Chapter 3

Topology Design of Distributed Local Area Networks

This chapter provides a detailed definition of the problem considered in this thesis. This information is required in order to understand the concepts, terminology, and related work described in subsequent chapters. Firstly, an informal discussion of network topology design and related issues are provided. This is followed by assumptions made for the purposes of this thesis and a formal statement of the distributed local area networks topology design problem. Design objectives and constraints are also discussed. Finally, the fuzzy logic approach for aggregating the individual design objectives is discussed.

3.1 Background

A computer communication network provides communication services to a large number of hosts. Hosts may include mainframe computers, mini systems, workstations, personal computers, printers, and other peripherals. To interconnect these hosts, network active elements such as routers, switches, and hubs are used. The large number of different hosts and network active elements result in a large number of ways in which hosts and network active elements can be interconnected. Such networks are referred to as internetworks. At an abstract level, a typical computer communication network can be divided into two stages. The first stage is known as the local access network, which allows users access to hosts or local servers. Local access networks are generally designed as *centralized* systems. The second stage consists of a backbone, which is responsible for the delivery of information from source to destination using *switching* elements. The backbone network can be designed as a *distributed* network which relies on switching technology.

In a modern organization, communication services are centered around a distributed local area network, or DLAN. In a DLAN, the backbone interconnects a number of local access networks via routers or layer 3 switches (refer to Figure 3.1). A local access network may be subdivided into smaller sub-groups, called *segments*. Thus, a hierarchically structured network topology is achieved. Such a hierarchical topology comprises a switched backbone that interconnects several local access networks via routers or layer 3 switches, where each local access network is an interconnected collection of segments. Distributed local area networks have several advantages over centralized networks, including

1. Broadcast traffic is confined to a single local access network. This prevents broadcast storms from sweeping across the entire network.

2. Highest network availability and lowest latency are ensured.

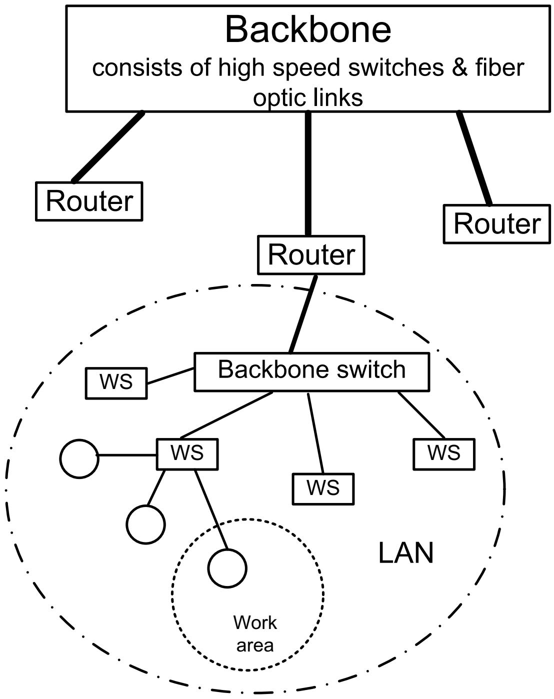

Figure 3.1: A typical distributed local area network (WS represents a workgroup switch)

3. Users are provided with the most appropriate connectivity.

4. Cost of administration is reduced. Equipment movement and changes are carried out more systematically. Moreover, diagnosis and troubleshooting of network problems are easier.

The objective of topological design of computer networks is to achieve a certain performance level through improvement in various parameters such as network delay, cost, reliability, and number of hops between communicating nodes. An over-dimensioned network is easy to design, while it is always difficult to design a cost-optimized network. The prime factors contributing to this difficulty are the size, the constraints, and obviously the cost parameters. Topology optimization usually involves a tradeoff between performance and cost. For example, management would like to consider financial aspects while designing a network, and would like the monetary costs to be minimized. On the other hand, a user would be more interested in a service where communication delay is minimized while network reliability is maximized. Thus, the notion of optimality becomes vague in the presence of various cost parameters and constraints. A rational approach to high-quality network design is to search for a solution that possesses a set of desirable attributes and does not violate well-established design principles. For example,

- there should be a physical path between any two nodes,
- the number of hops between any two nodes should remain within a defined upper limit, and
- utilization levels of links should always be below a given threshold.

The network topology design problem has received considerable attention by network designers and analysts. Extensive research has been done to develop efficient optimization techniques for this complex optimization problem [57, 58, 75, 98, 110, 111, 152, 204]. Topology design has been categorized as an NP-hard problem [75, 79, 98]. Similarly, topological design of DLAN can be considered as an equally complex problem with a huge search space. For a network with n nodes, there exist as many as $2^{(\frac{n(n-1)}{2}-1)}$ distinct topologies. Even for $n = 10$, there exist more than 10^{13} potential solutions. It is therefore clear that an exhaustive search is not desirable due to its huge computational cost. Rather, heuristic approximation methods are used to search for an optimal topology design. Heuristic methods produce good feasible solutions in a reasonable amount of time and focus the search on feasible topologies of desirable characteristics.

3.2 Assumptions and Problem Statement

This section describes the necessary assumptions adopted in this thesis, and also provides a formal statement of the DLAN topology design problem and design constraints.

3.2.1 Assumptions

For the purposes of DLAN topology design, the following are assumed:

- A tree topology is considered for network design.
- A "node" refers to a LAN (i.e. a local site). The node represents a layer 2, or above, networking device (i.e. switch, router, or gateway) connecting the LAN to another backbone node.
- The "root" node is a switch acting as a collapsed backbone with given required interfaces.
- Each link is bidirectional.
- The reliability of each link is known.
- Nodes are fault-free. Only links are susceptible to failure.
- Only fiber optic cable is used between two LANs.

- The maximum utilization of any link should remain within a desired threshold (e.g., 60%).
- Fast Ethernet is implemented on the backbone.
- The "capacity" of a network device is equal to the number of ports available on it. These ports are used to connect users to a network device.
- Distances between local sites (nodes) are known. The location of a node can be represented by its Cartesian coordinates with respect to some reference point.
- The internal topology of each LAN (i.e. a local site) is known a priori. That is, each LAN is assumed to already exist.

3.2.2 Problem Statement

The DLAN topology design problem can be stated as follows [268, 269, 270, 271, 272]:

"With the assumption that the internal topology of each LAN is already designed and known, find a quality feasible tree topology under a given set of design objectives and constraints. This tree topology will interconnect all nodes (LANs) in the network, thus forming a backbone topology of a DLAN."

The term "feasible topology" in the above statement refers to a solution that satisfies all design principles and constraints. The term "quality topology" refers to a solution that optimizes the design objectives. In this thesis, the quality of a topology is evaluated on the basis of four design objectives: monetary cost, average network delay per packet (network latency), maximum number of hops between any source-destination pair, and network reliability. The search targets feasible topologies which minimizes the first three objectives and maximizes the fourth objective.

3.3 Design Objectives and Constraints

This section describes the design objectives and constraints considered in this thesis. All objectives are summarized in Section 3.3.1, while the constraints are discussed in Section 3.3.2.

3.3.1 Design objectives

As mentioned earlier, four design objectives are considered. These objectives are:

Monetary cost

The aim is to find a topology with low cost, while satisfying the design constraints (discussed in Section 3.3.2 below). Since the number of network devices would be the same in any topology, the only entity that affects the monetary cost is the cost of cables. Cost is expressed as

$$cost = length \times c_{cable} \tag{3.1}$$

where *length* represents the total length of cable, and c_{cable} represents the cost per unit of the cable used.

Average Network Delay

The second objective is to minimize the average network delay incurred on a packet during transmission from a source node to a destination node.

To estimate the average network delay, the aggregate behavior of a link and network device is modelled by an M/M/1 queue [75]. If a link connects local sites i and j, then the delay per bit due to the network device feeding this link is $B_{i,j} = b_{i,j}/\omega$, where $b_{i,j}$ is the delay per packet, and ω is the average packet size in bits. If γ_{ij} is the total traffic through the network device between local sites i and j, then the average packet delay due to all network devices is:

$$D_{nd} = \frac{1}{\gamma} \sum_{i=1}^{d} \sum_{j=1}^{d} \gamma_{ij} B_{ij} \tag{3.2}$$

where d is the total number of networking devices in the network, and γ is the total traffic in the network, representing the summation of all γ_{ij}. The total average network delay is the summation of delays over all links and network devices, given as [75]:

$$D = \frac{1}{\gamma} \sum_{i=1}^{L} \frac{\lambda_i}{\lambda_{max,i} - \lambda_i} + \frac{1}{\gamma} \sum_{i=1}^{d} \sum_{j=1}^{d} \gamma_{ij} B_{ij} \tag{3.3}$$

where L is the number of links in the topology, λ_i is traffic on link i in bits per second (bps), and $\lambda_{max,i}$ is the capacity of link i in bps.

Maximum number of hops between any source-destination pair

The maximum number of hops between any source-destination pair needs to be minimized. A hop is counted as the packet crosses a network device. The reason for taking number of hops as an optimization objective is due to the restrictions imposed by the routing information protocol (RIP). RIP uses hop count to measure the distance between the source and a destination node. RIP implements a limit on the number of hops encountered in the path from a source to a destination to prevent routing loops from continuing indefinitely [233]. The maximum number of hops allowed in a path is 15. If the hop count exceeds this number, then the destination is considered unreachable [233].

Network reliability

Network reliability should be maximized. Network reliability can be defined as the probability of occurrence of an event in which each node communicates with every other node in the network [2]. For the purposes of this thesis, the topology is a tree. Thus, the reliability of such a topology is the product of the reliabilities of all links present in that particular topology [12, 136]. Mathematically,

$$R_s = \prod_{i=1}^{L} R_i \tag{3.4}$$

where R_i is the reliability of link i, and R_s is reliability of the network.

3.3.2 Constraints

Three types of constraints are considered in this thesis, namely:

1. The maximum number of nodes attached to network device i must not exceed the capacity, p_i, of that device. That is,

$$\sum_{j=1}^{n} t_{ij} < p_i, \qquad i = 1, 2, ..., n, \qquad \forall i \neq j \tag{3.5}$$

   where n is number of nodes in the network, and t_{ij} represents a connection between device (i.e. a node) i and device j.

2. Link bandwidth is limited. Therefore, a good network will employ "reasonably" utilized links, since links with high utilization levels experience delays, congestion, and packet loss. The traffic flow on any link i therefore must be limited by a threshold value, $\lambda_{max,i}$, as follows:

$$\lambda_i < \lambda_{max,i}, \quad i = 1, 2, ..., L \tag{3.6}$$

3. The last set of constraints are specified by the designer, and is used to enforce certain design guidelines and principles:

   (a) Certain nodes must be leaf/terminal nodes. For example, hubs should generally be placed as leaf nodes.

   (b) Certain nodes must be interior nodes of the tree, for example, nodes designated as switches or routers.

(c) Certain nodes cannot be directly connected to the backbone. For example, hubs should not be directly connected to the backbone (i.e. the root node).

3.4 Fuzzy Logic Approach to the DLAN Topology Design Problem

Topology design of LANs is a complex process which requires simultaneous optimization of a number of design objectives. Important objectives include monetary cost, maximum number of hops between a source-destination pair, network average delay, and network reliability. None of these objectives on their own gives sufficient information to decide the quality of the network topology. It is also the case that some of these objectives, such as the delay, can only be approximated. The complexity of the problem is further amplified by the conflicting nature of some of these objectives. Thus, a trade-off between the conflicting objectives is required. Fuzzy logic comfortably provides a mechanism to handle imprecise information since the logic provides a rigorous algebra for dealing with imprecise information. Furthermore, the logic is a convenient method of combining conflicting objectives and expert human knowledge.

The rest of this section describes how fuzzy logic is employed in combining the four conflicting objectives into a single overall objective. This overall objective estimates the quality of a solution in terms of membership of a given topology to the fuzzy set of quality topologies. For simplicity, the following discussion uses the terms cost, delay, hops, and reliability. The goal is to find a high quality solution, represented by a linguistic variable 'good topology'. A good topology consists of low cost, small number of hops, low delay, and high reliability, as summarized in Figure 3.2.

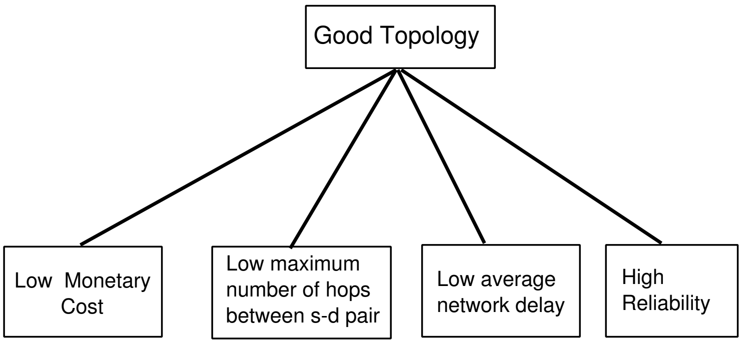

Figure 3.2: Basic components of a good topology

To evaluate the quality of the overall solution using fuzzy logic, the quality of individual objectives needs to be evaluated first through membership functions. Once this is done, fuzzy logic rules can be used to assess the quality of solutions with respect to the individual objectives. For the DLAN topology design problem, a solution should satisfy the four objectives mentioned above. Each of these objectives is evaluated using a membership function. Thus, a membership function needs to be defined for each objective. This process is described below.

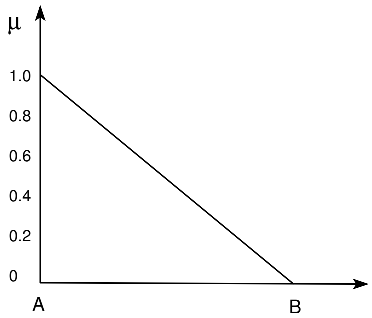

Figure 3.3: Membership function of the objective to be optimized

To find the membership function of cost, two extreme values, the minimum and maximum, are determined first. These values can be found mathematically or from prior knowledge. Figure 3.3 depicts the membership function of the objective to be optimized (cost in this case). In this figure, the two values shown as 'A' and 'B' refer to the minimum cost 'MinC' and the maximum cost 'MaxC'. The membership value for cost of solution x, $\mu_c(x)$, is computed as

$$\mu_c(x) = \begin{cases} 1 & \text{if } Cost(x) \leq MinC \\ \frac{MaxC - Cost(x)}{MaxC - MinC} & \text{if } MinC < Cost(x) \leq MaxC \\ 0 & \text{if } Cost(x) > MaxC \end{cases} \tag{3.7}$$

where the term $Cost(x)$ represents the cost of the solution. The membership function for delay, $\mu_d(x)$, can be defined in a similar way. The two extreme values of delay are 'MinD' and 'MaxD' for minimum and maximum values respectively. In Figure 3.3, 'MinD' corresponds to 'A' and 'MaxD' corresponds to 'B'. The membership value of delay is determined as

$$\mu_d(x) = \begin{cases} 1 & \text{if } Delay(x) \leq MinD \\ \frac{MaxD - Delay(x)}{MaxD - MinD} & \text{if } MinD < Delay(x) \leq MaxD \\ 0 & \text{if } Delay(x) > MaxD \end{cases} \tag{3.8}$$

where the term $Delay(x)$ represents the average delay of the solution. The membership function for number of hops, $\mu_h(x)$, is also illustrated by Figure 3.3, where the minimum (MinH) and maximum (MaxH) values correspond to 'A' and 'B' respectively. The membership value for number of hops is determined as

$$\mu_h(x) = \begin{cases} 1 & \text{if } Hops(x) \leq MinH \\ \frac{MaxH - Hops(x)}{MaxH - MinH} & \text{if } MinH < Hops(x) \leq MaxH \\ 0 & \text{if } Hops(x) > MaxH \end{cases} \tag{3.9}$$

Finally, the membership function for reliability, $\mu_r(x)$, can be determined by finding the maximum and the minimum bounds for the solution reliability. In Figure 3.3, the minimum (MinR) and maximum (MaxR) values correspond to 'B' and 'A' respectively. The membership value for reliability is determined as

$$\mu_r(x) = \begin{cases} 1 & \text{if } Rel(x) \geq MaxR \\ \frac{MaxR - Rel(x)}{MaxR - MinR} & \text{if } MinR < Rel(x) \leq MaxR \\ 0 & \text{if } Rel(x) < MinR \end{cases} \tag{3.10}$$

After obtaining the membership functions of the four objectives, the next phase is to combine these functions into an overall function (i.e. a single objective function) of "good topology" using fuzzy logic. A good topology is one that is characterized by a low cost, low delay, small number of hops, and high reliability. In fuzzy logic, this can be stated by the following fuzzy rule:

Rule 1: **IF** a solution X has *low cost* AND *low delay* AND *low hops* AND *high reliability* **THEN** it is a *good topology*

The expressions "low cost", "low delay", "low hops", "high reliability", and "good topology" are linguistic values, each defining a fuzzy subset of solutions. For example, "high reliability" is the fuzzy subset of topologies of high reliabilities. Each fuzzy subset is defined by a membership function μ. The membership function returns a value in the interval [0,1] which describes the degree of satisfaction with the particular objective criterion. The above fuzzy rule can be mathematically represented using the OWA-AND operator:

$$\mu(x)^O = \beta \; min\{\mu_1(x), \mu_2(x), \mu_3(x), \mu_4(x)\} + (1-\beta)\frac{1}{4}\sum_{i=1}^{4}\mu_i(x) \tag{3.11}$$

In Equation (3.11), $\mu(x)^O$ is the membership value for solution x in the fuzzy set *good topology* using the OWA-AND operator. Also in the same equation, μ_i for i = {1,2,3,4} represents the membership values of solution x in the fuzzy sets *low cost*, *low delay*, *low hops*, and *high reliability* respectively. The solution which results in the maximum value for Equation (3.11) is reported as the best solution found. However, it is also possible to get a set of best solutions having equal membership values (i.e. Pareto optimal solutions), in which case any one of such solutions is taken as the best solution.

3.5 Characteristics of Test Cases

The test cases used in this thesis have been used in other literature [271, 272, 268, 270, 269]. These test cases were used to evaluate the performance of all algorithms proposed in this thesis. The test cases represent networks consisting of local sites. Traffic generated by each local site for these test cases was collected from real sites, and costs of the network cables were collected from vendors. Other characteristics,

Table 3.1: Network characteristics assumed for experiments.

Parameter	Characteristic
Cost of fiber optic cable	$ 5 per meter
Delay per bit due to networking device	$250\mu sec.$
Maximum traffic on a link allowed	60
Average packet size	500 bytes
Type of networking device	Router, switch, or hub
Number of ports on a networking device	4, 8, or 12

such as the number of ports on a network device and its type are listed in Table 3.1. Five test cases were used for experimental work. For these test cases, the number of local sites ranged between 15 (denoted by *n15*) and 50 (denoted by *n50*). Table 3.2 summarizes the features of these test cases.

Table 3.2: Characteristics of test cases used in experiments. MinC is in US$, MinD is in milliseconds, and traffic is in Mbps.

Test Case	# of Local Sites	MinC	MinD	MaxR	Traffic
n15	15	4640	2.14296	0.868746	24.63
n25	25	5120	2.15059	0.785678	74.12
n33	33	8158	2.15444	0.72498	117.81
n40	40	9646	2.08757	0.675729	144.76
n50	50	11616	2.08965	0.611117	164.12

3.5.1 Upper and Lower Bounds for Objective Values

The extreme values of the objectives can be found as given below. Some of these values, such as MinC, MaxC, MinD, MinH, and MaxR, can be pre-calculated, while others such as MaxD, MaxH, and MinR are computed during the initialization step of the optimization algorithm.

For the cost objective, the minimum value, 'MinC', is found by using the Esau-

Williams algorithm [80], with all the constraints completely relaxed. This guarantees that the minimum possible cost of the topology is obtained. The value of 'MinC' for the five test cases is given in Table 3.2. The maximum value for cost, 'MaxC', is taken to be the cost generated by the initialization step of the optimization algorithm. Once the two extreme values are available, the membership value, μ_c, of cost is computed using Equation (3.7).

The minimum value for the delay objective, 'MinD', is found by connecting all the nodes directly to the root node, ignoring all constraints. The value of 'MinD' for the five test cases is given in Table 3.2. The maximum value of delay, 'MaxD', is taken to be the delay generated by the initialization step. Once these values are obtained, the delay membership value, μ_d, is calculated using Equation (3.8).

For the number of hops objective, the minimum value, 'MinH', is taken to be 1 hop. The maximum value, 'MaxH', is taken to be the maximum number of hops between any source-destination pair generated by the initialization step. The membership value of hops, μ_h, is computed using Equation (3.9).

Finally, for the reliability objective, the maximum reliability, 'MaxR', is found using the Esau-Williams algorithm [80], with all constraints relaxed. This guarantees the maximum possible reliability that could be attained. The value of 'MaxR' for the five test cases is given in Table 3.2. The minimum reliability, 'MinR', is taken to be the reliability of the initial solution. The membership value for reliability, μ_r, is found using Equation (3.10).

3.6 Conclusion

This chapter defined and discussed the problem of topology design of distributed local area networks formulated as a multi-objective optimization problem. Necessary background information along with important concepts, design objectives, and problem constraints were discussed in sufficient detail. Moreover, aggregation of individual objectives into a combined single fuzzy function using the ordered weighted averaging operator was also covered. In addition, this chapter provided characteristics of test cases used to evaluate the performance of each proposed algorithm, along with the upper and lower bounds of the objective values. The next chapter discusses a new fuzzy aggregating operator proposed in this thesis.

Chapter 4

The Unified AND-OR Fuzzy Operator

A new fuzzy operator is proposed and discussed in detail in this chapter. This new operator is shown to have mathematical properties similar to that of the ordered weighted averaging operator. A new preference handling approach is also proposed. To illustrate the effectiveness of the proposed operator and the preference scheme, empirical analysis is performed using some examples, and a comparison with the OWA operator is done.

4.1 Definition of the Unified AND-OR Operator

This section focusses on a new operator proposed in this thesis, namely the *unified AND-OR* operator or UAO. As will be seen, this operator uses a single equation (unlike the two separate equations for AND and OR of Yager's OWA operator), yet it is capable of behaving either as the OWA-AND or the OWA-OR operator. The behavior is controlled by a variable $\nu \geq 0$, whose value decides whether the function will behave as AND or OR. The operator is defined as

$$f(a,b) = \frac{ab + \nu \max\{a,b\}}{\nu + \max\{a,b\}} = \begin{cases} \mathrm{I}_{\star} = \mu_{A \cup B}(x) & \text{if } \nu > 1 \\ \mathrm{I}^{*} = \mu_{A \cap B}(x) & \text{if } \nu < 1 \end{cases} \tag{4.1}$$

where a represents the membership value of μ_A (i.e. $a = \mu_A$), b represents the membership value of μ_B (i.e. $b = \mu_B$), and $f(a,b)$ represents the value of the overall objective function (i.e. $f(a,b) = \mu_{AB}$). I^* represents the AND operation using the UAO operator, and $I_\star$ denotes the OR operation using the UAO operator.

In Equation (4.1), parameter ν is used to orient the equation such that the UAO behaves either as the AND or the OR operator. With $\nu < 1$, the UAO behaves as the OWA-AND operator. A value of $\nu = 0$ gives the pure-AND behavior. On the other hand, a value of $\nu > 1$ shifts the behavior of the UAO towards the OWA-OR operator. As $\nu \longrightarrow \infty$, the ORing becomes more 'rigid'. The UAO behaves as pure-OR when $\nu = \infty$. However, experimentation with different values of ν suggests that the behavior of the UAO is quite similar to that of the OWA-OR with $1 \leq \nu \leq 100$. Figure 4.1 depicts instances of the UAO with different values of ν. As illustrated in Figure 4.1(a), a value of $\nu = 0$ for UAO gives exactly the same behavior as that of OWA-AND with β =1 (see Figure 2.4(f)). Similarly, Figure 4.1(b) and (c) depict almost the same behavior illustrated in Figure 2.4. When ν = 1 (Figure 4.1(d)), the UAO operator behaves almost the same as the OWA-AND with $\beta = 0$ (or equivalently, as OWA-OR with $\beta = 0$).

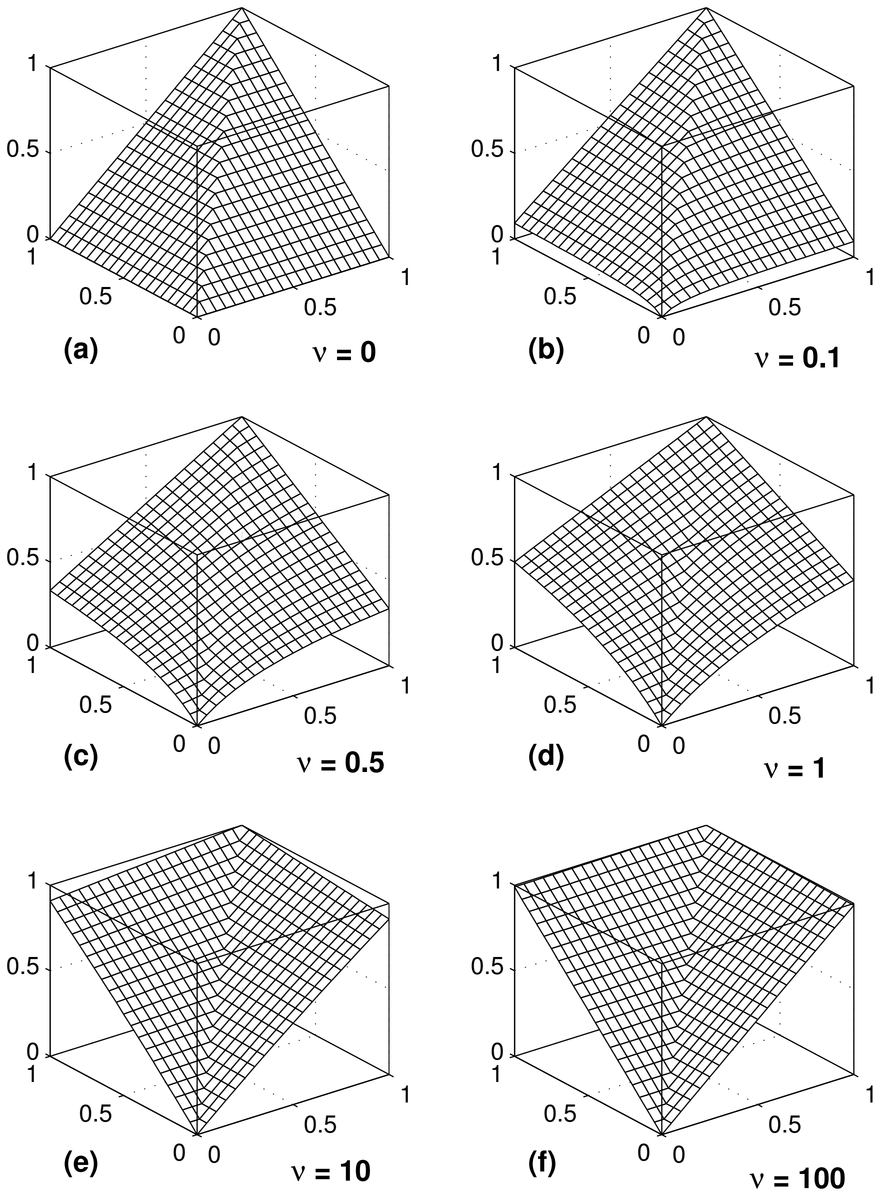

Figure 4.1: Effect of ν on Unified AND-OR operator

4.2 Mathematical Properties

This section proves that the UAO operator also satisfies the properties of monotonicity, symmetry, and idempotency, similar to OWA. Assume a value, $\Delta a > 0$, which represents the change in μ_A (i.e. $\Delta a = \Delta\mu_A$).

1. **Monotonicity:** The claim is that the UAO operator is monotonic. Thus, it is necessary to prove that

$$f(a,b) \geq \; f(\ddot{a},\ddot{b}) \text{ if } a \geq \ddot{a} \text{ and } b \geq \ddot{b}$$

To prove this, several cases need to be considered. The proof of each case is given below:

Case 1: $a + \Delta a < b \Rightarrow a < b$. It is to be proven that $f(a+\Delta a, b) > f(a,b)$.

Proof: From Equation (4.1),

$$f(a+\Delta a, b)\Big|_{a+\Delta a<b, a<b} = \frac{(a+\Delta a)b + \nu b}{\nu + b}$$

$$= \frac{ab+\nu b}{\nu+b} + \frac{\Delta ab}{\nu+b} \tag{4.2}$$

$$f(a,b)\Big|_{a<b} = \frac{ab+\nu b}{\nu+b} \tag{4.3}$$

Comparing Equations (4.2) and (4.3), and since $\frac{\Delta ab}{\nu+b} > 0$, it is concluded that $f(a+\Delta a, b) > f(a,b)$.

Case 2: $a + \Delta a > b$ and $a > b$. It is to be proven that $f(a+\Delta a, b) > f(a,b)$.

Proof: Since $a > b$, take $a = b + k_2$ where $k_2 > 0$. Also $a + \Delta a > b$, which implies that $a + \Delta a = k_1 + b$, where $k_1 > 0$. We find $\Delta a = k_1 - k_2$. From

Equation (4.1),

$$f(a+\Delta a,b)\Big|_{a+\Delta a>b,\ a>b} = \frac{(a+\Delta a)b+\nu(a+\Delta a)}{\nu+(a+\Delta a)} = \frac{(a+\Delta a)(b+\nu)}{\nu+(a+\Delta a)} \quad (4.4)$$

$$f(a,b)\Big|_{a>b} = \frac{ab+\nu a}{\nu+a} = \frac{a(b+\nu)}{\nu+a} \quad (4.5)$$

Substituting $\Delta a = k_1 - k_2$ and $a = b + k_2$ in Equations (4.4) and (4.5) respectively, yields:

$$f(a+\Delta a,b)\Big|_{a+\Delta a>b,\ a>b} = \frac{(b+k_1)(b+\nu)}{\nu+b+k_1} \quad (4.6)$$

$$f(a,b)\Big|_{a>b} = \frac{(b+k_2)(b+\nu)}{\nu+b+k_2} \quad (4.7)$$

Now, compare Equations (4.6) and (4.7) as follows:

$$f(a+\Delta a,b)\Big|_{a+\Delta a>b,\ a>b} \overset{?}{\gtrless} f(a,b)\Big|_{a>b} \quad (4.8)$$

That is,

$$L.H.S. \overset{?}{\gtrless} R.H.S.$$

$$\Rightarrow \frac{(b+k_1)(b+\nu)}{\nu+b+k_1} \overset{?}{\gtrless} \frac{(b+k_2)(b+\nu)}{\nu+b+k_2} \quad (4.9)$$

$$\therefore \frac{b+k_1}{\nu+b+k_1} \overset{?}{\gtrless} \frac{b+k_2}{\nu+b+k_2} \quad (4.10)$$

Cross multiplication yields:

$$(b+k_1)(\nu+b+k_2) \overset{?}{\gtrless} (b+k_2)(\nu+b+k_1) \quad (4.11)$$

Solving Equation (4.11) results in:

$$\nu(b + k_1) \overset{?}{\gtrless} \nu a \tag{4.12}$$

$$\therefore b + k_1 \overset{?}{\gtrless} a \tag{4.13}$$

Since $a + \Delta a = k_1 + b \Rightarrow a = k_1 + b - \Delta a$, Equation (4.13) becomes

$$(b + k_1) \overset{?}{\gtrless} (b + k_1 - \Delta a) \tag{4.14}$$

Simplifying Equation (4.14) yields:

$$\Delta a > 0 \tag{4.15}$$

From Equation (4.15), since L.H.S. > R.H.S., it is concluded that $f(a + \Delta a, b) > f(a, b)$.

Case 3: $a + \Delta a > b$ and $a < b$. It is to be proven that $f(a + \Delta a, b) > f(a, b)$.

Proof: Since $a < b$, assume $a + k_2 = b$ where $k_2 > 0$. Also $a + \Delta a > b$, which implies that $a + \Delta a = k_1 + b$, where $k_1 > 0$. Therefore $\Delta a = k_1 + k_2$. From Equation (4.1),

$$f(a + \Delta a, b)\Big|_{a+\Delta a>b,\ a<b} = \frac{(a + \Delta a)b + \nu(a + \Delta a)}{\nu + (a + \Delta a)} = \frac{(a + \Delta a)(b + \nu)}{\nu + a + \Delta a} \tag{4.16}$$

$$f(a, b)\Big|_{a>b} = \frac{ab + \nu b}{\nu + b} = \frac{b(a + \nu)}{\nu + b} \tag{4.17}$$

Substituting $\Delta a + a = b + k_1$ and $a = b - k_2$ in Equations (4.16) and (4.17) respectively, yields

$$f(a+\Delta a, b)\Big|_{a+\Delta a>b,\ a<b} = \frac{(b+k_1)(b+\nu)}{\nu+b+k_1} \tag{4.18}$$

$$f(a, b)\Big|_{a>b} = \frac{b[(b-k_2)+\nu]}{\nu+b} \tag{4.19}$$

Now, compare Equations (4.18) and (4.19) as follows:

$$f(a+\Delta a, b)\Big|_{a+\Delta a>b,\ a<b} \overset{?}{\gtrless} f(a, b)\Big|_{a>b} \tag{4.20}$$

That is,

$$L.H.S. \overset{?}{\gtrless} R.H.S.$$

$$\Rightarrow \frac{(b+k_1)(b+\nu)}{\nu+b+k_1} \overset{?}{\gtrless} \frac{b[(b-k_2)+\nu]}{\nu+b} \tag{4.21}$$

Cross multiplication yields:

$$(b+k_1)(\nu+b)^2 \overset{?}{\gtrless} b(b-k_2+\nu)(\nu+b+k_1) \tag{4.22}$$

Taking $b+\nu = b'$ reduces Equation (4.22) to:

$$(b+k_1)(b')^2 \overset{?}{\gtrless} b(b'-k_2)(b'+k_1) \tag{4.23}$$

$$\therefore (b')^2 k_1 + bk_2(b'+k_1) \overset{?}{\gtrless} bb'k_1 \tag{4.24}$$

$$\therefore ((b')^2 + bk_2)k_1 + b'bk_2 \overset{?}{\gtrless} b'bk_1. \tag{4.25}$$

Substituting $b' = b + \nu$ back into Equation (4.25) gives:

$$((b+\nu)^2 + bk_2)k_1 + (b+\nu)bk_2 \overset{?}{\gtrless} (b+\nu)bk_1 \quad (4.26)$$

$$\therefore b^2k_1 + \nu^2k_1 + 2\nu bk_1 + bk_1k_2 + b^2k_2 + \nu bk_2 \overset{?}{\gtrless} b^2k_1 + \nu bk_1 \quad (4.27)$$

$$\Rightarrow \nu^2k_1 + \nu bk_1 + bk_1k_2 + b^2k_2 + \nu bk_2 > 0 \quad (4.28)$$

Since L.H.S. $>$ R.H.S. for all b and for $\nu \geq 0$, it is concluded from Equation (4.28) that $f(a + \Delta a, b) > f(a, b)$.

The above proofs considered the case $f(a + \Delta a, b) > f(a, b)$ with different possible scenarios. The case where $f(a, b + \Delta b) > f(a, b)$ can be proven in a similar manner. Moreover, the proof can be extended to more than two variables. Thus, it can be claimed that the UAO operator is monotonic for any number of variables.

2. **Symmetry (generalized commutativity):** It is obvious from the structure of the UAO operator that the order of arguments does not matter. Thus the operator is symmetric.

3. **Idempotency:** It is to be proven that

$$f(a, a) = \frac{a \cdot a + \nu\ max(a, a)}{\nu + max(a, a)} = a \quad (4.29)$$

Proof: From Equation (4.29),

$$f(a, a) = \frac{a^2 + \nu\ a}{\nu + a} = \frac{a(a + \nu)}{\nu + a} = a \quad (4.30)$$

□

4.3 Fuzzy Rules for Topology Design

The four objectives can be combined in a number of ways using fuzzy operators to generate rules with a single consequent, i.e. "good topology". One possibility is the extreme where it is required that all the objectives are simultaneously optimized, thus implying the AND operation among all objectives, as mentioned earlier in Chapter 3. Another extreme is where optimization of any one objective would suffice. A number of combinations exist between these extremes. These cases, as well as the application of the proposed UAO operator on them, are discussed next.

4.3.1 Case 1: Simultaneous Optimization of All Four Objectives

For the extreme case where all objectives have to be optimized, the fuzzy rule is:

- R1: IF cost is *low* AND delay is *low* AND hops is *low* AND reliability is *high* THEN the topology is *good*.

The corresponding mathematical representation using the UAO operator is:

$$\mu_s = I^*(\mu_c, \mu_d, \mu_h, \mu_r) \tag{4.31}$$

4.3.2 Case 2: Simultaneous Optimization of Three Objectives

A number of cases can be defined when any three of the four objectives need to be considered. Some of these cases and their UAO representations are:

- R2a: IF cost is *low* AND delay is *low* AND (hops is *low* OR reliability is *high*) THEN the topology is *good*.

$$\mu_s = I^*(\mu_c, \mu_d, I_\star(\mu_h, \mu_r)) \tag{4.32}$$

- R2b: IF cost is *low* AND (delay is *low* OR hops is *low*) AND reliability is *high* THEN the topology is *good*.

$$\mu_s = I^*(\mu_c, \mu_r, I_\star(\mu_d, \mu_h)) \tag{4.33}$$

- R2c: IF cost is *low* AND hops is *low* AND (delay is *low* OR reliability is *high*) THEN the topology is *good*.

$$\mu_s = I^*(\mu_c, \mu_h, I_\star(\mu_d, \mu_r)) \tag{4.34}$$

- R2d: IF (cost is *low* OR reliability is *high*) AND delay is *low* AND hops is *low* THEN the topology is *good*.

$$\mu_s = I^*(\mu_d, \mu_h, I_\star(\mu_c, \mu_r)) \tag{4.35}$$

4.3.3 Case 3: Simultaneous Optimization of Two Objectives

When only two objectives need to be optimized, a number of different cases can be defined. Some of these cases and their UAO representations are given below.

- R3a: IF cost is *low* AND (delay is *low* OR hops is *low* OR reliability is *high*)

THEN the topology is *good*.

$$\mu_s = I^*(\mu_c, I_\star(\mu_d, \mu_h, \mu_r)) \tag{4.36}$$

- R3b: IF delay is *low* AND (cost is *low* OR hops is *low* OR reliability is *high*) THEN the topology is *good*.

$$\mu_s = I^*(\mu_d, I_\star(\mu_c, \mu_h, \mu_r)) \tag{4.37}$$

- R3c: IF (cost is *low* AND delay is *low*) OR (hops is *low* AND reliability is *high*) THEN the topology is *good*.

$$\mu_s = I_\star(I^*(\mu_c, \mu_d), I^*(\mu_h, \mu_r)) \tag{4.38}$$

- R3d: IF (cost is *low* AND reliability is *high*) OR (delay is *low* AND of hops is *low*) THEN the topology is *good*.

$$\mu_s = I_\star(I^*(\mu_c, \mu_r), I^*(\mu_d, \mu_h)) \tag{4.39}$$

- R3e: IF (cost is *low* OR delay is *low*) AND (hops is *low* OR reliability is *high*) THEN the topology is *good*.

$$\mu_s = I^*(I_\star(\mu_c, \mu_d), I_\star(\mu_h, \mu_r)) \tag{4.40}$$

- R3f: IF (cost is *low* OR hops is *low*) AND (delay is *low* OR reliability is *high*)

THEN the topology is *good*.

$$\mu_s = I^*(I_\star(\mu_c, \mu_h), I_\star(\mu_d, \mu_r)) \tag{4.41}$$

4.3.4 Case 4: Optimization of Any One Objective

For the extreme where any one of the four objectives is optimized, the fuzzy rule is given as:

- R4: IF cost is *low* OR delay is *low* OR hops is *low* OR reliability is *high* THEN the topology is *good*.

The corresponding mathematical representation using the UAO operator is:

$$\mu_s = I_\star(\mu_c, \mu_d, \mu_h, \mu_r) \tag{4.42}$$

4.4 Preferences and UAO

Many of the fuzzy rules mentioned above treat the objectives equally (i.e. giving no preference to any objective). For example, in rule R1, all the objectives get equal preference. This type of situation limits the accuracy in decision-making. To enhance the precision in decision-making, an additional set of rules is necessary. This set can be used to give preference to one criterion or another to emphasize or de-emphasize certain objective(s) involved in the decision-making process.

As an example, consider the set of rules given in Section 4.3.2. In all the rules, the three objectives are equally weighted. If the designer wants to emphasize cost in all cases, he/she has no way of doing so. However, with the availability of preference rules, it will be easier to give more significance to cost.

To formulate the preference rules, preference terms need to be defined. These terms are associated with the main linguistic terms. For the fuzzy rules defined above, the linguistic terms "low", "high", and "good" have been used. The literature [18, 131, 132] has reported some approaches to find preference terms and preference rules based on membership functions. These approaches map a fuzzy preference relation P to a fuzzy membership function μ_P in the range [0,1] as follows:

$$\mu_P(\mathrm{s}_i, s_j) = \begin{cases} 1 & \text{if } s_i \text{ is definitely preferred to } s_j \\ c \in (0.5, 1) & \text{if } \mathrm{s}_i \text{ is slightly preferred to } s_j \\ 0.5 & \text{if there is no preference (i.e., indifference)} \\ d \in (0, 0.5) & \text{if } \mathrm{s}_j \text{ is slightly preferred to } s_i \\ 0 & \text{if } \mathrm{s}_j \text{ is definitely preferred to } s_i \end{cases}$$

The above representation is impractical for decision-makers since the degree of preference for a certain objective is not precise [166]. For example, the term "slightly" represents indefinite numbers of the numeric preference values which are within the range (0.5, 1) or (0.0, 0.5) [166]. To overcome this problem, it is suggested that the numerical representation be extended by using a wider variety of linguistic terms which includes seven linguistic terms. Thus, the above set of preferences is extended as follows:

$$\mu_{Pm}(\mathrm{s}_i, s_j) = \begin{cases} 1 & \text{if } s_i \text{ is definitely preferred to } s_j \\ c \in (0.917, 1) & \text{if } s_i \text{ is strongly preferred to } s_j \\ d \in (0.834, 0.917) & \text{if } \mathrm{s}_i \text{ is highly preferred to } s_j \\ e \in (0.751, 0.834) & \text{if } \mathrm{s}_i \text{ is considerably preferred to } s_j \\ f \in (0.668, 751) & \text{if } \mathrm{s}_i \text{ is moderately preferred to } s_j \\ g \in (0.585, 0.668) & \text{if } \mathrm{s}_i \text{ is slightly preferred to } s_j \\ h \in (0.5, 0.585) & \text{if } \mathrm{s}_i \text{ is mildly preferred to } s_j \\ 0.5 & \text{if there is no preference (i.e., indifference)} \\ i \in (0.417, 0.5) & \text{if } \mathrm{s}_j \text{ is mildly preferred to } s_i \\ j \in (0.337, 0.417) & \text{if } \mathrm{s}_j \text{ is slightly preferred to } s_i \\ k \in (0.254, 0.337) & \text{if } \mathrm{s}_j \text{ is moderately preferred to } s_i \\ l \in (0.171, 0.254) & \text{if } \mathrm{s}_j \text{ is considerably preferred to } s_i \\ m \in (0.088, 0.171) & \text{if } \mathrm{s}_j \text{ is highly preferred to } s_i \\ n \in (0, 0.088) & \text{if } s_j \text{ is strongly preferred to } s_i \\ 0 & \text{if } s_j \text{ is definitely preferred to } s_i \end{cases}$$

Notice that the preference values are reflective-reciprocal at the point of 'indifference'. The evaluation values can be seen as consisting of two groups: 1) s_i preferred to s_j, and 2) s_j preferred to s_i. The value corresponding to 'indifference' may belong to both groups. Since the two groups are similar, and each group includes seven preference terms (including the 'indifference'), the above representation of preference values can be curtailed to a representation of only seven values, instead of fourteen. Thus, it can be said that $c \Longleftrightarrow n$, $d \Longleftrightarrow m$, and so on, where "$\Longleftrightarrow$" means that the two preference terms are equivalent. The interpretation of the "two terms being equivalent" is that if the first objective is preferred to the second, then the second objective is not preferred to the first. For example, if we say that s_i is *strongly preferred* to s_j, then it is equivalent to saying that s_j is *strongly not preferred* to s_i, etc. The seven-point evaluation is also advocated by Miller's [174] observation that the human mind can deal with around seven items at a time.

The proposed set of preferences can be seen as a subset of the approach presented by Cvetković *et al.* [47]. The approach in [47] suggests a more comprehensive scheme for the use of preferences in multi-objective optimization. However, the scheme in [47] seems more suitable for situations where the number of objectives is high (the authors have applied their approach to a multi-objective problem with thirteen objectives). Their approach involved more complex steps than are proposed in this thesis.

To continue, preference terms proposed in this thesis can be used to define a number of preference rules. These preference rules are divided into several categories, as described below.

4.4.1 Preference rules involving all four objectives:

Examples of preference rules that contain all four objectives are:

- PR1a: Cost is strongly preferred over the other three objectives
- PR1b: Delay is highly preferred over the other three objectives
- PR1c: Reliability is strongly preferred over the other three objectives
- PR1d: Cost is slightly preferred over the other three objectives

4.4.2 Preference rules involving three objectives:

It is also possible to develop preference rules which include three objectives, for example,

- PR2a: Cost is strongly preferred over hops and reliability
- PR2b: Delay is highly preferred over reliability and cost
- PR2c: Reliability is slightly preferred over hops and delay
- PR2d: Cost is slightly preferred over reliability and hops

4.4.3 Preference rules involving two objectives:

Examples of preference rules involving two objectives are:

- PR3a: Cost is strongly preferred over hops
- PR3b: Delay is mildly preferred over reliability
- PR3c: Reliability is considerably preferred over hops
- PR3d: Hops is slightly preferred over cost
- PR3e: Delay is highly preferred over reliability
- PR3f: Cost is mildly preferred over delay

4.4.4 Combining the main rules with preference rules

Once the main rules and the preference rules have been defined, the next step is to combine them together. This may lead to a number of scenarios, for example:

1. **Ex1**: If rule R1 is used at the first level, and *Cost* is emphasized over the other three objectives with PR1a at the second level, then the value of the objective function using the UAO operator is computed by the following formula:

$$\mu_s = I^*(0.917\mu_c, 0.088\mu_d, 0.088\mu_h, 0.088\mu_r)$$

2. **Ex2**: When rule R1 is used at the first level, and the emphasis is given to *reliability* using PR1c on the second level, then the value of the objective function using the UAO operator is computed by the following formula:

$$\mu_s = I^*(0.088\mu_c, 0.088\mu_d, 0.088\mu_h, 0.917\mu_r)$$

3. **Ex3**: If rule R1 is used at the first level and PR3c is used at the second level, thus giving emphasis to *reliability* over *number of hops*, then the value of the objective function can be computed as follows:

$$\mu_s = I^*((\mu_c, \mu_d, 0.25\mu_h), 0.75\mu_r)$$

4. **Ex4**: When rule R2d is used at the first level and PR2c is used at the second level, thus emphasizing *reliability* over *number of hops*, the value of the objective function can be computed as follows:

$$\mu_s = I^*(0.35\mu_d, 0.35\mu_h, I_\star(\mu_c, 0.65\mu_r))$$

5. **Ex5**: If rule R1 is used at the first level and both PR3b and PR3d are used at the second level, the emphasis is given to *delay* over *reliability* and to *number of hops* over *cost*. In this case the value of the objective function is computed by the following formula:

$$\mu_s = I^*(0.40\mu_c, 0.55\mu_d, 0.60\mu_h, 0.45\mu_r)$$

6. **Ex6**: If rule R3c is used at the first level and both PR3c and PR3f are used at the second level, the emphasis is given to *cost* over *delay* and to *reliability* over *number of hops*. In this case the value of the objective function is computed by the following formula:

$$\mu_s = I_\star(I^*(0.55\mu_c, 0.45\mu_d), I^*(0.20\mu_h, 0.80\mu_r))$$

7. **Ex7**: If rule R1 is used at the first level, PR2a is used at the second level, and both PR3b and PR3c are used at the third level, the emphasis is given to *cost* over *number of hops* and *reliability*. Then, the preference is given to *delay* over *reliability* and to *reliability* over *number of hops*. In this case the value of the objective function is computed by the following formula:

$$\mu_s = I^*(0.95\mu_c, 0.55\mu_d, 0.05 \times 0.20\mu_h, 0.05 \times 0.80 \times 0.45\mu_r)$$

The examples above provide different scenarios of using main rules with preference rules. Examples 1 to 4 suggest using rules at two levels. On the first level, the main rule is used, whereas at the second level one preference rule is used. An extension of this two level case is observed in examples 5 and 6, where a main rule is used at the first level, and two preference rules are used at the second level. Similarly, example 7 illustrates how rules at three levels might be used. In this example, it is observed that a main rule is used at the first level, followed by a preference rule involving three objectives at the second level, and two preference rules at the third level with each involving two objectives. Apart from the aforementioned scenarios, there exist many other possibilities where multi-level rules are possible and multiple preference rules could be used at a certain level.

4.5 Application of UAO to Topology Design

A multi-objective SimE algorithm for network topology design was proposed in [271, 268]. This algorithm was engineered to optimize three design objectives: cost, delay, and number of hops. The main focus of the algorithm was to incorporate fuzzy logic in the **allocation** phase to compute the overall goodness (or fitness) of the solution.

For the purposes of this chapter, the above SimE algorithm has been adopted to evaluate the performance of UAO and OWA. This evaluation is done by using rules Ex1 and Ex2 presented in Section 4.4.4. Although SimE was used for this experiment, any stochastic optimization algorithm, such as a genetic algorithm, simulated annealing, stochastic evolution, ant colony optimization, or particle swarm optimization, can be used instead.

The main focus of the fuzzy SimE algorithm in [271] and [268] was to incorporate fuzzy logic in the allocation phase to compute the overall goodness (or fitness) of the solution. This overall goodness was calculated using a fuzzy rule similar to rule R1 (refer to Section 4.3.1). The only difference between the rule in [271] and [268], and R1 is that the latter also includes reliability as an optimization factor along with the other three objectives. The overall goodness is used to compare the quality of two solutions. Results are presented in the following section.

4.6 Empirical Results and Discussion

This section discusses empirical results obtained by applying the UAO and OWA operators to SimE using rules Ex1 and Ex2 given in Section 4.4.4. The two operators were applied to the allocation phase of the SimE algorithm, as described in detail in Chapter 6.

The five test cases as described in Chapter 3 were used for performance evaluation. As mentioned earlier, SimE uses a problem dependent bias parameter, B. An appropriate selection of bias is important for convergence of SimE to near-optimal or optimal set of solutions. Thus, it is essential to find the appropriate bias value. Tables 4.1 to 4.4 provide these values, which were obtained after running several tests with different bias values ranging from 0.0 to 0.4 for each test case. Also, once the appropriate bias value was found, thirty runs were executed for each test case for each of the two operators (i.e. UAO and OWA). The average of these runs is also reported in Tables 4.1 to 4.4 for each case. Moreover, through trial runs it was found that the algorithm converges within 4000 iterations. Each run was therefore executed for 4000 iterations.

4.6.1 Application of UAO and OWA to Ex1

In Ex1 of Section 4.4.4, the rule requires optimization of all four objectives, with strong preference given to cost. Tables 4.1 and 4.2 summarize the results obtained after application of UAO and OWA to SimE using rule Ex1. These results reflect the percentage improvement of the final solution (average of thirty runs) with respect to the initial solution. Simulations were run using values of $\beta = 0.5$ and $\nu = 0.5$ for OWA and UAO respectively. It is observed from these tables that both OWA and UAO demonstrated a noticeable improvement in cost for all test cases, as validated by the t-test for statistical significance. These improvements range from 25.46% to 32.42% for UAO and from 25.26% to 32.57% for OWA. The average cost improvement for UAO was 29.17 %, while OWA showed an improvement of 28.38%. Thus, as far as cost is concerned, both operators performed almost equally well, with OWA having, in general, a slightly higher standard deviation than UAO. However, although strong preference was given to cost, there was a remarkable improvement in reliability, with improvements of 4.53% to 77.74% for UAO and 13.16% to 86.45% for OWA. In general, the improvement in reliability was also statistically significant (as validated by a t-test), with the exception of *n15* when UAO was used. One possible argument for this behavior is that the search space could be constricted such that no further improvement in cost is possible, irrespective of the emphasis placed on the cost objective. However, reliability might have a tendency to improve remarkably in most parts of the search space, with the improvement being more significant in some regions. Thus, for the reliability objective, OWA performed better than UAO with almost the same level of standard deviation for most of the test cases, with the exception of *n50*.

Table 4.1: Results for UAO for Ex1. N= number of local sites in the network, B=Bias, C=Cost, D=Delay, H=Hops, R=Reliability, Avg= Average percentage improvement of the five test cases. Statistically significant improvements are in italics.

Case	N	B	Percentage improvement				Standard deviations			
			C	D	H	R	C	D	H	R
n15	15	0.1	*25.46*	6.67	-6.31	4.53	5.24	11.09	18.56	11.13
n25	25	0.3	*30.85*	-7.60	*-37.20*	*40.14*	7.19	6.16	17.71	15.43
n33	33	0.3	*31.50*	-8.52	*-33.70*	*67.42*	4.38	13.43	15.62	14.30
n40	40	0.2	*32.42*	7.68	*-26.55*	*68.33*	3.82	52.68	26.49	15.54
n50	50	0.3	*25.62*	-17.96	*-30.01*	*77.74*	6.37	28.74	19.16	17.63
Avg			29.17	-3.95	-26.75	51.63				

As far as delay and number of hops are concerned, it should be kept in mind that rule Ex1 requires optimization of all four objectives, and that the objectives are conflicting in nature. Therefore, it is also noticed in Tables 4.1 and 4.2 that improvement in cost (and reliability) is achieved at the price of deterioration in number of hops and delay objectives. However, it is apparent from the tables that, on average, the percentage of deterioration in the delay and number of hops objectives was less with UAO than that of OWA (i.e. -3.95% for UAO versus -24.38% for OWA for delay, and -26.75% for UAO compared with -28.76% for OWA for number of hops). As far as standard deviations are concerned, OWA showed relatively higher values than UAO for the delay, specifically for *n33* and *n50*. For number of hops, the deviations were comparable for both UAO and OWA. Collectively, it can be fairly claimed that UAO showed a better overall performance than OWA.

Table 4.2: Results for OWA for Ex1. N= number of local sites in the network, B=Bias, C=Cost, D=Delay, H=Hops, R=Reliability, Avg= Average percentage improvement of the five test cases. Statistically significant improvements are in italics.

Case	N	B	Percentage improvement				Standard deviations			
			C	D	H	R	C	D	H	R
n15	15	0.4	*25.26*	-3.39	*-6.42*	*13.16*	9.25	7.67	16.45	31.02
n25	25	0.3	*26.55*	-7.91	*-33.43*	*54.02*	7.06	8.70	30.70	11.76
n33	33	0.2	*32.57*	-15.94	*-29.02*	*72.73*	5.69	46.72	24.43	12.18
n40	40	0.0	*29.27*	*-29.67*	*-36.91*	*76.26*	5.09	53.79	15.53	15.28
n50	50	0.1	*28.22*	*-64.97*	*-38.02*	*86.45*	9.03	67.36	10.96	6.06
Avg			28.38	-24.38	-28.76	60.52				

4.6.2 Application of UAO and OWA to Ex2

Rule Ex2 of Section 4.4.4 has a structure similar to that of rule Ex1, since Ex2 also requires the optimization of all four objectives. However, for rule Ex2, strong preference is given to reliability. The results of the application of UAO and OWA to SimE for Ex2 are given in Tables 4.3 and 4.4. Values of $\beta = 0.5$ and $\nu = 0.5$ were used for OWA and UAO respectively. The tables depict the expected behavior of UAO and OWA when reliability was given a strong preference over the other three objectives. For both UAO and OWA, a significant improvement was observed for almost all test cases, with the exception of test case *n15* while UAO was used. For example, the improvement in reliability for UAO ranges from 78.71% (for *n25*) to 93.51% (for *n50*). These improvements were also statistically significant. Only one case (*n15*) deviated from this trend, achieving an improvement of only 7.48%. A similar behavior was observed for OWA, where a statistically significant improvement ranging from 17.85% to 93.56% was achieved for test cases *n15* to *n50* respectively. On average, both UAO and OWA showed the same level of improvement in reliability, with

69.89% for UAO and 72.24% for OWA. The deviations in both cases were almost of the same magnitude. As for the cost objective, the obtained results suggest that both UAO and OWA were able to attain almost the same level of improvement for all test cases. On average, UAO reduced the cost by 11.85% compared to 11.74% by OWA. The standard deviations were also comparable for the cost objective.

For the delay and number of hops objectives, the tables show that, on average, both objectives deteriorated for almost all test cases. However, in general, the level of deterioration for the number of hops objective was not statistically significant. The reason for the deterioration in the delay and number of hops objectives is the same as for Ex1, i.e. the four objectives are conflicting in nature, and improvement in some objectives is achieved at the cost of deterioration in the others. However, average deterioration in delay for UAO (-9.2%) is less than that in OWA (-21.81%). Similarly, for the number of hops objective, the deterioration for UAO (-16.42%) is also less than that of OWA (-28.39%). However, for the hops objective, OWA generally showed slightly higher deviations than UAO. Overall, it could be claimed that UAO showed a better performance than OWA for Ex2.

The analysis of results presented above for Ex1 and Ex2 shows that both UAO and OWA effectively handled the multi-objective aspects of the DLAN topology design problem in presence of preferences. On the one hand, there was a case where strong preference was given to the cost objective, while on the other, the situation required strong preference for reliability. The two examples were specifically chosen to reflect the view that preferences diversified the search into different solution subspaces.

Table 4.3: Results for UAO for Ex2. N= number of local sites in the network, B=Bias, C=Cost, D=Delay, H=Hops, R=Reliability, Avg= Average percentage improvement of the five test cases. Statistically significant improvements are in italics.

Case	N	B	Percentage improvement				Standard deviations			
			C	D	H	R	C	D	H	R
n15	15	0.1	6.88	2.55	*-9.98*	7.48	8.97	6.97	14.70	18.66
n25	25	0.3	*15.54*	-5.78	*-23.33*	*78.71*	11.68	10.61	15.23	6.58
n33	33	0.3	*14.56*	17.90	*-6.64*	*79.35*	4.75	36.74	12.48	7.85
n40	40	0.2	*10.88*	-9.10	*-20.85*	*90.39*	9.59	59.51	17.66	3.29
n50	50	0.3	*11.38*	*-51.60*	*-21.32*	*93.51*	3.17	15.00	16.40	3.76
Avg			11.85	-9.20	-16.42	69.89				

4.7 Conclusions

This chapter presented a new fuzzy operator named the unified And-Or (UAO) operator. The operator was shown mathematically to be similar to the OWA operator. The proposed UAO operator was able to aggregate the four design objectives discussed earlier. Moreover, a new 7-point preference scheme was also proposed to handle preferences between the objectives and facilitate the decision-making process. The performance of UAO along with the preference scheme was evaluated by their application to a simulated evolution algorithm (discussed in detail in Chapter 6) using some test examples. An empirical comparison of UAO was also done with the OWA operator. Results suggest that the UAO operator performed better than the OWA operator. Moreover, the proposed fuzzy rules and 7-point preference rules scheme allowed the integration of individual objectives into a single objective function with a good degree of precise decision-making.

The next chapter proposes and discusses a fuzzy stochastic evolution algorithm for the multi-objective DLAN topology design problem. The effectiveness of the

Table 4.4: Results for OWA for Ex2. N= number of local sites in the network, B=Bias, C=Cost, D=Delay, H=Hops, R=Reliability, Avg= Average percentage improvement of the five test cases. Statistically significant improvements are in italics.

Case	N	B	Percentage improvement				Standard deviations			
			C	D	H	R	C	D	H	R
n15	15	0.4	5.19	-11.88	-20.12	*17.85*	19.86	24.22	26.59	19.12
n25	25	0.3	*15.26*	-5.75	*-24.74*	*72.61*	8.90	17.05	26.64	9.67
n33	33	0.2	*11.78*	-5.56	*-29.93*	*88.24*	5.91	32.21	22.98	3.72
n40	40	0.0	*11.92*	-40.06	*-30.60*	*88.94*	6.48	65.37	14.62	5.72
n50	50	0.1	*14.54*	-45.78	*-36.56*	*93.56*	6.93	48.88	20.20	2.87
Avg			11.74	-21.81	-28.39	72.24				

proposed algorithm is evaluated through empirical study.