tion can again be made with the best

ne repository of cases for the soil

se need to be stored. Determining
epends on how constraining it is, and

rization of some of the dimensions
one of the form "at least as" or "at

an be stored with some degree of
of the case.

nay not be unreasonable for human

ories, and their subsequent use in
erience. The three steps involved in
e system to make decisions based on
for validating the system, and their
nd (3) allow the system to make

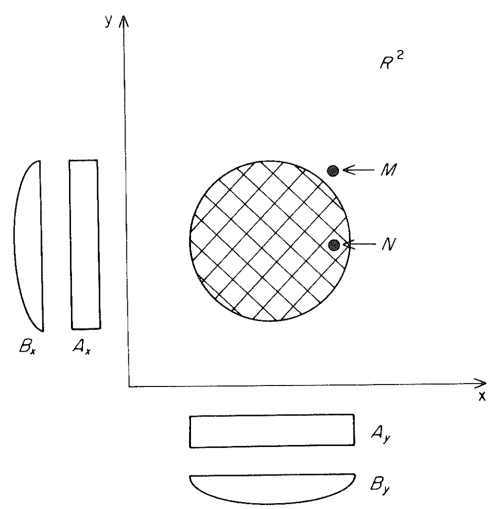
mplemented with FUZWIN. In this
s is available (i.e. a "training set"),
matically develop the database of
neric product or "adaptable mold"
1·0's, that is modified by adding
isfactorily matches the phenomenon

as represented by the database of cases. Such approach resembles learning by proceduralization, and can also be automated.

Limitations of windows

Windows are one dimensional knowledge structures. When combined in stacks, they interact in a conjunctive mode, without actually modeling an n-dimensional space. This representation transforms a classification problem into a shallow reasoning model of three levels, OR-AND-OR. This distinction is of importance; for example, given a two dimensional space defined in X and Y, and two stacks of two windows each (A_x, A_y) and (B_x, B_y), which is the correct representation for the circle (Fig. 5)? If the windows are uniform membership functions with $\mu = 1{\cdot}0$ along the projections of the circle on the axis, and 0·0 elsewhere, a point like "M" would be classified as belonging to the disc. On the other hand, if the windows are defined by the normalized intensity of the projection of the circle on each axis, only the center of the circle would be fully acceptable as belonging to the circle, and any other point in the disc such as "N" would have $\mu < 1{\cdot}0$; in fact, four points on the circle would have $\mu = 0{\cdot}0$.

This representation can create problems in situations where two or more variables are correlated. For example, in the soil improvement system that uses fuzzy windows, geotextiles could be used to reinforce both foundations and walls, but they would control deformations with very different levels of efficiency. To say that geotextiles cannot be used for foundations is incorrect, and to say that geotextiles adequately control deformations is correct for walls but not for foundations. However, there are several alternative solutions to such a problem. Particular cases could be handled with a production system architecture, or by creating separate stacks (e.g. one for structures and one for walls). Another alternative is to extend the one-dimensional representation of windows to n-dimensions.



$A_x - A_y$ or $B_x - B_y$?

FIG. 5. Limitation of the window representation.

Final comments and conclusions

Fuzzy windows were proposed to represent knowledge in classification systems. The formalism is simple but powerful, and allows for the modeling of knowledge and its uncertainty in a unique structure. This form of knowledge representation can be easily combined with a best-first search algorithm to provide an efficient knowledge engineering tool, such as FUZWIN. The limitation of this knowledge representation is that it provides a multi one-dimensional representation of a domain.

The fuzzy window approach and associated system, FUZWIN, can support the features of more classical system, e.g. explanation capabilities. In addition, it has applications in several important areas of artificial intelligence and knowledge systems. In particular the technique allows for (1) development of composite solutions; (2) search for lacunae; (3) case based representation of knowledge (an avenue for modeling learning); (4) use of different evaluation functions; and (5) model relaxation in decision making.

References

Chameau, J. L. & Santamarina, J. C. (1987). Membership functions I: comparing methods of measurement. *International Journal of Approximate Reasoning,* **1,** 287–301.

Chameau, J. L. & Santamarina, J. C. (1989). KBS for Soil Improvement. *ASCE Journal of Computing in Civil Engineering*. July.

Dubois, D. & Prade, H. (1980). *Fuzzy Sets Theory and Applications,* Orlando: Academic Press.

McDaniel Johnson, R. (1980). Phases of decision building. In S. & S. D. Ferguson, Eds., *INTERCOM, Readings in Organizational Communication,* New Jersey: Hayden Book.

Santamarina, J. C. & Chameau, J. L. (1987). Membership functions II: Trends in fuzziness and implications. *International Journal of Approximate Reasoning,* **1,** 303–317.

Wallsten, T. S. (1980). Process and models to describe choice and inference behavior. In T. S. Walleston, Ed., *Cognitive Processes in Choice and Decision Behavior,* pp. 125–237. Hillsdale.

Zadeh, L. A. (1972). A fuzzy-set-theoretic interpretation of linguistic hedges. *Journal of Cybernetics,* **2,** 4–34.

Zimmermann, H. J. (1985). *Fuzzy Set Theory and Its Applications.* Kluwer–Nijhoff.

Appendix: excavation methods—knowledge base

```
(setq *DB* '(   ; This is the database of stacks
  (drill-and-blast 1 none
    capital-investment (0·2 0·4 1 1)
    required-advance-rate (1 1 0·2 0 0 0)
    acceptable-vibrations (0 0 0 0·1 0·5 0·7 0·8 1 1))
  (full-face-TBM 1 none
    capital-investment (0 0 0·1 1)
    tunnel-length (0 0 0·2 0·9 1)
    variability-in-rock (1 1 0·9 0·75 0·6 0·3 0·1 0 0)
    need-to-access-face (1 1 0·8 0·6 0·3 0 0 0 0))))
```



```
(setq *DIMENSIONS* '(   ; This is th
  capital-investment |1000  10000  10
  tunnel-length |100  500  1000  500
  variability-in-rock linguistic
  required-advance-rate |0  300  600
  acceptable-vibrations linguistic
  need-to-access-face linguistic))

(setq *GENERAL* '(   ; General exp
  SELECTION OF ESCAVATION
  of a proper selection of an excav
  economical execution of the job))

(setq *WHY* '(   ; List of built-in exp
  capital-investment
    |the tunnel boring machines requi
    make the choice of this solution
  tunnel-length
    |the cost of the initial capital inv
    for tunnels of length about 5000
  variability-in-rock
    |the design of tunnel boring ma
    prove not possible–Blasting is no
  required-advance-rate
    |the cyclic process of blasting
    achieved by TBM|
  acceptable-vibrations
    |vibrations caused by blasting ma
  need-to-access-face
    |full face TBM do not permit
    important you are adviced to c
    blasting|))
```

```
(setq *DIMENSIONS* '(   ; This is the database of supports
  capital-investment |1000   10000   100000   1000000$|
  tunnel-length |100   500   1000   5000   10000 m|
  variability-in-rock linguistic
  required-advance-rate |0   300   600   1200   1500 ft/week|
  acceptable-vibrations linguistic
  need-to-access-face linguistic))

(setq *GENERAL* '(   ; General explanation for system
  SELECTION OF ESCAVATION METHOD FOR TUNNELS: the importance
  of a proper selection of an excavation method lies in a more successful and
  economical execution of the job))

(setq *WHY* '(   ; List of built-in explanations
  capital-investment
    |the tunnel boring machines require quite high initial capital investment and can
    make the choice of this solution not feasible giving preference to blasting|
  tunnel-length
    |the cost of the initial capital investment of a tunnel boring machine is justified
    for tunnels of length about 5000 meters or more|
  variability-in-rock
    |the design of tunnel boring machine for conditions with high variability may
    prove not possible–Blasting is not very sensitive to this parameter|
  required-advance-rate
    |the cyclic process of blasting may lead to lower advance rates than that
    achieved by TBM|
  acceptable-vibrations
    |vibrations caused by blasting may lead to avoiding the use of this method|
  need-to-access-face
    |full face TBM do not permit access to the face–If access to the face is
    important you are adviced to consider other types of excavation machines or
    blasting|))
```