
Applications of Hybrid Fuzzy Expert Systems in Computer Networks Design

Hany I. Fahmy and Christos Douligeris*
Dept. of Electrical and Computer Engineering
University of Miami
PO Box 248294
Coral Gables, FL, 33124-0640
email:{fahmy,christos}@ece.miami.edu
Tel (305) 284-3597, Fax (305) 284-4044

Abstract

The task of designing and configuring large Computer Networks most suited to a certain application and environment is difficult, as it requires highly specialized technical skills and knowledge, as well as a deep understanding of a dynamic commercial market. Current expert systems have made solid achievements in supporting decision makers, they use prior experience to solve problems in different domains. Hybrid fuzzy expert systems have appeared all over the world proving that integrated fuzzy expert systems/neural networks methods replaces classical hard decision methods and providing better performance. In this paper, we present an integrated fuzzy expert system, machine learning, and neural networks approach to large structured computer networks design and evaluation. After presenting an overview of the system and the major research choices, we describe in detail the system's modules and present examples of its potential use.

I. Introduction

The current most significant trend in the computing world is the growth of distributed processing, a technique that puts computing power closer to users rather than in large, central mainframes. Computer communication networks are the key for such distributed systems as they ease the share of information between cities, building complexes, buildings, departments and networked nodes.

Computer communication networks are generally classified into three broad categories each differentiated primarily by the distances they span:

- Local Area Networks (LANs) are short-distance networks (usually with a range of less than one mile) typically used within a building or building complex for high-speed data transfer between computers, terminals, and shared peripheral devices,
- Metropolitan-Area Networks (MANs) are medium-distance, high-speed networks with range of from 1 mile to 50 miles. MANs often transmits voice and video in addition to data, and
- Wide Area Networks (WANs) are primarily long-distance networks used for the efficient trans-

fer of voice, data, or video between local, metropolitan, campus, and premise networks. WANs typically use lower-transfer rates, and common-carrier services or private networking via satellite and microwave facilities.

In this age of internetworking the ability to effectively communicate is the key to the business success. Efficient and optimal network design is necessary to make communication networks usable and affordable. By network design, we mean the selection of various network devices and connections to accomplish an organization's operational objectives. A network's configuration can greatly affect its performance and cost. It is, therefore, vital that the best combination of equipment, connections, and placement of network connections for end-user nodes be made to satisfy an organization's objectives. These objectives may include a multitude of factors other than the prices of the computers and the networks, such as the reliability, the response, the availability, and the serviceability.

Professional designers with intensive knowledge and experience are needed for large computer communication network design, modeling and simulation. Such designers must be well informed about the most recent updates in this rapidly advancing field to be able to handle the available state-of-the-art technologies. Since designers of such caliber are difficult to find and usually very expensive, we proposed the use of hybrid fuzzy expert systems to play their role and/or assist them in their task.

In this paper, we focus on presenting the design, the knowledge representation, and the operation of a network design hybrid fuzzy expert system. The paper is structured as follows: Section II, compares the hybrid fuzzy expert system to similar activities in the field of computer communication network design. Section III, presents the general architecture of the hybrid fuzzy expert system and a design example. Section IV, presents the application of machine learning in acquiring fuzzy knowledge and a learning example. Finally, in section V, we draw conclusions.

II. Expert Systems in Network Design

ELAND, an Expert Design of Local Area Networks, has been the first activity in applying expert systems in network design. The ELAND's problem decomposition approach to the computer network design problem was suitable for solving the problem five to eight years ago.

In COMNED, we proposed a modern approach for using expert systems for network design, keeping in mind: 1) the system openness and modularity in-order to allow the system future updatability[10] and functionality, and 2) the usage of the available powerful network simulation tools.

COMNED has been fully implemented in the Telecommunication and Networking Laboratories, of the University of Miami, the expert system recommends the network feasible solutions most suited to the user's application and environment. A network simulation package receives the configuration of the network solutions from the expert system to be modeled, simulated and evaluated, after which the values of the performance indices are reported back to the expert system.

One of the most significant properties of the computer communication market is its rapid advancement and change in a very short periods of time. In [10] we presented a neural network/knowledge acquisition learning approaches to improve the time-efficiency of the COMNED's network design problem solver and allow COMNED to learn the new emerging network technologies, modern net-

work design techniques, and the updated specifications of the existing technologies.

Using classical expert systems and machine learning approaches to learn new emerging technologies, we have to learn the values of all the design variables for this technology and create all the facts/rules required to make the technology available as a design option the next time the system is used. Consider learning the value of a design variable like the “*noise_resistance*” for a newly emerging cabling system, then the knowledge engineer should enter the degree of truth for the low, medium, high, and very_high values of this design variable for this cabling system. The problem with this approach is that when using the experience of different design experts or knowledge engineers their estimates for the degrees of certainty could highly differ, then with this approach there will be multiple basis for the estimation process.

A second problem emerged with this approach. Consider, for example, that a 1.0 degree of truth was given to the ATM topology support for 500 nodes, and now the system was required to learn a new topology which has better capabilities in supporting the 500 nodes, then the degree of truth of the ATM topology support for this number of nodes should change to a lower value for the 1.0 to be given to the newly learned topology. On what basis this change should happen? If we depend on the knowledge engineer estimate then we are magnifying the problem of multiple estimation basis. If we consider a fixed change like 0.1 decrease (the degree of truth of the ATM topology support for 500 nodes will change to 0.9), we then find that many cases are far from realistic.

Also during the development of COMNED, a significant potential was found in integrating fuzzy sets, fuzzy logic, and neural networks into the knowledge representation and the reasoning process of the expert system. In particular we realized that:

1. The nature of knowledge representation and its suitability to undergo the well-founded fuzzy expert system theory. 60% to 70% of the used expert system facts/rules were found to be of fuzzy nature.
2. Hybrid fuzzy reasoning, was found to give better reasoning performance.

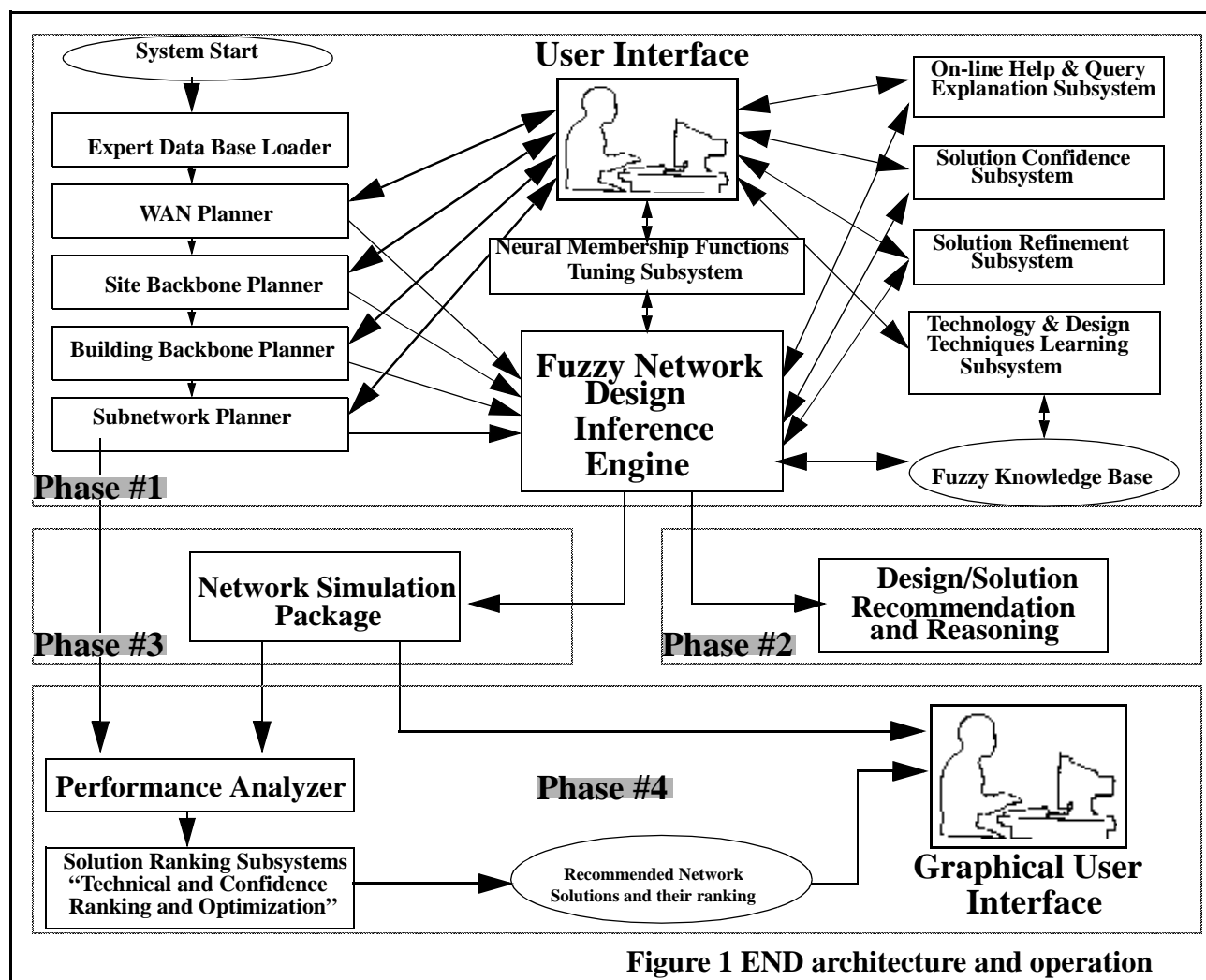
In addition, neural networks could be integrated with fuzzy expert systems to tune the shapes of fuzzy membership functions of the different design variables, this will improve the reasoning and confidence performance of the expert system and make the fuzzy expert system approach more justifiable.

For all the above reasons, we found ourselves motivated to present a hybrid fuzzy expert system approach for network design, keeping all the objectives of COMNED, and using fuzzy logic/neural networks to improve the system reasoning, manage the confidence calculation and estimation, and solve the above mentioned machine learning problem.

III. Network Design Hybrid Fuzzy Expert System: Architecture and Operation

The hybrid fuzzy expert system proposed in this paper is a part of a global network design system called “END”, An Expert Network Designer. The internal structure of the hybrid fuzzy expert system

is shown in **Phase #1, Phase #2, and Phase #4** areas of Figure 1. END is divided into four distinct



phases- the configuration entry phase, solution recommendation phase, model simulation phase, and the performance analysis phase.

1. Configuration Entry Phase:

In this phase, the hybrid fuzzy expert system interacts with the user through a user interface to obtain a general description of the networking project. The description is obtained through a group of planners which issue a number of hierarchical questions going from the highest possible network level, which is the number of network sites, the WAN interconnectivity between the different sites, passing by the number of buildings in each site, the number of floors in each building. etc., and ending with the number of workstations and servers in the departmental LANs. The system questions are designed to be as simple as possible for any person, not necessarily a network specialist, who is aware of the network general functionality and layout. The system uses a fuzzy knowledge base consisting of a database of information necessary for providing expert advice, a set of assertions relating these pieces of information, domain-specific information about the network configuration, traffic character-

istics, and measures for determining the suitability of various models in its model library, as well as information about modeling, analysis, and simulation in general. The answers of the configuration questions and the other design guidelines are used to find the network feasible solutions through a fuzzy network design inference engine. This inference engine uses fuzzy rule-based heuristics on an IF-THEN formalism. In the case of finding multiple solutions for a certain user configuration, the system interacts with the user, in a solution refinement session, with a new set of questions, depending on the different solutions obtained, to be able to filter the solutions to the most suitable solutions for the user environment. The user answers are used to revise the design guidelines initially assumed by the system. The Technology & Design Techniques Learning is a neural network/knowledge acquisition learning subsystem used to improve the time-efficiency of END's network design problem solver and allow the hybrid fuzzy expert system to learn the new emerging network technologies, modern network design techniques, and the updated specifications of the existing technologies. Finally, the Neural Membership Functions Tuning subsystem is a neural network to tune the membership functions of the fuzzy network design variables.

2. Solutions Recommendation Phase:

In this phase, the hybrid fuzzy expert system reports the best feasible topologies and cabling systems most suited to the user's application and environment (entered in the last phase) for each subnet, backbone and Wide Area Network, in addition to the confidence rank in each solution. At the end of the user interaction session, a graphical layout of each global feasible network solution is given to the user on a GUI. An optional full solution reasoning is available if the operator is interested in knowing why these solutions were chosen.

3. Model Simulation Phase:

If the operator chooses to run the optional network simulation ranking, END will generate a separate model for each feasible solution under a network simulation package [11][12] with the aid of a communications-oriented simulation language, run the simulations, and report the simulation results to the expert system part.

4. Solutions Analysis Phase:

In this phase a Performance Analyzer receives the simulation results from previous phase, in conjunction with the global network solutions from the Solution Recommendation Phase, to start classifying the different solutions with respect to their significance in each measured performance parameter.

The hybrid fuzzy expert system is designed to have two solution ranking subsystems:

1. A **Technical Ranker** which ranks the solutions according to the measured performance parameters.
2. A **Confidence Ranker** which simply ranks the solutions according to the user confidences in their satisfaction to the solution properties.

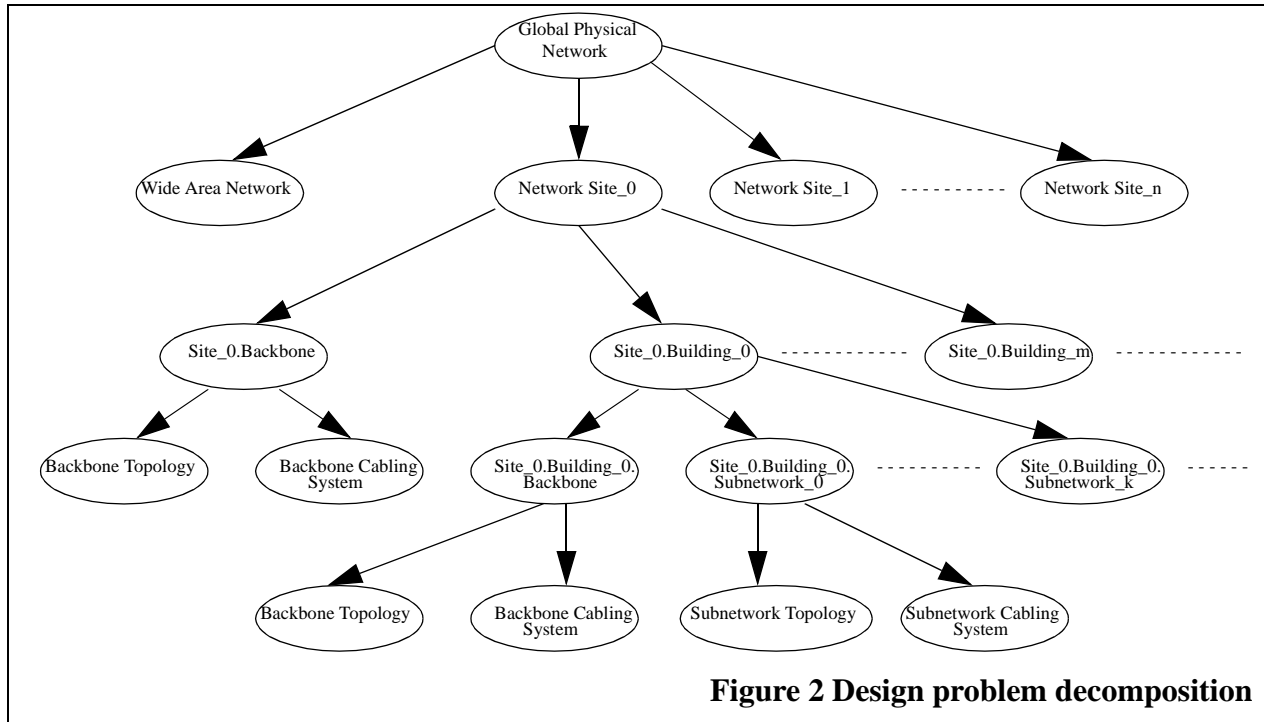
All solutions with confidence less than a pre-assigned value are initially eliminated as feasible solutions. All the other solutions are ranked first according to the Technical Ranker, and second according to the Confidence Ranker.

In this paper, we will concentrate on the hybrid fuzzy rule-based expert system. The details of the

other system components are described in [11] [12].

III.1 Network Design Problem

The network design problem solving structure is a tree traversal [13]. Figure 2 presents the decom-



position of the network design problem which introduces in each step a sequence of subproblems that must be solved. The global physical network problem “root node” is decomposed into subproblems for each network site and the Wide Area Network connecting these sites. In the design tree lower levels each network site problem is decomposed into subproblems for each building in the site and the site’s backbone. Then each building problem is further decomposed into subproblems for each subnetwork in the building and the building’s backbone. Finally, all the backbone and the subnetwork problems are decomposed into topology and cabling system subproblems.

The global network design problem is finally decomposed at the tree leaves into a number of topology and cabling system subproblems. For solving these two problems, COMNED uses a single general purpose subnetwork design inference engine which is invoked every time the system reaches the design tree leaves to obtain the suitable topology and cabling system for the currently active problem.

This problem decomposition represents the formalization of the practical approach used by design experts for computer networks design.

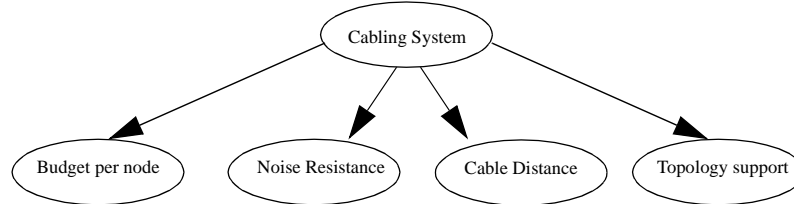
III.2 Operation and Knowledge Representation

The expert system uses fuzzy fact-like and constraint-like structures to represent the specification of

the different network topologies and cabling systems. In addition, it uses conceptual application-oriented design rules to represent the logic of network design.

There are three main fuzzy inferencing rules, the first for finding feasible cabling systems, the second for finding feasible topologies, and the third for finding WAN feasible solutions. A prolog-like format of the main inference rules and design decision trees are shown in Figure 3. The subnetwork

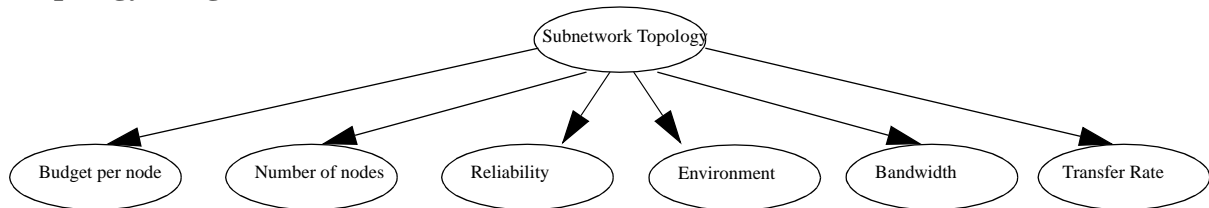
(a) Cabling system design tree and inference rule



```

cable(Cable_media,Budget,Distance,Noise):-
noise_resistance(Cable_noise,Cable_media,Noise<=Cable_noise_reistance,
cable_with_budget_per_station(Cable_media,Cable_budget),Budget>=Cable_budget,
cable_with_distance(Cable_media,Cable_distance), Distance<=Cable_distance)::95.
    
```

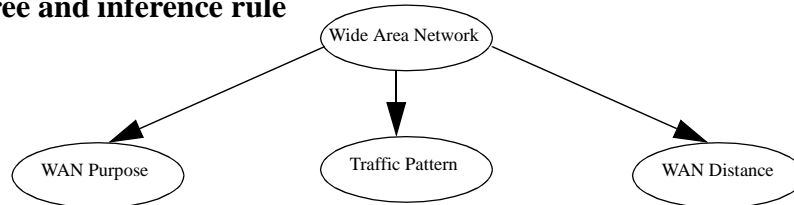
(b) Topology design tree and inference rule



```

topology(Budget,Stations,Reliability,lbm,Expand,Speed,Topology) :-
topology_budget_per_station(Topology,Min_budget),
Budget>=Min_budget,
number_of_stations(Max_stations,Topology),
Stations=<Max_stations,
Expand=<Max_stations,
lan_reliability(Reliability,Topology),
ibm_environment(lbm,Topology),
high_speed(Speed,Topology)::95.
    
```

(c) WAN design tree and inference rule



```

wan_type(WAN_purpose,Traffic_nature,Distance,WAN_Type):-
wan_purpose(WAN_purpose,WAN_Type),
traffic_nature_and_distance(Traffic_nature,Distance,WAN_Type)::95.
    
```

Figure 3 Inference engine design trees and rules

general specifications (obtained from the user and represented by the expert planners as design guidelines) are passed as arguments to these three inference rules which therefore will obtain, using the fuzzy predefined knowledge, the cabling systems “Cable_media”, the network topologies “Topolo-

gy”, and WAN types “WAN_Type” satisfying the passed subnetwork specifications.

It is clear from the cabling system design decision tree and the inferencing rule shown in Figure 3a, that the chosen cabling system depends on three design variables, the level of noise resistance “noise_resistance”, the budget required to connect one node using this cabling system “cable_with_budget_per_station”, and the distance supported by the cabling system “cable_with_distance”. The values of such variables for the chosen cabling system (Cable_noise_resistance, Cable_budget, and Cable_distance) should be, compared to the general specifications obtained from the user (Noise, Budget, and Distance), of equal or higher noise resistance “Noise=<Cable_noise_resistance”, of less budget requirements “Budget>=Cable_budget” and able to support wider network span “Distance=<Cable_distance”.

Such design variables are chosen in our system as fuzzy variables with different membership functions. For example, the “noise_resistance” is a fuzzy variable with four membership functions low, medium, high and very_high.

$$\text{low}(\text{noise_resistance}) = F1(\text{cabling_system})$$

$$\text{medium}(\text{noise_resistance}) = F2(\text{cabling_system})$$

$$\text{high}(\text{noise_reistance}) = F3(\text{cabling_system})$$

$$\text{very_high}(\text{noise_reistance}) = F4(\text{cabling_system})$$

The four membership functions F1, F2, F3, and F4 were chosen to be gaussian functions, as shown in Figure 4. The values of such functions represents the degree of truth of the satisfaction of these

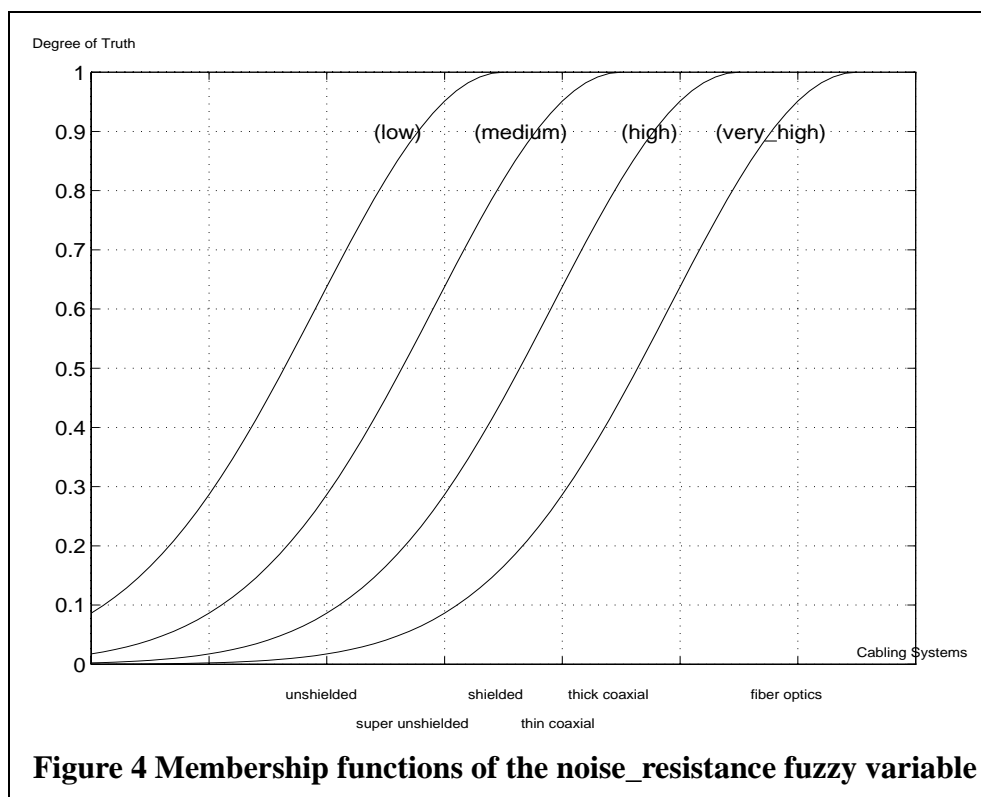


Figure 4 Membership functions of the noise_resistance fuzzy variable

membership functions of the “**noise_resistance**” fuzzy design variable. The membership functions are chosen such that the functions will have a contour similar to that estimated by a single knowledge engineer or a design expert.

Consider that the user entry for a subnetwork media noisiness (part of any subnetwork general specification obtained from the system user during the **Configuration Entry Phase**) was level high. From Figure 4, a cabling system like the `unshielded_twisted_pair` will have a degree of truth equal to 0.09 while another cabling system like the `thick_coaxial` will have a degree of truth equal to 0.82.

Similar membership functions are used for the other two fuzzy design variables “**cable_with_budget_per_station**” and “**cable_with_distance**”.

Obtaining the degree of truth for each of the three fuzzy design variables (**Cable_noise_resistance**, **Cable_budget**, and **Cable_distance**) say d_1, d_2 , and d_3 , for a specific cabling system, we can realize that the degree of truth of choosing this particular cabling system “**Cable_media**” (which is the same like the degree of linguistic certainty of the rule used in its choice) is:

$$\text{Conf} = \min(\min(d_1, d_2, d_3), 95)$$

where 95 is the linguistic certainty of the premise of the rule “**cable(Cable_media, Budget, Distance, Rule)**”.

The above method is used in the MILORD [13] computation method for calculating the degree of linguistic certainty of fuzzy rule.

All the above applies for all the design variables in the “**topology**” and “**wan_type**” design inference rules shown in Figure 3b & 3c. For example, the membership functions of the fuzzy design variable “**ibm_environment**”, which shows the extent of suitability of a specific networking topology for an IBM mainframe environment (used as a fuzzy design variable in the last row of the topology design inference rule), has two membership functions yes and no.

$$\text{yes}(\text{ibm_environment}) = G_1(\text{topology})$$

$$\text{no}(\text{ibm_environment}) = G_2(\text{topology})$$

The two membership functions G_1 and G_2 were chosen as gaussian and constant functions respectively. The two functions are shown in Figure 5. The values of such functions represents the degree of truth of the satisfaction of these membership functions of the “**ibm_environment**” fuzzy design variable.

III.3 Expert System Shell and Solution Reasoning

The hybrid fuzzy expert system uses an optimized fuzzy expert system shell based on MILORD[14] system shell which provides an inference engine to supervise the execution of the system rules. It uses standard backward chaining with uncertain reasoning capabilities based on fuzzy logic to satisfy the top-level goal. The engine allows the application of degrees of certainty by means of expert-defined linguistic statements. It also provides the features of accumulating proofs for query explanation and solution reasoning.

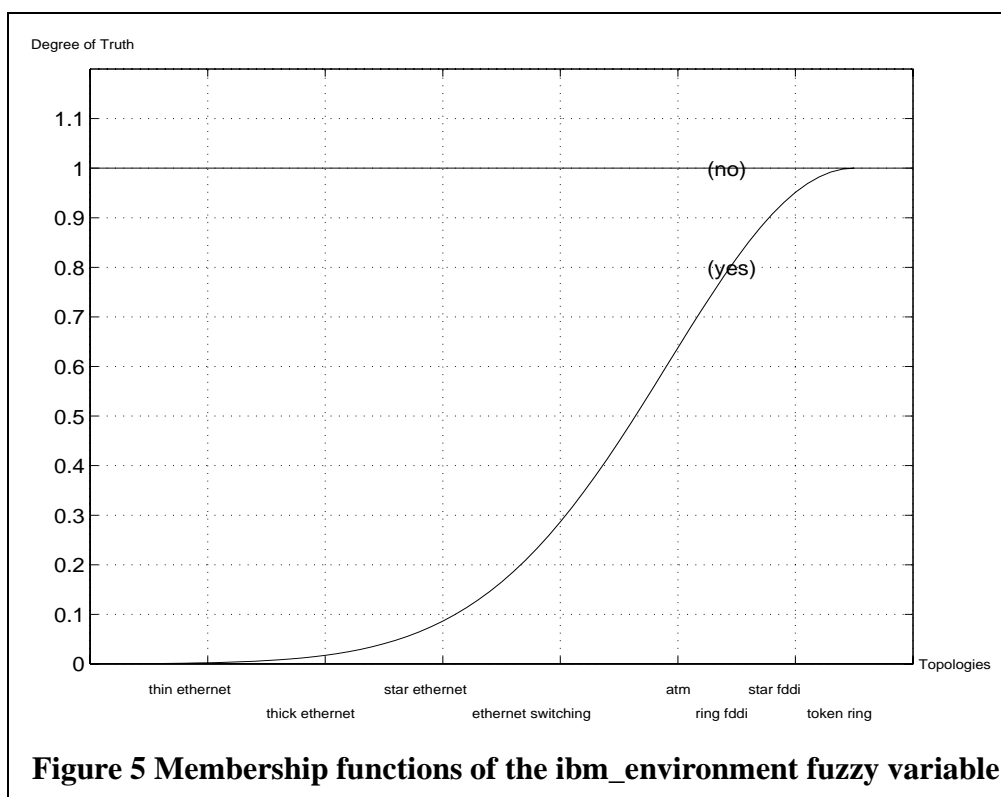


Figure 5 Membership functions of the ibm_environment fuzzy variable

The system stores proofs and explanations for the system predefined knowledge, the user collected knowledge, and the system queries as shown in Figure 6. The explanations for the system predefined knowledge and the user collected knowledge are used by the Design/Solution Reasoning subsystem to explain the system decisions. The system query explanations are used by the On-line Help & Query Explanation subsystem to explain the purpose and the role of these queries to the system operators whenever requested.

III.4 System Output

The hybrid fuzzy rule-based expert system reports the best feasible topologies and cabling systems for each subnet, backbone and Wide Area Network during its operation, in addition to the confidence rank in each solution. At the end of the user interaction session, a graphical layout of each global feasible network solution is given to the user on the GUI. Finally, if the operator chooses to run the optional network simulation ranking, the system will generate a separate model for each feasible solution under the network simulation package, run the simulations, and report the best solution to the operator. In addition an optional full solution reasoning is available if the operator is interested in knowing why these solutions were chosen.

III.5 Topology Design Example

Consider the following operator's dialogue with the expert system:

(a) Storage clauses for the explanations of the system predefined knowledge

```
system_fact(lan_reliability(A,_),
['The topology supports cabling systems of ',A,' fault tolerance.']).
system_fact(topology_support_cable_with_budget_per_station(A,B,C),
['The budget required to network a node in a ', A,' topology and ',B,' cabling system is ',C,' $ per node.']).
```

(b) Storage clauses for the explanations of the user collected knowledge

```
user_fact(user_num_buildings(A,_),['the number of buildings in the network is: ',A]).
user_fact(user_expandability(A,_),['the max. number of nodes in the network will not exceed: ',A]).
```

(b) Storage clauses for the explanations of the system queries

```
why('An estimate of the budget in $ for networking a workstation or Unknown(u)', %system query%
'Based on the answer of this question the system will estimate the category
of the network Topologies and cabling systems that are suitable for your available budget.
If you do not know an estimate value of the budget type u and your budget in this case will be
considered to be very high giving all the solutions satisfying the other specifications,
from which you can choose the solution suitable for your budget.').
```

```
why('Enter the number of workstations or Unknown(u)', %system query%
'Based on the answer of this question the system will estimate the category of the network
Topologies that can support this number of workstations. If you do not know a round value of
the number of workstations that will be connected to the network type u and in this case
a medium number of workstations will be assumed by the system giving all the solutions
satisfying the other specifications, from which you can choose the suitable solution.').
```

Figure 6 Explanation/proof storage clauses

```
+An estimate of the budget in $ for networking a workstation or Unknown(u): 300
+Enter the number of workstations or Unknown(u): 3
+Enter the number of Departmental/Floor servers or Unknown(u): 1
+Max number of workstations in any future expansion or Unknown(u): 50
+The network environment is an IBM mainframe environment [Yes(y) or No (n)] ? y
```

In the above query session, the system operator entered that the estimated budget to network a single node is \$300, the number of workstations to be connected to the network is 3, the number of network servers is 1, the maximum number of nodes in any future expansion is 50, and that the network environment is an IBM mainframe environment. Based on the above configuration the expert system will recommend the following feasible solutions:

```
The above Network description has 2 Solutions

The LAN Topology would be: ibm_token_ring_16mbps
The suggested LAN Cabling media: unshielded_twisted_pair_cable
The suggested LAN Access method: ibm_token_passing

Do you want more Design Solutions [Yes(y) or No(n)] [y]: y

The LAN Topology would be: ethernet_star
The suggested LAN Cabling media: unshielded_twisted_pair_cable
The suggested LAN Access method: csma/cd

no more solutions were found
```

As an explanation of how the expert system reached the above decisions for the subnetwork feasible solutions: First, the system collects/approximates the following user specifications through the questions issued by the user interface (the arguments to be passed to the topology choice rule in Figure 3b):

*Budget=300,
Stations=4,
Expand=50,
Ibm=yes,
Reliability=medium.*

The expert system has the following system technical facts for the *ethernet_bus*, *ethernet_star*, and *token_ring* topologies built-in its knowledge-base:

*topology_budget_per_station(ethernet_bus,200)::100
topology_budget_per_station(ethernet_star,250)::100
topology_budget_per_station(token_ring,300)::100*

*number_of_stations(30,ethernet_bus)::100
number_of_stations(100,ethernet_star)::100
number_of_stations(150,token_ring)::100*

*lan_reliability(low,ethernet_bus)::80
lan_reliability(low,ethernet_star)::100
lan_reliability(medium,ethernet_star)::90
lan_reliability(high,ethernet_star)::80
lan_reliability(low,token_ring)::90
lan_reliability(medium,token_ring)::80*

*ibm_environment(no,ethernet_bus)::100
ibm_environment(yes,ethernet_bus)::2
ibm_environment(no,ethernet_star)::100
ibm_environment(yes,ethernet_star)::9
ibm_environment(no,token_ring)::100
ibm_environment(yes,token_ring)::100*

Given the network topology choice rule, the network design inference engine will choose the *ethernet_star* and *token_ring* topologies (*ethernet_bus* topology was not chosen because it does not support both 50 nodes and the required cable reliability).

For the *ethernet_star* topology, the following variables will be bound as follows

*Min_budget=250,
Max_stations=100.*

The fuzzy certainty on each part of the rule is as follows:

*topology_budget_per_station(ethernet_star,250),300≥250 → {100}
number_of_stations(100,ethernet_star), 4≤100, 50≤100 → {100}
lan_reliability(medium,ethernet_star) → {90}
ibm_environment(yes,ethernet_star) → {9}
The Overall certainty → {min(100,100,90,9)=9%}*

For the *token_ring* topology, the following variables will be bound as follows

Min_budget=300,

Max_stations=150.

The fuzzy certainty on each part of the rule is as follows:

topology_budget_per_station(token_ring,300),300 \leq 300 \rightarrow {100}
number_of_stations(150,token_ring), 4 \leq 150, 50 \leq 150 \rightarrow {100}
lan_reliability(medium,token_ring) \rightarrow {80}
ibm_environment(yes,token_ring) \rightarrow {100}
The Overall certainty \rightarrow {min(100,100,80,100)=80% }

From the above obtained results it is clear that the system's confidence on the *token_ring* topology as a feasible networking topology {80% } is higher than that on the *ethernet_star* topology {9% }. This is due to the fact that the *token_ring* topology is more suitable for IBM mainframe environment than the *ethernet_star* topology.

The advantage of using fuzzy expert systems appears in this example. In a classical expert system the “**ibm_environment**” design variable will have only a value of “no” for the *ethernet_star* topology {*ibm_environment(no,ethernet_star)*}, so the *ethernet_star* topology will not show up as a feasible solution for the above configuration, in spite the fact that, practically, the *ethernet_star* topology could be used for the above configuration. As shown above, in fuzzy expert systems, the *ethernet_star* topology will show up as a feasible topology but with lower system confidence. This justifies the opinion that fuzzy variables (in fuzzy expert systems) are best suited to represent estimated values, in contrast to non fuzzy variables (in classical expert systems) which are best suited to represent rigid facts.

The problem with the fuzzy expert system approach is in the larger number of feasible solution the system will obtain. This increases the volume of the filtering process and the difficulty of the filtering questions. Our methodology to deal with such problem is to use a pre-specified cut off degree of truth below which solution with lower confidence will not be reported as feasible solutions.

In the above given example, it is clear that MIN inferencing is used for the computation of the degree of linguistic certainty in the network feasible topologies for a specific network configuration. Other inferencing methodologies like PROD and AVG inferencing were considered. It was found that MIN inferencing has better solution filtering capabilities than AVG inferencing but PROD inferencing was found to have similar solution filtering capabilities and more reasonable results. This is mainly due to the problem in MIN inferencing that low confidences eliminate the effect of other larger confidences, but in product inferencing the effect of all the confidences will appear.

Using PROD inferencing the results of the above given example will change as follows:

For the *ethernet_star* topology, the overall fuzzy certainty is:

The Overall certainty \rightarrow {PROD(100,100,90,9)=8.1% }

For the *token_ring* topology, the overall fuzzy certainty is:

The Overall certainty \rightarrow {PROD(100,100,80,100)=80% }

III.6 Sensitivity to Changes in the Membership Functions

As mentioned before the shapes and the values of the membership functions are chosen such that the functions will have a contour similar to that estimated by a single knowledge engineer or a design

expert. An important research question came up, how sensitive is the fuzzy expert system output to the changes in the membership functions of fuzzy network design variables?

Using different shapes, centers and values for the membership function, it was found that there is a general increase in the number of feasible solution with the increase in the values of the membership functions of the different fuzzy design variables and vice-versa.

The change in the number of solutions is not directly proportional to the change in the membership functions but it depends on many other factors, like the user's network configuration, the nature of the feasible solutions and the operating point in the formalized network design space.

Minimum inferencing was found to be less sensitive to the changes of the membership functions. Product and Average inferencing were found to be more sensitive to such changes.

IV. Machine-Learning and Network Design

As mentioned before, one of the most significant properties of the computer communication market is its rapid advancement and change in a very short periods of time.

In [10], we presented a neural network/knowledge acquisition learning approaches to improve the time-efficiency of the COMNED's network design problem solver and allow COMNED to learn the new emerging network technologies, modern network design techniques, and the updated specifications of the existing technologies.

With classical expert system approaches we faced the problem of estimating the confidence in the newly learned facts and rules. Since in fuzzy expert system predefined functions are used to manage such confidence factors (membership functions), we found that a fuzzy expert system approach could help in solving this problem.

Consider that the fuzzy expert system is learning a new topology named "xyz". First, the system will ask about the topology's name:

+ *Enter the name of the new topology? xyz*

Then, the following learning sessions will take place for the topology's fuzzy design variables.

IV.1 Learning the values of the membership functions of the "ibm_environment" fuzzy design variable

Suppose that the xyz topology was considered to be more suitable to be used in an ibm environment than the fddi_star topology.

The following knowledge acquisition session will take place between the system and the knowledge engineer when learning the value of the "**ibm_environment**" design fuzzy variable:

+ *The xyz topology is more suitable to be used in an IBM environment than the token_ring topology? n*

+ *The xyz topology is more suitable to be used in an IBM environment than the fddi_star topology? y*

The system compares the new topology with the pre-existing topologies starting from the topolo-

gies with the highest degree of truth, which is the “token_ring” in this case. Once the design expert agrees that the new topology is more suitable for an IBM mainframe environment than one of the pre-existing topologies, which is the “fddi_star” in this case, the system stops issuing any further comparison questions.

The membership functions diagram shown in Figure 5 for the pre-existing topologies will be updated to the diagram shown in Figure 7.

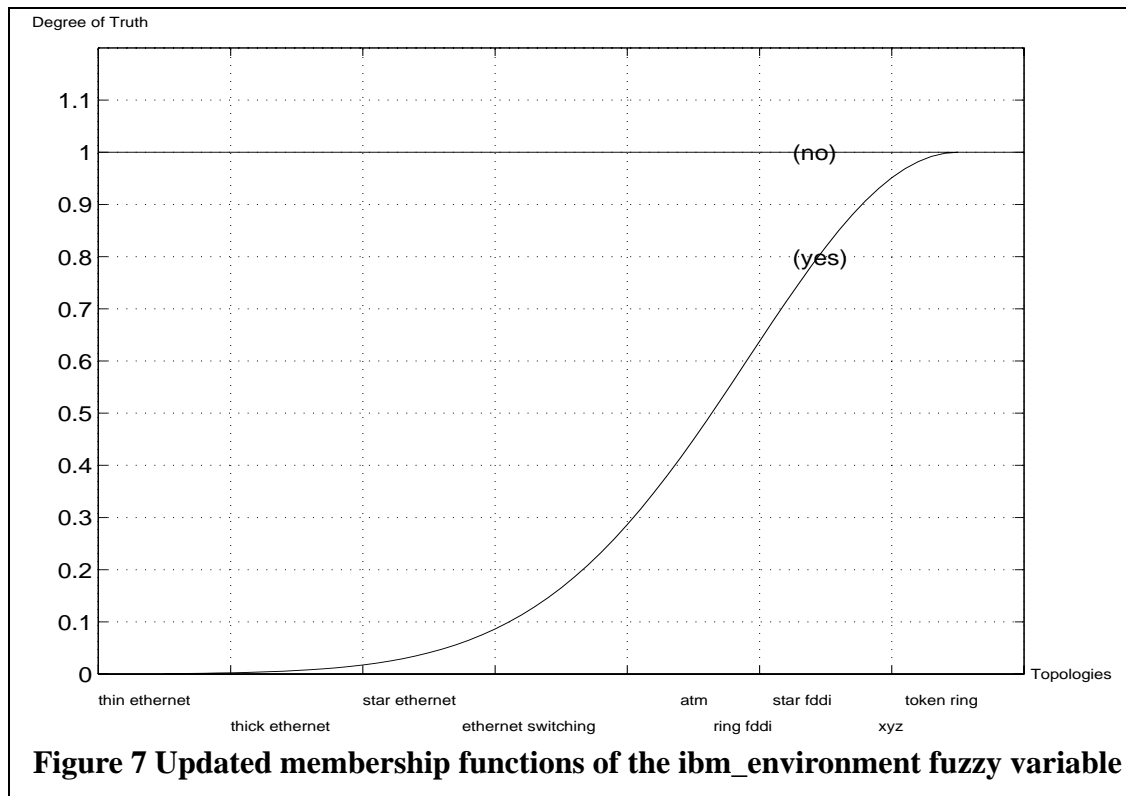


Figure 7 Updated membership functions of the `ibm_environment` fuzzy variable

The membership function “no” of the fuzzy variable “`ibm_environment`” is always equal to 100% due to the fact that all the entered topologies are suitable for networking nodes in non-ibm environment

It is clear that the learned xyz topology will occupy the same position of the star_fddi topology on the horizontal axis. All the topologies with ibm environment suitability less than the xyz topology will be shifted to the left on the horizontal axis. The axis shifting has no fixed rule. It just depends on the nature of the fuzzy design variable. We need to keep in mind that, the corresponding degrees of truth for all the topologies should make sense (relevant to each other).

The following prolog facts will be added to the knowledge base as well:

ibm_environment(no,xyz)::100.

ibm_environment(yes,xyz)::95.

IV.2 Learning the values of the membership functions of the “high_speed” fuzzy design variable

Figure 8 shows the membership functions “low” and “high” of the fuzzy design variable “**high_speed**”

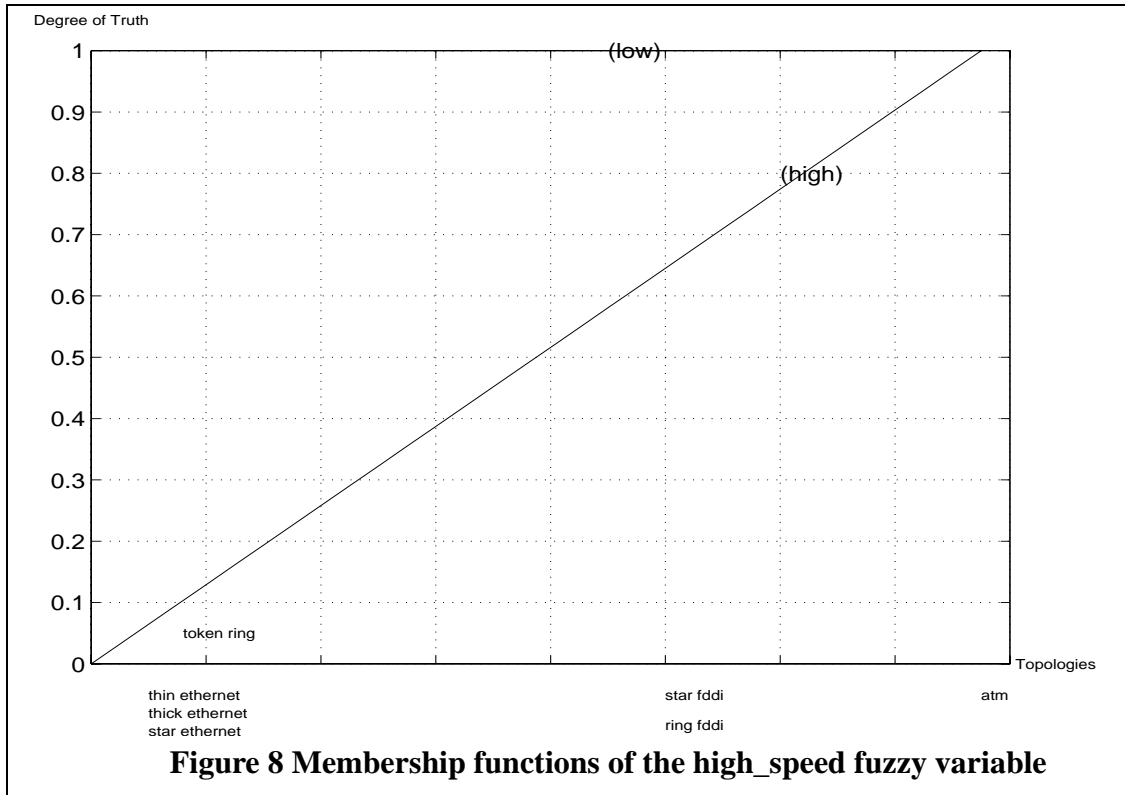


Figure 8 Membership functions of the high_speed fuzzy variable

It is clear that the 100% confidence was given to the atm topology (155 Mbps) {highest speed} and the degree of truth of the other topologies are computed according to the following equation:

$$\text{high}(\text{topology}) = \text{transfer_rate}(\text{topology}) * 100 / \text{max_transfer_rate}$$

The above computation equation is for the “high” membership function, the value of the “low” membership function is always 100% for any topology due to the fact that any topology can fit with 100% confidence in a low speed network requirements.

The following knowledge acquisition session will take place between the system and the knowledge engineer when learning the value of the “**high_speed**” design fuzzy variable:

+Enter the xyz topology transfer rate? 200

Suppose that the xyz topology has a transfer rate of 200Mbps. The membership functions diagram shown in Figure 8 for the pre-existing topologies will be updated to the diagram shown in Figure 9.

It is clear that the learned xyz topology will occupy the same position of the atm topology on the horizontal axis. All the topologies with transfer rates less than that of the xyz topology will be shifted to the left on the horizontal axis.

The following prolog facts will be added to the knowledge base as well:

high_speed(high,xyz)::100.
high_speed(low,xyz)::100.

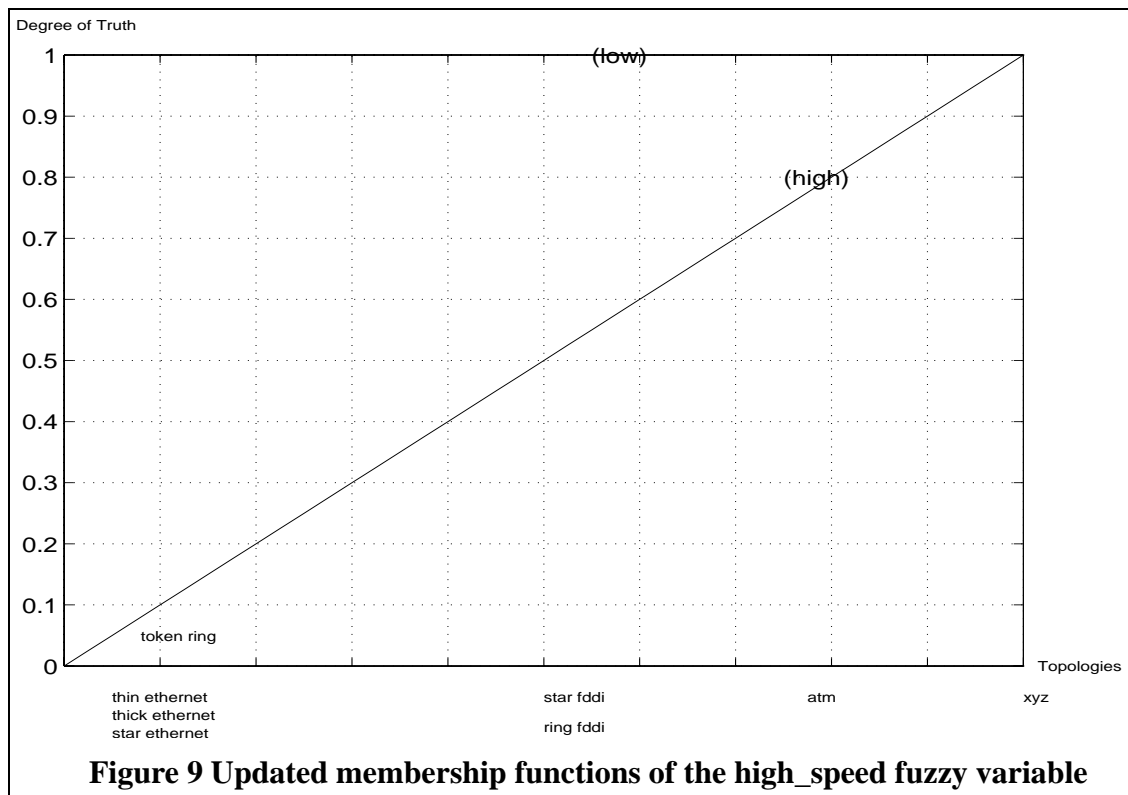


Figure 9 Updated membership functions of the high_speed fuzzy variable

IV.3 Learning the values of the “number_of_stations” design variable

The following knowledge acquisition session will take place between the system and the knowledge engineer when learning the value of the “number_of_stations” design variable:

+Enter the max number of nodes which could be supported by the xyz topology per physical network segment? 100

100% confidence is given to the topology’s support for 1/4 of its max number of nodes, 95% is given to the topology’s support for 1/2 of its max number of nodes, 90% is given to the topology’s support for 3/4 of its max number of nodes, and 85% is given to the topology’s support for its max number of nodes.

In the case of having a network configuration with number of nodes in between the 1/4, 1/2, 3/4, and the max number of nodes supported by the topology, interpolation is used to calculate the degree of truth in supporting this number of nodes (using the surrounding values).

The variable “number_of_stations” was not fuzzified due to the wide range of the supported max number of nodes by the available topologies, for example, the max number of nodes which could be supported by the “ethernet_thin” topology per physical network segment is 30, while the same value for the “atm” network topology is 500, that’s why we found that we are in need for many fuzzy membership functions to accurately represent this wide range. Based on this we decided to use the above method to calculate the degree of truth on this variable.

The above method guarantees a higher degree of truth on the support of any topology for any number of nodes than that of other topologies with lesser number of max supported nodes.

The following prolog facts will be added to the knowledge base after the above session:

```

number_of_stations(25,xyz)::100.
number_of_stations(50,xyz)::95.
number_of_stations(75,xyz)::90.
number_of_stations(100,xyz)::85.
    
```

IV.4 Learning the values of the membership functions of the “lan_reliability” fuzzy design variable

Figure10 shows the membership functions “low”, “medium”, “high” and “very_high” of the fuzzy design variable “lan_reliability”

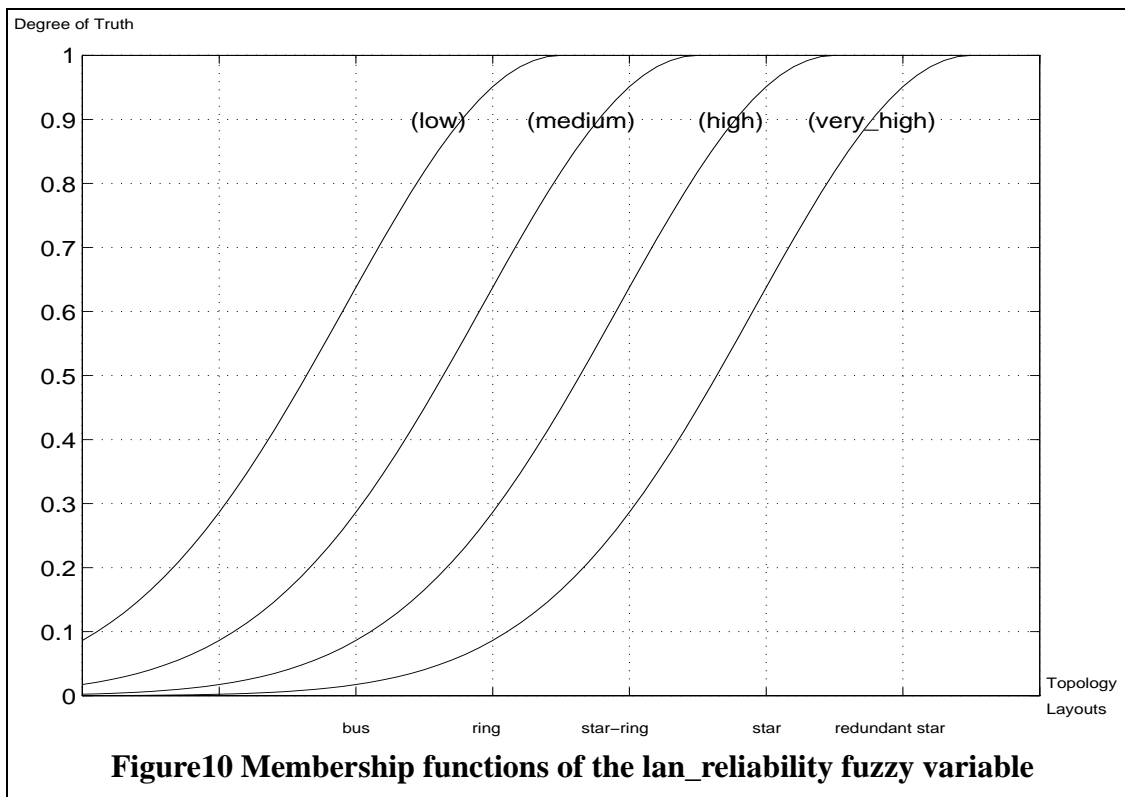


Figure10 Membership functions of the lan_reliability fuzzy variable

The following knowledge acquisition session will take place between the system and the knowledge engineer when learning the value of the “lan_reliability” design fuzzy variable:

```
+Enter the shape of the xyz topology layout [redundant-star(e), star(s), star-ring(t), ring(r), or bus(b)]? s
```

Entering the shape of the topology’s layout, the corresponding degrees of truth are obtained from Figure 10 and the following prolog facts will be added to the knowledge base:

```

lan_reliability(low,xyz)::100.
lan_reliability(medium,xyz)::100.
lan_reliability(high,xyz)::95.
    
```

lan_reliability(very_high,xyz)::64.

There is no axis shifting in this case due to the fact that the above layouts are general and unchangeable layouts {new layouts is not a possibility in this domain}. In addition, there is no way for any newly learned networking topology to have a shape different from the above mentioned ones.

V. Membership Functions Optimization in the Network Design Fuzzy Expert System

In the previous sections a fuzzy expert system was successfully applied in computer communication network design. The only problem with this approach was the choice of the membership functions of the different network fuzzy design variables and their shapes.

A neural network connectionist model [15] could be used to solve the problem of choosing the optimal shape of the fuzzy membership functions in the network design fuzzy expert system. The neural network connectionist model have just one fuzzy output which is the network feasible solutions for the input network configurations. Figure 11 shows the proposed neural-network connectionist model.

Layer 1: The nodes in this layer transmit network configuration input values (NC-1 to NC-n) to the

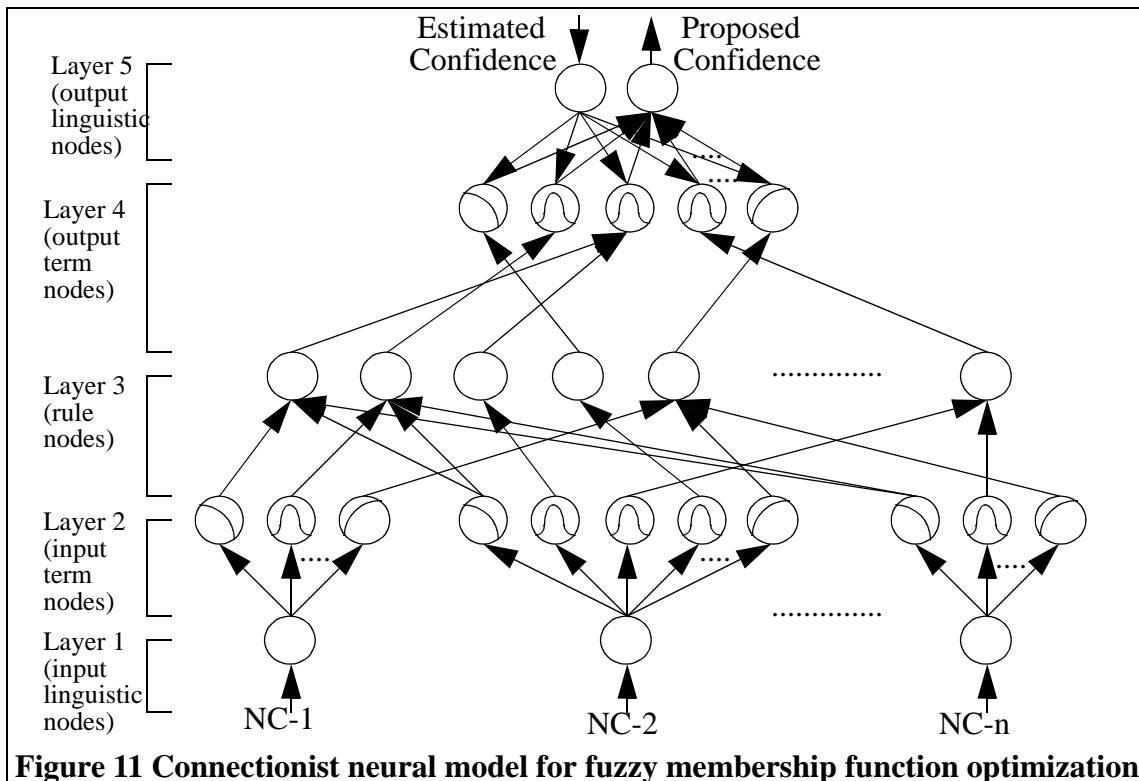


Figure 11 Connectionist neural model for fuzzy membership function optimization

next layer directly.

Layer 2: The output function of this node is the membership function of the different fuzzy design variables.

Layer 3: The links in this layer are used to perform the precondition matching of the fuzzy logic rules of the network design fuzzy expert system.

Layer 4: The nodes in this layer have two operation modes: down-up transmission and up-down transmission modes. In the down-up transmission mode, the links at layer four should perform the OR operation to integrate the fired rules which have the same consequence.

Layer 5: There are two kinds of nodes in this layer. The first kind of node performs the up-down transmission for the estimated confidence in the obtained feasible solutions (training data from a design expert based on the input network configuration and the output feasible solutions) to be fed into the network. The second kind of node performs the down-up transmission for the Proposed confidence in the same obtained feasible solutions.

Again a two-stage hybrid learning algorithm is used[15]. This learning algorithm will determine optimal centers and widths of the membership functions of the fuzzy design variables used in layers two and four.

In phase one of the hybrid learning algorithm, a self-organized learning scheme is used to locate initial membership functions. In phase two, a supervised learning scheme is used to optimally adjust the membership functions for desired outputs. To initiate the learning scheme, training data (the estimated confidence in each network feasible solution) and the desired or guessed coarse of fuzzy partition (i.e., the general membership function shapes of the different fuzzy network design variables) must be provided from the outside world. Before this network is trained, an initial form of the network is first constructed. Then, during the learning process, some nodes and links of this initial network are deleted or combined to form the final structure of the network.

With the connectionist neural network model for fuzzy membership function optimization, the category of the network design fuzzy expert system will move from Loosely-Coupled hybrid fuzzy expert system model to Fully-Integrated model [16].

VI. Conclusions

Fuzzy expert systems have proved to be very successful in formalizing the practical rules used by the design experts for computer networks design, formalizing the logic of solving computer network design problems, and initially choosing the most suitable solutions for a certain networking requirement.

By using fuzzy expert systems to generate the network models/simulations the user is not required to have any kind of background of the simulation package operation. Neither is the user required to be an expert in networks design. The user is only required to answer a group of general questions about the network requirement: he/she will get a network design, a description of the design, why it was chosen, a graphical diagram of the design, a design simulation, vector simulation results (curves), and even results analysis by the expert system. Automating the process of network modeling and simulation generation is very important as it can save the user time and expense.

Fuzzy Logic addresses several problems with current expert systems like, providing better knowledge representation, better reasoning performance, and better management of confidence factors.

Fuzzy logic was found to aid the easiness of the learning process, in this paper, it was clear how fuzzy logic solved the problem of multiple estimation basis by the usage of pre-defined membership functions and how the problem of saturating the membership functions was eliminated by the usage of the membership functions axis-shifting.

Proposing the full integration of neural networks to tune the shapes of fuzzy membership functions will improve the performance of the fuzzy expert system and make the fuzzy expert system approach more justifiable.

References

- [1] A.A. Covo, T.M. Mouzzi, and E.D. Peterson, "A.I. Assisted Telecommunication Network Management", IEEE GLOBECOM, March 1989, Dallas, TX, pp. 487-491.
- [2] Claude Muller, Evan H. Magill, Patrick Prosser and D. Geoffrey Smith, "A.I. in Telecommunication", IEEE GLOBECOM, December 1993, Houston, TX, pp. 883-887.
- [3] Shri Goyal and Ralph Worrest, "A.I. Directions in Telecommunications", IEEE GLOBECOM, November 1987, Tokyo, Japan, pp. 46.8.1-46.8.5.
- [4] Stefano Ceri and Letizia Tanca, "Expert System of LAN Design", IEEE Expert, Vol. 10, October 1990, pp. 23-33.
- [5] John J. Hannan, "Network Solutions Employing Expert Systems", IEEE GLOBECOM, November 1987, Tokyo, Japan, pp. 543-547.
- [6] Hany I. Fahmy and Christos Douligeris, "COMNED: COMputer Network Expert Designer", submitted to the IEEE Expert Magazine.
- [7] Miyamoto, S. and Yasunobu, S., "Predictive Fuzzy Control and its Applications to Automatic Train Operation Systems", Fuzzy Inf. Proc. Soc. Conf., 1984
- [8] Ogama, H., Fu, K., and Yoa, J.P.T., "Knowledge Representation, and Inference Control of SPERILLII", ACM Conf. Proc., November 1984.
- [9] Buckley, J.J., Siler, W., and Tucker, D., "FLOPS, A Fuzzy Expert System: Applications and Perspectives", Fuzzy Logics in Knowledge Engineering, Nongpita, C.V. and Prade, H., Eds., Verlag TUV Rheinland, Germany, 1986.
- [10] Hany I. Fahmy, George Develekos, and Christos Douligeris, "Applications of Neural Network and Machine Learning in Computer Networks Design", IEEE Journal of Selected Areas in Communications, January 1997, to appear.
- [11] Hany I. Fahmy and Christos Douligeris, "END: An Expert System Designer", IEEE Network Magazine, Vol. 9, No. 6, Nov/Dec 1995, pp. 18-27.
- [12] Hany I. Fahmy and Christos Douligeris, "NAMS: Network Automated Modeler and Simulator",

- IEEE/ACM 29th Annual Simulation Symposium, New Orleans, Louisiana, April 1996, pp. 65-70.
- [13] K.B. Irani and N.G. Khabbaz, "A Methodology for the Design of Communication Networks and the Distribution of Data in Distributed Supercomputer Systems," IEEE Trans. Computers, Vol. C-31, No. 5, 1982.
- [14] R. Lopez de Mantaras, J. Agusti, E. Plaza, and C. Sierra, "MILORD: A Fuzzy Expert System Shell", Fuzzy Expert Systems, Abraham Kandel, CRC Press, Boca Raton, FL.
- [15] Chin-Teng Lin and George Lee, "Neural-Network-Based Fuzzy Logic Control and Decision System," IEEE Trans. on Computers, Vol. 40, No. 12, December 1991, pp. 1320-1336.
- [16] Larry R. Medsker and David L. Bailey, "Models and Guidelines for integrating Expert Systems and Neural Networks," Hybrid Architecture for Intelligent Systems, Abraham Kandel, CRC Press, Boca Raton, FL.