THE FLORIDA STATE UNIVERSITY

COLLEGE OF ARTS AND SCIENCES

THE IMPLEMENTATION OF INTELLIGENT QoS NETWORKING BY THE DEVELOPMENT AND UTILIZATION OF NOVEL CROSS-DISCIPLINARY SOFT COMPUTING THEORIES AND TECHNIQUES

By

AHMED SHAWKY MOUSSA

A Dissertation submitted to the
Department of Computer Science
In partial fulfillment of the
requirements for The degree of
Doctor of Philosophy

Degree Awarded:
Fall Semester, 2003

The members of the committee approve the dissertation of
Ahmed Shawky Moussa
Defended on September 19, 2003.

Ladislav Kohout
Major Professor

Michael Kasha
Outside Committee Member

Lois Wright Hawkes
Committee Member

Ernest McDuffie
Committee Member

Xin Yuan
Committee Member

The Office of Graduate Studies has verified and approved the above named committee members

Optimization is not just a technique, or even a whole science; it is an attitude and a way of life. My mother’s life is a great example of optimization, making the best out of every thing, sometimes out of nothing. She has also been the greatest example of sacrifice, dedication, selfishness, and many other qualities, more than can be listed. I dedicate this work and all my past and future works to the strongest person I have ever known, the one who taught me, by example, what determination is, Fawkeya Ibrahim, my mother.

ACKNOWLEDGMENTS

The preparation for this work started with my birth. Every thing I learned since that day contributed to this work. I certainly have had many teachers, friends, and advisors who helped shaping my knowledge, skills, and attitudes. I value and thank them all, especially my mother who has given me all she has, her life, Mohamed Kamel, my mentor and spiritual father during my growing years, El-Raee Abdo Sheta, whose instructions played a great role in forming my attitudes.

At Florida State University I have had tremendous support and help, without which this work would have never been possible. My thanks are due to the staff of the International Center, the Thagard Student Health Center, and the Center for Music Research, especially Dr. Jack Taylor.

I am deeply thankful to all the faculty, staff, and colleagues who helped me in every way they could. I have taken an extraordinary number of classes in the Department of Computer Science and other departments too. These classes were building blocks in constructing my multidisciplinary scientific background. Certain classes have had special impact on this dissertation and were milestones in my progression: Soft Computing by Dr. Ladislav Kohout, Computational Science and Engineering, and Parallel Programming by Dr. Kyle Gallivan, Algorithm Analysis and Design by Dr. Stephen Leach, and the Seminar on QoS Networking by Dr. Xin Yuan.

My gratitude goes to Dr. Ladislav Kohout, my major professor. I thank him for his faith in me and for guiding me on the right track toward a career in computer science research. I appreciate his open mindedness and vast knowledge, which he always made available for me. I certainly am lucky to have learned Computational Intelligence from one of the leading pioneers in the field.

I am greatly indebted to all my committee members, who gave me their knowledge, support, and resources, even before they were assigned to my PhD committee. Special appreciation is due to Dr. Lois Wright Hawkes who has gone a long way beyond duty to support me in the difficult times, especially in the absence of my major professor, without receiving official credit or recognition.

No words would ever be enough to express my gratitude and appreciation for Professor Michael Kasha. I feel extremely privileged and proud to have known, and learned from, such a world-renowned top-notch scientist. He has been a very generous source of knowledge and support, and a role model to follow. I appreciate his tremendous enthusiasm in teaching, coaching, and helping me, and I hope I can live up to his expectation in my scientific career.

Finally, I would like to express my gratitude and love to my wife Shaheera, whose dedication, love, and assistance made this work possible. She has been the best partner in my journey to the PhD degree.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Soft Computing is the fusion of methodologies that were designed to model and enable solutions to real world problems, which are not modeled, or too difficult to model, mathematically. These problems are typically associated with fuzzy, complex, and dynamical systems, with uncertain parameters. These systems are the ones that model the real world and are of most interest to the modern science.

Among the methodologies of Soft Computing, two seem to, mistakenly, be considered alternatives, namely, Fuzzy Computing and Probabilistic Computing. The fusion of methodologies that characterizes Soft Computing suggests the complementarity, rather than the comparison, of the two systems. This dissertation proposes a model for the integration of the two paradigms to solve problems of fuzzy, complex, dynamical systems in which one field cannot solve alone.

However, the study of Fuzzy Computing and Probabilistic Computing revealed flaws in both systems that may lead to erroneous and misleading conclusions about the results of their applications. On the Fuzzy Computing side, this dissertation addresses the violation of the Law of Excluded Middle by Fuzzy Set Theory as a non-natural feature of the theory and proposes an extension of the theory to fix the deficiency. The dissertation also identifies the possible erroneous computations that may result from applying the crisp techniques of Probability Theory to fuzzy and complex systems. For a solution, the dissertation initiates the idea of Soft Probability, where a model for computing probabilities of fuzzy systems and events is constructed.

Quality of Service Networking is an example of the class of complex, fuzzy, and dynamical systems with uncertain parameters, which Soft Computing is intended to model and compute. The term Quality of Service is a fuzzy term. Its measures are typically fuzzy linguistic hedges. The uncertainty associated with the network state information is inevitable in terms of, both, fuzziness and randomness. Therefore, the integration of Fuzzy and Probabilistic Computing ought to be an ideal approach to the implementation of Quality of Service networks. This dissertation proposes a model for applying the integration of fuzzy and probabilistic techniques for building intelligent adaptive communication systems.

INTRODUCTION

This dissertation documents a hybrid research in the areas of *Soft Computing*, *Quality of Service Networking*, and their integration to develop intelligent adaptive communication systems. Therefore, the main objective, and major contribution, of the research is two-fold: conducting research in soft computing methodologies to improve their computational powers, and use these methodologies as means to provide a new approach for solving the problems of quality of service communications as an example of the class of complex, dynamical, multi-variable, multi-body, uncertain systems, for which, the author believes, soft computing is not only preferred but actually inevitable.

Soft Computing is a multi-disciplinary field. Quality of Service Networking and Distributed Computing are also multi-disciplinary fields. Combining those broad multi-disciplinary areas under one topic in a dissertation is a great writing challenge. The challenge here is not what may be written. Rather, it is how to structure such a dissertation and control the flow of topics in a way that does not lack the broad coverage, but does not drown in endless writing that may go beyond the scope.

To fulfill this purpose, this dissertation was structured conceptually to include four parts, which are physically organized into eight chapters. The first part, Chapters One and Two, covers the introductory background. The second part, Chapter Three, provides a review of the literature. The third part, Chapters Four and Five, includes one of the two major phases of the research, that is, the developments the dissertation contributes to the theories of the soft computing sciences. The fourth part, Chapters Six, Seven, and Eight, is the other major contribution, that is, a model for the implementation of intelligent quality of service networking by using cross-disciplinary integrated soft computing methods and describe solutions at the modeling, network state information maintenance, and routing stages. Finally, the dissertation ends with the conclusion and projected vision and plans for future continuation of the research.

Chapter One is an introduction to Soft Computing providing a brief history about its origins and what it may cover. It was concluded that an exact definition of what Soft Computing may include has not been made yet, since it is still an area newly shaping up. Therefore, an introduction to the basic soft computing as proposed by its inventor, Dr. Lotfi Zadeh, is given.

The four fields that constitute the area of Soft Computing are *Fuzzy Computing*, *Evolutionary Computing*, *Artificial Neural Networks*, and *Probabilistic Computing*. The current research has been focusing on the use of *Fuzzy Systems*. This explains why more coverage of the field of *Fuzzy Logic* and its elements is given in the first chapter.

The chapter starts with an introduction to *Fuzzy Sets* introducing historical background and motivation, the basic properties of Fuzzy Sets, the basic operations on Fuzzy Sets, and Fuzzy Arithmetic. The next section is on *Fuzzy Relations* introducing the properties of Fuzzy Relations, representation of Fuzzy Relations, and the basic operations on Fuzzy Relations. The last section is on *Fuzzy Logic* including its basic concepts and utilization of Fuzzy Sets and Fuzzy Relations, including the *BK-products* of Fuzzy Relations.

The second part of Chapter One introduces Evolutionary Computing. It covers the components of Evolutionary Computing with a historical coverage of how it started. Then, more elaboration on the most common Evolutionary Computing technique, namely, *Genetic Algorithms* is given.

The third part of the chapter is an introduction to the field of Artificial Neural Networks. The nature of the Artificial Neural Networks and their origins in the biological sciences is explained. Then a track down of how they progressed and how they work computationally is covered.

The forth part of Chapter One is on Probabilistic Computing. It is divided into two sections, namely, *Bayesian Belief Networks* and *Dempster-Shafer's Theory*, also known as the *Mathematical Theory of Evidence*.

The last part of this chapter elaborates on the fusion of the four methodologies to form Soft Computing. First, a comparison between the *probability theory* and *fuzzy set theory* is given to clear the confusion between the two theories and show how they can complement each other. Several hybrid soft computing algorithms are introduced, thereafter. The combinations introduced are probabilistic fuzzy systems, fuzzy neural networks, fuzzy genetic algorithms, and artificial neural networks combined with genetic algorithms. Finally, the chapter concludes with new and future directions on combining more than two soft computing fields.

Chapter Two is also an introductory chapter. It covers the area of Quality of Service Distributed Computing. It is divided into two main parts. Part one is a generic introduction to computer networks. It explains the different types of computer networks and their characteristics elaborating on network protocols, performance, and functionality. After that, two sections are given to cover network topologies and routing, respectively.

The second part of Chapter Two narrows down to Quality of Service (QoS) computer networks. It starts with basic concepts and definition, followed by an introduction to the basic QoS performance measures such as bandwidth, packet delay and jitter, and packet loss. Next, the main three levels of QoS are introduced and the chapter concludes with the most important problem in QoS networking, namely, QoS routing.

Chapter Three is the review of the literature of the interaction of Soft Computing and QoS Networking. It is divided into three main sections. The first is a review of the work done to develop intelligent networks using some soft computing techniques. The second section is a review of the work done on the worst obstacle in the way of implementing QoS networking, that is, the uncertainty in the network state information. The last section is a review of the work done on developing adaptive networking techniques with varying service levels and varying loads.

Chapter Four is the beginning of the actual contribution of this research. It introduces the part of the research that focused on Fuzzy Set Theory and points out the observed problem associated with the violation of the *Law of Excluded Middle* and the *Law of Contradiction*. Chapter Four proposes developments to the Fuzzy Set Theory conception and mathematical tools to solve the problem and prove that Fuzzy Set Theory does not have to violate these two laws.

Chapter Five is another development in one of the Soft Computing disciplines, namely, Probabilistic Computing. This chapter proposes and introduces the notion of *Soft Probability*, a proposed framework for the computation of probabilities of real-life systems. The proposed methodology is intended to provide more accurate, meaningful, and computationally feasible probability calculations for fuzzy, complex, and nonlinear systems. The chapter builds on two refereed papers, which has been accepted for publication and presentation in major conferences. The first is accepted at JCIS-2003[1], Cary, NC, USA, under the title: "Questions on Probability Theory and Its Effect on Probabilistic Reasoning" [Moussa 2003], and the second was accepted and selected for presentation at the VIP forum of IPSI-2003[2] in Sveti Stefan, Montenegro under the title: "Soft Probability: Motivation and Foundation" [Moussa 2003].

Chapter Six consists of the first phase of the work on this dissertation. It is basically the paper that was presented in IFSA/NAFIPS-2001[3] in Vancouver, Canada, in July 2001 under the title: "Using BK-Products of Fuzzy Relations In Quality of Service

[1] JCIS, The 7th Joint Conference on Information Sciences.

[2] IPSI, the international conference on Internet, Processing, Systems, and Interdisciplinaries

[3] IFSA is acronym for "International Fuzzy Systems Association". NAFIPS is acronym for "North America Fuzzy Information Processing Society". IFSA/NAFIPS-2001 is the Joint 9th IFSA World Congress and 20th NAFIPS International Conference.

Adaptive Communications" [Moussa and Kohout 2001]. The paper proposes a novel approach for modeling and attacking the QoS networking problems based on the use of fuzzy technology. The models and solutions in the paper are mainly given on the conceptual level and provide the foundation for developing the new paradigm.

Chapter Seven was published and presented at NAFIPS-2002 in New Orleans, LA, USA, under the title: "A New Network State Information Updating Policy Using Fuzzy Tolerance Classes and Fuzzified Precision" [Moussa and Kohout 2002]. This chapter proposes a new dynamic network state maintenance policy that aims at optimizing the network performance by optimally maintaining the balance between the overhead associated with updating the networking information and the degree of accuracy and reliability of the available information.

Finally, Chapter Eight proposes a new QoS routing algorithm based on a model for the integration of Fuzzy and Probabilistic Computing. The algorithm exploits the maintenance policy introduced in Chapter Seven. This chapter was also accepted for publication and presentation in JCIS/FT&T-2003[4] in Cary, NC, USA under the title: "Fuzzy Probabilistic QoS Routing for Uncertain Networks".

Figure (1) describes the general conceptual structure of the dissertation, identifying the order of dependency of the main chapters. Figure (2) is a detailed description of the flow of topics illustrating the division of the chapters into different sections

[4] JCIS is the 7th Joint Conference on Information Sciences. FT&T is the 9th international conference on Fuzzy Theory and Technology.

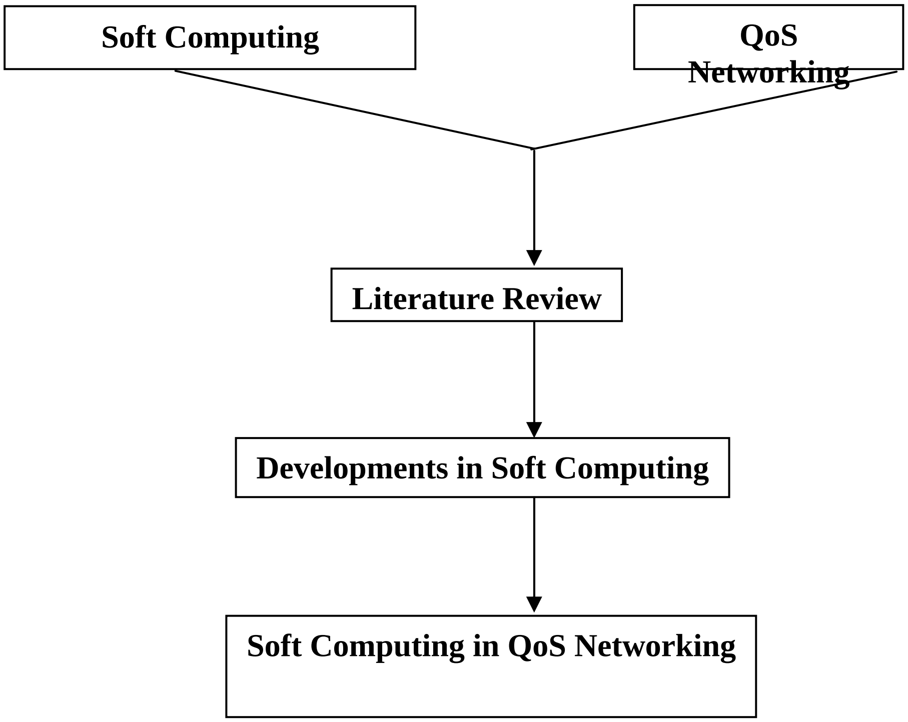

Soft Computing
QoS Networking
Literature Review
Developments in Soft Computing
Soft Computing in QoS Networking

FIGURE (1): The General Conceptual Structure of The Dissertation

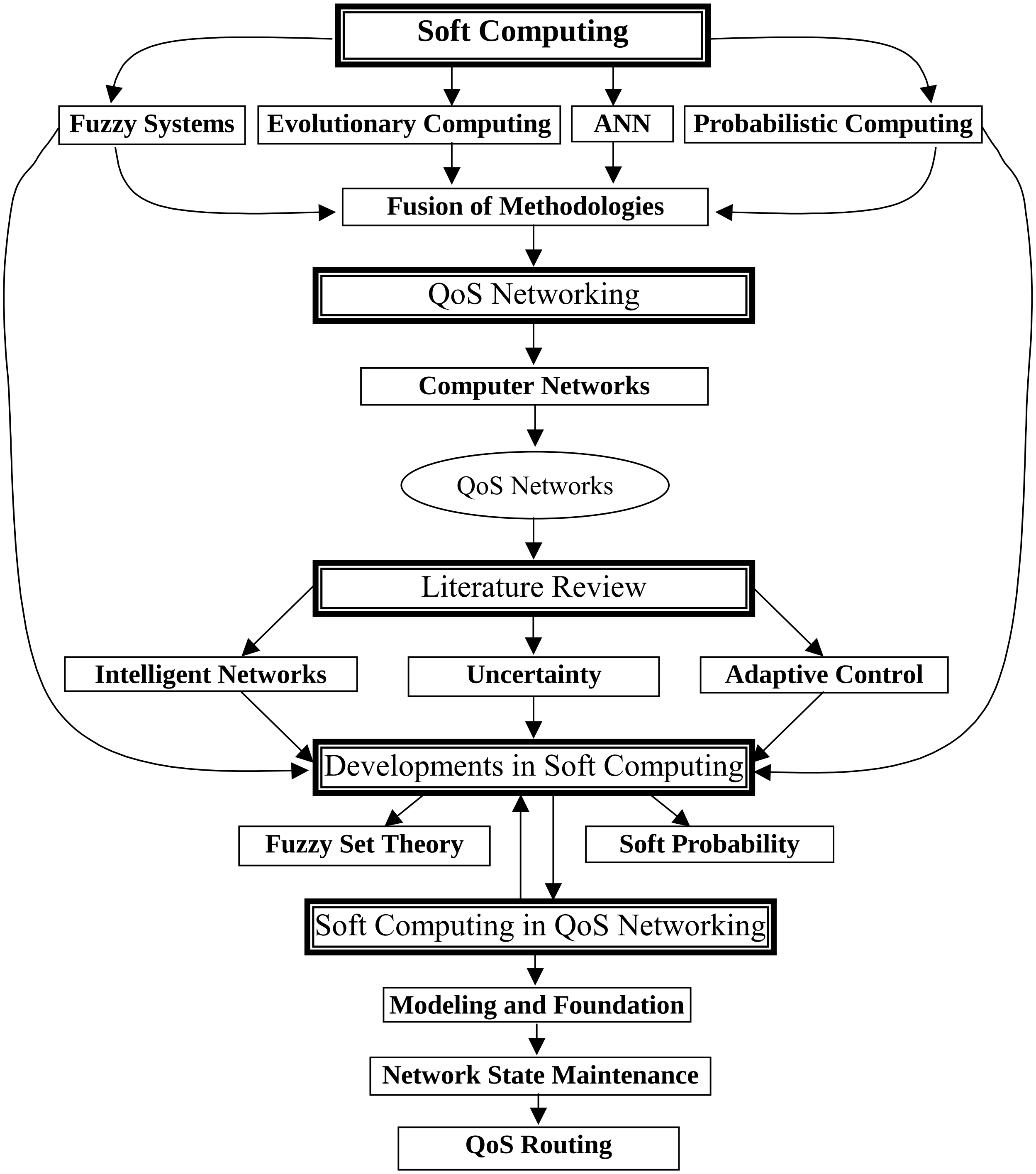


FIGURE (2): The Dissertation Flow-Chart

CHAPTER ONE

SOFT COMPUTING

Soft Computing is a new multidisciplinary field that was proposed by Dr. Lotfi Zadeh, whose goal was to construct new generation Artificial Intelligence, known as Computational Intelligence. The idea of Soft Computing was initiated in 1981 when Dr. Zadeh published his first paper on soft data analysis [Zadeh 1997]. Since then, the concept of Soft Computing has evolved. Dr. Zadeh defined Soft Computing in its latest incarnation as the fusion of the fields of *Fuzzy Logic*, *Neuro-computing*, *Evolutionary and Genetic Computing*, and *Probabilistic Computing* into one multidisciplinary system. The main goal of Soft Computing is to develop intelligent machines and to solve nonlinear and mathematically unmodelled system problems [Zadeh 1993, 1996, and 1999].

The applications of Soft Computing have proved two main advantages. First, it made solving nonlinear problems, in which mathematical models are not available, possible. Second, it introduced the human knowledge such as cognition, recognition, understanding, learning, and others into the fields of computing. This resulted in the possibility of constructing intelligent systems such as autonomous self-tuning systems, and automated designed systems.

Soft Computing is a new science and the fields that comprise Soft Computing are also rather new. Though, a tendency toward the expansion of Soft Computing beyond what Dr. Zadeh initiated has been rapidly progressing. For example, Soft Computing has been given a broader definition in the literature to include *Fuzzy Sets*, *Rough Sets*, *Neural Networks*, *Evolutionary Computing*, *Probabilistic and Evidential Reasoning*, *Multi-valued Logic*, and related fields [Kacpzyk 2001]. Other scientists [Dote *et al.* 2000] proposed the notion of *Extended Soft Computing* (ESC) as a new discipline developed by adding *Chaos Computing* and *Immune Network Theory* to the classical Soft Computing, as defined and proposed by Lotfi Zadeh. ESC was proposed for explaining complex systems and cognitive and reactive AIs. Moreover, Fuzzy Logic, which is the basis on which Soft Computing is built, has been expanded into what is known today as Type-2 Fuzzy Logic [Mendel 2001]. Now on the rise is the new science of *Bios* and *Biotic Systems*. The author of this dissertation proposes, and expects, the inclusion of *Bios Computing* to become one of the pillars of Soft Computing. The author also proposes the replacement of Soft Probability for the traditional probability computing techniques to process soft systems' computations.

From the above presentation of the subject, it is obvious that Soft Computing is still growing and developing. Hence, a clear definite agreement on what comprises Soft Computing has not yet been reached. Different views of what it should include have been proposed and more new sciences are still merging into Soft Computing. Therefore, only the four main components of Soft Computing as proposed by the founder, Dr. Zadeh, are considered in this introductory chapter. Furthermore, only Type-1 Fuzzy Logic, the original Fuzzy Logic, is presented.

Fuzzy Computing

Fuzzy Sets

Fuzzy Logic is built on The *Fuzzy Set Theory*, which was introduced to the world, for the first time, by Lotfi Zadeh in 1965 [Zadeh 1965]. The invention, or proposition, of Fuzzy Sets was motivated by the need to capture and represent the real world with its fuzzy data due to uncertainty. Uncertainty can be caused by imprecision in measurement due to imprecision of tools or other factors. Uncertainty can also be caused by vagueness in the language. We use linguistic variables often to describe, and maybe classify, physical objects and situations. Lotfi Zadeh realized that the Crisp Set Theory is not capable of representing those descriptions and classifications in many cases. In fact, Crisp Sets do not provide adequate representation for most cases.

A very classical example is what is known as the Heap Paradox [Klir, St. Clair, and Yuan 1997]. If we remove one element from a heap of grains or sand, we will still have a heap. However, if we keep removing single elements, one at a time, there will be a time when we do not have a heap anymore. At what time does the heap turn into a countable collection of grains that do not form a heap? There is no one correct answer to this question. This example represents a situation where vagueness and uncertainty are inevitable.

Throughout the history, until the end of the nineteenth century, uncertainty, whether due to vagueness or imprecision, were always considered undesirable in science and philosophy. Hence, the way to deal with uncertainty was to either ignore it and assume its non-existence, or to try to eliminate it. Though, obviously any investigation that involves a concept such as the heap will have to deal with vagueness. Moreover, the ever existing imprecision due to the physical limitations of measurement tools would disqualify any investigation that assumes zero uncertainty. Uncertainty in the macroscopic world is always viewed as lack of knowledge. Nevertheless, both excess knowledge and uncertainty lead to increased complexity. Let us take for example the computational model of a car driver. Using a manual transmission rather than an automatic one requires more knowledge to drive, which increases the complexity involved. However, the uncertainty caused by not knowing the road or some bad weather also increases the complexity of the computational model of the driver. Uncertainty and complexity result in the failure of Crisp Sets to represent many concepts, notions, and situations.

Crisp Sets can be ideal for certain applications. For example, we can use crisp sets to represent the classification of coins. We can list US coins and put a boundary on the set that encloses them so that other coins like French francs or English pounds are definitely out of the set and the US coins are definitely in the set, as illustrated in Figure (1-1).

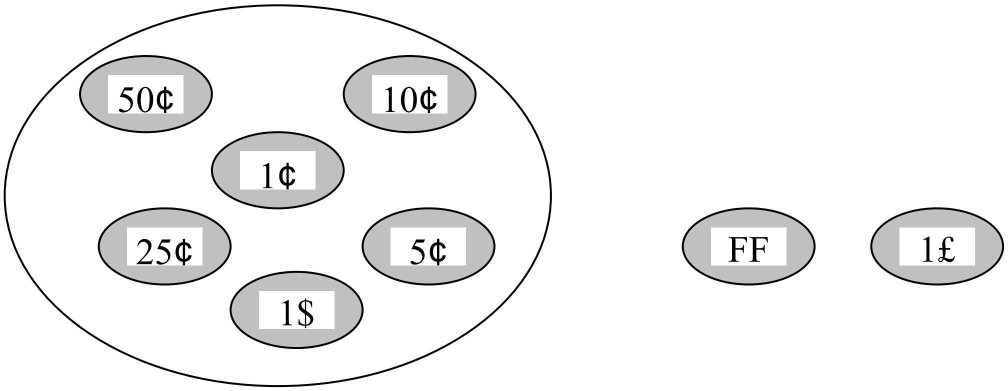


FIGURE (1-1): Example of Crisp Sets [Klir et al. 1997]

However, we cannot always define precise boundaries to describe sets in real life. In fact we often cannot do that. For instance, when we try to classify students in a school into tall students that qualify for a basketball team and short students who do not, if we consider students who are six feet and four inches tall to be qualified, should we then exclude a student who is one-tenth of an inch less than the specified height? Should we even exclude a student who is a whole inch shorter?

Instead of avoiding or ignoring uncertainty, Lotfi Zadeh developed a set theory that captures this uncertainty. The goal was to develop a set theory and a resulting logic system that are capable of coping with the real world. Therefore, rather than defining Crisp Sets, where elements are either in or out of the set with absolute certainty, Zadeh proposed the concept of a *Membership Function*. An element can be in the set with a degree of membership and out of the set with a degree of membership. Hence, Crisp Sets are a special case, or a subset, of Fuzzy Sets, where elements are allowed a membership degree of 100% or 0% but nothing in between. Figure (1-2) illustrates the use of Fuzzy Sets to represent the notion of a *tall* person. It also shows how we can differentiate between the notions of *tall* and *very tall*, resulting in a more accurate model than the classical set theory.

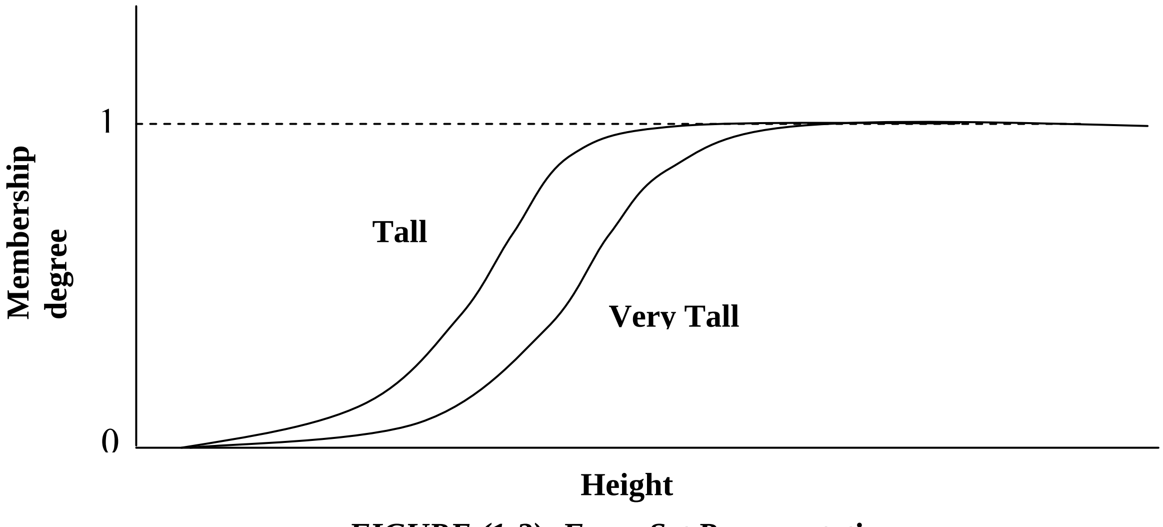


FIGURE (1-2): Fuzzy Set Representation

Basic properties of Fuzzy Sets: Fuzzy Sets are characterized by Membership Functions. The membership function assigns to each element *x* in a fuzzy set a number, *A(x)*, in the closed unit interval [0, 1]. The number, *A(x)*, represents the degree of membership of *x* in *A*. Hence, membership functions are functions of the form:

$$A: X \rightarrow [0, 1]$$

In the case of Crisp Sets, the members of a set are either out of the set, with membership degree of zero, or in the set, with the value one being the degree of membership. Therefore, Crisp Sets $\subseteq$ Fuzzy Sets or in other words, Crisp Sets are Special cases of Fuzzy Sets.

There are four ways of representing fuzzy membership functions, namely, *graphical representation*, *tabular and list representation*, *geometric representation*, and *analytic representation*. Graphical representation is the most common in the literature. Figure (1-2) above is an example of the graphical representation of fuzzy membership functions. Tabular and list representations are used for finite sets. In this type of representation, each element of the set is paired with its degree of membership. Two different notations have been used in the literature for tabular and list representation. The following example illustrates the two notations for the same membership function.

$A = \{ <x_1, 0.8>, <x_2, 0.3>, <x_3, 0.5>, <x_4, 0.9> \}$

$A = 0.8/x_1 + 0.3/x_2 + 0.5/x_3 + 0.9/x_4$

The third mothod of representation is the geometric representation and is also used for representing finite sets. For a set that contains *n* elements, *n*-dimentional Euclidean space is formed and each element may be represented as a coordinate in that space. Finally analytical representation is another alternative to graphical representation in representing infinite sets, e.g., a set of real numbers. The following example illustrates both graphical and analytical representation of the same fuzzy function:

$$A(x) = \begin{cases} x-5 & when\ 5 \le x \le 6 \\ 7-x & when\ 6 \le x \le 7 \\ 0 & otherwise \end{cases}$$

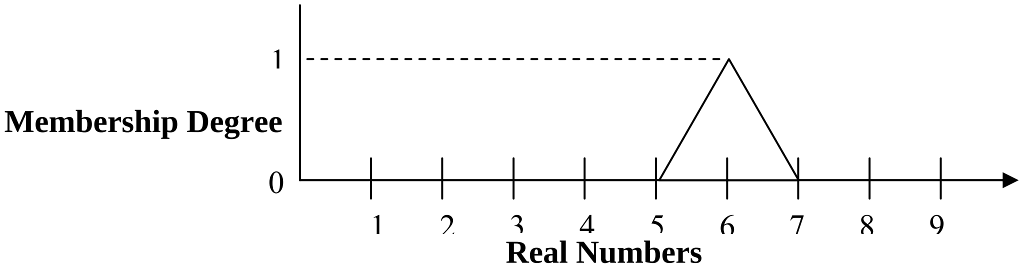


FIGURE (1-3): Graphical Representation of the Analytical Representation Given Above

The above example also illustrates the important notion of a *fuzzy number*. A Fuzzy Number is a fuzzy set represented by a membership function of the form
A: ℝ → [0, 1]
With the additional restriction that the membership function must capture an intiuitive conception of a set of real numbers surrounding a given central real number, or interval of real numbers. In this context, the example above illustrates the concept of the fuzzy number “about six”, “around six”, or “approximately six”.

Another very important property of fuzzy sets is the concept of *α–cut* (alpha cut). ∝-cuts reduce a fuzzy set into an extracted crisp set. The value α represents a membership degree, i.e., α ∊ [0, 1]. The α-cut of a fuzzy set *A* is the crisp set (*A* – α), i.e., the set of all elements whose membership degrees in *A* are ≥ α [Kohout 1999].

Basic Operations on Fuzzy Sets: The basic operations on fuzzy sets are Fuzzy Complement, Fuzzy Union, and Fuzzy Intersection. These operations are defined as follows:

Fuzzy Complement

$$\overline{A}(x) = 1 - A(x)$$

Fuzzy Union

$$(A \cup B)(x) = \max\,[A(x), B(x)] \qquad \textit{for all } x \in X$$

Example:

$A(x) = 0.6$ and $B(x) = 0.4$

$\therefore (A \cup B)(x) = \max\,[0.6, 0.4] = 0.6$

Notice that $A \cup \overline{A} \neq X$, which violates the *Law of Excluded Middle*.

Fuzzy Intersection:

$$(A \cap B)(x) = \min\,[A(x), B(x)] \qquad \textit{for all } x \in X$$

Example:

$A(x) = 0.6$ and $B(x) = 0.4$

$\therefore (A \cap B)(x) = \min\,[0.6, 0.4] = 0.4$

Also notice that $A \cap \overline{A} \neq \phi$, which violates the *Law of Contradiction*.

The above operations are the Standard operations on Fuzzy Sets. Different definitions have been developed for the fuzzy union, intersection, and complement, but a detailed listing or study of the different definitions is beyond the scope of this dissertation.

Fuzzy Arithmetic. Fuzzy Arithmetic uses arithmetic on closed intervals. The basic fuzzy arithmetic operations are defined as follows:

Addition:

$$[\alpha, b] + [c, d] = [\alpha + c, b + d]$$

Subtraction:

$$[\alpha, b] - [c, d] = [\alpha - d, b - c]$$

Multiplication:

$$[\alpha,b].[c,d] = [\min(\alpha c, \alpha d, bc, bd), \max(\alpha c, \alpha d, bc, bd)]$$

Division:

$$[\alpha, b] / [c, d] = [\alpha, b] . [1/d, 1/c]$$

$$=[\min(\alpha/c,\ \alpha/d,\ b/c,\ b/d),\max(\alpha/c,\ \alpha/d,\ b/c,\ b/d)$$

Figure (1-4) illustrates graphical representations of fuzzy addition and subtraction.

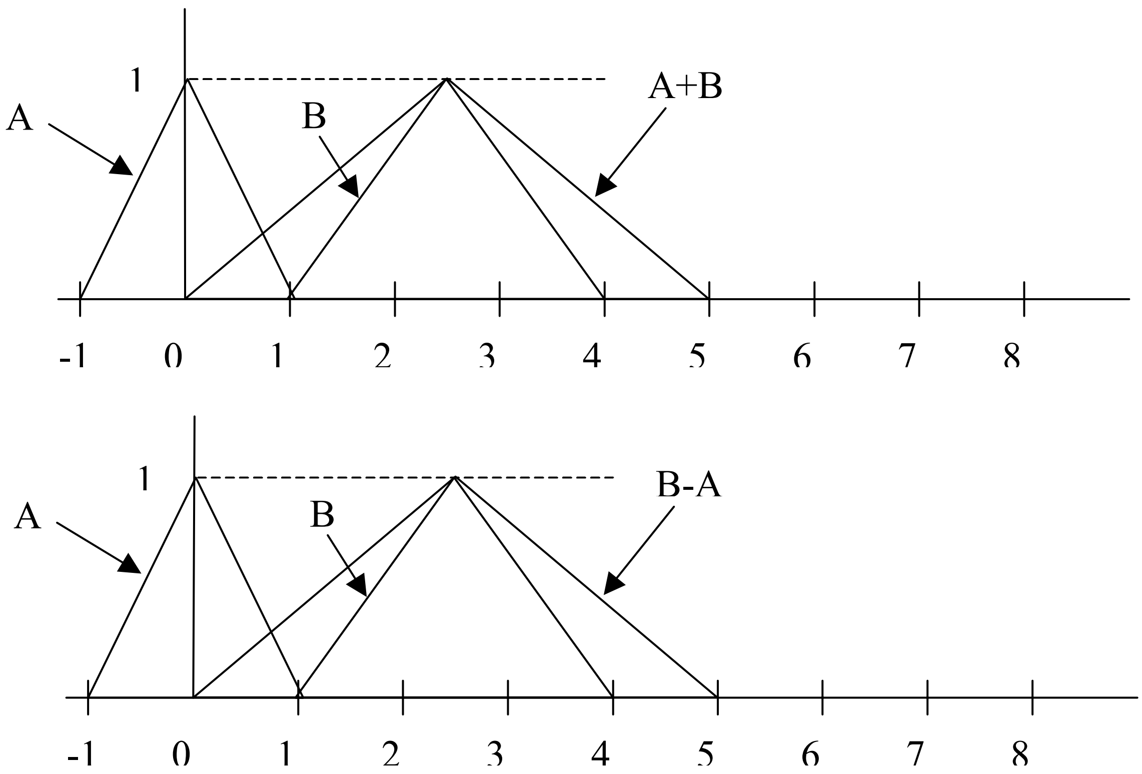


FIGURE (1-4): Example fuzzy addition and subtraction

Fuzzy Relations

Properties of Fuzzy Relations: Fuzzy Relations were introduced to supersede classical crisp relations. Rather than just describing the full presence or full absence of association of elements of various sets in the case of crisp relations, Fuzzy Relations describe the degree of such association. This gives fuzzy relations the capability to capture the uncertainty and vagueness in relations between sets and elements of a set. Furthermore, it enables fuzzy relations to capture the broader concepts expressed in fuzzy linguistic terms when describing the relation between two or more sets. For example, when classical sets are used to describe the equality relation, it can only describe the concept "*x* is equal to *y*" with absolute certainty, i.e., if *x* is equal to *y* with unlimited precision, then *x* is related to *y*, otherwise *x* is not related to *y*, even if it was slightly different. Thus, it is not possible to describe the concept "*x* is approximately equal to *y*". Fuzzy Relations make the description of such a concept possible. Table (1-1) provides comparison of the special properties of Crisp and Fuzzy relations (E_x is the Equality Relation and O_x is the Empty Relation) [Bandler and Kohout 1988].

TABLE (1 – 1): Properties of Crisp vs. Fuzzy Relations

Property	Crisp	Fuzzy
Covering	$\Leftrightarrow \forall i \in J, \exists j \in J \mid R_{ij} = 1$	$\Leftrightarrow \forall I \in J, \exists j \in J \mid R_{ij} = 1$
Locally reflexive	$\Leftrightarrow \forall i \in J, R_{ii} = \bigvee_j (R_{ij} \vee R_{ji})$	$\Leftrightarrow \forall I \in J, R_{ii} = \bigvee_j (R_{ij} \vee R_{ji})$
Reflexive	$\Leftrightarrow$ Covering and locally reflexive	$\Leftrightarrow$ Covering and locally reflexive
Transitive	$\Leftrightarrow R^2 \subseteq R$	$\Leftrightarrow R^2 \subseteq R$
Symmetric	$\Leftrightarrow R^T \subseteq R$	$\Leftrightarrow R^T \subseteq R$
Antisymmetric	$\Leftrightarrow R \cap R^T \subseteq E_x$	$\Leftrightarrow R_{ij} \wedge R_{ji} = 0$ if $j \neq I$
Strictly Antisymmetric	$\Leftrightarrow R \cap R^T = O_x$	$\Leftrightarrow \forall I, j \in J, R_{ij} \wedge R_{ji} = 0$

It is important to note here that the concept of local reflexivity was introduced for the first time in Crisp Relational Theory by Bandler and Kohout in 1977. The fast fuzzy relational algorithms that employ local reflexivity in fuzzy computing were introduced also by Bandler and Kohout in 1982 [Bandler and Kohout 1987], [Kohout 2001].

Representation of Fuzzy Relations: The most common methods of representing fuzzy relations are lists of *n*-tuples, formulas, matrices, mappings, and directed graphs. A list of *n*-tuples, i.e., ordered pairs, can be used to represent finite fuzzy relations. The tuple consists of a Cartesian product with its membership degree. When the membership has degree zero, the tuple is usually omitted. Suitable formulas are usually used to define infinite fuzzy relations, which involve *n*-dimensional Euclidean space, with $n \geq 2$. Matrices, or *n*-dimensional arrays, are the most common method to represent fuzzy relations. In this method, the entries of the matrix are the membership degrees associated with the *n*-tuple of the Cartesian product. The mapping of fuzzy relations is an extension of the mapping method of classical binary relations. For fuzzy relations, the connections of the mapping diagram are labeled with the membership degree. The same technique is used to extend the directed graph representation of classical relations to represent fuzzy relations.

Operations on Fuzzy Relations: All the mathematical and logical operations on fuzzy sets explained above are also applicable to fuzzy relations. In addition, there are operations on fuzzy binary relations that do not apply to general fuzzy sets. Those operations are the inverse, the composition, and the BK-products of fuzzy relations.

The inverse of a fuzzy binary relation *R* on two sets X and Y is also a relation denoted by R^{-1} such that $xR^{-1}y = yRx$. Therefore, for any fuzzy binary relation, $(R^{-1})^{-1} = R$. When using matrix representation, the inverse can be obtained by generating the transpose of the original matrix, i.e., swapping the columns and the rows of the matrix as in the following example.

$$R = \begin{bmatrix} 0.5 & 1 & 0 \\ 1 & 0.8 & 0.2 \\ 0.7 & 0 & 0.3 \end{bmatrix} \qquad R^{-1} = \begin{bmatrix} 0.5 & 1 & 0.7 \\ 1 & 0.8 & 0 \\ 0 & 0.2 & 0.3 \end{bmatrix}$$

The composition of two fuzzy relations is defined as follows:

Let *P* be a fuzzy relation from X to Y and *Q* be a fuzzy relation from Y to Z such that the membership degree is defined by *P*(x, y) and *Q*(y, z). Then, a third fuzzy relation *R* from X to Z can be produced by the composition of *P* and *Q*, which is denoted as *P*o*Q*. Fuzzy relation *R* is computed by the formula [Klir and Yuan 1995]:

$$R(\mathrm{x}, \mathrm{z}) = (P\mathrm{o}Q)\ (\mathrm{x}, \mathrm{z}) = \max_{y \in Y}\{\min[P(x,y), Q(y,z)]\}$$

The idea of producing fuzzy relational composition was expanded by Bandler and Kohout in 1977 when they introduced, for the first time, special relational compositions called the Triangle and Square products [Bandler and Kohout 1977, 80, 87]. The Triangle and Square products were named after their inventors and became known as BK-products. Since Bandler and Kohout introduced three new types of products, namely, Triangle sub-product, Triangle super-product, and Square product, a name was needed for the original composition. Therefore, it was called the Circle product. The four different types of fuzzy relational products are defined as follows [Kohout 2000]:

Circle product	x(RoS)z ⇔ xR intersects Sz
Triangle sub-product	x(R ◁ S)z ⇔ xR ⊆ Sz
Triangle super-product	x(R ▷ S)z xR ⊇ Sz
Square product	x(R□S)z ⇔ xR ≅ Sz

Fuzzy Logic

According to the Fuzzy Set Theory, a statement about the status of an element in a set may not be true or false with unlimited certainty. For example, the proposition "*x is in X*" may be 80% true and 20% false. Consequently, any reasoning based on such a proposition would have to be approximate, rather than exact. Such a reasoning system is the goal and the basis of Fuzzy Logic.

The attempts of philosophers and logicians to go beyond the classical two-valued logic, where propositions are either definitely true or definitely false, have been motivated by the need to represent and conduct reasoning on reality. Those attempts to expand the two-valued logic into a more realistic and flexible logic, where propositions may be partly true and partly false, have started very early in history. Aristotle presented the problem of having propositions that may not be true or false in his work *On Interpretation* [Klir and Yuan 1995]. He argued that propositions on future events would not have any *Truth-values*. The way to resolve their truth-values is to wait until the future becomes present. However, the indeterminate truth-value of many propositions may not be easily resolved.

With advent of the twentieth century, we realized that propositions about future events are not the only propositions with problematic truth-values. In fact, the truth-values of many propositions may be inherently indeterminate due to uncertainty. This uncertainty may be due to measurement limitations such as the one that resulted in the well-known *Heisenberg Uncertainty Principle*. It can also be caused by the intrinsic vagueness of the linguistic hedges of natural languages when used in logical reasoning.

Multi-valued logics have been invented to enable capturing the uncertainty in truth-values. Logicians such as Lukasiewicz, Bochvar, and Kleene devised three-valued logics, which relaxed the restrictions on the truth and falsity of propositions. In a three-valued logic, a proposition may have a truth-value of half, in addition to the classically known zero and one. This resulted in the expansion of the concept of a *tautology* to the concept of a *quasi-tautology* and *contradiction* into *quasi-contradiction*. This was necessary since under the truth-value of one half we cannot have a truth-value of one or zero on all the rows of a truth table in many three-valued logic systems.

The three-valued logic concept was generalized in the first half of the twentieth century into the *n*-valued logic concept, where *n* can be any number. Hence, in an *n*-valued logic, the degree of truth of a proposition can be any one of *n* possible numbers in the interval [0, 1]. Lukasiewicz was the first one to propose a series of *n*-valued logics for which $n \geq 2$. In the literature, the *n*-valued logic of Lukasiewicz is usually denoted as L_n, where $2 \leq n \leq \infty$. Hence, L_∞ is the infinite valued logic, which is obviously isomorphic to Fuzzy Set Theory as L_2 is the two-valued logic, which is isomorphic to the Crisp Set Theory [Klir and Yuan 1995].

In a sense, Fuzzy Logic can be considered to be a generalization of a logic system that includes the class of all logic systems with truth-values in the interval [0, 1]. "*In a broader sense, fuzzy logic is viewed as a system of concepts, principles, and methods for dealing with modes of reasoning that are approximate rather than exact.*" [Klir, St. Clair, and Yuan 1997].

The inference rules of classical logic are certainly not suitable for approximate reasoning. However, those inference rules can be generalized to produce an inferential mechanism that is adequate for approximate reasoning and is based on Fuzzy Logic. The generalized *modus ponens*, introduced by Lotfi Zadeh in 1979, is the basic inference mechanism in fuzzy reasoning [Bonissone 1997].

The classical *modus ponens* can be expressed as the conditional tautology:

$$[(p \Rightarrow q) \wedge p] \Rightarrow q$$

Alternatively, *modus ponens* can be represented by the schema:

Rule: $p \Rightarrow q$

Fact: p

Conclusion: q

The generalized *modus ponens* is formally represented as follows. Let χ and γ be variables taking values from sets X and Y, A and A′ are fuzzy sets on X , and B and B′ are fuzzy sets on Y.

Rule: χ is A ⇒ γ is B

Fact: χ is A′

Conclusion: γ is B′

If A′ is given, then B′ can be computed by the following equation:

$$B'(y) = \sup_{x \in X} \{\min[A'(x), R(x,y)]\}$$

where *R* is a fuzzy relation on $X \times Y$, and sup is defined to be *supremum* (the minimum upper bound).

The generalized *modus ponens* provided the beginning of the development of fuzzy inference rules. It was followed by the formation of the generalized *modus tollens* and the generalized *hypothetical syllogism*, which together with the generalized *modus ponens* served as the basis for the fuzzy logic based approximate reasoning [Klir and Yuan 1995].

The generalized *modus ponens* is the basis for interpreting fuzzy rule sets. The most common definition of a fuzzy rule base *R* was proposed by Mamdani and Assillian in 1975 and is represented by the formula:

$$R = \bigcup_{i=1}^{m} r_i = \bigcup_{i=1}^{m} (\overrightarrow{X} \to Y_i)$$

According to the definition, *R* is composed of a disjunction of a number *m* of fuzzy rules. Each rule r_i associates a fuzzy state vector $\vec{X}$ with the corresponding fuzzy action Y_i [Mamdani and Assillian 1975].

Using the generalized *modus ponens*, an inference engine of a Fuzzy Controller can be built. The output of the resulting fuzzy system can be described by the formula:

$$\mu_Y(y) = \bigvee_{\mathrm{i}}^{\mathrm{m}} \{\min[\lambda_i, \mu_{yi}(y)]\}$$

where λ_i is the degree of applicability of rule r_i, which is determined by the matching of an input vector $\vec{I}$ with the *n*-dimensional state vector $\vec{X}$. The degree of applicability, λ_i, is determined from the resulting degree of matching. Therefore, λ_i can be computed by the formula:

$$\lambda_i = \bigwedge_j^n \prod(X_{i,j}, I_j)$$

In this formula, $\prod(X_{i,j}, I_j)$ is the possibility measure representing the matching between the reference state variable and the input and can be computed as follows: [Bonissone 1997]

$$\prod(X_{i,j}, I_j) = \bigvee_{x_j}(\min[\mu_{x_{i,j}}(x_j), \mu_j(x_j)])$$

When utilizing the above inference mechanism for Fuzzy Control, an actuator is expected to be triggered eventually to perform some function. The action to be taken should be based on a single scalar value. Therefore, a defuzzification mechanism is needed to convert the fuzzy membership distribution into the required scalar value. A variety of defuzzification techniques exist. The selection of one defuzzification technique is application dependent and involves some trade off between elements of computational costs such as storage, performance, and time.

The BK-products of fuzzy relations proved to be very powerful not only as a mathematical tool for operations on fuzzy sets and fuzzy relations but also as a computational framework for fuzzy logic and fuzzy control. In addition to the set based definitions presented above, many valued logic operations are also implied and are defined as follows:

Circle product	$(R \circ S)_{ik} = \vee_j(R_{ij} \wedge S_{ik})$
Triangle sub-product	$(R \triangleleft S)_{ik} = \wedge_j(R \rightarrow S_{ik})$
Triangle super-product	$(R \triangleright S)_{ik} = \wedge_j(R \leftarrow S_{ik})$
Square product	$(R \square S)_{ik} = \wedge_j(R \equiv S_{ik})$

Where R_{ij} and S_{ik} represent the fuzzy degree of truth of the propositions x_iRy_j and y_jSz_k, respectively [Kohout 2000].

BK-products have been applied, as a powerful computational tool, in many fields such as computer protection, AI [Kohout 1990], medicine, information retrieval, handwriting classification, urban studies, investment, control, [Kohout *et al.* 1992], [Kohout 2000] and most recently in quality of service and distributed networking [Moussa and Kohout 2001].

Fuzzy Control

Fuzzy Control is considered to be the most successful area of application of Fuzzy Set Theory and Fuzzy Logic. Fuzzy controllers revolutionized the field of control engineering by their ability to perform process control by the utilization of human knowledge, thus enabling solutions to control problems for which mathematical models may not exist, or may be too difficult or computationally too expensive to construct.

A typical Fuzzy controller consists of four modules: the rule base, the inference engine, the fuzzification, and the defuzzification. A typical Fuzzy Control algorithm would proceed as follows:

1- Obtaining information: Collect measurements of all relevant variables.

2- Fuzzification: Convert the obtained measurements into appropriate fuzzy sets to capture the uncertainties in the measurements.

3- Running the Inference Engine: Use the fuzzified measurements to evaluate the control rules in the rule base and select the set of possible actions.

4- Defuzzification: Convert the set of possible actions into a single numerical value.

5- The Loop: Go to step one.

Several defuzzification techniques have been devised. The most common defuzzification methods are: the center of gravity, the center of maxima, the mean of maxima, and the root sum square.

Evolutionary Computing

Evolutionary Computing is a class of global search paradigms. It includes paradigms such as *Evolutionary Strategies* (ESs), *Evolutionary Programs* (EPs), *Genetic Algorithms* (GAs), and *Genetic Programming* (GP) [Bonissone 1997]. ES is concerned with continuous function optimization [Rechenberg 1965], [Schwefel 1965]. EP is a paradigm for generating strategies and behaviors by means of generating *finite state automata* [Fogel 1962]. GA was proposed by Holland [Holland 1975] and was inspired by processes observed in biological evolutionary systems [Klir 1995]. GP is a technique for generating approximate solutions to problems such as predicting time series by means of evolving computer programs [Koza 1992].

Once again, as a still new and developing science, an exact definition of what comprises Evolutionary Computing has not yet been set. For example, some references consider ES, EP, and GAs only to be the essentials of Evolutionary Computing [Mitchell 1999]. Moreover, in this context it is important to note that those fields are not always easily separable. There is a great deal of overlapping and similarities between the different paradigms of Evolutionary Computing that were noted by Fogel and others [Fogel 1995]. However, GAs are the most common in this class of paradigms. GAs are also the most widely integrated in hybrid systems with other fields, such as Fuzzy Systems and Artificial Neural Networks [Bonissone 1997].

As the name suggests, GAs represent a new programming paradigm that tries to mimic the process of natural evolution to solve computing and optimization problems. In a GA, a population of computer chromosomes, which are usually strings of bits, is randomly selected. This population is transformed into a new population by a sort of natural selection based on the use of operators inspired by the natural genetic operators. The three operators defined by Holland are the crossover, the mutation, and the inversion operators.

The natural selection is based on the output of a function called the Fitness Function. Only the fit chromosomes survive and are allowed to reproduce offsprings. Among those surviving chromosomes, the fitter chromosome reproduces more offsprings than the less fit ones. The crossover operator selects a feature, e.g., a bit location, from the two parents of an offspring and performs the crossover on the subsequences of the string before and after the chosen location. The mutation operator flips some of the bits in a chromosome. The inversion operator reverses the order of a subsequence of a chromosome.

When the generation of a new population is completed, stopping criteria are evaluated. If the stopping criteria were met, the algorithm stops. Otherwise, the fitness function is used again to obtain the fitness degree of the new population.

Variations on the above basic GA have been devised and a debate about what remains in the realm of GAs and what constitutes a different paradigm of Evolutionary Computation was raised. The areas of application of those GAs, and their variations, have been diverse. The most successful applications were in the areas of Optimization, Automatic Programming, Machine Learning, Economics, Immune Systems, Ecology, Population Genetics, Evolution and Learning, and Social Systems [Mitchell 1999].

Some variations on GAs aim at exploiting the points of strength, minimizing the shortcomings of GAs. Due to the global search nature, GAs are known to be robust, being trapped in local minima. However, GAs are also known to be inaccurate and inefficient in finding the global minimum. In 1994, Renders and Bersini proposed one of the variations on GAs, mentioned above, in the form of a hybrid GA based on the integration of GAs with Hill Climbing (HC) techniques. In this algorithm, they do not follow the traditional way of solution selection based on the instantaneous evaluation of the fitness function.

Instead, the fitness function is applied to a refined solution after applying HC techniques [Renders and Bersini 1994]. In the same paper, Renders and Bersini proposed another hybrid GA that depends on embedding optimization techniques in the crossover operator. Finally, they combined the two hybrid methods and showed that the combined hybrid GA outperformed each of the two hybrid GAs that it combines.

Artificial Neural Networks

Artificial Neural Networks (ANN) is another computing paradigm that originated in the biological world. Neural Computation does not have to be the computation carried out by nerve cells. An artificial system can emulate a simplified version of a neural computational system. ANN is an example of such an artificial neural system [Bossomajer and David 2000]. Even though the name ANN has been the most common but other names have been used synonymously as well. Examples of these names are Neural Computing, Connectionism, Parallel Distributed Processing, and Connection Science [Alexander and Morton 1990].

The multidisciplinary nature of the field of neural networks and its origin in biological science makes it difficult to state a rigorous definition for the field and what it addresses. This is the same problem with Evolutional and Genetic Computing. However, few references have attempted such a definition. A definition given by Igor Aleksander and Helen Morton is given as follows. *"Neural computing is the study of networks of* *adaptable nodes which, through a process of learning from task examples, store experiential knowledge and make it available for use"* [Aleksander and Morton 1990].

ANNs have often been used as an alternative to the techniques of standard nonlinear regression and cluster analysis to carry out statistical analysis and data modeling [Cheng and Titterington 1994]. In addition, computer scientists and engineers have seen ANNs, as providing a new experimental paradigm for Parallel Distributed Processing, rather than the algorithmic paradigm that dominated the field of machine intelligence prior to the ANN revolution [Gurney 1999].

Although scientists from various fields worked on the study of understanding and modeling of neuro-sciences, ANNs were actually realized in the 1940s. Warren McCulloch and Walter Pitts designed the first ANNs [McCulloch and Pitts 1943]. The first learning rule for ANNs was designed by Donald Hebb in McGill University [Hebb 1949]. In the 1950s and 1960s, ANNs entered their first flowering era. The most remarkable implementations of that era were the development of the *Perceptrons* and the *ADALINE* algorithm. After that, there was a rather quiet period in the 1970s, regardless of the works of Kohonen, Anderson, Grossberg, and Carpenter. The 1980s witnessed the second revival of ANNs. *Back-Propagation*, *Hopfield Nets*, *Neocognitron*, and *Boltzmann Machine* were the most remarkable developments of that era [Fausett 1994].

An ANN is a computational structure designed to mimic biological neural networks. The ANN consists of computational units called *neurons*, which are connected by means of weighted interconnections. The weight of an interconnection is a number that expresses the strength of the associated interconnection.

The main characteristic of ANNs is their ability to learn. The learning process is achieved by adjusting the weights of the interconnections according to some applied learning algorithms. Therefore, the basic attributes of ANNs can be classified into Architectural attributes and Neurodynamic attributes [Kartalopoulos 1996]. The architectural attributes define the network structure, i.e., number and topology of neurons and their interconnectivity. The neurodynamc attributes define the functionality of the ANN.

ANNs were developed in the 1960s after a series of developments, proposals, and implementations [Bonissone 1997]. The most remarkable foundational achievements are the work on Spontaneous Learning by Rosenbaltt in 1959 [Rosenbaltt 1959], Competitive Learning by Stark, Okajima, and Whipple in 1962 [Stark *et al.* 1962], and ADALINE/MADALINE algorithms by Widrow and Hoff in 1960 [Widrow and Hoff 1960], [Widrow 1990]. However, it is important to note that modeling a neuron mathematically has been a research problem for over a hundred years [Kartalopoulos 1996].

Even though an ANN is supposed to mimic the biological neural network, the structure of a neuron in an ANN is completely different from the structure of the neuron in a biological network. A basic artificial neuron consists of n inputs, numbered X_j, $1 \leq j \leq n$. Each X_j is weighted, i.e. multiplied by W_j, the connection strength of the relative connection. Other components of the neuron are the bias term W_0, a threshold value Θ, a nonlinearity function F, and two output signals R and O. the bias term W_0 acts as another input, e.g., X_0, whose weight is always one. We can increase the net input to a node by increasing the bias term X_0 [Fausett 1994]. The threshold value Θ, is used for setting the firing condition, i.e., a neuron can produce an output signal R only when the following condition is met:

$$\sum_{j=1}^{n} w_j x_j \geq \Theta$$

The first output R is the initial output of the neuron. The nonlinearity function F is a function applied to the initial output R to ensure a controlled neuron's response. The final output O is the output after applying the function F to the output R.

$$O = F(\sum_{j=1}^{n} w_j x_j)$$

After output O is produced, it becomes an input to another neuron.

The ability of ANNs to adapt to input changes until the output O reaches a desired value is what makes ANNs so powerful. The adaptation is accomplished by continuously adjusting the network parameters, called Synaptic Weights, in response to input stimuli until the output response converges to the desired output. This adaptation process is known in ANNs as the learning process, i.e., when the actual output response matches the desired one, the ANN is said to have completed the learning process.

Probabilistic Computing

The origin of Probabilistic Reasoning dates back to the eighteenth century. One of the two major paradigms of probabilistic reasoning is called *Bayesian Belief Networks*. This paradigm is based on the work of Thomas Bayes [Bayes 1763]. The other is called *Dempster-Shafer's Theory* (DST) of Belief, also known as the *Mathematical Theory of Evidence* (MTE), which was developed by Dempster and Shafer [Dempster 1967], [Shafer 1976].

Bayesian Belief Networks

As mentioned above, Bayesian Belief Networks are built on the eighteenth century work of Thomas Bayes. However, the first efficient propagation of belief on Bayesian Networks was proposed in the 1980s by Pearl [Bonissone 1997]. He proposed an efficient updating scheme for trees and poly-trees in a series of research publications [Pearl 1982, 1986, and 1988]. Despite the novelty of Pearl's work, a major drawback of his approach is the increasing computational complexity as the graph complexity increases from trees to poly-trees to general graphs. Table (1 – 2) illustrates this increasing complexity.

TABLE (1 – 2): Increasing complexity of Pearl's belief propagation on Bayesian Networks.

Graph Type	Complexity	Variables
Tree	$O(n^2)$	*n* is the number of values per node.
Poly-tree	$O(K^m)$	K is the number of values per parent node, *m* is the number of parents per child.
Multi-connected graphs	$O(K^n)$	K is the number of values per node, *n* is the size of the largest non-decomposable sub-graph.

Several techniques have been developed to decrease the above computational complexity. The common goal of those techniques is to decrease the value of *n*, decomposing the initial problem into a set of smaller sub-problems. The most common complexity reduction techniques are *moralization and propagation in a tree of cliques* [Lauritzen and Spiegelhalter 1990] and *loop cutest conditioning* [Suermondt *et al.* 1991].

In problems where the above decomposition is not possible, a different class of methods is used, namely Approximate Methods. The most common approximate methods are *clustering*, *bounding conditioning*, [Horvitz *et al.* 1989], and *simulation techniques*, of which, the most common are *logic samplings* and *Markov simulations* [Henrion 1989].

MTE/DST

MTE/DST is a generalization of the Bayesian theory of subjective probability [Shafer 1990]. It provides a mechanism for evaluating the outcome of systems with randomness and probabilistic uncertainties. Moreover, it makes it possible to update a previous outcome estimate when new evidence becomes available [Bonissone 1997]. MTE/DST has been widely accepted and used because, in addition to its unique points of strength, it is also compatible with the Classical Probability Theory, compatible with Boolean Logic, and has been feasible in terms of computational complexity [Wierzchoń and Kłopotek, still under preparation].

Dempster-Shafer Theory of Evidence is built on the definition of a space of propositions Θ. A function *m* is defined to map subsets of that space of propositions Θ on the [0, 1] scale. The function *m* is called a basic probability assignment if:

- $m(\phi) = 0$
- $0 < m(x) < 1, where\ x \subseteq \Theta$ [5]
- $\sum_{x \subseteq \Theta} m(x) = 1$

[5] This is the definition given by [Bonissone 1997]. [Ramsay 1999] defines it as $m(x) \geq 0$.

The value of the probability function *m(x)* represents the degree of evidence that x is true. Two important characteristics here define the distinction between a Dempster-Shafer probability assignment and a probability distribution. First, $m(x_1)$ and $m(x_2)$ may both equal zero even when $x = x_1 \bigcup x_2$ and *m(x)* ≠ 0. Second, $x_1 \subset x_2$ does not immediately imply that $m(x_1) < m(x_2)$. Another function called the credibility function $Bel(x)$ and is defined as follows:

$$Bel(x) = \sum_{y \subseteq x} m(y)$$

Regardless the above pointed out differences between the basic probability assignment and probability distribution, the credibility function $Bel(x)$ reduces to classical probability distribution if function *m* assigns values greater than 0 to singleton sets [Ramsay 1999]. Finally, the certainty of any proposition x is represented by the interval [$Bel(x), P(x)$], where $P(x)$ is defined as follows:

$$P(x) = \sum_{y \bigcap x \neq \phi} m(y)$$

and the following relation can be derived from the above definitions:

$$Bel(x) = 1 - P(\neg x)$$ [Bonissone 1997].

Fusion of Methodologies

The methodologies presented above have been theoretically and experimentally proven to be very powerful in many areas of application. However, they have not yet been up to their biological counterparts [Jain and Martin 1998]. Thus, Lotfi Zadeh proposed the idea of fusing those methodologies into one multidisciplinary field, namely Soft Computing, so that the merits of one technique can offset the demerits of another. Later, others proposed merging other techniques as well into Soft Computing or what they called Extended Soft Computing. However, the use of one of those methodologies still falls into the category of Soft Computing.

It is important here to point out two important characteristics of this fusion process. First, regarding the distinction between probability and fuzziness, it should be clear that these two terms, even though they both deal with uncertainty, but they capture two different types of uncertainty and address two different classes of problems. Second, we do not have to fuse all different techniques in one application in order to obtain what we can call a Soft Computing technique.

Probability Theory vs. Fuzzy Set Theory

To distinguish between Probability Theory and Fuzzy Set theory, we have to understand the type of uncertainty that each of them describes and processes. The uncertainty described by probability is randomness. On the other hand the uncertainty described by fuzzy set theory is known as fuzziness.

In probability, we have a sample space S and a *well-defined* region R of that space. The uncertainty results from the *non-deterministic* membership of a point P in the region R. The sample space S represents the set of all possible values of a random variable. The Region R represents the event that we want to predict. The point P is the outcome of a system. The characteristic function of the region R determines whether the statement "P $\subseteq$ R" is true or false. Hence, the membership value of P in R can be either zero or one. In other words, the function *maps the sample space to the set {0, 1}*. The probability value describes the *frequency* of such a value [Bonissone 1997].

In contrast, fuzzy set theory defines a space S′ and an *imprecisely defined* region R′ in that space. The uncertainty results from the *partial membership* of a point P in the region R′. The characteristic function of the region R′ defines a *mapping from the sample space to the interval [0, 1]*. The partial membership describes the *degree*, rather than the frequency, to which P $\in$ R′.

Another way of stating the distinction between probability and fuzziness is to state the class of problems to be solved by each. Probability theory deals with the prediction of a future event based on the information currently available. On the other hand, fuzzy set theory deals with concepts and status of variables, rather than events [Klir and *et al.* 1997].

Hybrid Soft Computing Algorithms

Several hybrid techniques that combine more than one soft computing technology have been developed. The remaining part of the chapter lists and presents the most common and successful soft computing hybrid technologies.

Probabilistic Fuzzy Systems: The above presentation of the differences between fuzzy set theory and probability theory points out the importance of the collaboration of the two to solve complex problems, rather than competing on solving the same problem. Many problems deal with both types of uncertainty, randomness and fuzziness, and require this collaboration. For example, in weather forecast, probabilistic techniques are needed to predict the future weather status, e.g., “it is highly probable that it will be cloudy (or partly cloudy) tomorrow”. However, fuzzy techniques are also needed to not only capture and quantify the concept of “cloudy”, or “partly cloudy”, but also the concept of “highly probable”.

Fuzzy Neural Networks: The need for developing hybrid systems that combine Fuzzy technology and Artificial Neural Networks was motivated by the shortcomings and the complementary nature of each of the two methodologies. The performance of Artificial Neural Networks becomes degraded and less robust when the inputs are not well defined, i.e., fuzzy inputs. Neurons in ANNs also do not function properly when network parameters are fuzzy. On the other hand, fuzzy systems are not capable of learning. Moreover, the fuzzy rules that define the input/output relationship must be known in advance. Therefore, combining the two technologies to create hybrid systems that fill the gabs of one paradigm by means of the other was very highly motivated [Kartalopoulos 1996].

S. Lee and E. Lee were among the earliest to propose the combination of Fuzzy Logic (FL) and Artificial Neural Networks (ANNs) [Lee and Lee 1974]. Since then, many hybrid FL/ANNs systems have been proposed or actually developed. Takagi conducted an exhaustive survey of those systems up to 1990 [Takagi 1990]. In this context, the combination of the two technologies has gone two ways. FL has been used to tune and control ANNs and ANNs have been used to tune FL controllers.

In the first direction, FL controllers have been used to control the learning rate of ANNs. The goal is to optimize the degrading performance that typically occurs when ANNs approach the local minimum. A fuzzy controller was developed to accomplish that task in 1992 by Arabshahi, Choi, Marks, and Caudell [Arabshahi *et al.* 1992].

In the second direction, ANNs have been used to tune FL controllers. In 1974 Lee and Lee proposed a novel model of a neuron with multi-input/multi-output, instead of the binary-step output function that was widely accepted then [Lee and Lee 1974]. Since then, the research has been very active in this area and a special issue of IEEE Communications Magazine was dedicated to Fuzzy Neural Networks [Plevyak 1992]. Another milestone in the field is the Adaptive Neural Fuzzy Inference Systems (ANFIS) by Jang [Jang 1993]. More recently, Costa Branco and J. Dente of the Mechatronics Laboratory, Department of Electrical and Computer Engineering, Instituto Superior Técnico, Lisbon, Portugal, designed an electro-hydraulic system using Neuro-Fuzzy techniques [Costa Branco and Dente 1999].

Many other implementations of FL controllers tuned by ANNs have been developed. Examples of those are described in [Kawamura *et al.* 1992], [Bersini *et al.* 1993], and [Bersini *et al.* 1995]. Khan carried out a more recent study of neural fuzzy systems, surveying the advantages and disadvantages of neural and fuzzy systems and the different types of Neural Fuzzy Systems, describing some real world implementations and applications [Khan 1999].

Fuzzy Genetic Algorithms: Genetic Algorithms (GAs) and FL have also been combined to generate the hybrid field of Fuzzy Genetic Algorithms (FGAs). Similar to the case of Fuzzy Neural Networks, the fusion has gone also two ways. GAs controlled by FL as well as FL controllers tuned by GAs.

Typically, GAs perform a global search in the solution space, called the exploration phase, followed by a localized search in the discovered promising region, called the exploitation phase. FL has been used to manage the resources of GAs such as population size and selection pressure during the transition between these two phases [Lee and Takagi 1993], [Herrera *et al.* 1995], [Cordon *et al.* 1996]. GAs resource management by FL resulted in adaptive algorithms, which significantly improved its efficiency and speed of convergence [Bonissone 1997]. Moreover this adaptability can also be used in setting parameters, selecting genetic operators, setting the genetic operators behavior, representing solutions, and setting the fitness function [Herrera and Lozano 1996].

Research has also been active on the use of GAs to tune FL controllers. An exhaustive survey of the research in this area was published in [Cordon *et al.* 1996]. The study surveyed over 150 research papers on using GAs in tuning designing FL controllers [Bonissone 1997].

Among the most important implementations in that trend were the techniques of modifying the membership functions of FL controllers by means of GAs [Karr 1991], [Karr and Gentry 1993]. Another trend is to use GAs to tune the rules used by FL controllers [Herrera *et al.* 1995]. Kinzel *et al.* used GAs to tune, both, the rules and the membership functions. Instead of the traditional string representation of the rules, they used a cross-product matrix. The general algorithm they used consists of three steps. First, they defined the initial rule base using intuitive heuristics. Second, they used GAs to generate a better rule base. Finally, they used GAs to tune the membership functions of the final rule base [Kinzel *et al.* 1994].

ANNs/GAs: GAs have been used in synthesizing and tuning ANNs in many ways. One way is to use the GAs to evolve the network topology before Back Propagation is used to tune the network. GAs have also replaced Back Propagation as a technique for finding the optimal weight. Another application of GAs in ANNs has been making the reward function adaptive by using GAs to evolve the reward function. However, combining more than one of those utilization techniques requires the GA chromosome to be too large, which would result in an inefficient global search. Therefore, combining more than one of the above approaches has been rare and was attempted only by using variable granularity to represent the weights [Maniezzo 1994], [Patel and Maniezzo 1994].

Many combinations of ANNs with GAs can be considered a continuation of the earlier discussions of the hybrid methods to exploit the advantages and overcome the disadvantages of GAs and ANNs. For example, ANNs using Back Propagation are able to exploit their local knowledge. Hence, they are faster to converge than GAs, but this is at the expense of risking the ANN getting stuck in the local search, which happens frequently and causes the whole ANN to get stuck in local minima. On the other hand, even though GAs are not exposed to this problem, but they are slower due to their global search characteristic. Therefore, GAs are efficient in finding the promising region where the global minimum is located, i.e., coarse granularity search, but they become very inefficient when the granularity is fine. This tradeoff was the motivation behind the hybrid algorithm proposed by Kitano in 1990. Kitano algorithm starts by using GA to find a good parameter region. The found parameter region is then used to initialize the ANN. Finally, Back Propagation is used for the final parameter tuning [Kitano 1990].

Chapter Conclusion

The above presentation shows that the areas of application of Soft Computing and its constituents are rapidly expanding. Besides the traditional application of control, many other applications in diverse areas have been proposed, implemented, and actually deployed. Khan states, *"Neural Fuzzy techniques can be applied to many different applications. Home appliances (vacuum cleaners, washing machines, coffee makers, cameras etc.), industrial uses (air conditioners, conveyor belts, elevators, chemical processes, etc.), automotive (antiskid braking, fuel mixture, cruise control, etc.), fast charging of batteries, and speech recognition are a few examples."* [Khan 1999]. Soft Computing technologies have been used to design electro-hydraulic systems [Costa Branco and Dente 1999]. Methods based on GAs and ANNs have been used to solve the Vehicle Routing Problem [Potvin and Thangiah 1999]. Another application is the use of

FL and ANNs in fault detection [Köppen-Seliger and Frank 1999]. FL and ANNs have also been applied to machine diagnosis [Tanaka 1999]. Another very innovative application is the use of Time-Delay Neural Networks for estimating lip movements from speech analysis, a research done on developing multimedia telephone for hearing-impaired people [Lavagetto 1999].

More recently, the tendency toward combining more than two soft computing techniques in one application has been growing. Koji Shimojima and Toshio Fukuda proposed a new hierarchical fuzzy-neural control system for an unsupervised Radial Basis Function (RBF) fuzzy system. This control system combines FL, ANNs, and GAs techniques. The hierarchical fuzzy-neural controller is based on a skill knowledge database consisting of the skills acquired by the fuzzy-neuro controller. Those skills were acquired through an unsupervised learning based on Genetic Algorithms [Shimojima and Fukuda 1999].

The list of applications of soft computing includes other fields as well such as chemistry, medicine, information engineering, computational science, networking and distributed computing, and many others. Such a list can be a very extended one and very difficult, if not impossible, to cover in one document.

CHAPTER TWO

QUALITY OF SERVICE DISTRIBUTED COMPUTING

Computer Networks

A *computer network* is a form of *communication networks*. A communication network is *"the set of devices, mechanisms, and procedures by which end-user equipment attached to the network can exchange meaningful information"* [Saadawi *et al.* 1994]. In a communication network, electric signals are transmitted over a path that has a mechanism for converting those signals to, and from, bits. The bits are usually grouped into frames, packets, or messages. A communication network must also incorporate methods to overcome path deficiencies and techniques for selecting and maintaining the paths.

The above characteristics are common between all communication networks, and computer networks are no exception. The evolution of communication networks started in the nineteenth century by the inventions of the telephone and telegraph. Since then, different types and technologies of communication networks have evolved. A computer network is one of those types. Today, information infrastructure is based on the interconnection of computer networks with other types of communication networks.

Two motivations were behind the development of computer networks. The first was the need to build efficient networks for information exchange. The second was the need to build efficient distributed computing systems to overcome the limitations of the localized sequential computers.

Types of Computer Networks

Computer networks are classified into four types according to the number of nodes and their proximity. The first type is called *Massively Parallel Processor* (MPP) network. In MPP, a large number of nodes, can be thousands, are interconnected in a small area, typically 25 meters. The traffic of MPP is, typically, all-to-all. The second type is called *Local Area Network* (LAN). The LAN interconnection can cover up to few kilometers and the traffic is typically many-to-one. The third type is called *Wide Area Network* (WAN). WANs can interconnect computers throughout the world. The forth, most recent, type is *System Area Network* (SAN). SAN falls between MPP and LAN. It was developed to be fast, cost-effective networks. The closed relatively small area makes it possible to use wider and faster connections without the need for the costly fiber optics. Two or more interconnection networks can be connected to form an Internetwork [Hennessy and Patterson 1996]. In addition, some references define another category, namely *Metropolitan Area Network* (MAN), that falls between LAN and WAN. A MAN can be used to connect a network of LANs within a city, or over a campus. The network that covers the university campus or a large hospital is an example of MAN [Dodd 2002].

The evolution of the above types of networks started with the development of WANs. The first WAN was built in 1969 and was called ARPANET. The success of

ARPANET led to the development of the first LAN in 1974. The Ethernet interface chips were used to connect a cluster of 8086 computers in a hyper-cube forming the first MPP in the 1980s. SAN is the most recent and began to be available in the 1990s [Hennessy and Patterson 1996].

Characteristics of Computer Networks

Protocol: A protocol that defines the sequence of steps to be followed by communication software to transmit messages is typically used to enable hardware and software of different manufacturers to communicate over the network. In this context, the two important terms *routed protocol and routing protocol* are often confused. Routed protocols such as Internet Protocol (IP), DECnet, Apple Talk, Novell Netware, OSI, Banyan VINES, and XNS are the protocols that are routed over a network. In contrast, routing protocols are protocols used to implement routing algorithms, i.e. path selection algorithms. Examples of commonly known routing algorithms are Interior Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF), and Exterior Getaway Protocol (EGP) [Cisco Systems *et al.* 2001].

Performance: The network performance is generically defined by the efficiency, effectiveness, and reliability of the network. However, some measurable parameters are needed in order to evaluate the network performance quantitatively. Performance parameters can be classified into four categories: delay parameters, throughput parameters, accuracy parameters, and availability parameters [Verma 1989]. The terminology for performance parameters is not standardized. Different references may name them, or categorize them, differently. However, an important performance criteria that does not seem to fit under any of the above four categories is scalability, which is an important factor to consider when comparing between different types of networks such as IP networks versus ad hoc networks [Perkins 2001].

Functionality: The operational characteristics of a network include (1) Interconnection Topology, dynamic or static. (2) Timing Protocol, synchronous or asynchronous. (3) Switching Method, circuit or packet switching. (4) Control Strategy, centralized or distributed [Gallivan 1998].

Interconnection Topology

Ideally, from performance and programming point of view, a network should be built with full connectivity, i.e. each node is connected to every other node, as in Figure (2-1)[6]. Unfortunately, this cannot be a practical solution for networks with more than few nodes. Therefore, various network topologies have been proposed [Pacheco 1997].

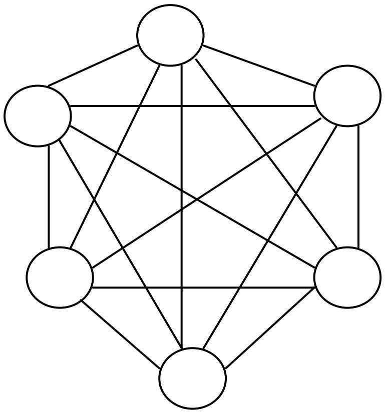


FIGURE (2-1): Full Connectivity

[6] In the topology figures, circles represent nodes and squares represent switches.

Network topology has been a very active research area. An enormous number of topologies have been proposed in research publications. Though, only few have been practically used. The most popular network topologies are crossbar, omega, fat tree, ring, mesh, torus, and hypercube. Those network topologies can be categorized into two broad categories, namely, static networks and dynamic networks. Static networks are networks that, when represented with a graph, each vertex in the graph would represent a node in the network. Dynamic networks are networks in which some vertices represent nodes and others represent switches.

A crossbar network topology is a dynamic implementation of the fully connected network. Instead of connecting each node to every other node directly, switches are used to implement this connectivity. Although crossbar networks are less costly than networks with full connectivity, they are still rather expensive because of the number of switches required, $O(n^2)$, where n is the number of nodes in the network. Figure (2-2) illustrates a crossbar network topology.

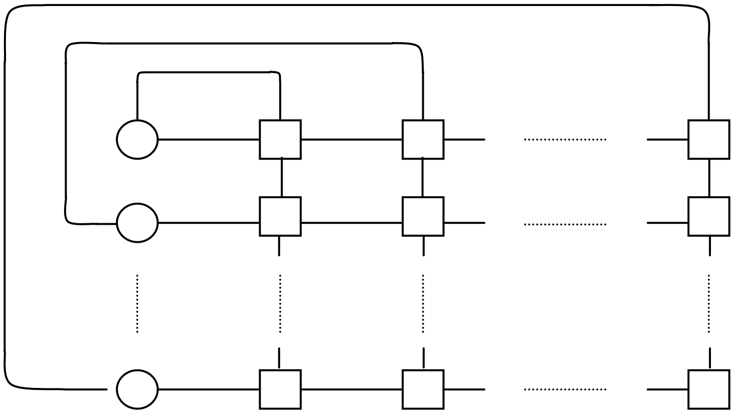


FIGURE (2-2): Crossbar Network Topology

Omega networks were developed as a solution to the high cost problem of crossbar networks. Instead of the $O(n^2)$ switches used by crossbar networks, the number of switches in an omega topology is reduced to $O(n\log_2 p/2)$. However, this is at the expense of the increased delay due to the less number of switches and the multistage switching mechanism. Omega and crossbar networks are often combined to compromise the cost versus the performance.

Another dynamic network topology that aims at less complex, less costly network with high or full connectivity is the fat tree. Typically, the number of links in each level is increased to allow higher bandwidth by allowing multiple paths between any pair of nodes.

The above topologies are all dynamic network topologies since they combine the interconnection of switches and nodes to construct the network. The other class of network topologies is the static networks, where nodes are directly connected without switches. The simplest and least connected example of that class is a linear array of nodes. In such a linear array, a node cannot reach more than two neighboring nodes directly. The ring topology is a slight enhancement on the linear topology that is done by connecting the first and last nodes to create a ring as illustrated in Figure (2-3).

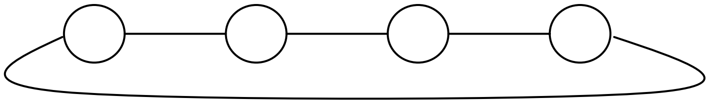


FIGURE (2-3): Ring Topology

A mesh topology is a higher-dimensional version of the linear array. Typically, a mesh can be two-dimensional or three-dimensional. A torus is actually a mesh with the first and last nodes in a row, or column, connected. In other words, a torus is a combination of the mesh and ring into one network topology.

The most powerful topology among static networks is the hypercube. A hypercube is a recursive construction of a *d*-dimensional hypercube from *(d-1)*-dimensional hypercubes. Hence, a 0-dimensional hypercube consists of only one node, a 1-dimensional hypercube of two connected nodes, a 2-dimensional hypercube consists of two 1-dimensional hypercube, and so on as illustrated in Figure (2-4). In a *d*-dimensional hypercube each node is connected to exactly *d* nodes.

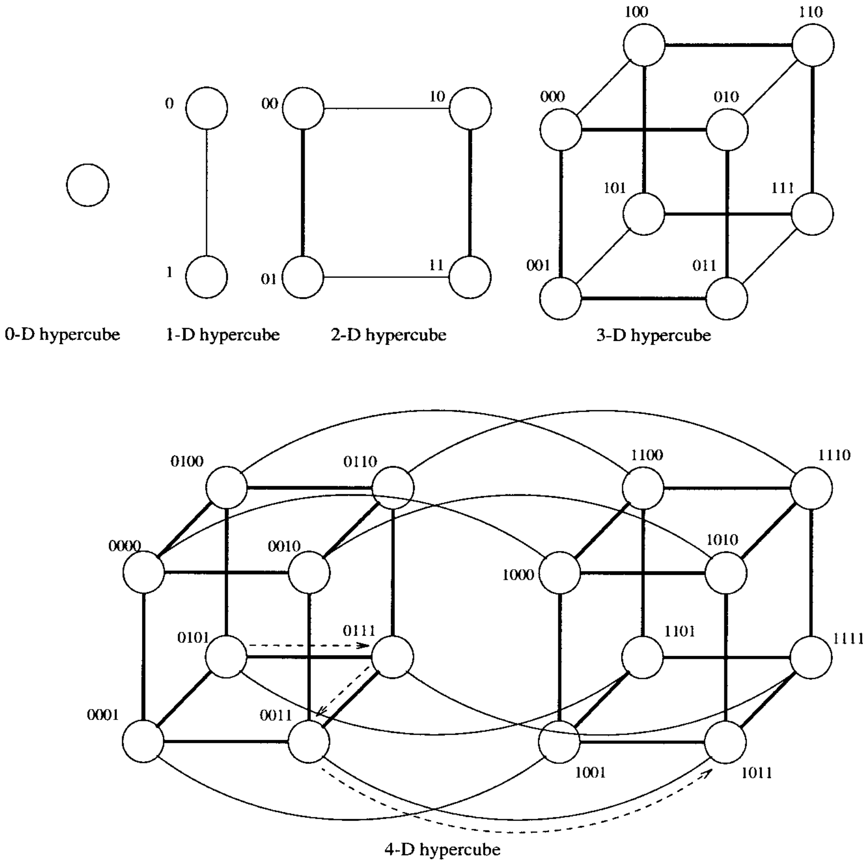


FIGURE (2-4): Hypercube Topology, scanned from [Gallivan 1997]

Finally, the parameters of a network interconnection topology are the Diameter, the Arc connectivity, and the Bisection width. The diameter is defined to be the maximum distance between any two nodes in the network, where the distance is defined as the least number of nodes that needs to be passed to reach one node from another. Connectivity is defined to be a measure of the multiplicity of paths between different nodes, the higher the connectivity the less the network contention. Bisection width is defined to be a measure of the traffic volume a network can handle [Gallivan 1997], [Pacheco 1997].

Routing

Routing is the process of transmitting information over a network from a source node to a destination node. A routing device, called router, is a device that directs packets to their designated destination. Routers also maintain the routing tables used to determine the correct paths [Slattery and Burton 2000].

Most packet switching networks use the hop-by-hop paradigm for routing packets over the network. In this paradigm, the routing process starts by running the routing protocol program that exchanges route information with other routers in the network. The most common routing protocols today are based on either the *link state algorithm* or the *distance vector algorithm*. The next step for the router is to use the information obtained by the routing protocol to populate the routing tables. Next, the router scans the routing tables to select the best path to each destination. When the router receives a packet, it determines the destination address from the packet's header. Then, it checks the forwarding table to determine the outgoing interface and the address of the next hop in the path to the destination. Finally, the router forwards the packet to the appropriate device and updates its status and settings [Halabi and McPherson 2000]. Routing is a major problem in Quality of Service networks and requires further special consideration. Therefore, it will be readdressed again in the following section.

Quality of Service Computer Networks

The network Quality of Service (QoS) is a relatively new term, which is defined as: *“The capability to control traffic-handling mechanisms in the network such that the network meets the service needs of certain applications and users subject to network policies”* [Bernet 2001]. Alternatively some references define QoS by the capability to define and specify the level of performance in a system [Freedman 2001]. To provide the capabilities of measure and control required by either definition, QoS networks must have mechanisms to control the allocation of resources among applications and users.

The notion of QoS came up as a response to the new demands imposed on the network performance by modern applications, especially multimedia real-time applications. Those applications made it necessary to set limitations on what can be defined as an *acceptable time delay* when routing information over a network. Those time demands are classified into three main categories. The first is the subjective human needs for interactive computing such as chatting sessions and other interactive web applications. The second is the automated tasks under time constraints such as the automated once-per-day backups during a limited pre-assigned time period. The third category is the need of some applications for a transmission rate with limited jitter along with a temporal ordering of the transmitted packets. This is the case when streaming multimedia over a network. The transmission rate is needed to keep the transmitted material meaningful and perceptible while the preserved temporal order is needed for synchronization [Armitage 2000].

The temporal requirements presented above are intrinsic to QoS that some references define QoS in terms of those requirements. Webster's New World Dictionary of Computer Terms defines QoS to be *"the guaranteed data transfer rate"* [Pfaffenberger 2000]. The word "guaranteed" is of special importance since QoS can only be implemented through guarantees on the limits of some network parameters as will be explained below.

It is important here to note that although QoS became an issue only in the past few years, but the idea of QoS had been envisioned earlier before new applications mandated the use of QoS. In the initial IP specification, a Type of Service (ToS) byte is reserved in the IP header to facilitate QoS. Until the late 1980s, almost all IP implementations ignored the ToS byte since the need for QoS was not yet obvious [Vegesna 2001].

QoS Performance Measures

In order to provide QoS, some quantitative measures of what constitutes QoS must be defined. As mentioned above, QoS is quantitatively defined in terms of guarantees or bounds on certain network performance parameters. The most common performance parameters are the bandwidth, packet delay and jitter, and packet loss.

Bandwidth: The term *bandwidth* defines the transmission capacity of an electronic line [Freedman 2001]. Theoretically, it describes the range of possible transmission rates, or frequencies [Sackett 2000]. In practice, it describes the size of the pipe that an application program needs in order to communicate over the network [Vegesna 2001]. The significance of a channel bandwidth is that it determines the channel capacity, which is the maximum information rate that can be transmitted [Saadawi *et al.* 1994]. The relationship between channel capacity and information transmission rate was set in the *Information Theory* of Claude Shannon in the 1940s.

According *Shannon's information theory*, if information rate is R and channel capacity is C, then, it is always possible to find a technique to transmit information with arbitrarily low probability of error provided $R \leq C$ and, conversely, it is not possible to find such a technique if $R > C$ [Nielsen and Chuang 2001].

Packet Delay and Jitter: The *delay*, also known as *latency*, consists of three different types, namely, *serialization delay*, *propagation delay*, and *switching delay*. Serialization delay, also called *transmission delay*, is the time it takes a device to synchronize a packet on a specified output rate. This transmission delay is a function of the bandwidth and the packet size. For example, a packet with size 64 bytes would take 171 μs when sent at the rate of 3Mbps. The same packet would take 26 ms when sent at the rate of 19.2 kbps.

Propagation delay is the time it takes a bit to travel from a transmitter to a receiver. Physics set upper limits on the speed of such a bit, making at best a fraction of the speed of light. Hence, propagation delay is a function of the distance traveled and the link medium.

Switching delay is the time lag between receiving a packet and starting to retransmit it. The switching delay is a function of the device speed.

In addition to those three types of delay, other delays also contribute to the overall performance of the network. Depending on the traffic, network condition, and the nature of the information being transmitted, different packets will experience different delays. The term packet jitter refers to this variation in packet delay. When the network is congested, queues will build up at the routers and start affecting the end-to-end delays [Vegesna 2001].

Queuing delay may be negligible when the network is fast and not experiencing congestion. However, when the network is congested, the queuing delay grows and becomes significant. The number of clients in a queue is a random variable and its distribution depends on *r*, the ratio of arrival rate to service rate. The probability *p* of having *n* clients in the queue is computed as follows:

$$p(n) = (1 - r) \times r^n$$

The queuing delay is a function of the number of packets in the queue and the service time for each queue. When the service rate is μ, the *average queuing delay aqd* can be computed as follows [Huitema 2000]:

$$aqd = \frac{1}{(1 - r)\mu}$$

Packet Loss: Packet loss is another important QoS performance measure. Some applications may not function properly, or may not function at all, if the packet loss exceeded a specified number, or rate. For example, when streaming video frames, after certain number of lost frames, the video streaming may become useless. This number may be zero in certain cases. Therefore, certain guarantees on the number of rate of lost packets may be required by certain applications for QoS to be considered. Packet loss can occur because of packet drops at congestion points when the number of packets arriving significantly exceeds the size of the queue. Corrupt packets on the transmission wire can also cause packet loss [Vegesna 2001].

QoS Levels

Even after realizing the inevitable need to QoS networking, there are still many applications that do not require any QoS. Moreover, those applications that do require QoS differ in the degrees of priorities and guarantees that they require to implement QoS. Therefore, on one extreme we have tasks that do not require any guarantees. On the other extreme, we have tasks that require absolute guarantees that may not be compromised. In between those two extremes there are numerous levels of QoS. However, those levels of QoS have been grouped into three main categories: best effort service, soft QoS, and hard QoS.

Best effort service: The level of best effort service provides no guarantees at all. It represents the first extreme mentioned above. It cannot really be considered a QoS. Many network applications work very well with best effort service. An example of such applications is the File Transfer Protocol (FTP). No guarantees or performance measures are assumed for FTP. The only criterion is whether the transfer was completed successfully or not.

Soft QoS: Soft QoS is also known as *differentiated service*. In this QoS level, no absolute guarantees are given. Rather, different priorities are assigned to different tasks. Hence, applications are grouped into different classes of priorities. Many application traffics work very well with this policy when absolute guarantees are not needed. For example, network control traffic should always be given higher priority over other data communications to ensure the availability of, at least, the basic connectivity and functionality at all times.

Hard QoS: Hard QoS is also called *guaranteed service*. It represents the level of QoS for applications that require absolute guarantees on the minimum needed resources of the network in order to function correctly, or in order to function at all. Prior network resource reservation over a path is usually performed to enable the network to provide, or deny, the required guarantee. Applications that require Hard QoS include multimedia applications, where streaming audio and/or video data is done in real-time.

It is important to note here that the only level the Internet is currently able to provide is the first level. Although the IP protocol supports QoS but still the Internet does not offer any of the other two levels of QoS. However, other networks such as ATM networks do support QoS [Vegesna 2001].

QoS Routing

The emergence of QoS networking created many challenges to network developers of many fields. It is true that physicists and engineers contributed significantly to the development of faster networks. On the horizon now are optical networks that use *Wavelength Division Multiplexing* (WDM) technology to increase the capacity of optical transmission systems by transmitting multiple wavelengths over a single fiber, reaching transmission rates on the order of *terabits per second* [Gerstel 1996], [Tomsu and Schmutzer 2002]. However, as the physical capabilities of the networks grow, the demands by new applications to exploit those capabilities also grow. This necessitates the need for computer scientists to constantly develop algorithms and solutions to provide the needed exploitation.

Many factors complicate the problem of QoS routing. One of those factors is the diversity of the requirements and guarantees of different distributed computing applications running simultaneously. This problem expands to include applications with zero constraints requirements, making it necessary to develop routing mechanisms that handle all the three levels of QoS presented above. The other major factor is the impossibility of maintaining accurate network state information in a large dynamically changing network. This latter factor will be a very important theme in the current dissertation.

To maintain network state information, each node in the network needs to maintain its local state. Then all local states can be combined to form the global state information. Typically, a node maintains the network global state information using one of two algorithms: *link-state* algorithm and *distance-vector* algorithm. This is done by using the chosen algorithm to exchange the local states between all nodes in the network periodically. The resulting global state information cannot be accurate due to many factors that will be discussed later. Dealing with this uncertainty about the network global state is one of the main problems this dissertation is trying to solve.

QoS routing algorithms can be classified according to the cardinality of the destination of the searched path into two main categories: *unicast routing* algorithms and *multicast routing* algorithms. In a unicast routing algorithm, the problem is to find the best feasible path from a source node to a destination node, satisfying a pre-designated set of constraints. In multicast routing, the problem is to find the best feasible tree that covers a source node and a set of destination nodes, satisfying a pre-designated set of constraints [Chen and Nahrstedt 1998]. The nature of the problem necessitates the use of algorithms from one category or the other with no trade-off between the two.

Another way of classifying QoS routing algorithms is according to the path search and deployment strategy. Accordingly, there are three routing strategies: *source routing*, *distributed routing*, and *hierarchical routing*. These three strategies can be used interchangeably in many cases with the involvement of trade-offs between the advantages and disadvantages of each strategy.

In the source routing, the feasible path is computed locally at the source node, which is expected to have its own maintenance mechanism of the network global state information. The main advantage of source routing is the localized storage of the network state information and the centralized computation of the path. The local maintenance of the global state enables the source node to compute the path locally as well. The computational complexity in this case is much smaller than that of the distributed computing. This, in turn, makes source routing algorithms much easier to design and implement. In addition, it guarantees loop-free routing. However, source routing suffers from two major problems. The first is the inaccuracy of state information. The degree of precision of the global state at each node is directly proportional to the frequency of updates. Nevertheless, the updating frequency is also directly proportional to the updating overhead, which is inversely proportional to the availability of resources for the actual network activities. This inevitable imprecision in the global state information may result in the failure of finding an existing feasible path [Shaikh *et al.* 1997].

The second routing strategy, distributed routing, depends on using distributed computing to compute the path. The computation is done by exchanging control messages and the global state information stored locally at each node [Salama *et al.* 1997], [Sun and Langendorfer 1997], [Wang and Crowcroft 1996]. However, some distributed routing algorithms do not require the maintenance of global state at all [Shin and Chou 1995], [Chen and Nahrstedt 1998]. The main advantage of distributed routing is the distributed computation of the path, which enables shorter response time and better scalability [Chen and Nahrstedt 1997]. The shorter response time and higher scalability are achieved at the expense of higher network traffic due to more message exchanging. Furthermore, distributed routing cannot be loop-free, especially when global states at different nodes are inconsistent.

In the last routing strategy, namely, hierarchical routing, nodes are grouped into clusters, which are further grouped into a higher level clusters. This recursive clustering continues to build up forming a multi-level hierarchy. Instead of maintaining global state information at each node, the aggregated state is maintained, where an elected node in each cluster maintains the global state of the nodes in the cluster in which it is local to in addition to the aggregated states of the other clusters. The use of partial global states maintained by logical nodes enhances the scalability of the hierarchical routing significantly over other routing schemes. In addition, the overall traffic in the network does not get so intense as it does in distributed routing. Thus, hierarchical routing combines advantages of both source and distributed routing. The only noticeable problem with hierarchical routing, which is not a trivial one, is that the aggregation of the network states introduces additional imprecision. It has been shown that the performance of routing algorithms that are not designed specifically to take imprecision into account degrades significantly as the imprecision grows [Yuan and Zheng 2001], [Apostolopoulos *et al.* 1998], [Shaikh.*et al.* 1998].

Handling the uncertainty in the network state information is among the major concerns of this dissertation and will be addressed again in the following chapters. Most routing algorithms available today do not take this uncertainty into account. Instead, they assume it does not exist, regardless of the inherent nature of this uncertainty. However, some research has been done to evaluate the impact of neglecting this uncertainty on the performance of different routing algorithms. In addition, few routing algorithms were proposed with the main objective of handling the intrinsic imprecision and reducing its effect [Chen and Nahrstedt 1998], [Guerin and Orda 1997], [Lorenz and Orda 1998]. Those studies will be reviewed, evaluated, and hopefully built upon to develop new algorithms throughout the dissertation.

CHAPTER THREE

REVIEW OF THE LITERATURE OF THE FUSION OF NETWORKING AND SOFT COMPUTING

In the coverage of the broad background material in chapters one and two the literature was reviewed for each of the two broad fields, namely, soft computing and QoS networking, separately. This chapter covers the preliminary review of the literature of the related studies that attempted to combine the two fields in order to implement QoS distributed computing environment based on soft computing techniques.

Since uncertainty is a major theme in each of the fields of soft computing, the fields of QoS distributed computing, and, consequently, this dissertation, the previous approaches and studies about handling this uncertainty will also be reviewed.

Another direction that will be investigated is the network adaptive control. Adaptive control is an essential tool for controlling a dynamic environment such as large networks. The uncertainty in the global state information comes as a result of the transient dynamic nature of such a network. Thus, static control that does not respond to those dynamic changes cannot be optimal. However, fuzzy control, which is adaptive by nature, has been the most successful application of soft computing in general and fuzzy technology in particular. Therefore, a review of the literature of network adaptive control will be carried out in preparation for a possible development of adaptive control based on soft computing.

Intelligent Networking

The preliminary review shows that very few attempts have been made to utilize soft computing techniques in the area of networks and distributed computing. A statement about ATM networks and Neuro-Fuzzy technology in particular was made by Douligeris and Palazzo, who state that *"It seems that the Neural Networks and Fuzzy Logic literature, and, of course, that for Neuro-Fuzzy, have addressed a variety of issues in system design, stability, and convergence that have not been evaluated extensively in ATM traffic control"* [Douligeris and Palazzo 1999].

Intelligent Wireless Management: Improving TCP Performance Over Wireless Networks [Suarez 2001]**:** This study is a Ph.D. dissertation by Tiki Suarez at the Department of Computer Science, Florida State University. The main objective of Suarez' dissertation was to modify the *Transmission Control Protocol* (TCP) in order to improve its performance by elevating the throughput over wireless networks.

The main problem of the research is the performance degradation of the TCP protocol when used over wireless networks. The main reason for this performance degradation is the congestion control mechanisms used by TCP to overcome, or alleviate, the problem of frequent link errors and packet losses typically associated with wireless networks. Suarez attempts to solve this problem by modifying the transport layer and the link layer protocols of the TCP.

The main contribution of Suarez's dissertation is the development of the *Intelligent Wireless Management* (IWM), a knowledge-base system that maintains the dynamically changing characteristics of the wireless channel. Based on the received information, the IWM modifies the protocols of the transport layer and the link layer in order to tune the TCP to the underlying network. The maintenance of channel error characteristics by IWM results in efficient retransmission of the lost, or dropped, packets. In addition, the utilization of IWM limits the triggering of the TCP congestion control mechanisms by controlling the errors at the physical medium. Suarez used *activity structures* for the design and validation of the IWM.

The project experimentation was carried out by using the *Network Simulator* (NS), which is a discrete event simulator for network research. Experimental monitoring was implemented by visualizing the packets transmission and drops. *Network Animator* (NAM) is the visualization tool used to create an animated trace. Suarez reported an increase of 24% to over 100% in the overall end-to-end performance after implementing the proposed IWM and performing the simulations.

An Artificial Neural Network Approach for Routing in Distributed Computer Networks [Samuel *et al.* 2001]**:** In this paper, the authors propose a new routing algorithm for packet switching computer networks. The proposed algorithm is based on the Hopfield model of Artificial Neural Networks (ANNs). They modeled the network as a non-directed graph, where links are considered bi-directional. The ANN used consists of $n(n-1)$ neurons, where n is the number of nodes in the network. The neurons are represented as a $n \times n$ matrix with neurons on the diagonal eliminated. A neuron at the (i, j) position of the matrix is characterized by its output V_{ij}, which is defined as:

$$V_{ij} = \begin{cases} 1 & \text{if the arc}(i,j)\,\text{is in the route} \\ 0 & \text{if not} \end{cases}$$

The proposed algorithm proceeds as follows:

1- Obtaining the network data, which includes the number of nodes, the matrix of links, and the traffic matrix.

2- Initialize the neurons output matrix. Each V_{ij} in the matrix is assigned a random value between –0.0002 and +0.0002.

3- Trigger the process of minimization in the ANN to solve the differential equations and stabilize the network.

4- Round the values in the output matrix

The last step leads to the routing matrix, which is used to assign flow to each link of the network configuration by distributing the traffic between pair of nodes. The network configuration is represented by a matrix M, where an element $M_{ij} \in \{0,1\}$; the diagonal of the matrix is also eliminated. In this matrix,

$$M_{ij} = \begin{cases} 1 & \textit{if the link}(i,j)\textit{exists} \\ 0 & \text{if not} \end{cases}$$

Once the network configuration and related data are obtained, the routes between pair nodes are determined, the flow and capacity are assigned to each link, and the mean delay of the network is computed.

The authors of the paper investigated the effect of varying the traffic and network size, and compared their algorithm with other optimal routing algorithms. The proposed algorithm gave delay results that are slightly higher but in execution times that are considerably smaller. Therefore, they concluded that, *"the proposed neural network approach is suitable to be integrated into overall topological design processes, for moderate and high-speed networks subject to quality of service constraints as well as to changes in configuration and link costs"*.

Fuzzy Expert systems in ATM Networks [Douligeris and Palazzo 1999]**:** This paper was published as a chapter in a book with the title "Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms: Industrial Applications". The paper shows that fuzzy rule based systems can be used effectively in admission control, policing, rate control, and buffer management of ATM networks.

In their paper, the authors introduce ATM networks as *"the most attractive information transport technique within broadband integrated networks supporting multimedia services, because of its flexibility"*. The authors then provide analysis and justification of the advantages of using fuzzy expert systems to solve ATM network problems. The main contribution of the paper is the proposal of a novel controller scheme for peak rate regulation and cell loss rate reduction. The proposed scheme is designed to work with the *Leaky Bucket* algorithm. The authors end the body of their paper by showing how their fuzzy controller can be, further, enhanced by combining it with ANNs. Finally, they recommend that it is time that researchers should start looking at more elaborate models and techniques to develop better algorithms for ATM networks using Neuro-Fuzzy systems.

The Uncertainty in Network State Information

Distributed QoS Routing with Imprecise State Information [Chen and Nahrstedt 1998]**:** The authors here start by presenting the source of uncertainty by stating that the network state information maintained at every node is often imprecise in a dynamic environment. This is caused by the propagation delay of state messages, periodic updates (more updates more overhead), and the hierarchical state aggregation.

The network is modeled as a set V of nodes that are interconnected by a set E of full-duplex, directed communication links. The delay and cost of a path $P = i \rightarrow j \rightarrow ... k \rightarrow l$ are defined as:

$$delay(p) = delay(i, j) + ... + delay(k, l)$$

$$\cos t(p) = \cos t(i, j) + ... + \cos t(k, l)$$

The following information is required to be maintained at every node i for every possible destination t.

- **Connectivity** $R_i(t)$ **:** a routing table entry, keeping a subset of adjacent nodes that can be used to route data from i to t.
- **Delay** $D_i(t)$ **:** the delay of the least-delay path from i to t.
- **Cost** $C_i(t)$ **:** the cost of the least-cost path from i to t.
- **Delay Variation** $\Delta D_i(t)$ **:** keeps the estimated maximum change of $D_i(t)$ before the next update.

All these, especially $D_i(t)$ are inherently imprecise in a dynamic network. Therefore, in order to overcome the imprecision problem, the authors developed a multi-path distributed routing scheme, called *ticket-based probing*. In this scheme, routing messages (probes) are sent from the source *s* to the destination *t* to search for a low-cost path that satisfies the delay requirement. Certain number of tickets is issued at the source according to the contention level of network resources. The total number of probes is bounded by the number of available tickets, since each probe is required to carry, at least one, ticket. The maximum number of paths is also bounded by the number of tickets, since each probe searches a path. The routing scheme uses the state information at intermediate nodes to guide the limited tickets along the best paths to the destination. The authors listed several advantages of the proposed scheme:

1. The routing overhead is controlled by the number of tickets, which allows the dynamic trade-off between the overhead and the routing performance.

2. Tolerates information imprecision by searching multiple paths in parallel, increasing the chance of finding a feasible path.

3. The scheme considers the optimality of the selected path as well as the QoS requirement.

However, two problems had to be dealt with. The first is: how to determine the number of tickets (No). The second is: how to distribute the tickets of a received probe among the new probes. The number of tickets problem was solved by designating Yellow tickets (Yo), and green tickets (Go). Then, No = Yo + Go, where Yo and Go are determined by the delay factor D. For the distribution of tickets problem, each probe accumulates the delay of its path. If node k sends a probe p to node i then

$$\text{Let } R_i^p(t) = \{j \mid delay(p) + delay(i,j) + D_j(t) - \Delta D_j(t) \le D, j \in R_i(t) - \{k\}\}$$

If $j \in R_i(t) - R_i^p(t)$, then no need to send tickets. If $R_i^p(t) = \phi$, invalidate all received tickets and discard them. Otherwise, for every $j \in R_i^p(t)$, i makes a copy of p, denoted as p_j. p_j has $Y(p_j)$ yellow tickets and $G(p_j)$ green tickets such that

$$\sum_{j \in R_i^p(t)} Y(p_j) = Y(p) \text{ and } \sum_{j \in R_i^p(t)} G(p_j) = G(p)$$

$$Y(p_j) = \frac{(delay(i,j) + D_j(t))^{-1}}{\sum_{j' \in R_i^p(t)} (delay(i,j') + D_{j'}(t))^{-1}} \times Y(p)$$

$$G(p_j) = \frac{(\cos t(i,j) + D_j(t))^{-1}}{\sum_{j' \in R_i^p(t)} (\cos t(i,j') + D_{j'}(t))^{-1}} \times G(p)$$

The search terminates when all probes have either reached destination or been dropped by the intermediate nodes. All the tickets are then submitted to the destination and the invalid tickets are discarded. The valid tickets designate a feasible path. If more than one valid ticket arrived at a destination, the cost is checked to use the lowest cost path. Using this scheme, the authors attempt to overcome the imprecision problem indirectly by conducting a multiple searchs in parallel. However, the scheme is still classical in the sense that it does not take the imprecision into account when doing computation.

QoS-based Routing in Networks with Inaccurate Information: Theory and Algorithms [Guérin and Orda 1997]**:** This study presented a different approach based on using probabilistic methods for dealing with the inaccuracy of the network state information. The Goals, as stated by the authors, are to explore the impact of the inaccuracy in the available network state information on the route selection process, and to establish efficient routing schemes for connections with QoS requirements that are capable of operating efficiently in the presence of inaccurate information. The authors chose to focus on the inaccuracy in the bandwidth/delay information.

First, Connections with Bandwidth Requirements are considered. To deal with the inaccuracy, for each link $l \in E$, the source node only knows the quantities *pl*(*x*), where *pl*(*x*) is the probability that link *l* can accommodate a connection which requires *x* units of bandwidth. Then the problem becomes: find a path P* such that for any path P:

$$\prod_{l \in p^*} pl(w) \geq \prod_{l \in p} pl(w)$$

This problem can be solved by modifying the shortest path algorithm by adding properly selected weights.

Second, for Connections with End-to-End Delay Requirements, The solution is based on the rate-based delay model. A source node determines the end-to-end delay guarantee *d*(*p*) it expects to obtain from an *n*-hop path *p* as follows:

$$d(p) = \frac{\sigma}{r} + \frac{\sum_{l \in p} cl}{r} + \sum_{l \in p} d_l$$

where σ is the size of the connection's burst, cl is a fixed quantity at link *l*, typically the maximum packet length for the connection, and *r* is minimal rate that can be guaranteed for the connection along the path. Now, if we Let $\pi D(p)$ denote the probability that $d(p) \leq D$, where *D* is the maximum delay requirement, then the problem becomes: find a path p* such that, for any path p: $\pi D(p^*) \geq \pi D(p)$.

The main problem with this paper is that it never explained the method to construct the probability distribution function.

A Comparative Study of Quality of Service Routing Schemes That Tolerate Imprecise State Information [Yuan, Zheng, and Ding 2002]**:** This is paper is based on the Master's project of Wei Zheng at the Department of Computer Science, Florida State University. The importance of this work is that it provides methodology for comparing the performance of different routing algorithms that tolerate the imprecision in the network state information.

Empirical Probaility Based QoS Routing [Yuan and Yang 2003]**:** This paper is based on the Master's project of Guang Yang at the Department of Computer Science, Florida State University. The research is done on emperical probabilistic routing schemes, i.e., schemes that make routing decision based on empirical resource availability probability information, rather than the resource availability information. It is known that those probabilistic shcemes provide better performance than the latter ones when the global network state information is inaccurate. However, probabilistic schemes usually require the exchange of much larger data structure to maintain the probability information distribution. The study acknowledges and confirms these two facts. Moreover, it describes an approach similar to the one in [Gosh *et al.* 2001] for obtaining the empirical availability independent probability information.

The conclusions of the two studies, and other probabilistic routing research papers will be crucial to this dissertation as will be shown in the last chapter on the proposed fuzzy probabilistic routing scheme.

Adaptive Control

End-host Architecture for QoS-Adaptive Communication: In their paper, Abdelzaher and Shin present new communication subsystem architecture for QoS-adaptive real-time applications. The main contribution of that architecture is its dynamic QoS-optimization mechanism and support for flexible QoS contracts. Those QoS contracts between the communication client and server specify multiple levels of QoS with acceptable degradation mechanism. Under overload, the communication subsystem degrades connections predictably according to their respective contracts. The contracts also associate rewards with each level. The reward may be commensurate with the client's perception of the acceptability of each level. If it is impossible to service the client even at the lowest acceptable QoS level, the client must be reimbursed for the broken contract, which may also specify QoS-violation penalty to be paid by the service provider in that case. The system takes the total rewards as its success criteria. Hence, the system aims at maximizing its aggregate reward by choosing clients' QoS levels that optimize the aggregate reward based on resource availability and constraints and given the information about rewards and QoS-violation penalties specified by the contracts.

The proposed end-host communication subsystem was designed to achieve four goals: Provide per-flow or per-service-class guarantees on the end-host, maximize the aggregate reward across all clients, adapt responsively to transient load fluctuations and resource shortage, and avoid starvation.

A QoS level can be viewed as a communication-capacity reserve C_i, specified by a number of bytes M_i, a period P_i, and a buffer size B_i. The reserve can be of sender or receiver type. At the sender's side, the reserve holds enough computing and buffer resources for M_i bytes to be transmitted every P_i units of time. At the receiver's side, the reserve holds enough computing resources to receive M_i bytes every P_i time units. The buffer-size parameter specifies the maximum number of bytes to be buffered. In addition, $N_{max}(t)$ is a function defined as the maximum number of bytes that can be generated within an interval of length *t.* and is computed as follows:

$$N_{max}(t) = (B_i + \lfloor t/P_i \rfloor)\ M_i$$

Since $\lfloor t/P_i \rfloor M_i$ of these bytes will have been transmitted by time *t*, then at most $B_i\ M_i$ can be awaiting transmission at the sender at any given time. A byte generated at time *t* is considered *conformant* if (*total number of bytes generated by time t*) $< N_{max}(t)$. Guarantees are given to conformant bytes only. Moreover, transmission deadline of a byte generated at time *t* is $D_i = t + B_i\ P_i$. At the receiver end, the buffer size B_i specifies the maximum size of the delivery buffer, measured as a multiple of M_i. A byte is considered *delivered* when it has been processed by the protocol stack at the receiver and deposited in a delivery buffer.

A QoS contract is signed between the server and client upon connection establishment and specifies three things:

1- Alternative QoS levels acceptable to the client.

2- The levels' corresponding rewards.

3- QoS violation penalty.

In addition, the server may specify a QoS rejection penalty to be incurred if a new request gets rejected. Once the QoS contract is signed by the server, the client is guaranteed to receive the service at one of the levels agreed upon in the contract. The selection of the QoS level is left up to the server, or more accurately, the communication subsystem. Furthermore, the communication subsystem may change to any of the levels agreed upon at any time without consulting the client.

The user or the server, depending on the type of application, may set the rewards. However, the problem of specifying the rewards and the levels is that when a real-time application interacts with a human user rather than a physical environment, QoS levels and rewards may be subjective rather than objective. For example, the parameters of QoS levels may be set to “poor”, “good”, and “excellent” performance. Also, the rewards may be expressed as a corresponding degree of satisfaction. This is a very important area for fuzzy logic to step in. Terms like “poor”, “good”, “excellent”, and level of satisfaction are fuzzy terms to be used for metrics. To do effective and efficient computation on these terms, we should use fuzzy logic. Otherwise, the only thing to do is to translate them into non-fuzzy terms, which results in computation that may not be representative of the required metrics. The authors did not use fuzzy logic to determine the QoS levels and their corresponding rewards in their paper.

The system architecture is geared directly toward the four main objectives stated previously. To achieve per-client or per-service-class QoS on the server, each client (or service class) is handled within the communication server by a separately schedulable entity called *Adaptive Negotiation Agent* (ANA). The ANA expresses the client’s QoS contract terms to the communication server and is scheduled in accordance with the client’s assigned QoS level. The association of different ANAs with different clients allows servicing clients concurrently at different QoS levels. The second objective is to maximize the server’s aggregate reward. To achieve this, a proper choice of QoS levels that optimally utilizes available resources is needed. This is translated into proper ANA scheduling and schedulability analysis. To adapt responsively to transient load changes, which is the third goal of the system, a fast feedback loop is provided. This feedback loop detects transient overload or under-utilization conditions and makes small incremental changes to the QoS level currently available. Finally, to prevent starvation, the periodic optimization, together, with the fast QoS level adjustment heuristic ensure that during overload clients are gracefully degraded but not starved.

The two main modules of the system are the *Load Control* module and the *Monitoring-feedback* module. QoS levels are selected by *Load Control* module and feedback loops are implemented by *Monitoring-feedback* module [Abdelzaher and Shin 1998].

CHAPTER FOUR

NOTABLE DEFICIENCY IN FUZZY SET THEORY

As hybrid research, the author of this dissertation approached the two major fields involved in the dissertation in a bi-directional manner. The study of Soft Computing and its applications offer solutions to QoS Networking problems that, the author believes, are unmatchable by the current traditional approaches. On the other hand, the study of QoS Networking as an example of the class of complex, dynamical, nonlinear, and uncertain systems triggered motivations for improving, and developing, some of the Soft Computing sciences.

Among the Soft Computing sciences, this dissertation focused on Fuzzy Computing and Probabilistic Computing. This part of the dissertation, Chapters Four and Five, covers developments proposed by the author for improving the computational powers, and correcting possible deficiencies, in both paradigms. This chapter contains the work done on Fuzzy Set Theory.

Fuzzy Set Theory and the Observed Deficiency

Introduction

The operations on Fuzzy Sets that were presented in Chapter One are known as the Standard Fuzzy Set Operations. Since Dr. Zadeh introduced the concept of fuzzy sets to the world in 1965, many scholars studied and developed various functions to provide other definitions of the Fuzzy Union and Fuzzy Intersection operations. Today, we have a class of Fuzzy Union Functions, known as *t-conorm*, and a class of Fuzzy Intersection Functions, known as *t-norm*. The goal of developing such functions has always been to form functions that possess the appropriate properties required to ensure that the fuzzy sets produced by these functions are intuitively acceptable and meaningful.

As mentioned in Chapter One, the Law of Excluded Middle and the Law of Contradiction do not hold for Fuzzy Union and Fuzzy Intersection, respectively. The Standard Fuzzy Union operation is defined as follows:

$$(A \cup B)(x) = \max\,[A(x), B(x)] \qquad \textit{for all } x \in X$$

According to the Law of Excluded Middle, we should have:

$A \cup \overline{A} = X$, but this is not the case with fuzzy sets.

Also, Standard Fuzzy Intersection is defined as:

$$(A \cap B)(x) = \min\,[A(x), B(x)] \qquad \textit{for all } x \in X$$

The Law of Contradiction states that:

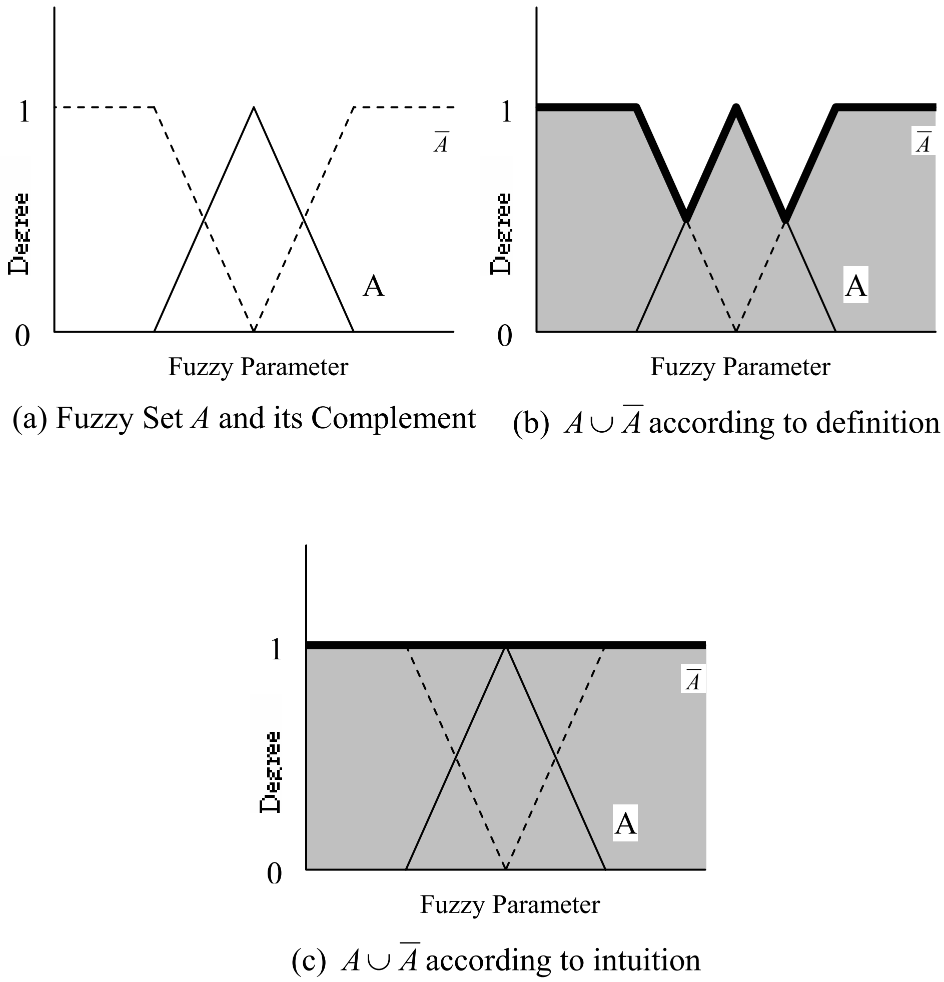
$A \cap \overline{A} = \phi$, but this does not hold for fuzzy sets either.

The Problem Observation

The above violations of the two laws hold for all of the classes of t-norm and t-conorm. As a matter of fact, these violations of the laws of excluded middle and contradiction have been regarded in the literature as properties of Fuzzy Sets. However, the question here is: are we justified when we take these two violations as normal properties of Fuzzy Sets? The author of this dissertation contends that the answer is: No, we are not justified. The author also contends that at least one of the two violations goes against logical reasoning and intuition.

Advocates of the violations being normal properties of fuzzy sets explain the violations as the result of the partial, and gradual, membership, i.e., when *x* has a partial membership in set *A*, say 0.7, it is conceivable that *x* also has a partial membership in the set's complement $\overline{A}$, 0.3 in this case by the definition of Fuzzy Complement. Therefore, *A* and $\overline{A}$ are not mutually exclusive like the case of Crisp Sets.

The above justification explains the violation of the Law of Contradiction but does not explain the violation of the Law of Excluded Middle. It is intuitive that if the two sets *A* and $\overline{A}$ overlap, then their intersection may not be empty. Nevertheless, this does not explain why their union should not be the set space, i.e., the Universal Set. It is intuitive that if the set space consists of only two sets, then the union of the two sets should be the entire set space, the Universal Set for that particular case. Figures (4 – 1) (a), (b), and (c) visualize the sets *A* and $\overline{A}$, their union as currently defined, and as intuitively should be, respectively.



(a) Fuzzy Set A and its Complement (b) $A \cup \overline{A}$ according to definition

(c) $A \cup \overline{A}$ according to intuition

Figure (4 – 1): Fuzzy Union and the Law of Excluded Middle

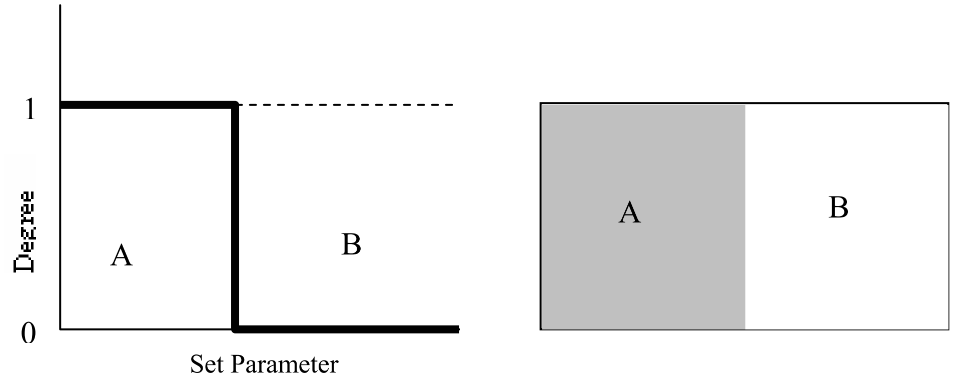
As serious as it is, the problem of violating the Law of Excluded Middle in Fuzzy Set Theory is not limited to going against logical and intuitive thinking. The violation also results in a serious computational weakness, that is, the lack of the modeling and representation of the set $S – A$, the set of all sets in the Universal Set other than, and excluding, set A.

The Problem Identification

The above presentation of the problem makes clear that we do have a serious deficiency in the Fuzzy Set Theory, which is the violation of the Law of Excluded Middle. This violation obviously goes against nature and logical intuitive reasoning. Now the question becomes: what is the solution to this problem? Before searching for, or developing, a solution, further investigation of the problem is necessary in order to identify the origin and nature of the problem.

After a thorough study of the problem, the author of the current dissertation concluded that the problem originated from an inherited property of Crisp Set Theory and Boolean Logic. That property is natural and intuitive in Crisp Set Theory and Boolean Logic but loses credibility, in the author's opinion, when ported to Fuzzy Set Theory and Fuzzy Logic without questioning and/or modification. The property in question is the identity relation between the Set Negation and Set Complement.

In Crisp Set Theory and Classical Binary Logic, the negation of a set, say set *A* – denoted as $\neg A$ –, is the same as the set complement – denoted as $\overline{A}$. With the accurate, well defined, and non-overlapping boundaries of crisp sets, this identity between the negation and complement of a set is justifiable and provable. It is true that there is one thing in common between the set complement and the set negation, that is they both refer to something as not in *A*. However, an important distinction between the two notions must be set according to their linguistic and intuitive meanings. The complement of a set *A* refers to every thing else in the set space other the set *A*. On the other hand, the negation of set *A* refers to the set, or sets, that counter the fundamental features characterizing set *A*. Although these two notions are identical in crisp sets, and may be in some cases of fuzzy sets too, but they are largely different in most real-life situations. The difference does not only depend on the set *A*, but also on the setting of the universal set of each particular case. Figure (4 – 2), (a) and (b), illustrates two different visualizations of a situation where the universal set consists of two mutually exclusive crisp sets, *A* and *B*.



(a) Membership in Crisp Sets *A* and *B* (b) Conceptual illustration

Figure (4 – 2) $\neg A$ ***and*** $\overline{A}$ ***are identical in crisp sets***

In the example illustrated by Figure (4 – 2), set *B* complements set *A*. Hence, when it is not *A*, it must be *B*, and nothing else. In reverse, when it is *B* it must be not *A*, and nothing else. Thus, we have:

$\neg A \textit{ iff } B$, that is, $\neg A \Leftrightarrow \overline{A}$.

The situation does not change when we have more than two sets in the set space, because *B* will still be the union of all the sets in the Universal Set except set *A*, that is:

$B = S - A$, where *S* is the Universal Set.

When the notion of Fuzzy Sets was proposed, a mathematical model of the theory was required. Since Fuzzy Sets were seen as a generalization of Crisp Sets, the approach for building a mathematical model was to generalize the mathematical model of crisp sets. Similarly, the mathematical model for Fuzzy Logic was constructed by generalizing the mathematical model of Many-Valued Logic, which is, in turn, a generalization of the mathematical model of the Boolean, Two-Valued, Logic. Hence, for example, we do have the *modus ponens* of classical logic and we do have the *generalized modus ponens* of fuzzy logic. Similarly, when the mathematical model of fuzzy sets was under construction, for each operation, or notion, in classical sets, a corresponding operation, or notion, built as a generalization of its counterpart in classical sets, was developed for fuzzy sets. These mathematical operations, and operators, were studied later and alternatives were proposed but what was not studied was the possibility that new concepts, and operations, may have been required in order to build a sound model, which is intuitively provable, or at least justifiable.

This one to one correspondence between the concepts and operations of crisp sets and fuzzy sets resulted in the persistency of the identity relation between the Set Negation and Set Complement in fuzzy sets too. But the question is: is it really true that the negation is the same as the complement in the case of fuzzy sets?

The Fuzzy Complement operation is defined as:

$$\overline{A}(x) = 1 - A(x)$$

Figure (4 – 1) above illustrates one possible case of such a complement. Since the power of Fuzzy Sets is best demonstrated when modeling real life situations, especially when dealing with subjective measures and linguistic variables, let us take a concrete, real, example, rather than the abstract example of Figure (4 – 1). We study the modeling of the notion of “experienced student”, depending on the number of credit hours taken [Klir et al. 1997]. Figure (4 – 3) illustrates one possible model of that notion. In the example, set *A* represents the set of "experienced students", set $\overline{A}$ represents the complement of set *A*, which is the set of "inexperienced students", according the current definition of fuzzy complement. The bold curve represents the union of the two sets, and the area between the bold curve and dashed line is the area in question of whether it should belong in the complement or not.

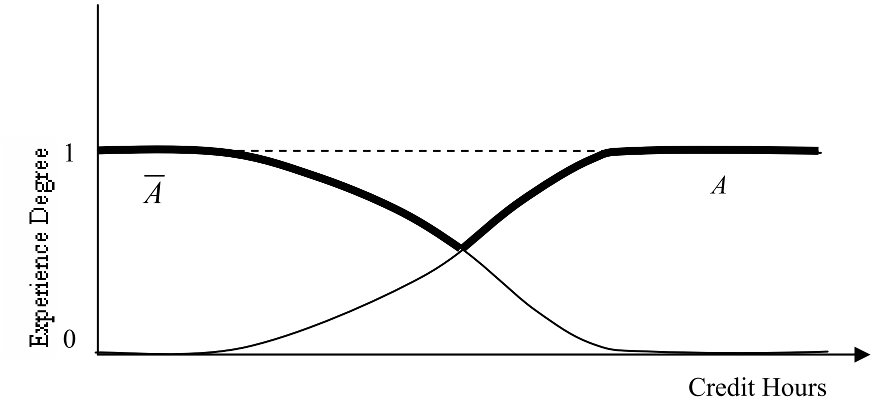


Figure (4 – 3): The Union of a Fuzzy Set and its Complement [Klir et al. 1997]

For the area between the bold curve and dashed line in Figure (4 – 3) to be in the union, it should be either added to the set of experienced students, which would be totally against any intuitive or plausible classification, or be part of the complement of the set *A*. But how could it be part of the complement, where the complement is defined and represented as above?

The Problem Specification and Justification

The author identifies, and specifies, the problem here as the erroneous definition in the literature of Fuzzy Set Theory of the concept of Fuzzy Complement. What is represented above as the complement of set *A*, according to the standard definition of Fuzzy Complement, is not really the complement. Rather, it is the *opposite* or, better called, the *inverse* of set *A*. To justify this statement, let us consider variations of the set $\overline{A}$ as given above such as the set of "very inexperienced students", say A_{ve}, and the set of "slightly inexperienced students", A_{sl}. Figure (4 – 4) visualizes the new situation with the added sets represented by the dotted curves. The dotted curve below $\overline{A}$ represents A_{ve} and the dotted curve above $\overline{A}$ represents A_{sl}.

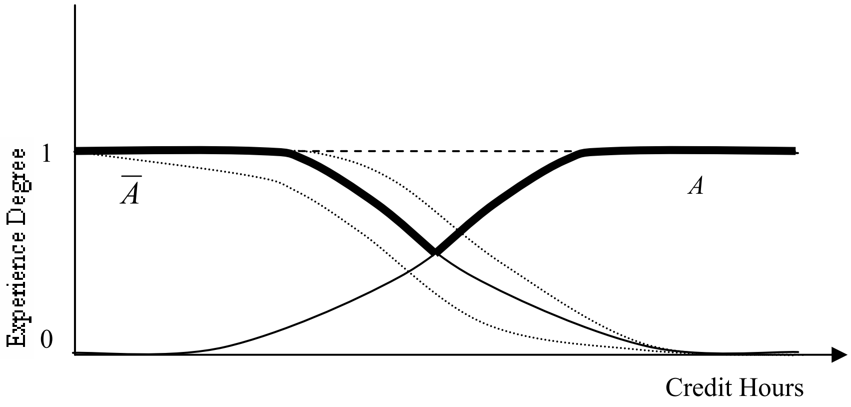


Figure (4 – 4): The Union of a Fuzzy Set and its Complement

Clearly the set of “very inexperienced students” is a subset of the set of “inexperienced students”, i.e., $A_{ve} \subset \overline{A}$. Therefore, we concentrate on A_{sl}, the set of “slightly inexperienced students”. Obviously, in Figure (4 – 4), A_{sl} covers part of the area in question between the bold curve and the dotted line. An important question to be raised here is: why are we not taking A_{sl} as the complement of *A*, or at least as part of that complement? It is clear here that all of the sets A_{sl}, $\overline{A}$, and A_{ve} fit the criteria of complementing *A*, and they all overlap with it with different degrees. Furthermore, if we take another set, say the set of “very slightly inexperienced students”, it will cover even more of the area between the bold curve and the dashed line in Figures (4 – 3) and (4 – 4) above. This shows that the complement, $\overline{A}$, as currently defined, and as illustrated in Figures (4 – 3) and (4 – 4), is just one set of a class of sets in which each set complements *A* with a different degree of complementarity. The union of all the sets in that class should be the total complement of *A*.

The Problem Solution and Proposed Model

In order to solve the problem presented above and fix the obvious deficiency in Fuzzy Set Theory, we need to introduce new conception to the theory and new elements to the mathematical model of Fuzzy Set Theory. First of all, we need to acknowledge that the identity relation between the Set Negation and Set Complement cannot hold when fuzziness is introduced to set theory. Hence, we need to differentiate between two different types of negation, that is, when the negation of a set means the set, or sets, characterized by opposing the characteristics of the set under negation, and when the negation of a set means every thing else except that set. The author of the dissertation calls the first type to be the set negation; the latter is the set complement. Furthermore, another distinction is needed between the notions of strict negation and ranged negation, as will be introduced and explained below. In the rest of this chapter, the author presents the proposed model for fixing this deficiency in Fuzzy Set Theory.

The author proposes the introduction of three different concepts to Fuzzy Set Theory, namely, *Fuzzy Complement*, *Fuzzy Negation*, and *Fuzzy Inverse*. The following notation can be used for the set *A*, the Fuzzy Negation of set *A*, the Fuzzy Complement of set *A*, and the Fuzzy Inverse of set *A*, respectively:

$$A, \neg A, \overline{A}, A^{-1}$$

The Fuzzy Inverse is defined as the currently known Fuzzy Complement, that is:

$$A^{-1}(x) = 1 - A(x)$$

Accordingly, the Fuzzy Inverse is the direct strict negation of a set, concept, or notion.

The Fuzzy Complement of set *A*, i.e., $\overline{A}$, is defined as follows:

$\overline{A}(x) = S(x) - A(x)$, where S is the Universal Set.

Set Subtraction is not commonly known as a typical operation on fuzzy sets. However, the definition of Set Subtraction of Crisp Sets is still valid and can be used with slight modification to accommodate the fuzziness and possible infinity as follows:

$$S - A(x) = S \cap [(\bigcup_{i=1}^{\infty} A_i) \mid A_i \neq A]$$

The Fuzzy Negation of a set A, i.e., $\neg A$, is the union of the sets with elements above and below the elements of A^{-1} with a pre-specified value, called β value (Beta value). The β value quantifies the range, or fuzziness, of the negation when it is not strict negation, i.e., inverse. Therefore, the Fuzzy Negation is defined as follows:

$\neg A = \int A_u(x) - A_l(x)dx$, where $A_u(x) = A^{-1}(x) + \beta$ and $A_l(x) = A^{-1}(x) - \beta$.

Alternatively, $\neg A = \bigcup_{i=-\beta}^{\beta} A_i \mid A^{-1}(x) - \beta \le A_i(x) \le A^{-1}(x) + \beta$

In words, this means that the Fuzzy Complement of a set *A* will consist of the intersection of the Universal Set with the union of all the sets in the set space, except the set *A*. Using the example of modeling the notion of “experienced student” presented above as set *A*, Figure (4 – 5) illustrates the proposed differentiation between $\overline{A}$ and A^{-1} by visualizing the graphical representation of the set of “experienced students”, the set of “inexperienced students”, and the set of “all students who may not fall in the set of experienced students”, i.e., sets *A*, A^{-1}, and $\overline{A}$, respectively.

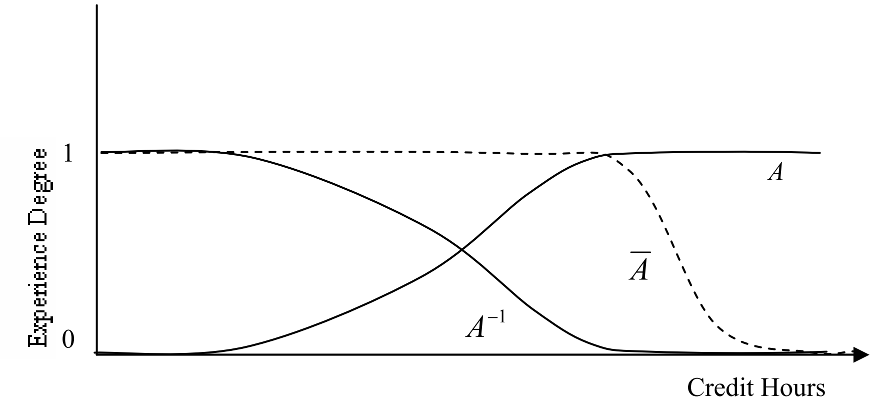


Figure (4 – 5): Fuzzy Complement and Fuzzy Inverse

The set $\overline{A}$ in Figure (4 –5) is computed using Standard Fuzzy Intersection. Figure (4 – 6) illustrates the resulting Fuzzy Union of A and $\overline{A}$, which does not violate the Law of Excluded Middle.

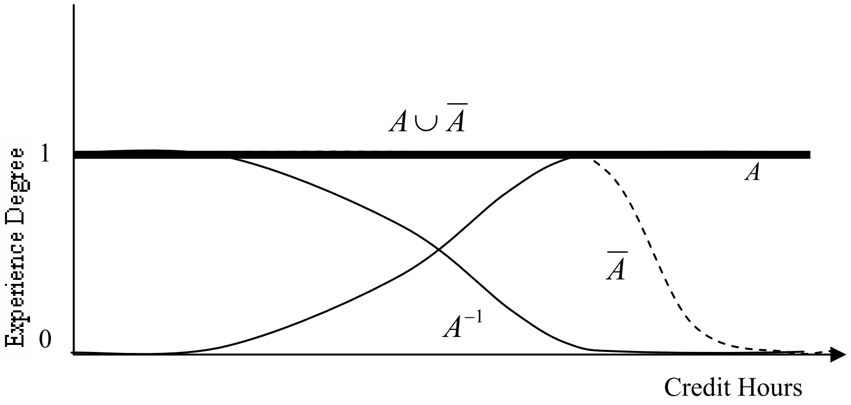


Figure (4 – 6): The Resulting Fuzzy Union

Regarding Fuzzy Negation, the definition is based on the intuitive expectation of the negation statements. The Fuzzy Inverse, introduces the strict negation, e.g., *experienced student* vs. *inexperienced student*, by application of the same example. We also have to acknowledge that both the concepts of *slightly inexperienced student* and *very inexperienced student* also negate the concept of experienced student with different degrees of negations. However, this applies only to a certain range of negation. For example, there is no point of talking about the level of experience of a student who just admitted but have not taken any classes yet. This would fall into the category of the Set

Complement, if the Universal Set were the set of all students who are admitted and have not yet graduated, for instance. The negation of set A in the example above would be the union of all the sets in the area between the dotted lines in Figure (4 – 4).

An important axiom of the proposed model is that the Set Inverse is a subset, that is a special case, of the Set Complement, which in turn is a subset of the Set Negation.

$$A^{-1} \subseteq \neg A \subseteq \overline{A}$$

In conclusion, the author hopes that the proposed model is natural, justifiable, and improves the computational power of Fuzzy Set Theory by computationally enabling the natural distinction between the set inverse, set negation, and the set complement.

CHAPTER FIVE

SOFT PROBABILITY

Probability Theory has been perceived to be, at the bottom, common sense reduced to computation. For historical reasons, probability theory has been concerned with the computation of the probability of discrete events on crisp sets. The probabilistic computing techniques were initially motivated by gambling problems. Those techniques usually assume crisp events with unlimited certainty about the event categorization. The computing is then performed to deal with the uncertainty associated with the random nature of the sought event. Today, the applications of probability theory pervaded into every field of our daily life. However, those fields do not always involve crisp sets with precisely measurable parameters; in fact they seldom do. Most applications of probability theory in our lives involve fuzzy events with linguistic descriptors. Nevertheless, the application of crisp techniques to fuzzy functions with uncertain parameters does not seem to be "common sense". When crisp techniques are applied to fuzzy sets, they result in misunderstanding, confused beliefs, and poor computational conclusions. Therefore, new probabilistic computing techniques are needed to cope with the fuzziness, complexity, and nonlinearity of real life.

In this chapter the author questions the limited assumptions upon which the probability theory was developed, the possible erroneous conclusions that may result from applying crisp techniques in fuzzy applications, and the need for developing an extended probability theory that is capable of processing fuzzy, as well as crisp, events. Following that, the author proposes and describes a model for probabilistic computing of fuzzy events. The scope of the research is not limited to compound events as previously done in the literature. The researcher actually tries to reform and remodel the theory of probability.

Introductory Background

The traditional tie between probability theory and gaming, especially gambling related games, is no coincidence. This relationship has historical basis. Girolamo Cardano (1501 – 1576) was a physician, a mathematician, and a gambler, in addition to practicing other fields too. His contribution to the science of mathematics included his novel systematic analysis of gambling problems [Cardano – Stoner 1931], [Ore 1956]. Blaise Pascal was another famous scientist who became interested in some dice problems in the seventeenth century when another gambler, the Chevalier de Mere, approached him about bad luck in gambling. Therefore, it is no surprise that we see probability texts always start with the a chapter on combinatorial analysis, where the basic, and generalized, principles of counting is introduced, and find them full of examples related to the flipping of a coin, the rolling of a die, the drawing of a card from a deck, the blind pick up of a ball from a pool, etc.

However, over the years, the theory of probability that was born out of the gaming rooms, and its applications, developed into an important branch of mathematics that pervaded into all branches of science and almost every aspect of our life, where the vast majority of events and studies involve fuzzy sets with overlapping, uncertain, parameters.

The famous French mathematician and astronomer Pierre Simon (1749--1827), Marquise de Laplace, who is considered to be the "Newton of France", described the theory of probability as "common sense reduced to calculations". This is probably the reason the probability theory found applications in almost every field of science, and daily life. Nevertheless, the birth of the theory of probability out of the gambling rooms affected, both, the scope and computational techniques of the theory.

We notice that the class of problems typically addressed in Probability textbooks involves random variables, where an estimate of the prediction that the variable will take certain value, or values, is the fundamental problem of consideration. In this class of problems, partial truth and varying degrees of truth are not of concern. For instance, when we roll a die and attempt to compute the probability that the outcome will be 2 or 3, $P(2 \vee 3)$, we do not consider the case when the die ends on an edge where both 2 and 3 partially face up. Rather, we assume a flat surface and a symmetric die with one number per throwing. We discard the repetition if any of these conditions is violated.

According to probability theory, in problems such as the above games, the probability of the event is calculated as the relative frequency of an event $E \in S$, where *S* is the sample space. In a different interpretation, the probability of the event is calculated as a measure of belief that the event will occur, or has occurred. Although some scholars may disagree, the author of the current dissertation believes, and claims, that the two interpretations are interrelated, as will be shown below.

The frequentistic interpretation of probability theory is defined as follows. We have a sample space S and a well defined region $R \subset S$. Then, we are trying to obtain the probability that a point $P \in S$ is in the region R. S, the sample space, represents the set of all possible values of a random variable, *R* represents the event whose probability is to be predicted, and *P* is the outcome of the system. The characteristic function of *R* determines whether the statement *"$P \in R$"* is true, i.e., truth-value of *"$P \in R$"* = 1, or false, i.e., truth-value of "$P \in R$" = 0. Hence, the function maps the set S to the set {0, 1} and the probability value describes the frequency of $"P \in R" = 1$ [Bonissone 1997]. Figure (5 – 1) depicts the situation.

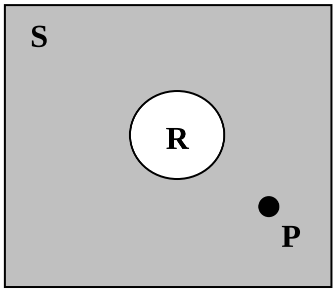


Figure (5 –1): $P \in R$ **or** $P \notin R$

The interpretation of probability theory as a measure of belief, though computable, implies the subjective belief in the likelihood of the event. Next, the author explains the interrelationship between the two major interpretations of probability theory, questions the limited assumptions upon which probability theory is developed and how they limit the domain of applications of the resulting theory, and presents the justification for the need for novel developments in the theory of probability to be more comprehensive and be more computationally feasible and powerful.

The Dual Interpretation

Let us consider the situation illustrated in Figure (5 – 1) and take event A to be $A = "P \in R"$. Let us also take the probability of event A to be, say, 90%, that is $P(A) = 0.9$. The frequentistic interpretation of this probability implies that if the experiment, or abstract situation, illustrated in Figure (5 – 1) is repeated 100 times under the same conditions, in 90 times the statement "$P \in R$" will be true, and if repeated 1000 times, A will be true in 900 times, and so forth.

According to the interpretation of $P(A) = 0.9$ as a measure of belief, we are 90% sure that P belongs in R. However, recall that R is NOT a fuzzy set. Hence, the uncertainty is not the result of how far P is from R, or of not knowing the exact boundaries of R. The uncertainty is the result of the fact that the event may have not occurred yet, or has occurred but we do not have a decisive evidence of whether $A = "P \in R"$ is true or false. Therefore, the value 0.9 indicates how strong our belief that $"P \in R"$ is true as a prediction of a future event, or based on a non-decisive evidence of a past, or present, event.

Now, the question to be asked is: if the boundary of R is well defined, which means that P can be either definitely in, or definitely out, of R, what does it mean to believe with a 90% certainty that P is in R? Another related question is: how can we test our belief for correctness?

It is intuitive that, when predicting a future event, the only way to test the correctness of our belief is to run the experiment. But, would it suffice to run the experiment once? It would if our belief was 100%, but recall that we assumed the belief of 90% with no partial membership of P in R. Thus, the only remaining way is to run the experiment a large number of times and record the ratio of $P \in R$ to the total number of occurrences of *P*. If the ratio converges to 0.9 as the number of repetitions grows larger, then our initial belief was correct, otherwise it is not. This is one way to establish the correlation between the two interpretations.

It is important to note here that the above discussion does not mean that both interpretations are the same or that the frequency of the event can determine with certainty the correctness of the belief. The main reason for this is that we do not know for sure what number is large enough for the number of repetitions. If the frequency did not converge to our belief measure, we may think it would if we continue to repeat more. On the other hand, if it did converge initially, this does not guarantee that the convergence will continue with more repetitions.

Moreover, probability, as a measure of belief, does not always involve future prediction. Let us consider the following statements for example: "*it is 90% probable that Shakespeare actually wrote Hamlet*" and "*the probability that Oswald acted alone in assassinating Kennedy is 0.8*" [Ross 1994]. Here, the individual making the statements is not predicting a future event. But we do not have decisive evidences that the statements are correct with absolute certainty either. Notice, again, that the uncertainty is not the result of fuzziness. Either Shakespeare did, or did not, write Hamlet, and Oswald did, or did not, act alone. Partial truth is not being considered here. For this type of application, the model and mathematics of Figure (5 – 1) can still be applied as follows. *S* is the sample space of all circumstances and evidences surrounding the event, *R* is the event in which the statement is made about, and *P* is evidence in support, or against, the statement. Then, rather than counting the number of times in which $"P \in R"$ is true in successive repetitions, we compute the sum of $"P \in R"$, which denotes the cumulative evidence in support of the belief. This shows that the two interpretations are interrelated since the calculation of the measure of belief is based, at bottom, on the ratio of the evidences in support of the belief to the pool of, or the opposing, evidences. Hence, the measure of belief indicates the willingness of a rational agent to bet on the event with no guarantee of even a partial success [DeFinetti 1937], [Bonissone 1997]. Therefore, probability, whether based on statistical frequency, subjective belief, or a combination of both, is concerned with predicting future events, in which no decisive evidence exists that they will happen, or past events, in which no decisive evidence exists that they have happened. As a consequence, taking a decision based on probabilistic reasoning will always involve the risk of complete failure, which is a fundamental difference between fuzzy control and probabilistic computing.

Important Questions

As stated above, for historical reasons, probabilistic techniques originated in problems associated with discrete mathematics and where all possible outcomes are equally likely. For example, when we flip a coin, the outcome is either head (H) or tail (T), which are both equally likely to begin with. When rolling a die, we assume rolling on a flat table where just one side of the die will end facing up and all sides are equally likely to begin with. Dealing with card decks is a similar problem with different numbers.

However, probabilistic techniques emerged out of the gaming rooms and pervaded into almost every field of science, including those whose possible states are not discrete and possible outcomes are not equally likely. For instance, what if we are considering the rolling and/or flipping of a non-symmetric object such as a rock or a body in the outer space like an asteroid or meteoroid? Another important example is the weather forecast, where the rain, the winds, and the clouds are not symmetric or discrete. These questions can be formulated in the following mathematical question: Revisiting

Figure (5 – 1), what if either, or both, of the sets *S* and *R* are fuzzy sets? Figure (5 – 2) illustrates the situation.

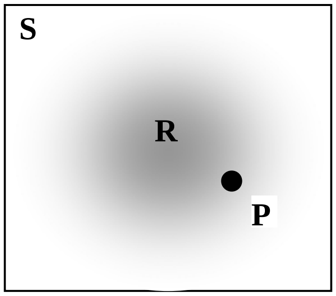


Figure (5 – 2): $P \in R$ ***is not certain or deterministic***

We have shown above that $P(A)$, the probability of event *A*, such that $A = "P \in R"$ can be computed by counting the frequency of $"P \in R"$ or computing the sum of $"P_i \in R"$ for all *i*, where P_i is an evidence in support of A. Both jobs are straightforward when *R* is a crisp set such as the case illustrated in Figure (5 – 1). But what if *R* is a fuzzy set, as in the case abstracted in Figure (5 – 2)?

Previous Attempts

In 1968, rather shortly after proposing his theory of fuzzy sets, Lotfi Zadeh published a paper on computing the probability of fuzzy events. He recalls the classical computation of $P(A)$, the probability of event *A*, as:

$$P(A) = E(\mu_A)$$

where A is the fuzzy event, μ_A denotes the characteristic function of A (0 or 1 in traditional probability), and $E(\mu_A)$ is the expectation of μ_A, which is computed as follows:

$$E(\mu_A) = \int_{R^n} \mu_A(x) dP = P(A)$$

In his computation, Dr. Zadeh replaced the Boolean characteristic function by the generalized function $\mu_A : R^n \rightarrow [0,1]$. However, we notice here that the integration is done over *dP*, which can be interpreted as:

$$P(A) = \sum_{x_i \in X} p(x_i)\mu_A(x_i)$$

where X is the sample space, x_i is the ith singleton in X, and $p(x_i)$ is the probability of the ith singleton. This implies that (1) A is a compound event, and (2) the computation assumes the pre-existence of $p(x_i)$. In words, the computation of the probability of the compound event A is done by summing the probabilities of its constituents, with each probability multiplied by the degree of the singleton event x_i in A [Zadeh 1968].

In a similar framework, Philippe Smets extended the theory of belief functions by defining the belief of a fuzzy event also by the assumption of the pre-existence of the probability mass function of each singleton in the set of evidences in support of event A [Smets 1981].

Both of the above papers presented a model for the computation of the probability of compound fuzzy events by integrating the composing simple events. Those singletons, supposedly, have pre-computed probabilities and membership functions in the compound event.

Indeed, two types of events exist in probability theory, namely, *simple* and *compound* events. Neither of the above papers did approach the fundamental problem of calculating the probability of the simple, atomic, fuzzy events. This problem was pointed out in a most recent publication by the author of the current dissertation. The paper laid out a set of questions and examples with the purpose of exposing the shortcomings of probability theory and its applied techniques when computing the probability of events with fuzzy, complex, parameters. The paper also set the criteria for a desired solution to the problem. The exposition of the shortcomings of the theory of probability was presented above under the section titled: Important Questions. The next section below provides the problem definition and mathematical formulation. The motivation for a new solution will then follow and the criteria for a successful solution will be listed under the success criteria section before the solution is presented at the end of the chapter [Moussa 2003].

Problem Definition and Formulation

Probability theory is concerned with the inference of information about the likelihood of a particular event from the available information about the overall population [Mendenhall 1968]. Hence, the basic mathematical formulation of a probability problem is that we have a sample space, *S*, which consists of the set of all sample points, P_i, representing the population. A region $R \subset S$ is considered and the problem, in words, is to try to compute the probability that a sample point P_i, which is known to be in *S*, is also in *R*. Mathematically, we are trying to calculate the probability of event *A*, *P(A)*, such that:

$$A = [(P_i \in S) \in R] \text{or } A = [P_i \in R \mid P_i \in S]$$

When the size of R is equal to the size of P_i, A is a simple event, and when the size of R is larger than the size of P_i, A is a compound event. The difference between crisp and fuzzy events is that the sets S and R are well defined with precise boundaries in the case of crisp events, but may not be so in the case of fuzzy events due to the lack, or absence, of precise specification, or the dependency on linguistic descriptors, with their intrinsic vagueness and subjectivity. Figures (5 – 1) and (5 – 2) above visualize the difference between crisp and fuzzy events, respectively.

Examples of crisp events as illustrated in Figure (5 – 1) may be, when rolling a die, the event A = "observe number 3" (simple), or A = "observe odd number" (compound). Examples of fuzzy events, visualized by Figure (5 – 2) within the same experiment, may be A = "observe odd numbers *significantly* more than even numbers", or A = "observe the average of the outcome is *approximately* 5".

From the latter examples above, obviously, what seem to be crisp experiments may still result in fuzzy events. Moreover, surprisingly, in science and in every day experience we are more concerned with the probability of fuzzy events than crisp events. For example, in science, we are more interested in the outcome of the motion of uncontrolled, non-symmetric, objects, or particles, the weather forecast, or the status of the economy, all fuzzy concepts with fuzzy outcomes. In every day life, we also are more often concerned about the probability that "it will be a *warm* day", "certain variable, say x, is *approximately* 19", or "certain path in town would have a *rather heavy* traffic *around* certain time".

The Motivation for a Solution

From the introduction and the review of the literature presented above we see that, for historical reasons, the basic probabilistic techniques are designed for crisp, mutually exclusive, events of regular simple systems. Even when Zadeh and Smets attempted to define the probability of fuzzy events, they built on the same techniques and proposed models for computing the probability of compound fuzzy events only. Nevertheless, applying those techniques to fuzzy systems built essentially on top of simple fuzzy events does not only produce results that are erroneous, but also misleading.

Let us, for example, take the weather forecast and concentrate on the rain prediction in particular. We are all familiar with statements such as, *"there is a 30/% chance of rain tomorrow"*. According to the relative frequency interpretation of probability theory, the statement means that if the current weather development occurred 100 times, in 30 of them rain would happen, and if 1000 times it would rain in 300 of them, and so forth. According to the interpretation of probability theory as a measure of belief based on evidence, the statement means that the cumulative evidence, in number and strength, that it will rain tomorrow to the total cumulative evidence about the weather forecast in general, and the rain in particular is 3 to 10. The result of this computation is that people repeatedly mistake the probability value for the heaviness of rain. Even those with scientific and mathematical background, who are supposed to understand the difference between the low probability and the amount, or heaviness, of rain, reason that if the probability that it may rain is low, then, we do not have strong conditions that would lead to heavy rain. Therefore, if it rains, it is unlikely that it would heavily. The reasoning sounds plausible, but we still frequently get heavy rain when the probability is low and get light rain when the probability is high. The real problem is in the application of probability computing techniques, which were designed for crisp events to a fuzzy, complex system with uncertain, and often subjective, parameters [Moussa 2003], [Moussa – Kohout 2003].

The Success Criteria

The above analysis, as well as the analysis of the problems of Quality of Service Networking that will follow in the remaining part of the dissertation, justify the need to revisit, revise, and develop the theory of probability by innovation or renovation. However, before presenting the proposed model, it is important to determine the set of criteria for a successful proposed model. The author proposes the following set in fulfillment of that objective:

1- Since probability theory has been proven, and successfully in use for crisp computation, then an important criterion of the success of any proposed model is to be reducible to the axioms of the current theory when fuzziness reduces to crisp sets, recalling that crisp sets are a special case of fuzzy sets.

2- The revision of probability theory should start at the base level starting from the counting principle of simple events. The counting and arithmetic of fuzzy events cannot be expected to be reliable if the fuzziness of the variables is ignored, and we have seen that the previous attempts of Zadeh and Smets did not have any contribution on that basic level. However, criterion one must be preserved.

3- The proposed model should be computationally feasible.

4- The proposed model should offer better probability estimation. The word “better” here means: more accurate and meaningful estimation. After all, the end goal is to reach the accurate computational model of “common sense”.

5- Since the computational gains obtained from the use of fuzzy set theory are highly dependent on the choice of the fuzzy membership function and the partitioning scheme, the desired probability theory should be sensitive to these two factors.

6- Although the probability of an event should not be confused with the fuzzy membership function of its variable parameters, but the proposed model should promote the intuitive correlation between the two. Criterion 6 can be expanded as follows:

> 6-a: It is intuitive that higher probability should be obtained from higher frequency, than from lower frequency, of comparable memberships in the event.
>
> 6-b: It is intuitive that higher probability should be obtained from higher membership values than from lower membership values of comparable frequencies; the higher the membership the stronger the positive-ness about the occurrence of the event.

7- The fundamental axiom of probability should always be preserved, that is: $0 \leq P(E) \leq 1$.

The Proposed Model

In this section, the author of the dissertation proposes a novel model for the computation of probability, which satisfies the above success criteria. The model proposal is split into two parts. As specified in criterion two, the proposed model starts, in the first part, at the base level by redefining the basic counting principle upon which the relative frequency of events is calculated, and the probability of an event is based. The second part deals with the problem of combining the cumulative probability of events in a fuzzy complex system, where the additive method is intractable and may be erroneous.

Counting and Frequency

The first step in any fuzzy and/or probabilistic computing task is the definition of the sets, classes, membership functions, and events of interest. Choice, design, or both can do this definition. Class and set partitioning and the design, or choice, of membership functions can change the computation and the outcome significantly. However, the author believes that this part has been sufficiently covered in the literature and is beyond the current treatment of the topic. This dissertation will not add to this phase but will attempt to exploit what has been done.

Our proposed model starts by the redefinition of $P(E)$, the probability of a simple event *E*. Classically, $P(E)$ is defined as follows:

$$P(E) = \lim_{\substack{n \to \infty \\ N \to \infty}} \frac{n(E)}{N} \qquad (5-1)$$

where $n(E)$ is the number of times E has occurred in N repetitions of the experiment. This is the classical definition as given textbooks, possibly with different notations [Ross 1994], [Mendenhall 1968], and many others. Clearly, Equation (5 – 1) does not account for any partial occurrence of *E*. The occurrence of *E* is a Boolean variable; either *E* occurred, increment *n* and *N*, or *E* did not occur, increment *N* only. The author proposes the following general formula to replace Equation (5 – 1).

$$P(E) = \lim_{\substack{i \to \infty \\ n \to \infty \\ N \to \infty}} \frac{\sum_i n_i \mu_{Ei}}{N} \qquad (5-2)$$

where n_i is the number of times E occurred with a membership degree of μ_{Ei} in N repetitions of the experiment.

According to Equation (5 – 2), the probability of event *E* is computed, as a relative frequency, by counting the number of times *E* occurred with each degree of membership, multiplying each counted number by its corresponding membership value, summing up the products, and divide the sum by the total number of repetitions. The limit of the outcome as the number of repetitions grows to infinity represents the overall probability of event *E*.

Now, let us study a concrete abstract example to compare the probability values computed using Equation (5 – 2) with those computed by Equation (5 – 1).

Suppose we have an experiment in which the outcome could be any positive real number. Let us suppose further that we are only interested in the first few numbers, which are partitioned into 4 classes, or sets. We take the class size to be 4 units. For simplicity, and without loss of generality, we assume that we ran the experiment, say, 23 times, and that the outcome was 3.25 five times, 3.8 six times, and 5 twelve times. Figure (5 – 3) illustrates the example with crisp tangent classes.

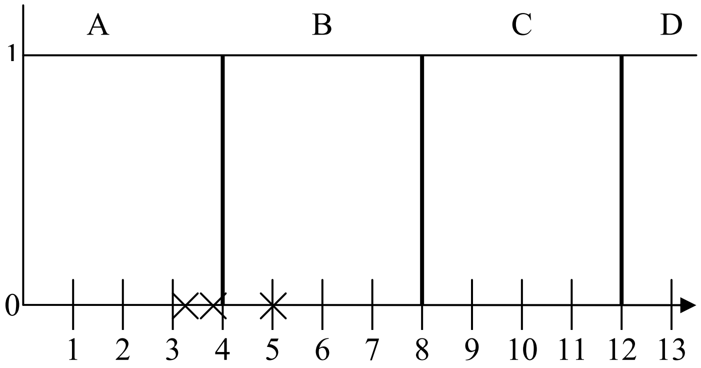


Figure (5 – 3): Computing probability with crisp tangent classes.

Now let us compute the probability of the outcome being in class B, $P(B)$, and the probability of the outcome being in A, $P(A)$.

Using Equation (5 – 1):

$$P(B) = \frac{12}{23} \approx 0.52 \text{, and } P(A) = \frac{11}{23} \approx 0.48$$

Using Equation (5 – 2):

$$P(B) = \frac{(5\times 0)+(6\times 0)+(12\times 1)}{23} = \frac{12}{23} \approx 0.52,$$

and $P(A) = \frac{(5\times 1)+(6\times 1)+(12\times 0)}{23} = \frac{11}{23} \approx 0.48$ (both (5 –1) and (5 – 2) yield the same value).

Now, we change the class partitioning into crisp overlapping classes using the same class sizes, as illustrated in Figure (5 – 4), and redo the computation.

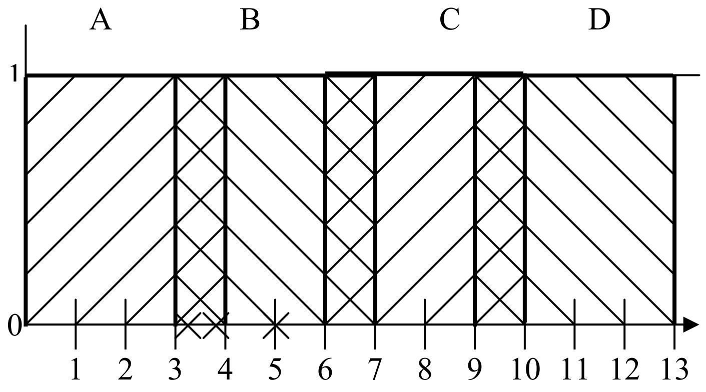


Figure (5 – 4): Computing probability with crisp overlapping classes.

Using Equation (5 – 1):

$P(B) = \frac{23}{23} = 1$, and $P(A) = \frac{11}{23} \approx 0.48$

Using (5 – 2):

$$P(B) = \frac{(5 \times 1) + (6 \times 1) + (12 \times 1)}{23} = \frac{23}{23} = 1,$$

and $P(A) = \frac{(5 \times 1) + (6 \times 1) + (12 \times 0)}{23} = \frac{11}{23} \approx 0.48$ (again, both Equations (5 – 1) and (5 – 2) yield the same probability values).

Now we use fuzzy overlapping classes and still use the same class sizes. We use the trapezoidal membership functions as in Figure (5 – 5).

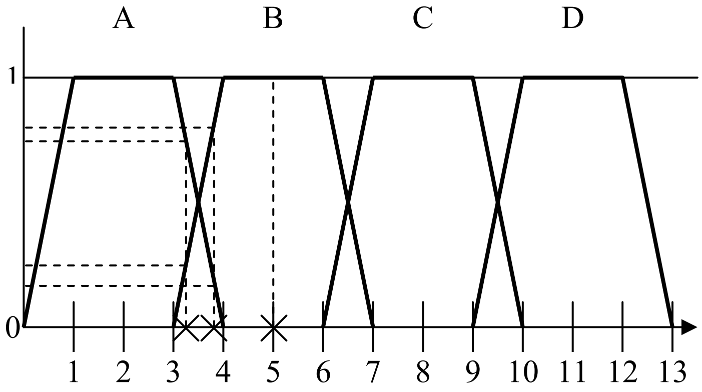


Figure (5 – 5): Computing probability with fuzzy overlapping classes.

Now we compute the probabilities of the same events, *A* and *B*. If we try to do the computation using Equation (5 – 1), we have to, first, make a decision on what counts as an occurrence of the event and what does not. For instance, there is no question about the outcome of 5 being in class *B*. But, when the outcome of the system is 3.25, the membership function specifies this as being in class *A* with 75% certainty and in class *B* with 25% certainty. We obviously do not have a mechanism of defining what constitutes an occurrence of the system output in a class according to Equation (5 – 1) and, consequently, we cannot perform the computation of the probability of an event that is not defined. However, using Equation (5 – 2), the computation is straightforward.

Using Equation (5 – 2):

$$P(B) = \frac{(5 \times 0.25) + (6 \times 0.8) + (12 \times 1)}{23}$$

$$= \frac{1.25 + 4.8 + 12}{23} = \frac{18.05}{23} \approx 0.785$$

and

$$P(A) = \frac{(5 \times 0.75) + (6 \times 0.18) + (12 \times 0)}{23}$$

$$= \frac{3.75 + 1.08 + 0}{23} = \frac{4.83}{23} = 0.21$$

By studying the example above, we see that the proposed model meets the seven criteria preset for a successful model in the previous section. When Equation (5 – 2) was used to compute the probabilities of crisp classes, it yielded the same probabilities as Equation (5 – 1) for both the tangent and overlapping classes; this satisfies criterion one. Equation (5 – 2) computes the expectation of *simple* events by counting both the frequency and membership function of each event, in accordance with criterion two. Criterion three is satisfied since the computation, although more costly than in Equation (5 – 1), but is still feasible; no intractability is expected. The computation also satisfies criterion four in several ways: (1) it provides a model for the computation of the probability, which would be otherwise incomputable. Most modern science and real-life systems are fuzzy, complex and nonlinear, which cannot be modeled by crisp partitioning. Computing the probability of such systems using Equation (5 – 1) would yield results that are erroneous, meaningless, or misleading. For example, if we use Equation (5 – 1) as we did in the case of crisp overlapping classes, ignoring the fuzziness, $P(B)$ would be ONE, which is erroneous, meaningless, and against the intuitive hypotheses of criterion six. From the computation we also see that the results are highly dependant on the membership function and the partitioning scheme as set in criterion five. Finally, it is clear that the result will always be proportional to n_i and μ_{Ei}, while the computable probability will always satisfy the fundamental axiom of probability, that is $0 \le P(E) \le 1$, which satisfies criterions six and seven.

Cumulative Probability

It should be understood that neither of the limiting formulas specified by Equation (5 – 1) or Equation (5 – 2) is guaranteed to hold or yield accurate results at every experiment. After all, probability theory is about uncertainty, not determinism. Hence, we do not have a way of guaranteeing that if an experiment is repeated a large number of times, the outcome will converge to a limiting constant. Furthermore, if it does, we would not know if it would converge again if we re-carry the sequence another time, nor do we know that the convergence will continue if we continue to repeat. The limiting value specified by Equation (5 – 1) in classical probability is an axiom that goes with the natural intuition. Equation (5 – 2) was built on the same assumption with the goal of enabling more accurate, and covering wider range of, computation.

To support the intuitive, but complex, assumption that the above limiting value really exists, the modern axiomatic approach to Probability Theory is to assume a set of simpler and self-evident axioms, then, attempt to prove that the sought constant exists in some sense, or approximation.

Taking $P(E)$ to denote the probability of event *E*, and *S* to denote the sample space, the three axioms of Classical Probability are:

Axiom 1: $0 \le P(E) \le 1$

Axiom 2: $P(S) = 1$

Axiom 3: $P(\bigcup_{i=1}^{\infty} E_i = \sum_{i=1}^{\infty} P(E_i)$, where events E_i must be all mutually exclusive, that is: $\forall i \forall j(i \neq j \Rightarrow E_i \cap E_j = \phi)$.

Axioms 1 and 2 above should remain untouchables. Any change to these two axioms would result in a meaningless probability theory. However, the author of the dissertation questions Axiom 3, which provides the method for calculating cumulative probabilities of multiple events.

The author of the dissertation contends that Axiom 3 suffers from two serious drawbacks, which become compelling when dealing with complex and/or fuzzy systems. The first problem with Axiom 3 is that it assumes the additive synthesis of probabilities. Accordingly, for two events, say E_1 and E_2, with probabilities $P(E_1)$ and $P(E_2)$, respective, we compute the combined probability of the union of the two events by, simply, linearly adding up the two probabilities. Although this assumption is intuitively appealing when dealing with simple systems, it does not seem to be a priori. There are many complex systems where the linear additive synthesis model fails when the system grows larger than few, possibly only two, elements. Systems of particle physics are one such class of systems.

The second drawback, which aggravates the first significantly when dealing with complex systems, is the mutual exclusion condition. To additively synthesize the probability of a union of events, those events must be mutually exclusive. In fuzzy systems with overlapping membership functions, we may never find any mutual exclusive events. Even in crisp systems, when the complexity of the system grows to a certain degree, the extraction of mutual exclusion may be become an intractable task.

The mutual exclusion condition is required to guarantee that the value of one will be an upper limit on any computed probability, in accordance with Axioms 1 and 2. Otherwise, some probabilities may be added more than once, which results in the total probability to be more than one. Therefore, as a consequence of Axiom 3, if events

$E_1$and $E_2$are not mutually exclusive, that is, $E_1 \cap E_2 \neq \phi$, the probability of the intersection of the two events must be subtracted from their additive union as follows:

$$P(E_1 \cup E_2) = P(E_1) + P(E_2) - P(E_1 \cap E_2) \quad (5-3)$$

When the number of events is generalized to *n*, rather than 2, the computation of the probability of the union of the *n* events becomes as follows:

$$\begin{aligned} P(E_1 \cup E_2 \cup \ldots \cup E_n) = \sum_{i=1}^{n} P(E_i) - \sum_{i_1 < i_2} P(E_{i_1} \cap E_{i_2}) + \ldots \\ + (-1)^{r+1} \sum_{i_1 < i_2 < \ldots < i_r} P(E_{i_1} \cap E_{i_2} \cap \ldots \cap E_{i_r}) \qquad (5-4) \\ + \ldots + (-1)^{n+1} P(E_1 \cap E_2 \cap \ldots \cap E_n) \end{aligned}$$

In Equation (5 – 4), the summation $\sum_{i_1 < i_2 < \ldots < i_r} P(E_{i_1} \cap E_{i_2} \cap \ldots \cap E_{i_r})$ is taken over all of the $\binom{n}{r}$possible subsets of size *r* of the set {1, 2, …, n}, where $\binom{n}{r} = \frac{n!}{(n-r)!r!}$ [Ross 1994].

The above computation basically states that to compute the probability of the union of n events, we need to compute the sum of the probabilities of all the events taken one at a time, minus the sum taken two at a time, plus the sum taken three at a time, and so forth.

Combinations of three types of systems, which unfortunately are common in many sciences and fields of study, can make the above computation very impractical, if possible at all. The first type is a large-scale complex system. It is clear that in such a case the computation can be very costly, possibly intractable. The second type is a fuzzy system, where we may never have any mutually exclusive elements. The third type is an infinite system. There are cases where we study a system where the state space is not well known, or uncountably infinite. In such a system, we compute the probabilities of the measurable elements, or events. When new elements, or events, occur, emerge, or are discovered, according to the above method, the entire system must be recomputed.

After computing the probabilities of the individual events using the alternate method proposed in this dissertation, the author of the current dissertation proposes an alternate method for computing the cumulative probability of the union of multiple events as follows:

$$P(\bigcup_{i=1}^{\infty} E_i) = \left[P(E_1)^{\prod_{i=2}^{\infty} P(E_i)} \right]^c \qquad (5-5)$$

where $P(E_1) \geq P(E_i)$ for all $2 \leq i \leq \infty$, and c is a *completeness uncertainty factor*.

Instead of summing up the probabilities of the events, we take the highest probability to be the base probability and raise it to the power of the product of all other available probabilities. If a new probability becomes available, we simply raise the current cumulative probability to the power of the new one, if the new one is smaller than $P(E_1)$. Otherwise let $P(E_1)$ be the new probability and recomputed the cumulative probability. Finally, the *completeness uncertainty factor* is optionally used as the last power. When the completeness uncertainty factor, *c*, is equal to zero, we are certain that the elements calculated represent the complete sample space, i.e., the probability computed is $P(S)$, and *c* = 0 rounds it up to 1. When *c* is equal to one, we are certain that the elements calculated represent their portion of the sample space, and *c* = 1 ensures the computed probability is unmodified. In between the two cases, different values can be used for fine tuning depending on the type of the system and the degree of uncertainty.

The author contends the proposed computation is advantageous over the additive technique for the following reason:

1- The proposed computation is an alternative that is also intuitively natural and yields comparable results when the system is crisp and simple.

2- The computation is tolerant of the mutual exclusion condition. This feature makes it more suitable for fuzzy and complex systems. When we recall that it also returns comparable results with simple and crisp systems, it supersedes Axiom 3 of traditional probability.

3- The computation is fault tolerant in general. Errors in the computation of mutual exclusion in the additive model can cause the whole probability computation to fail if some probability is added more than once. This failure will never happen when using Equation (5 – 5). The worst that can happen if some probability value is counted more than once is a slight change in the outcome. The worst thing that can happen with the proposed computation is when the base probability is not the largest, and even then, the probability computation may deviate from the intended one but will not collapse and run out of bounds.

4- Since a probability value will always be between zero and one, in accordance with Axiom 1, it can be represented, or approximated by a rational number. Consequently, the cumulative probability computed by the proposed model is guaranteed to comply with Axioms 1 and 2, by the theory of mathematical analysis, which states that: $\mathit{if}\ x > 0, \lim_{n\to\infty} \sqrt[n]{x} = 1$ [Rudin 1964].

5- Looking at Equations (5 – 4) and (5 – 5), it is clear that the proposed model is much more computationally feasible and not exposed to intractability, despite the fact that it yields comparable results in the simple cases and the fact that it is fault tolerant, which makes it more automatable and programmable.

6- The guarantee of being within the meaningful bounds, in addition to its fault tolerance, makes it ideal for infinite and uncertain systems.

7- The function used in the computation is a strictly increasing function, which minimizes the deficiency that may result from unexpected emergence of new components in the case of a large-scale, complex, dynamical, or infinite system.

8- Finally, and most importantly, the proposed computation satisfies all of the success criteria specified above.

The author of the dissertation believes, and hopes, that the proposed Soft Probability Theory would significantly improve the computational power of probabilistic computing. The application example presented above is just one of a large class of complex, fuzzy, nonlinear, dynamical, and large-scale, possibly infinite, systems with important applications in physical and chemical sciences, economics, meteorology, distributed computing, just to name a few. Chapter Eight will present a model for applying fuzzy probabilistic computing in the area of Quality of Service Networking.

CHAPTER SIX

USING BK-PRODUCTS OF FUZZY RELATIONS IN QUALITY OF SERVICE ADAPTIVE COMMUNICATION

The term Quality of Service (QoS) is a fuzzy term. When applied to data communication, the measures of the QoS are also fuzzy linguistic measures. Hence, fuzzy logic ought to be exploited as a tool for modeling the communication system and solving its problems. In this chapter the author of the dissertation provides a foundation and overview of the possible utilization of Fuzzy Logic in the area of QoS data communication. The chapter is based on a published paper under the same title [Moussa and Kohout 2001]. The chapter starts by an introduction to the subject, which provides justification of the need for Fuzzy Logic to be utilized in this area. This will be followed this by a review of the previous approaches for solving the problems of QoS data routing in general and the problems associated with the inaccuracy in the network state information in particular. The author follows this by presenting the new proposed approach, which provides the foundation for building a new paradigm for modeling and solving QoS computation problems. Finally, the chapter concludes by pointing out several new avenues for research and list some open problems that will be dealt with in the chapters that follow.

Introduction

The term Quality of Service (QoS) has been introduced to the field of data routing/communication in computer science only in the past few years. The concept of QoS has been proposed to capture the qualitatively and/or quantitatively defined performance contract between the service provider and the user applications [Chen and Nahrstedt 1998]. These qualitative and quantitative metrics may not be of importance in some applications but may be critical to others. For example, when we FTP a file, quality of service data transmission is not so important; the issue is just whether the file transfer is successfully completed or not. On the other hand, transferring or broadcasting a transaction in a distributed database system may require a minimum speed and a certain order to be considered successful. Furthermore, when streaming video or implementing video/audio real-time communication, not only the speed but also the synchronization of visual and audio data is essential to the success of data communication

Therefore, QoS must be considered when building any communication system in order for that system to meet the requirements of an integrated service network that carry heterogeneous data traffic. The QoS requirement of a connection is usually given as a set of constraints. It can be *Link* constraints, *Path* constraints, *Tree* constraints, or any other pre-defined constraints. Dealing with such QoS requirements and the resulting problems is a multi-disciplinary field. A variety of problems need to be dealt with by scientists from diverse areas, such as physics, hardware engineering, and computer science. In physics, research has been running on discovering new media through which data can be transmitted as well as trying to enhance the use of the currently available media. Engineers have been working on using the currently available physics to design the best possible network in terms of speed and reliability. Meanwhile, computer scientists have been developing algorithms and architectures that optimally utilize the currently available technology.

Speaking of the computer science solutions to the QoS networking, the author of the current dissertation noticed the absence of utilizing fuzzy logic and soft computing as tools for developing solutions to QoS problems. We cannot conclusively say that no work has been done in that field, but upon the review of many research papers in the area of QoS, none in this direction has been found, except for recent proposals for using typical Fuzzy Controllers to control job admissions.

Yet, the field of QoS is a natural area for applying fuzzy systems and soft computing techniques. The following points provide the justification for this claim:

1- The term QoS itself is a fuzzy term. We do not have a precise generalized specification of the term QoS.

2- We usually describe the quality of service, especially the soft type, as: acceptable, not acceptable, good, bad, etc., which are obviously all fuzzy linguistic terms, the type of measures Fuzzy Logic was invented to deal with.

3- A number of research papers have been published about problems such as dealing with imprecise state information [Chen and Nahrstedt 1998] and inaccurate information [Guérin and Orda 1997]. Those problems are predominantly subtle and important problems in large-scale network communications and are inherently fuzzy logic problems.

Previous approaches

QoS data communication and its metrics have always been specified in the literature as a set (or sets) of constraints. Even though the QoS metrics have been described in the fuzzy linguistic terms mentioned above, they have not been modeled as such. The quality of service and its metrics have only been modeled as constraints that can be satisfied or dissatisfied. However, some attempts have been done to deal with the imprecise and inaccurate state information when taking the routing decisions. Guérin and Orda [Guérin and Orda 1997] published a paper on QoS-based routing in networks with inaccurate information. Their approach was based on the probability distribution functions. To deal with the inaccuracy, for each link, the source node only knows the quantities *pl*(*x*), where *pl*(*x*) is the probability that link *l* can accommodate a connection which requires *x* units of bandwidth. Similarly, the source node maintains, for every link *l*, the quantities *pl*(*d*) where *pl*(*d*) is the probability that link *l* have a delay of *d* units, where *d* ranges from zero to maximum possible value. The goal is to maximize the probability of finding a feasible path. However, they did not address the problem of how to maintain the probability distribution.

Another research by Chen and Nahrstedt [Chen and Nahrstedt 1998] was published on distributed QoS routing with imprecise state information. They developed a multi-path distributed routing scheme, called *ticket-based probing*. They claim that their algorithm tolerates information imprecision by searching multiple paths in parallel, increasing the chance of finding a feasible path. We believe that they developed a good scheme, which considers the optimality of the selected path as well as the QoS requirement. However, we do not believe that searching multiple paths in parallel is a proper solution to the problem of imprecise state information. The imprecision will still be there, affecting all the paths being searched in parallel.

The Proposed Approach

As explained above, the term QoS is a fuzzy term and its metrics are determined in terms of linguistic fuzzy measures and are often subjective. Hence, the author believes that Fuzzy Set Theory, and its consequent Fuzzy Relations and BK-products, is an ideal tool for dealing with this type of computation. As stated by Klir: *"An important feature of fuzzy set theory is its capability to capture the vagueness of linguistic terms in statements of natural language"* in [Klir 1995]. Consequently, the author proposes a new paradigm for modeling and solving the QoS computation problems by utilizing Fuzzy Logic.

In the proposed model, a QoS data communication system is modeled as a system whose variables range over states. Those states are represented with Fuzzy Numbers. These fuzzy numbers may represent the linguistic variables that provide the QoS metrics, such as very poor, poor, acceptable, good, very good, excellent, and so on. Each of those linguistic variables is defined in terms of a base variable whose values are real numbers within a specific range. As specified by Klir, each linguistic variable consists of [Klir 1995]:

- A name to capture the meaning of the base variable involved.
- A base variable with its range of values (an interval of real numbers).
- A set of linguistic terms that refer to values (or states) of the base variable.
- A semantic rule, which assigns to each linguistic term its meaning.

In this context, our model is exemplified in Figure (6 - 1). The name of the linguistic variable is “QoS”. The base variable whose meaning is captured by QoS is ranged from 0 to 100 as a percentage of being satisfactory. There may be a value, possibly below 100, in which all communication requirements are satisfied. This value would obviously be system and problem dependent. In our example, the value is 80. The linguistic values (states) of the linguistic variable are *very poor*, *poor*, *acceptable*, *good*, and *very good*. Each of these linguistic terms is assigned one of the trapezoidal-shape fuzzy numbers by specified semantic rules.

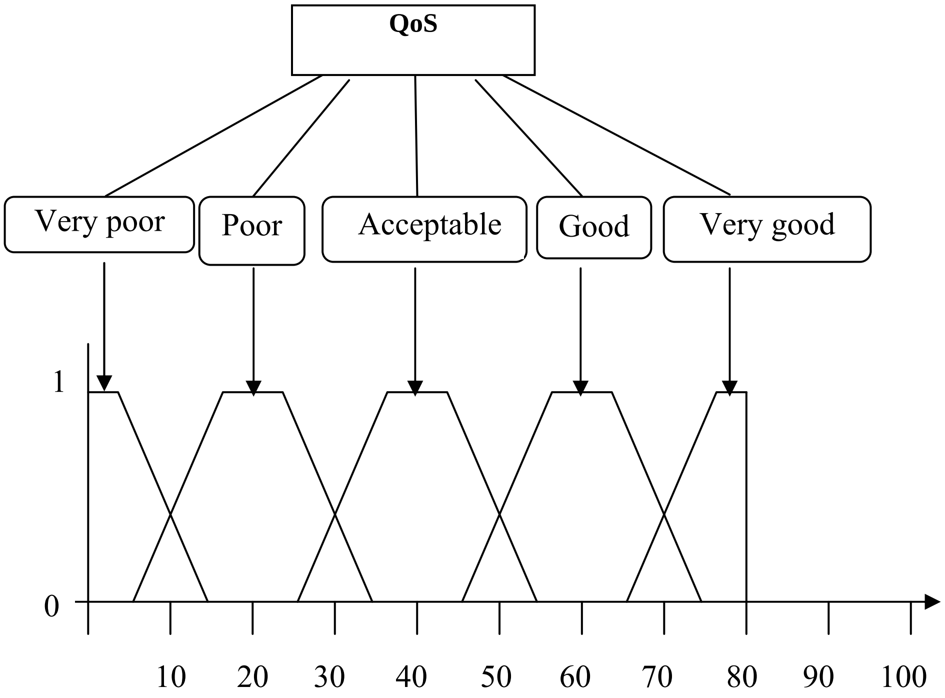


FIGURE (6-1): An example of our system model

In this regard, it is important to note that the fuzzy number functions do not have to be trapezoidal. In fact, the author of the dissertation believes that this could be one of the open research areas in the utilization of Fuzzy Logic in QoS networking. It could also be application dependent. However, the trapezoidal function is used here because of its popularity in Fuzzy Computing due to its efficiency and reduced computational cost. Moreover, the power and flexibility of the trapezoidal function will be exploited later in Chapters Seven and Eight when designing the network state updating policy and the routing algorithm.

The main system modeled above is an aggregation of several subsystems that can be modeled the same way. For example, the delay requirement, which is part of the QoS requirements, can be specified as in Figure (6 - 2). The name of the linguistic variable is “The Delay”. The linguistic values (states) of the linguistic variable are *very good*, *good*, *acceptable*, *bad*, and *not acceptable*. The fuzzy number represents the degree of truth in the statement. The delay is *variable* (where the *variable* in this example can be *very good*, *good*, *acceptable*, *bad*, and *not acceptable*, respectively). Therefore, in the example of Figure (6 – 2), a delay of 8 milliseconds is considered to be 75% very good, 25 % good, and 0% otherwise. A delay of 20 milliseconds is 100% good and 0% otherwise.

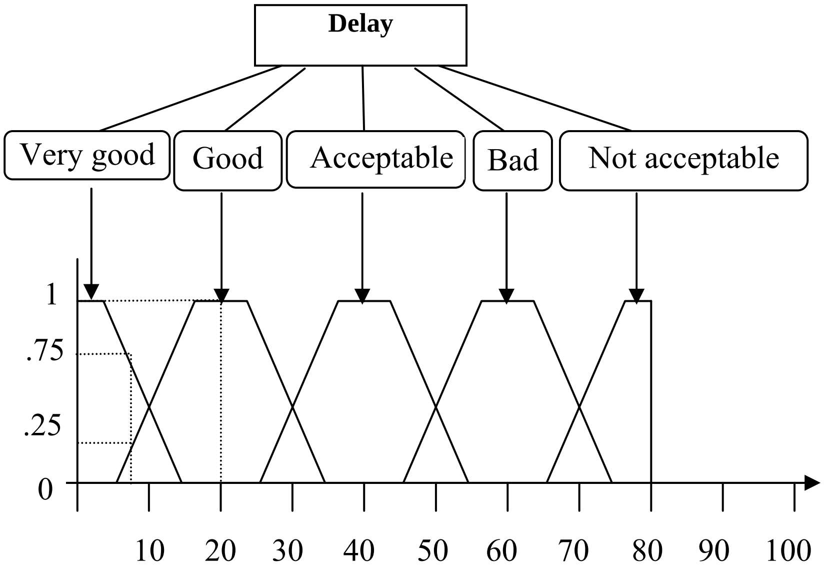


FIGURE (6-2): An example of our system model

The same technique (possibly with different fuzzy functions) can be used to model other components of the system such as the bandwidth, the cost, etc. When the various components have been modeled and the corresponding computations have been done, the system can be aggregated using the BK-products [Kohout 2001], Kohout and Bandler 1992] to compute the QoS available.

Let us now describe a concrete example of the computation of a routing selection decision using BK-products to aggregate the system. For simplicity, and without loss of generality, we assume the above scheme has been used to model delay, bandwidth, and cost. Furthermore, we assume that each parameter has been computed for an entire path and only three paths are available. Let the three paths be denoted I, J, and K. The fuzzy sets specifying the delay (D), bandwidth (B), and cost (C) will be defined by the system designer. The level of the service is defined by the contract, for this example we assume the service is required to be *good*. The membership grades of each path in the three sets express the degree of the path being able to provide the service with the level *good*. We encode all the available network information in the following relation R.

Path	Delay	Bandwidth	Cost	Degree of good
I	**.58**	**.3**	**.68**	**.30**
J	**.9**	**.89**	**.93**	**.89**
K	**.97**	**.15**	**.43**	**.15**

To compute the routing decision in this case, we aggregate the system by computing the relation $(R \circ R)_{xz} = \vee_y (R_{xy} \wedge R_{yz})$. The fuzzy sets intersection is needed to guarantee the minimum satisfactory system parameter. Then, the path with highest degree of overall performance is the one to be selected, in this case path J. In a 2-way synchronized traffic that requires two paths with equivalent states, the square product ($R \square R$) would be needed.

In some cases, aggregation may not be even needed, if only one QoS requirement needed to be satisfied. For example, if certain application is only bandwidth requirement dependent, then we can only model the bandwidth and compute the QoS as the outcome of the bandwidth computation without further aggregation of other factors. However, if a path change needs to be computed then the triangle products $(R \triangleright R)$ and $(R \triangleleft R)$ of the two paths must be computed in order to guarantee the inclusion of the parameters of the replaced path in the newly selected one. The definitions of all the BK-products were introduced in Chapter One under the Fuzzy Relations section. Finally, when certain numerical guarantees are required, the computation can be done on the α-cut at the required value.

Justification of Validity

The author believes that adapting this paradigm will be an ideal approach for the solution of the problem of QoS computation for the following reasons:

1. Representing the states of the variables by fuzzy numbers provides good and realistic quantization. This quantization is capable of capturing the limited resolution of measuring the system features. In addition, it also captures the error and imprecision in the network state information by avoiding the

computation on single values or intervals with sharp divisions between them, which are known to be unrealistic.

2. It is very difficult, if not impossible, to maintain and update a precise global network state like most routing algorithms require [Chen and Nahrstedt 1998], [Ma and Steenkiste 1997], [Salama *et al.* 1997], [Wang and Crowcroft 1996]. It will be shown in the next chapter that the global state will always be imprecise in a dynamic network, and will even be worse in large wide-area networks [Chen and Nahrstedt 1998]. The sources of imprecision can be one or more of the following: propagation delay of state messages, periodic updates (more updates means more overhead), hierarchical state aggregation, reliability of nodes, and imprecision in the links. This shows that the imprecision will be always present and our best bet is to live with it. If the imprecision will always be there, why then do the computation assuming precise state information?

3. Other than the forced quantization due to the lack of accurate information, and since the level of accuracy will always be limited, we can utilize the opted quantization, which is desirable when lower precision is sufficient for a given task [Klir and Yuan 1995], [Klir *et al.* 1997]. Therefore, performing the computation on the quantized variable states significantly reduces the computation demands, which in turn frees the system for more optimized solutions. The resulting drop in computational cost is badly needed in the field of QoS networking. For example, finding a feasible path with two independent path constraints is NP-complete [Garey and Johnosn 1979]. Other examples are the problem of finding the least-cost path with bounded delay and the problem of finding a path with both bounded delay and bounded delay jitter. These are examples of two classes of problems, namely, Path-Constrained Path-Optimization routing (PCPO) and Multi-Path Constrained routing (MPC), respectively. The entire two classes are NP-complete [Chen

and Nahrstedt 1998]. This shows how badly reducing the computational cost is needed here. The design of the proposed paradigm is aimed at the reduction of the complexity of some of these problems. However, even if the level of complexity remains the same, the use of quantized states of variables will certainly reduce the computation, which spares more computational power for dealing with more complex problems.

4. When specifying the QoS requirements and metrics, users always define the service in linguistic measures such as good, bad, … etc. Translating these linguistic measures into precise numbers without fuzzification may unnecessarily reduce the system effectiveness and waste considerable computations. For example, suppose we specify one QoS constraint such as the delay to be, say 9 microseconds. If the system performs the necessary computations and reaches the conclusion that the best path available has a delay of 9.01 microseconds, it is likely that the system will reject the connection even though the 9.01 microseconds could be satisfactory. The problem is that the constraint had been precisely specified as a smaller value and the model used does not make room for the jitter. These jitters can be captured and dealt with by our paradigm.

5. In addition to the previous point, if a precise, non-negotiable, value is ever required as a constraint, the proposed model can still deal with it by taking the α-cut at this value, returning to the crisp relation's computation. Hence, we propose a more flexible paradigm that could deal with different degrees of satisfaction of requirements, which can still be turned to the known crisp model whenever needed. This makes the proposed paradigm a superset of the currently known one.

After all, a main contribution of this approach is the substantial computational savings that are achieved by developing equivalence classes of QoS measures [Bandler and Kohout 1998]. Therefore, the computation can be done on equivalence classes rather than individual values. Let us take *delay* as an example. If the delay is considered to be good when it is between 5 and 9 nanoseconds, then all the values falling between these two values can be dealt with as an equivalent class. Changes within this range should not result in the re-computation of information for the whole system. Hence, computation can be done only when one QoS measure changes status from equivalence class to another. So, by employing fuzzy equivalence classes we reduce the cardinality of the ordered set, which provides the scale for measurement of delay, hence computational saving is achieved. In the examples of Figures (6 – 1) and (6 – 2) above, we only have five overlapping equivalence classes to deal with. This chapter is more of an overview of the proposed paradigm. Detailed algorithms and applications to maintaining the network state information and computing the routing decisions will be introduced in the following two chapters.

Fuzzy Control

Beyond the scope of the classes of problems and points of strength presented above, other research has been done on designing new architectures for QoS-adaptive communication such as the one presented by Abdelzaher and Shin in [Abdelzaher and Shin 1998]. In their paper, Abdelzaher and Shin present a new communication subsystem architecture for QoS-adaptive real-time applications. The main contribution of that architecture is its dynamic QoS-optimization mechanism and support for flexible QoS contracts. Those QoS contracts between the communication client and server specify multiple levels of QoS with acceptable degradation mechanism. Under overload, the communication subsystem degrades connections predictably according to their respective contracts. The contracts also associate rewards with each level. The communication subsystem consists of three basic modules: Load Control, Protocol Processing, and System monitoring. This system was reviewed in more detail in Chapter Three, the literature review.

The author of this dissertation proposes the use of Fuzzy Control to enhance this system. Fuzzy control has been the most successful area of applications of Fuzzy Set Theory by far [Klir *et al.* 1997] and the proposed fuzzy controller will have several advantages:

1- The use of a fuzzy controller should be ideal for dealing with subjective and objective QoS levels and rewards.

2- The multi-level control implemented by the *load control* module requires smooth and controlled transitions from one level to another. The capability of fuzzy sets to capture gradual transitions from one set to another, and even from a set to its own complement, considerably enhances their expressive power in comparison with crisp sets. Therefore, fuzzy set theory can effectively deal with this class of problems.

3- The dynamic optimization using the *monitoring and feedback* module incurs three types of difficulties: additional complexity and computational costs, the previously mentioned smooth transition between states, and the need for a bridge between the mathematical model and the associated physical reality. These are three areas that Klir specifically addressed in [Klir] when discussing the capabilities of fuzzy set theory and fuzzy logic.

4- The proposed fuzzy controller is not designed to only handle the degradation of QoS levels. Instead, it is designed to also upgrade the levels whenever possible. Hence, maintaining optimal performance as will be shown by the concrete example below.

Here, the author proposes a fuzzy network controller that does not only take the current network state into account, but also generates and maintains a curve representing the history of the difference between the target and current states of the underlying network. Therefore, by computing the derivative of the curve at some point, we can predict system changes, overloads, and under-utilization before they occur. This results in a continuously optimized network. Here we present a basic network fuzzy control system for resource management. The system uses the network parameters recorded periodically and on demand to generate a curve for the corresponding parameter, the bandwidth for example. The data is subtracted from the target to determine the error. Then the derivative with respect to time can be computed in order to determine the slope (rate of change) of the error. The fuzzy controller' outputs are used to start the process of initiating further resource reservation if the fuzzy engine decides that overload is present (or projected) or start the process of releasing some resources for other agents of the network if the fuzzy controller decides that under-utilization is present (or projected). The controller may also keep the current network state unchanged if the system is in an optimum performance. Let E be the error, T be the target, and C be the current state value. Then, E = T – C. We have the following three cases: $E > 0 \Rightarrow$ system under-utilization, $E < 0 \Rightarrow$ system overload, and $E = 0 \Rightarrow$ optimum performance. Furthermore, we compute dE/dt which also results in the following three cases: $dE/dt > 0 \Rightarrow$ system is getting overloaded, $dE/dt < 0 \Rightarrow$ system is getting under-utilized, and $dE/dt = 0 \Rightarrow$ system is stable. The next step is to define the fuzzy inference rules in which the correct action will be based upon. Let Rel = Release Resource, Rq = Request Resource, and NC = No Change. Then, the rule base is as follows:

Rule 1: $\mathit{If}\, E < 0\, \mathit{and}\, dE/dt < 0\, \mathit{then}$ OUTPUT = Rq.

Rule 2: $\mathit{If}\, E = 0\, \mathit{and}\, dE/dt < 0\, \mathit{then}$ OUTPUT = Rel.

Rule 3: $\mathit{If}\, E > 0\, \mathit{and}\, dE/dt < 0\, \mathit{then}$ OUTPUT = Rel.

Rule 4: $If\, E < 0\, and\ dE/dt = 0\ then$ OUTPUT = Rq.

Rule 5: $If\, E = 0\, and\ dE/dt = 0\ then$ OUTPUT = NC.

Rule 6: $If\, E > 0\, and\ dE/dt = 0\ then$ OUTPUT = Rel.

Rule 7: $If\, E < 0\, and\ \ dE/dt > 0\ then$ OUTPUT = Rq.

Rule 8: $If\, E = 0\, and\ dE/dt > 0\ then$ OUTPUT = Rq.

Rule 9: $If\, E > 0\, and\ dE/dt > 0\ then$ OUTPUT = Rel.

The controlling algorithm would then run as follows:

1- Set the target value, T, for the specified parameter.

2- Obtain the current value, C and record the time t.

3- Computer the error E = T – C.

4- Computer dE/dt .

5- Apply the appropriate rule from the rule base.

6- Go to step 2.

Figure (6 - 3) and Figure (6 - 4) provide examples of possible membership functions for error and error curve, respectively, of the *good* bandwidth with hypothetical numerical values. In Figure (6 - 3), E is represented by the solid line and $\overline{E}$ is represented by the gray line. Similarly, in Figure (6 - 4), dE/dt is represented by the solid line and $\overline{dE/dt}$ is represented by the gray line.

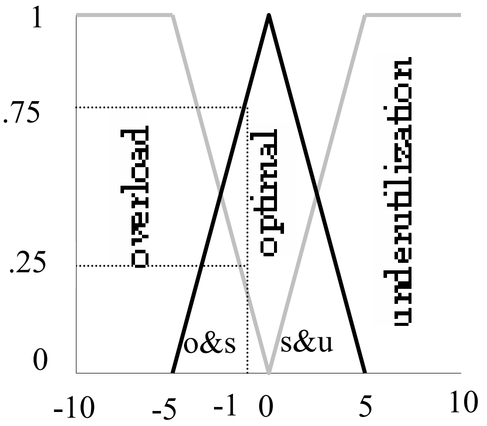


FIGURE (6-3): Error

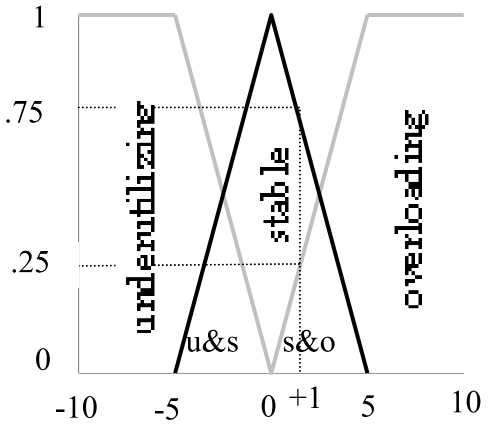


FIGURE (6-4): Error curve

Now, let the inputs to the fuzzy controller be as depicted in Figures (6 - 3) and (4), $E = -1$ (this means the system is 25% overloaded and 75% optimal) and $dE/dt = +1$ (this means the system is getting overloaded with a rate of 25% but is still stable with a degree of 75%). Then, we apply rules number 4, 5, 7, and 8. By Rule 4, OUTUT = Rq with degree of membership $= \min(0.25,0.75) = 0.25$. By Rule 5, OUTPUT = NC with degree of membership $= \min(0.75,0.75) = 0.75$. By Rule 7, OUTPUT = Rq with degree of membership = $\min(0.25,0.25) = 0.25$. By Rule 8, OUTPUT = Rq with degree of membership $= \min(0.75,0.25) = 0.25$. The next step is to perform the defuzzification in order to convert the overall conclusion into a real number. The most common defuzzification methods are the Center of Gravity Defuzzification (CGD) and the Root Sum Square (RSS). According to CGD, given a fuzzy set A defined on the interval [a1, a2], $CGD = \int_{a1}^{a2} xA(x)dx \Big/ \int_{a1}^{a2} A(x)dx$. According to RSS, for outputs o1, o2, and o3, $RSS = \sqrt{o1^2 + o2^2 + o3^2}$. Since we have few discrete values here, we suggest the use of RSS. Then we have: Rq = $\sqrt{0.25^2 + 0.25^2 + 0.25^2} = 0.433$ (by rules 4, 7, and 8) and NC = $\sqrt{0.75^2} = 0.75$ (by rule 5). Then the final output is computed as follows:

$$output = \frac{-100 \times Rq + 0 \times NC + 100 \times \mathrm{Re}l}{Rq + NC + \mathrm{Re}l} = \frac{-100 \times 0.433 + 0.0 \times 0.75 + 100 \times 0.0}{0.433 + 0.75 + 0.0} =$$

-36.6%.

Then, the conclusion is that the system is in overload state and needs to request 36.6% of the allowed bandwidth.

Chapter Conclusion and Future Directions

This chapter is based on the first phase of the research on the application of Soft Computing to the area of Quality of Service Networking. It is foundational in nature as the first phase was exploratory. Hence, the solutions and approaches are mainly on the conceptual level. The chapter aimed at introducing the modeling of the problem and presenting the ideas for conceptual solutions with some abstract examples. The Proposed model will be exploited in the next two chapters, where specific solutions to the major problems of QoS Networking in the presence of uncertainty will be presented and evaluated. The two major issues that will be addressed are the maintenance of global network state information and QoS routing, respectively.

CHAPTER SEVEN

A NEW NETWORK STATE INFORMATION UPDATING POLICY USING FUZZY TOLERANCE CLASSES AND FUZZIFIED PRECISION

Maintaining global network state information with unlimited precision in a dynamic large-scale network is practically impossible. The nature and the level of the inevitable imprecision vary significantly when different policies are used for updating the global network state information. This has a direct effect on the routing decisions and the network performance. In this chapter the author discusses the concept of precision of a network state, presenting a model for the impossibility of reaching, not only unlimited precision, but also the desired level of precision. The author, then, proposes a novel network state updating policy to achieve two goals: bounded imprecision and much lower overhead by its ability to impede frequently unneeded updates without the need for a hold-down timer. The proposed policy is based on overlapping fuzzy tolerance classes and is partitioning-mechanism-independent. Hence, it can be used on top of currently available class-based policies.

Introduction

This chapter is a continuation of the work we started in [Moussa and Kohout 2001] and presented in Chapter Six. The previous chapter was built on a foundational paper for establishing a new approach for attacking Quality of Service (QoS) communication problems by utilizing fuzzy and soft computing techniques.

Maintaining global network state information with unlimited precision in a large dynamic network is practically impossible. However, the performance of QoS routing algorithms that do not assume the existence of this imprecision, i.e., assume precise state information, drops significantly when the state information is imprecise [Apostolopoulos *et al.* 1998], [Shaikh *et al.* 1997, 2001]. Therefore, the solution to this problem is to develop techniques to capture and tolerate the imprecision, rather than neglecting it. Those techniques can be on the modeling level, the state information maintenance level, and on the routing level. Chapter Six focused on the modeling and conceptual level. This chapter continues to build the QoS network model and concentrates on the network state information maintenance level.

On the modeling level, the author had developed and presented a generalized fuzzy model of the network with its QoS parameters [Moussa and Kohout 2001]. This chapter is based on another published paper [Moussa and Kohout 2002], where the author pursued a deeper investigation of the proposed model. Since the precision of the network state is not only a concept but also actually a parameter that the network performance depends on, the author addresses this precision first, tracing its sources and the different types and degrees it may take. The author then presents a fuzzy model of this imprecision, aiming at quantifying the imprecision and limiting it within computable bounds. This modeling can be used to decrease the impact of the unavoidable imprecision on the network performance when developing updating protocols and routing algorithms.

On the maintenance of the network state information level, this chapter includes a survey of the previous approaches for maintaining the network state, identifying the advantages and disadvantages of each approach and justifying the need for a new approach. The author of the dissertation then proposes a new approach for updating the network state information based on the use of fuzzy overlapping tolerance classes. The new policy aims at optimizing the trade-off between the precision of the network state information and the updating overhead.

Network State Updating Policies

An updating policy specifies how and when a node is triggered to initiate the update and the contents of such an update. Two classes of update policies have been in use: *periodic policies* and *triggering policies*.

Periodic Policies

Periodic policies are also called *timer-based policies*. Each router updates the state of its links periodically by broadcasting the current state to the other routers in the network. The interval between updates is an important parameter of this policy. A short time interval results in more frequent updates. This means increased precision in the network state information. Nevertheless, it also means increased update overhead. On the other hand, larger time intervals reduce the overhead at the expense of increasing uncertainty about the network state. Despite the trade-off between the interval size and the associated overhead and their effect on the network state performance, timer-based policies have the general advantage of being predictable in terms of frequency and overhead cost, since both the times and sizes of the updates are known a priori [Shaikh *et al.* 1997, 2001]. However, the main disadvantage of periodical update policies is that the degree of imprecision is completely unpredictable. Between time ticks of the timer, there is no way to find or infer any values, or ranges of values, for any metric. This type of imprecision is referred to in the literature as *random imprecision* [Apostolopoulos *et al.* 1998], [Yuan, Zheng, and Ding 2002]. The author uses the term *unbounded imprecision* in this dissertation. This terminology will be addressed further and justified in the next section.

Triggering Policies

Instead of relying on a timer to trigger the updates, a change in the value of a link metric is used to trigger the updating process. A variety of methods exist for triggering the update. The most widely used are the *threshold-based policy* and the *class-based policy* [Apostolopoulos *et al.* 1999].

Threshold-based Policies: In this policy, a constant threshold value *th* is used to determine when an update should be performed. Take a change in the bandwidth *bw* for example. According to the threshold-based policy, an update is triggered when $\frac{|bw^l - bw^c|}{bw^l} > th$, where bw^l is the last advertised value for the available bandwidth and bw^c is the currently available bandwidth. The threshold value *th* is the main parameter of this policy. Smaller threshold values result in higher accuracy but usually higher overhead due to more frequent updates. Larger threshold values lead to less frequent updates, which reduces the overhead, but reduces the accuracy of the network state information as well. The general advantage of this policy is the controlled imprecision, termed in the literature as *deterministic imprecision* [Yuan *et al.* 2002] and *systematic inaccuracy* [Apostolopoulos *et al.* 1998]. In this dissertation, the author maintains the term *bounded imprecision*. The author will also explain and justify this proposed terminology in the next section. The imprecision is bounded because it is localized between upper and lower bounds. For example, if th = 0.1 and an advertised bandwidth has the value 10 Mbps, until the next update the bandwidth remains in the range [9 Mbps, 11 Mbps]. The main disadvantage is the unpredictable overhead.

Class-based Policies: In class-based policies the range of all possible values of a particular link metric is partitioned into classes. When the current available value crosses a class boundary a network state update is triggered. Two types of class-based policies are commonly used, namely, *equal class-based* updating and *exponential class-based* updating. The equal class-based policy is characterized by a constant B, which represents the class size. Therefore, the range of possible values of a metric, e.g., bandwidth, is partitioned into equal size classes: [$0, B$], [$B, 2B$], [$2B, 3B$], etc. The exponential class-based policy is characterized by two constants B and f, where $f > 1$. The range of all possible values is partitioned into classes of unequal sizes as follows: [$0, B$], [$B, (f + 1)B$], [$(f + 1)B$, $(f^2 + f + 1)B$], …, etc. In class-based policies, B and f are the main parameters and their values are critical to the frequency of updates and the associated overhead. Like threshold-based policies, the imprecision is bounded in class-based policies. Between updates, the parameter values stay within the class boundaries.

Triggering policies in general have the potential problem of generating too frequent, possibly meaningless, updates when the metric value fluctuates around a class boundary in class-based policies, or when the threshold value is small in threshold-based policies. The most common solution to this problem is the use of *hold-down timers*, also called *clamp-down timers*. When a hold-down timer is used, updates cannot be carried out before a specified minimum time period passes. Clearly, although the hold-down timer controls the overhead, it takes away the main advantage of triggering policies, i.e., the bounded imprecision. Another solution is sometimes utilized with class-based policies. Instead of triggering the update as soon as the value crosses the class boundary, a minimum change must be detected before an update is initiated. For example, updates are not performed unless the new value reaches the middle point of a neighboring class. In threshold-based policies the only way to reduce the updating overhead without a hold-down timer is to use a larger threshold constant.

Precision Fuzzification

Precision is a fuzzy concept. When we make a statement such as “X is precisely Y”, the word “precisely” may be interpreted completely differently in different fields and/or applications. For example, when measuring the distance between lines in a football field, a difference of 0.1 millimeter may not alter the meaning of “precisely” in the above statement. Hence, “precisely” still indicates unlimited precision. However, if we were measuring differences between wavelengths of light waves, the same difference would mean zero precision.

This means that, depending on the area of applications and type of measurement, a higher degree of precision may be desirable to a certain point, after which it may be increasingly undesirable. Therefore, when we use the term *unlimited precision* we do not really mean infinite precision. What we actually mean is the degree of precision after which added precision is not needed, or undesirable due to the cost paid for an unneeded gain. In some applications the satisfactory degree of precision may be reachable; in others it may not be.

This raises some critical questions about the precision of a network state. (1) What is a precise network state? (2) Is the point of unlimited precision, as defined above, reachable in this case? A third question to be answered after answering the first two is: what is the nature of imprecision in network state information?

To answer the first question, a network state is accurate with unlimited precision when all the parameters of the network state are known and consistent with the physical state of the network at all the nodes all the time. As for the answer to the second question, it has been stated in the literature that the maintenance of accurate network state information in large dynamic networks is very difficult, if possible at all [Guérin and. Orda 1997], [Chen and Nahrstedt 1998], and [Yuan *et al.* 2002]. The author of this dissertation contends that answer to the question comes in a much stronger statement, in the form of a provable theorem.

Theorem (7 – 1): Maintaining network state information with unlimited precision in any network is actually impossible.

Proof: A network state update consists of three steps: a triggering step, an update-sending step, and update-receiving step. When performing an update, node N1 is triggered to send an update message at time T1. Node N2 receives the message at time T2, where T2 – T1 is the time needed to transmit the update message. Since T2 – T1 can never be equal to zero and since the network state cannot freeze during the time T2 – T1 (otherwise, the update itself cannot be performed), there is no guarantee that the state advertised at time T1 is still the same at time T2, when the update procedure is complete.

In addition, when node N1 advertises the current bandwidth, it sends an update message. The update message itself consumes a portion of the bandwidth that it is advertising. This means that even before the update message reaches node N2, the information it carries is already partly outdated.

The above discussion proves the impossibility of maintaining a network state with absolute precision, regardless of the maintenance method. This brings the third question:

what is the nature of the inevitable imprecision and how is it related to the updating policies? As noted above, two types of imprecision exist. There is no standard terminology for the two different types of imprecision, but the author of the current dissertation disagrees with the terminology currently in use in the literature, namely, *random imprecision* for one type [Yuan *et al.* 2002], [Apostolopoulos *et al.* 1998] and *deterministic* [Yuan *et al.* 2002], or *systematic* [Apostolopoulos *et al.* 1998], [Apostolopoulos *et al.* 1999] for the other. The author further believes that the linguistic term is very important here and should be informative.

The type of uncertainty is directly related to the network state update policy in use. The author proposes, and uses, the terminology *bounded imprecision* and *unbounded imprecision* in this dissertation. Unbounded imprecision is the type of imprecision that results from the use of timer-based policies. Since no link state metric is used to trigger the update, there is no way to predict or infer even an approximate value, or range of values, between the updates. This is why it is *unbounded*. The term *random* is not distinctive because even in the case of bounded imprecision the individual values in the range between the upper and lower bounds of all possible values are still random. On the other hand, in triggering policies such as the threshold-based and class-based policies the values are not *deterministic* or *systematic*. All we know about the expected values in this case are the upper and lower bounds that the real values would fall in between. Therefore, the author uses the term *bounded imprecision*.

Clearly, since the imprecision cannot be zero in network state, it is desirable to limit the imprecision to the bounded type whenever possible. Furthermore, the narrower the range of imprecision the better, provided the increasing precision is not at the expense of the network performance.

A New Network State Update Policy

From the above discussion in the previous two sections, it is clear that the trade-off in any network state update policy is between the overhead generated by the update policy and the precision of the network state information available to path selection mechanisms. The overhead cost of an update policy can be classified into two components: frequency of updates and contents of the update messages [Apostolopoulos *et al.* 1998].

Several studies have been carried to evaluate the routing performance under different network state update policies [Apostolopoulos *et al.* 1998, 1999], [Shaikh *et al.* 1997, 2001], and [Yuan *et al.* 2002]. From those studies the author infers the criteria of a desired network state update policy to be (1) the policy that achieves the best trade-off between precision and overhead cost, (2) a policy that results in bounded imprecision, since it has been proven that zero imprecision is impossible, and (3) the size of update messages is not significantly increased.

None of the currently available policies achieve all those criteria combined in practice. All current policies violate criterion one. In addition, timer-based policies violate criterion two and triggering policies also violate criterion two with varying degrees depending on the setting of the hold-down timer used to control the update traffic volume.

In the course of the current research, the author believes that static policies cannot achieve criterion number one since it involves the control of two dynamic variable parameters. Furthermore, criterion number two cannot be fully achieved as long as hold-down timers are used. Hence, the desired updating policy ought to be dynamic and non-timer-based. Next, the author proposes a novel update policy to meet all the desired criteria.

To maintain the balance between the dynamically changing traffic overhead and network state precision, the author proposes the use of a dynamic class-based policy that can be applied on top of any partitioning scheme of choice. The dynamic policy uses fuzzy tolerance classes, which are not necessarily of equal size. The tolerance classes are characterized by the properties of *symmetry* and *reflexivity* but not the *transitivity* [Bandler and Kohout 1988], [Klir *et al.* 1995] in order to relax the conditions on the membership of a class to enable a flexible mechanism that allows for classes of variable sizes and overlapping bounds. We use a trapezoidal membership function defined as follows:

$$F(x) = \begin{cases} \dfrac{a-x}{a-b} & when\ a \le x \le b \\ 1 & when\ b \le x \le c \\ \dfrac{d-x}{d-c} & when\ c \le x \le d \\ 0 & otherwise \end{cases}$$

Figure (7 – 1) is a graphical representation of the class membership function.

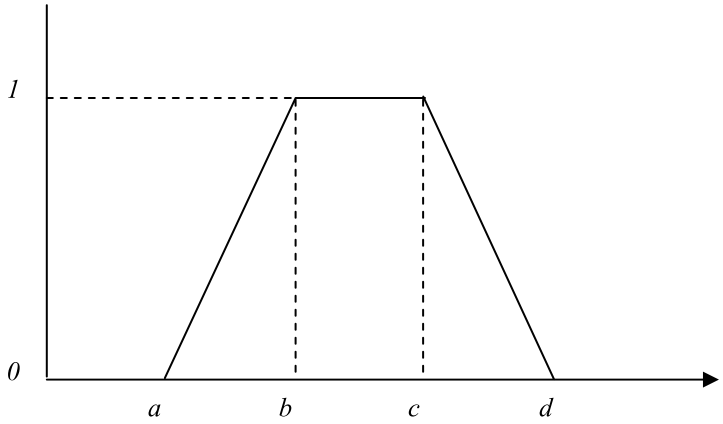


Figure (7 - 1): Trapezoidal Membership Functions

Taking the bandwidth as an example, in Figure (7 – 1) a bandwidth value *bw* would be in the class with full membership if *bw* is in [*b*, *c*]. Bandwidth *bw* has zero membership in that class if $bw < a$ or $bw > d$. When $a < bw < b$ or $c < bw < d$, *bw* is in the class with degree $0 < F(x) < 1$. In comparison with traditional class-based partitioning, two constants *A* and *B* are used to specify the class size when equal class size is used. $A = c - b$, and $B = b - a = d - c$. When exponential class size is used, the third constant *f* will be used along with the two constants *A* and *B*. The range of all possible values for a link metric, e.g., the bandwidth, is to be partitioned into overlapping tolerance classes. Figure (7 – 2) illustrates the partitioning of the bandwidth into three classes.

In this proposed policy we eliminate the need for a hold-down timer to suppress the unnecessary too frequent updating. Instead, the network state updates are not initiated

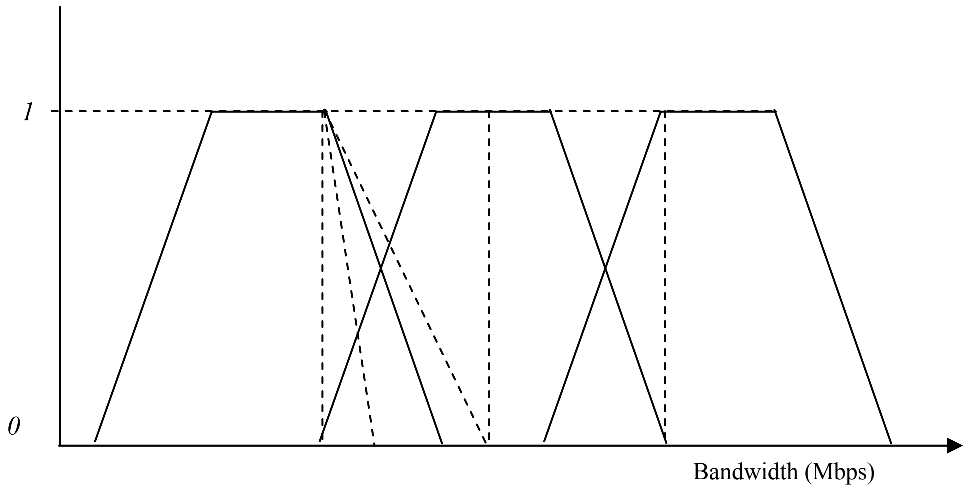


Figure (7 - 2): Bandwidth Class Partitioning

until the current class membership becomes zero. Once, the class changed and the update is triggered, no further updates are performed until the membership in the new class becomes zero again. Therefore, no fluctuation around a boundary point would result in flooding the network with meaningless update messages.

The second criterion is implemented in this policy, since the uncertainty is bounded by the upper and lower bounds of the class. Additional optimization can still be implemented without the need for timers by using dynamic classes. To implement the dynamic classes, the two constants *A* and *B* are replaced by two variables *A* and *B*. The link state table keeps track of the ratio *R* of the number of update messages to the total number of messages in the link traffic. Two constants *Ru* and *Rl* are specified for upper and lower limits of acceptable ratios and two levels of dynamic optimization are utilized by changing the value of variables *B* and *A*, respectively. If $R \geq Ru$, the ratio *R* is too high and the slope of the membership function can be changed by increasing the value of variable *B* to suppress the frequency of the updates. A class boundary cannot exceed the middle of a neighboring class. If class boundaries reach the middle of the neighboring classes, then flattening the slope is not sufficient. Then, *B* returns to the default value and *A* is changed to provide coarser granulation. Similarly, when the ratio $R \leq Rl$, the ratio is too low, the slope can be changed by one unit to allow for more frequent updates. Also, when *B* reaches a lower limit, slope change is not sufficient. Hence, *B* returns to the default value and *A* is decremented to allow finer class partitioning.

The convergence to the optimal condition can be implemented faster by using a classical fuzzy control mechanism as presented in Chapter Six [Moussa and Kohout 2001]. After setting the desired ratio *R*, the current ratio can be stored periodically and the error and error curve can be computed to decide the needed modification directly rather than incrementally. However, this will be at the expense of tremendous additional cost that may overweigh the convergence speed. Therefore, at this stage, the author prefers the updating algorithm as proposed above for its minimal cost, which maximizes the gain.

By utilizing the proposed mechanism an optimal precision-overhead relation can be maintained. The frequency of updates is controllable without the need for hold-down timers. Therefore, maintaining bounded imprecision at all time. The additional cost involved to maintain this optimal control over the trade-off between overhead and imprecision is minimal. Two constants, *Ru*, and *Rl* are stored and three variables, *A*, *B*, and *R* are computed. The computation of *A* and *B* can only be increment, decrement, or reset. *R* is a division operation. This clearly shows that the optimal control and bounded imprecision is gained at the expense of negligible additional cost. Finally, when performing the update, the author of the dissertation proposes only one addition to the traditional update message, which is the current degree of membership in the class that triggered the update. In fact this one addition is of no use with the current available routing algorithms. It will be of use only when we complete the fuzzy routing algorithm, which exploits the proposed fuzzy tolerance classes-based update policy. That routing algorithm will be presented in Chapter Eight.

Chapter Conclusion and Future Directions

In this chapter, it was shown, and proven, that it is provably impossible to maintain network state information with unlimited precision. The imprecision is inevitable and should be dealt with by devising techniques to capture it and tolerate it since ignoring the inherent imprecision results in drastic degradation in routing performance and network utilization.

The imprecision in the network state information can be bounded or unbounded. It is highly desirable to deal with bounded rather than unbounded imprecision. With bounded imprecision, there are methods to compute and infer a range of imprecision to be dealt with accordingly. Unbounded imprecision can take any degree and, when present, routing results and network performance become unpredictable.

Network state update policies determine the type and degree of imprecision. Current available policies either introduce the undesirable unbounded imprecision by using periodical updates, or utilizing hold-down timers with triggering policies, or may become the source of tremendous overhead if triggering policies are used without a hold-down timer.

A well-devised policy can eliminate the need for hold-down timers. However, optimal control of the relation between imprecision and update overhead cannot be obtained by using static update policies. Therefore, we proposed a novel network state update policy to cover the weaknesses of the current available policies. The results of using our proposed policy are (1) bounded imprecision with computable range, (2) elimination of the need for hold-down timers, (3) dynamic optimal control of the overhead-imprecision relation, and (4) the additional cost associated with our policy over the current available policies is minimal.

The immediate future plan is to develop a novel fuzzy routing algorithm that exploits the proposed policy, by which we complete the fuzzy model we intended in our foundation research.

CHAPTER EIGHT

FUZZY PROBABILISTIC QOS ROUTING FOR UNCERTAIN NETWORKS

Routing algorithms have been the ultimate objective for the computer science researchers on computer and communication networks. Therefore, the trend in the literature has been to address the routing problem first, then attempt to solve the other surrounding problems in order to bring the devised routing algorithm into realization. For example, the researchers used to develop routing algorithms, then, design network state maintenance updating policies as supplements to those algorithms to make them realizable. The researcher on this dissertation reversed this approach by focusing on the development of a novel, routing algorithm-independent, network state information maintenance policy first. This policy, introduced in Chapter Seven, enable optimal control of the trade-off between the accuracy of the information about the network and the overhead associated with the sought maintenance, regardless of the routing algorithms used over the network nodes and clusters. However, an intelligent routing scheme can exploit the previously devised policy for maintaining the global network state information. This chapter introduces a novel routing scheme with that objective. The chapter is based on a paper that has been refereed and accepted for publication and presentation by JCIS/FT&T-2003, the 7th Joint Conference on Information Sciences/9th

International Conference on Fuzzy Theory and Technology, Cary, NC, USA, in September of 2003 [Moussa and Kohout 2003].

Fuzzy Computing and Probabilistic Computing are two of the fields that constitute Soft Computing. The integration of both systems can result in powerful tools that enable solutions for difficult computational problems, which involve uncertain complex dynamical systems. QoS routing in large-scale dynamical networks with uncertain parameters is one such a problem. In this chapter the author discusses the domains, differences, and merits of each paradigm and the importance of their integration for solving problems of complex dynamical systems under uncertainty. For application, the author presents the problem of QoS routing in large-scale dynamical networks with uncertain parameters, the main area of application for the current dissertation. The author, then, proposes a novel multi-constraint routing algorithm that builds on the network state updating policy presented in Chapter Seven [Moussa and Kohout 2002]. The proposed algorithm utilizes fuzzy modeling to capture the uncertainty in the network state information, probabilistic computing to compute routing paths with highest probability to satisfy a QoS communication request, and BK-products of fuzzy relations to aggregate the computation and select the optimal path.

Introduction

The cross-disciplinary field of Soft Computing is built on the fusion of several methodologies, among which two disciplines seem to frequently run into collision and confrontation. These two fields are the Fuzzy Set Theory (FST) and Probability Theory (PT) or, more generally, Fuzzy Computing (FC) and Probabilistic Computing (PC). Advocates of each system repeatedly challenge their counterpart in meetings and publications. However, these challenges have always revealed the lack of understanding of one of the two disciplines or the other.

The fusion of methodologies that was first proposed by Dr. Zadeh implies the complementarity of each paradigm to the other rather than the replacement of one another. This chapter starts by revisiting this issue, discussing its motivations, and elaborating on the importance of integrating the two fields to form powerful tools for solving comp1ex problems, where complete solutions may not be possible by either discipline alone. This discussion is presented in the next section “Fuzzy Probabilistic Computing”. In the following section, the author presents the problem of Quality of Service (QoS) Routing in large-scale networks as an example of this class of complex dynamical chaotic systems, reviews the literature to study and evaluate other approaches for dealing with the problem of QoS routing under uncertainty, and reviews the previous attempts to use probabilistic routing. The author, then, proposes a novel algorithm for computing multi-constraint QoS paths based on the integration of Fuzzy Computing and Probabilistic Computing in the following section, the main contribution of the chapter and an ultimate objective of the dissertation. The proposed algorithm is designed to capture, and account for, the inherent uncertainty in network state information. Another goal for the proposed algorithm is to solve the path computation problem without introducing high, or any, additional overhead. The proposed algorithm then uses BK-products of fuzzy relations [Kohout and Bandler 1992], [Kohout 2002], and [Moussa and Kohout 2001] to compute the selection decision of a multi-constraint optimal path. Finally, the last section concludes the chapter.

Fuzzy Probabilistic Computing

The clash between Fuzzy Set Theory, and its resultant computing system, Fuzzy Computing, on one hand and Probability Theory, and its descendant computing system, Probabilistic Computing, on the other hand is caused, mainly, by the fact that a primary objective of both systems is to deal with uncertainty. To understand the difference between Probability Theory and Fuzzy Set Theory, it is important to understand the type of uncertainty that each describes and processes. The uncertainty described by Probability Theory is *randomness* while the uncertainty described by Fuzzy Set Theory is *fuzziness* [Bonissone 1997].

Recalling the introduction of both computational systems from the previous chapters, in probability, we have a sample space S and a *well-defined* region $R \subset S$. The uncertainty results from the *non-deterministic* membership of a point $P \in S$ in the region R. S represents the set of all possible values of a random variable, R represents the event whose probability is to be predicted, and the point P is the outcome of the system. The characteristic function of R determines whether $P \in R$ or $P \notin R$. The probability value describes the frequency of, or the belief that, $P \in R$.

In contrast, Fuzzy Set Theory defines a space S' and an *imprecisely defined* region $R' \in S'$. The uncertainty results from the *partial membership* of a point P in the region R'. The partial membership describes the degree, rather than the frequency or belief, to which $P \in R'$.

Another major distinction between Fuzzy Set Theory and Probability Theory is the class of problems to be solved by each. Probability Theory deals with the prediction of an event whose occurrence is uncertain based on the information currently available.

On the other hand, Fuzzy Set Theory deals with concepts and status of variables, rather than events [Klir *et al.* 2003]. However, the predictability nature of Probability Theory, if not well understood, can cause it to be confused, not only with Fuzzy Logic, but also with any logic system.

A major characteristic of a logic system is the inference mechanism, which results in the reasoning engine. If the time domain of a premise of an inference rule is present, or past, time, and the time domain of the conclusion of the rule is future time, then the logic system can be used for predicting future values. For example, in the classical *modus ponens* expressed as $[(p \Rightarrow q) \wedge p] \Rightarrow q$, if $p \Rightarrow q$ is a rule established from past information, or events, p is a present information, or event, and q is a future outcome of a system, then the *modus ponens* would be used to infer, or predict, future results. When the *fuzzy modus ponens* is applied similarly, it can be used for uncertain prediction, which may result in the false belief that Probabilistic Computing and Fuzzy Computing can be used interchangeably.

The fundamental difference between using Probabilistic Computing and Fuzzy Computing for future prediction is that PC tries to predict the likelihood of the occurrence of an event whereas Fuzzy Computing, if used for prediction, would be trying to predict, or determine, the status of the system if the sought event occurred. For a real-life example, let us take the weather forecast process. To produce the statement “it is highly probable that it will rain tomorrow”, Probabilistic Computing is used to compute the probability of which rain may fall the next day based on the status of the current weather system. The computed probability value, say 0.9, means that if the same weather system occurs 100 times, rain will fall in 90 of them. Another different, but related, interpretation is that the 0.9 represent the strength of our belief that it will rain tomorrow, which would be based on the ratio of the number of evidences in support of our belief to the total number of evidences about the status of the weather system. The implication, in both cases, is that we do not know for sure that it will rain tomorrow and over a span of 100 times we do not know at which 90 times rain will fall. Moreover, the computed value, even though so high, does not indicate in anyway the heaviness of the rain. Fuzzy Computing on the other side can be used to model and compute the status of the system out of which the probability value is to be computed. Furthermore, FC can be used to fill in the gap between mathematical modeling and human perception in order to provide a computational model for such terms as *highly probable* or *heavy* rain. This is an example of a system where Probabilistic Computing and Fuzzy Computing should be integrated, rather than compared, in solving problems.

QoS Routing and Related Work

The emergence of QoS networking created many challenges to network developers of many fields. It is true that physicists and engineers contributed significantly to the development of faster networks. Optical networks that use Wavelength Division Multiplexing (WDM) technology to increase the capacity of optical transmission systems by transmitting multiple wavelengths over a single fiber, reaching transmission rates on the order of *terabits per second* [Ori 1996], [Ghosh and Acharva 2001], represent the latest such breakthrough. However, as the physical capabilities of the networks grow, the demands by new applications to exploit those capabilities also grow. This necessitates the constant development of algorithms and solutions to provide the needed exploitation.

The uncertainty in the network state information is an intrinsic problem. It was proven in Chapter Seven [Moussa and Kohout 2002] that maintaining global network state information with unlimited precision in any network, regardless of size, is actually impossible. It was also shown in several studies that the performance of QoS routing algorithms, which do not account for the inherent imprecision, drops drastically when the network state information is inaccurate [Apostolopoulos *et al.* 1999], [Shaikh *et al.* 2001], and [Shaikh *et al.* 1997]. Hence, a routing algorithm that does not capture and tolerate the intrinsic uncertainty is deemed to be unsuccessful as a QoS routing algorithm. Regretfully, most routing algorithms available today do not take this uncertainty into account. Instead, they assume it does not exist, regardless of the inherent nature of this uncertainty. However, few routing algorithms were proposed with the main objective of handling the intrinsic imprecision and reducing its effect [Guérin and Orda 1997], [Lorenz and Orda 1998], [Chen and Nahrstedt 1998], and [Apostolopoulos *et al.* 1999]. Nevertheless, some Probability Theory-based approaches described how to compute with Probability Theory but never explained how to infer and maintain the Probability Distribution [Guérin and Orda 1997]. More comprehensive probabilistic approaches were published in [Apostolopoulos *et al.* 1999], [Ghosh and Acharva 2001] but they proposed a form of empirical probability based routing that requires very high cost computation and network traffic, which would defeat the original purpose of the algorithm. Alternatively, a uniform distribution function was used to avoid the high overhead associated with exchanging the empirical probability information. The overall conclusion of this class of studies, which was confirmed in a most recent study at Florida State University [Yuan and Yang 2003], is that probabilistic routing schemes offer better performance in terms of the rate of successful routing decision, but they require the exchange of large data structure, which generates a flooding overhead in the network traffic.

QoS routing in a large-scale dynamical network with uncertain state is a perfect area of application for Fuzzy Probabilistic Computing. In this chapter, the author continues the research, which started on the conceptual and foundation level in Chapter Six [Moussa and Kohout 2001] and the network state maintenance level in Chapter Seven [Moussa and Kohout 2002]. This new step is on the routing level and strongly builds on the network state information policy proposed in the previous chapter [Moussa and Kohout 2002].

The New Algorithm

In this section, the author of the dissertation presents a novel routing algorithm based on Fuzzy Probabilistic Computing. The proposed algorithm builds on the fuzzy tolerance classes based network state information updating policy proposed in Chapter Seven [Moussa and Kohout 2002], which is essential to the maintenance of the empirical probability information required for the final refinement of the algorithm. The approach uses Fuzzy Computing to model the network state and to capture the inherent uncertainty as described in [Moussa and Kohout 2001] and [Moussa and Kohout 2002]. Fuzzy Computing is also used to implement the updating mechanism that limits and quantifies the uncertainty to enable better routing decisions and optimizes the trade-off between the network state uncertainty and the overhead associated with the updating mechanism. The routing module, then, infers the probability distribution with which a requested communication requirement may be met. If the routing involves multi-parameter QoS requirement, multi-path solutions are computed, the solutions are encoded into a fuzzy relation, and BK-products computation is performed to select the optimal path.

The System State

When the network state information is computed and maintained as proposed in Chapter Seven, the global network state at each node will be maintained with *bounded uncertainty*. The bounds on the uncertainty will be controlled by the optimal control of the balance between the dynamically changing traffic overhead and the network state precision. Thus, the levels of QoS of some parameter, such as the bandwidth (*bw*), will be maintained as fuzzy tolerance classes as illustrated in Figure (8 – 1).

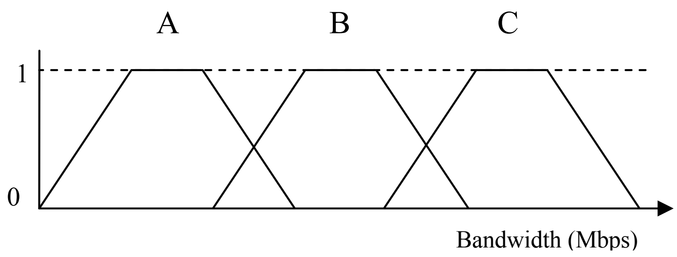


Figure (8 – 1): Bandwidth Class Partitioning

Figure (8 – 1) describes the partitioning of the available bandwidth on a link into three classes: A, B, and C. According to the updating policy in Chapter Seven, a node would not broadcast a state update until the available *bw* goes out of class bounds, i.e., *bw* has membership value of zero in the class. Hence, at any time, we know with unlimited precision the upper and lower bounds on the currently available bandwidth *bw*, i.e., the uncertainty is bounded by the limits of the current class. However, we have no way of knowing the exact *bw* available at anytime.

The Goals of The Algorithm

Having maintained the network state as above, the proposed algorithm is expected to answer the following fundamental questions:

1- Given the bounded uncertain state of the currently available bandwidth *bw* on a link *l*, represented by a tolerance class, e.g., class *B*, and the value of a requested bandwidth *x*, what is the probability $p_l(x)$ that link *l* can satisfy the request?

2- In doing the computation for all possible links, what is the range of the acceptable probability?

3- How to compute feasible paths?

4- In the presence of multiple paths and multiple constraints, how to select an optimal path?

Probability Distribution Computation

It is not hard to see that, when *bw* and *x* are in the same class, the relation between *bw* and *p(x)* is proportional, whereas the relation between *x* and *P(x)* is inversely proportional. In other words, if *lb* is the lower bound of a particular class, e.g., class B, and *ub* is the upper bound, the closer *bw* to *ub* the more likely to be able to support bandwidth requests such as *x* and the closer *bw* to *lb* the less likely to be able to support the demanded bandwidth. Mathematically formulated:

$$\lim_{bw \to ub} p(x) = 1 \text{ and } \lim_{bw \to lb} p(x) = 0$$

Taking *x* to be in a particular class, e.g., class B, i.e., $lb \le x \le ub$, since we know with unlimited certainty that $lb \le bw \le ub$, then:

$$\lim_{x \to lb} p(x) = 1 \text{ and } \lim_{x \to ub} p(x) = 0 .$$

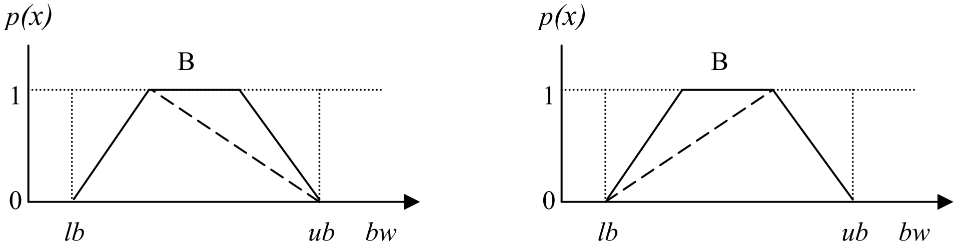


Figure (8 – 2a): p(x) vs. x

Figure (8 – 2b): p(x) vs. bw

Figure (8 – 2): The Initial Probability Distribution Per Class

For the lack of evidence of otherwise, when the system is initiated, we choose linear functions to describe these relations as illustrated in Figure (8 – 2).

Using geometric computation on Figure (8 – 2a), the probability distribution function inside any class can be defined as follows:

$$p(x) = \begin{cases} 1 & when\ x < lb \\ \dfrac{ub - x}{ub - lb} & when\ lb \le x \le ub \\ 0 & when\ x > ub \end{cases}$$

Using the probability distribution function above, the probability $p_l(x)$ that link l can support a connection with bandwidth requirement x can be computed internally at the source node, resulting in Zero traffic overhead. The probability that a link can support other QoS parameters can be computed similarly.

The Acceptable Probability Values

Answering question (2) above in the set of goals for the algorithm will be application dependent and should be set by the source node, or as agreed upon by a QoS contract. If the source node were requesting a connection with hard QoS measures, then only links with $p_l(x) = 1$ would be taken into consideration in the path computation. In the case of soft QoS requirements, the source node sets either the minimum acceptable value for the required parameter to compute the corresponding probability at each link or the minimum acceptable probability to compute the available value for a specific parameter at the link, depending on the type of application. The first scheme can be used when there is a lower, or upper, limit on the acceptable value of a parameter such as the case when the communication is time-sensitive due to real-time tasking and/or synchronization requirements. The latter scheme can be used when there is an upper limit on the rate of packet loss, after which the communication cannot be considered successful.

Algorithm Refinement

It is important to note that when we talk about the routing mechanism here we mean the whole system, that is, the updating policy from Chapter Seven [Moussa and Kohout 2002], which is designed to optimize the network performance regardless of the routing algorithm, plus the above probabilistic routing algorithm that exploits it. However, thus far, we used a linear probability function, which is the theoretical function assuming uniform distribution of the probability between all possible values of bandwidth in a class. The problem with this approach is that the theoretical linear distribution may not be a faithful representation of the actual distribution of probability over certain link, or set of links.

An alternative to this approach is to use empirical probability, where the network state is sampled periodically and a counter vector is used to store a histogram for the link. Details of this scheme are in [Apostolopoulos *et al.* 1999] and were developed further in several studies; [Ghosh and Acharya 2001] is an example, which was studied further most recently in [Yuan and Yang 2003]. However, the drawback of this approach is that it requires periodic sampling at each node and the exchange of large data structures to maintain the probability information, in addition to the network state information traditionally being maintained. This extra traffic could degrade the network performance.

As a solution to this problem, the author of the current dissertation proposes a solution that optimizes the probability function without overloading the network traffic. In this context, the author contends that the instantaneous evaluation of the probability distribution is not necessary. Three factors significantly diminish the credibility of such a distribution:

1- The dynamic nature of the network.

2- The inevitable non-trivial delay, which is expected to grow worse when all these histograms are exchanged periodically.

3- Exchanging the large data structures over the network will consume bandwidth that was not accounted for at the time the probability distribution was computed.

Therefore, rather than exchanging detailed distributions stored in large data structures, which would flood the network traffic, the author proposes two solutions to the problem. The first solution is to use the following heuristic as an alternative:

The Heuristic

1- Each node maintains its own empirical probability locally. A link parameter, such as the bandwidth, can be sampled periodically, or whenever possible.

2- Discrete probability values can be computed as $p(x_i) \approx \frac{f(x_i)}{n}$, where $f(x_i)$ is the frequency of bandwidth x_i being available and n is the total number of samples.

3- For each tolerance class of the partitioning scheme, the collected empirical data is approximated into one of the common distribution functions, e.g., normal, uniform, gamma, etc.

4- When state propagation is due, only a variable representing the distribution function type and the function parameters is added to the broadcast information.

5- Using the broadcast information, the receiving nodes reconstruct the advertised distribution function internally, without the need for all the large data structures to flood the network.

The above proposed heuristic enable the maintenance and exchange of empirical probability information without the need to exchange large data structure. Therefore, we gain the all advantages of probabilistic routing identified in the literature without paying the typical computational cost.

The second solution to be proposed by the author of the current dissertation is actually a refinement and an optimization of the above heuristic. The idea is that since the probabilistic data are not maintained for the entire network metric and is maintained only for each tolerance class by virtue of policy of Chapter Seven, an accurate density function to describe the distribution may not significantly different from a linearly approximated one. Therefore, computing the Expectation of x, $E(x)$, and drawing straight lines from $E(x)$ to each of the lower and upper bound of the current class may suffice. The optimized heuristic is as follows:

The Optimized Heuristic

1- Each node maintains its own empirical probability locally. A link parameter, such as the bandwidth, can be sampled periodically, or whenever possible.

2- Discrete probability values can be computed as $p(x_i) \approx \frac{f(x_i)}{n}$, where $f(x_i)$ is the frequency of bandwidth x_i being available and n is the total number of samples.

3- For each tolerance class of the partitioning scheme, the expected value $E(x) = \sum_{i=1}^{n} x_i p_i$ is maintained locally.

4- When state propagation is due, only $E(x)$ is added to the broadcast information.

5- The actual probability distribution for a class is calculated as the superposition of the linear distribution and the triangular function resulting from $p(E(x)) = 1$.

The resulting distribution is illustrated in Figures (8 – 3), (8 – 4), and (8 – 5). When the system is initialized, empirical data are not available. Hence, Figure (8 – 3), the linear distribution will be used.

As the system runs, the empirical distribution becomes available and *E*(*x*) becomes computable empirically, as in Figure (8 – 4).

Since the empirical distribution is changeable at anytime, the superposition of the linear and empirical distributions is computed to optimize the success of the probability, as in Figure (8 – 5).

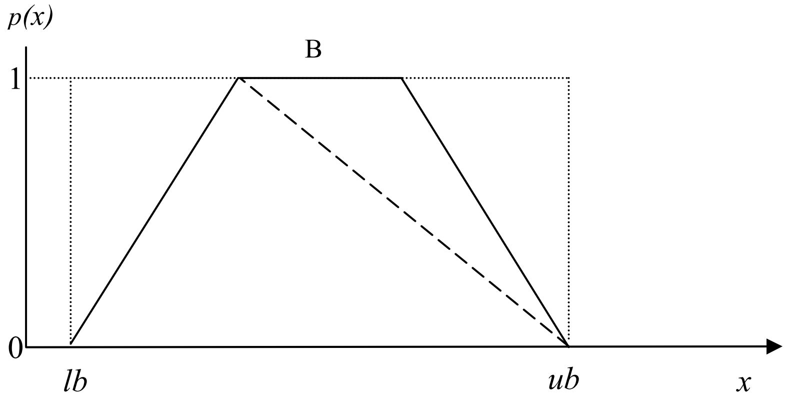


Figure (8 – 3): Linear Probability Distribution

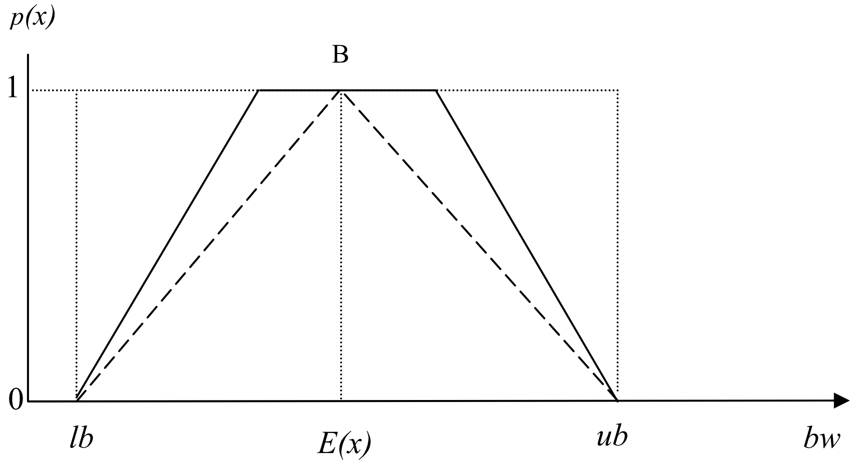


Figure (8 – 4): Probability Distribution Using The Expectation E(x)

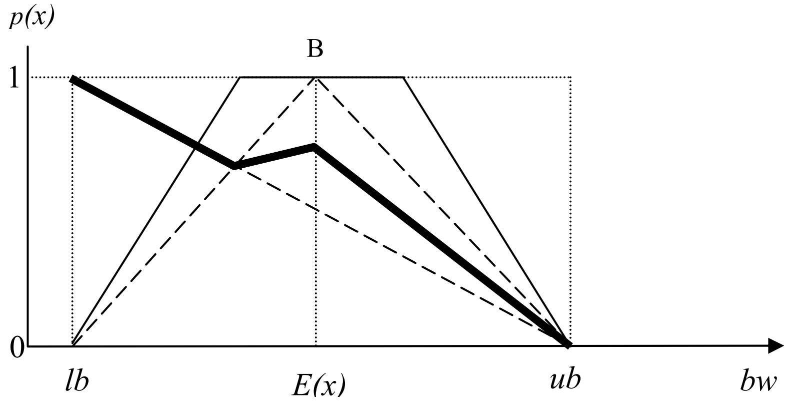


Figure (8 – 5): The Supper Position of the Previous Two Distributions

Path Computation

A path is a set of connected links that starts at the source node and ends at the destination node. Therefore, to compute a path, we need to aggregate the feasible links that provide the end-to-end connection between the source and the destination. The probability that a path P can support a communication request with QoS parameter of x units can be computed as follows: $\prod_{l \in P} p_l(x)$

Optimal Path Selection

When multiple feasible paths are found the optimal path will be the path with highest probability to support the specified QoS requirement. In this case, the problem can be formulated mathematically as, find path PP such that for any path P :

$$\prod_{l \in PP} p_l(x) \geq \prod_{l \in P} p_l(x) .$$

However, the problem may not be so simple if multi-constraint QoS requirements were in demand. In that case we will have a set of paths and a set of parameters. The problem will be to select the path that has the highest probability and most available resources to meet the minimum QoS requirements. To compute this decision, we revert back to Fuzzy Computing. We encode the set of computed paths and their parameters in a fuzzy relation R and use BK-Products to compute the selection of the optimal path. For a concrete example, let us assume the following situation. We have three computed paths I, J, and K and three parameter QoS requirements bandwidth (*bw*), delay (*d*), and cost (*c*). We also assume linguistic measures, such as *bad*, *acceptable*, *good*, *very good*, etc., are chosen to describe the QoS levels and that the current connection is required to be *good*. We encode the available network information in the following relation R [Moussa and Kohout 2001].

Path/parameter	**D**	**B**	**C**	Degree of good
I	.58	.3	.68	.30
<u>J</u>	.9	.89	.93	.89
K	.97	.15	.43	.15

To compute the routing decision in this case, we aggregate the system by computing the relation:

$$(R \circ R)_{xz} = \vee_{y} (R_{xy} \wedge R_{yz})$$

The fuzzy sets intersection is needed to guarantee the minimum satisfactory system parameter. Then, the path with highest degree of overall performance is the one to be selected, in this case path J.

Chapter Summary and Conclusion

Fuzzy Computing and Probabilistic Computing are powerful tools in the arsenal of Soft Computing. Each system excels in solving a certain type of problems. However, there is a class of problems in which neither of the two can optimally solve alone. QoS routing in a large-scale dynamical network, with the intrinsic uncertainty in the global network state information is an example.

The literature shows that probabilistic routing leads to higher success rate, but suffers from the drawback of exchanging large data structures, which results in a tremendous overhead. The high-cost overhead may overwhelm the network traffic and cancels out, at least partly, the gains of the higher success rate achieved by applying probabilistic routing.

This chapter proposed routing computation algorithms to exploit the network state information updating policy proposed in the previous chapter. The proposed algorithms are based on the integration of fuzzy and probabilistic computations. Fuzzy Computing is used to model the network and capture the inherent uncertainty about the network state information. It is used further to dynamically optimize the trade-off between the level of accuracy and overhead associated with the maintenance of the network state information. Probabilistic Computing is then used to exploit the modeled and maintained information about the network to compute the probabilities of successfully routing data from the available information about the network. The proposed algorithm is cost-effective because it provides a scheme for probabilistic routing without the need to exchange the large data structures typically used for empirical probabilistic routing.

The chapter also included a mechanism for computing multi-constraint routing. In this case, multiple feasible paths may be found and a mechanism for the path selection may be needed. To solve this problem, Fuzzy Computing was used again. This chapter introduced a mechanism for the path selection based on the use of BK-products of Fuzzy Relations that enables the selection of an optimal path.

FINAL CONCLUSION, SUMMARY, AND FUTURE PLANS

Dissertation Summary and Conclusion

This dissertation documents a hybrid research on the integration and utilization of Soft Computing sciences with the area of Quality of Service Networking. As a hybrid research, the author of the current dissertation took a bi-directional approach to the problem. In one direction, the researcher studied the Soft Computing science and investigated, and identified, their powers, and suitability, as a new approach for solving the problems of Quality of Service Networking. The approach leads to the construction of intelligent, self-adaptive, networks. Due to the cross-disciplinary nature of the dissertation, the research had to be overlapping multi-directional. Hence, the order of topics in the dissertation does not in any way imply the chronological order the research went through.

The research idea started as the author was able to realize the suitability of, and relate, the two cross-disciplinary fields toward each other. Since then, Soft Computing offered powerful promising solutions to the field of Quality of Service Networking, and the problems of Quality of Service Networking opened research avenues within the sciences of Soft Computing too. The research and application of Soft Computing in this dissertation concentrated on the use of Fuzzy Computing and Probabilistic Computing.

In Fuzzy Computing, the dissertation contributed the identification of a flaw in Fuzzy Set Theory that, for the lack of alternative, has been viewed in the literature as a normal property of the theory. In this dissertation, the author exposed the problem of the violation of the Law of Excluded Middle in Fuzzy Set Theory, and proposed the solution by pointing out the distinction between the concepts of a Set Complement, Set Negation, and Set Inverse. The author also constructed the mathematical model for the theory expansion.

The erroneous, misleading, and sometimes meaningless information obtained from computing the probabilities of real-life systems are the result of using inappropriate probabilistic techniques. Most probabilistic computing techniques are based on the mathematics of crisp sets on simple non-fuzzy systems. However, the real world is not crisp or linear and the events real-life systems are seldom mutually exclusive or independent. Therefore, new probabilistic computing conception and methodology are needed to deal with the dynamics, nonlinearity, complexity, and fuzziness of the real world.

This dissertation proposed and described the basis of such a computational paradigm. The introduction clarified the historical background of probability theory that led to the currently available, insufficient probabilistic computing techniques. The review of the literature presented the previous attempts to develop probability computing to deal with fuzzy events. It shows that those attempts were made on compound events only, which is not enough to solve the problem, especially when the singletons contained in the compound events are also fuzzy. Therefore, after the problem was clearly defined, mathematically formulated, motivation for a solution was pointed out, and criteria of success was set, a computational model was proposed. Since the fundamental problem of probability is to extract predictions about a specific event from the available information about the whole population, the model starts by using Fuzzy Set Theory to model, arrange, and classify the startup data. Then, a formula of counting that takes into account, both, the frequency and membership of each simple event in the population is proposed. From the abstract example, we see that the results obtained by using the proposed model reduce to the results obtained by the traditional model when the degree of fuzziness modeled by the system reduces to zero. This is an important result and is in accordance with the fact that Crisp Set Theory is a subset of Fuzzy Set Theory. The outcome of the system was evaluated against the set of seven points of preset criteria. Finally, a model for computing the rain forecast was presented as an example application.

After establishing the method for computing the basic probability of an event, the dissertation also introduced a model for computing the probability of a union of events. The proposed model does not suffer from the problems and computational intractability associated with non-mutually exclusive events of fuzzy, complex, nonlinear, and infinite systems. The author of the current dissertation hopes that this will be the beginning of establishing the proposed field of Soft Probability, which, when further developed in future work, it should provide valuable tools for computing the probability of large fuzzy, complex, and nonlinear systems. Examples of such systems would be the weather forecast, economy, and motion and flow of uncontrolled, irregular objects, just to name a few.

QoS routing in a large-scale dynamical network is a complex problem of a complex, dynamical, and fuzzy system of the above class. The uncertainty in the global network state information complicates the problem further. Two types of uncertainty exist in the dynamics of a network: *fuzziness* of the QoS levels and the states of the network parameters, and *randomness* of communication requests arrival patterns and the chaotic unpredictable dynamics of the network parameters.

The uncertainty is an inherent problem that cannot be eliminated. However, it was possible to reduce, both, the level and effect of the uncertainty by limiting it to *bounded uncertainty* by using fuzzy techniques to store and update the global network state information. A well-devised policy for the maintenance of network state information can eliminate the need for hold-down timers. However, the relation between the imprecision in the network state information and the update overhead is a dynamic problem for which the optimal control cannot be obtained by using static algorithms. Therefore, the author of the current dissertation proposed a novel network state update policy to cover the weaknesses of the current available policies. Unlike the traditional approach of designing update policies as a supplement to the routing algorithm, this dissertation regarded the update policy as a design target by itself. The results of using the proposed policy are:

1- Bounded imprecision with computable range.

2- The Elimination of the need for hold-down timers.

3- Dynamic optimal control of the overhead-imprecision relation.

4- The additional cost associated with the proposed policy over the current available policies is minimal.

The bounded uncertainty can be further dealt with on the routing algorithm level by using fuzzy probabilistic techniques. Fuzzy probabilistic routing uses Fuzzy Computing to maintain the global network state information while controlling the trade-off between the accuracy and the overhead associated with the global state maintenance. Probabilistic Computing can then be used to predict the probability of the network response to the dynamic changes. Routing decisions can then be taken based on the definition and agreement of the acceptability of probability. In the case of multi-constraint routing, the problem becomes a multi-variable dynamic problem. BK-Products of fuzzy relations can be a powerful tool for solving the problem. The multi-parameter probabilities can be encoded as a fuzzy relation between paths and parameters. The fuzzy relational composition, and the BK-Products of fuzzy relations can then be used to compute an optimal path.

Future Plans

Five refereed papers of this dissertation have been already published or accepted, registered, and scheduled for publishing and presentation at the time the dissertation is ready for the defense. As the research is cross-disciplinary and multi-directional, the future plans are also cross-disciplinary and multi-directional. On the Soft Computing science, the immediate plan is further development of the Soft Probability to cover all the types of discrete, continuous, and conditional probabilities. On this track, further study development of equation (5 – 5) will be required. The advantages of the equation were listed in Chapter Five. However, one weakness of the equation as it currently is still exists. The equation does not yield the desired results when all the probabilities being cumulated are low. The reason is that the formula has a slow starting growth rate. When at least one of the probabilities being cumulated is rather high, the computation escapes the slow growing section. The author of the current dissertation expects that the solution will be in the composition of equation (5 – 5) with another term that speeds up the growth rate of the lower section without causing the having the result exceed one. The ultimate test of Soft Probability will then be to eventually either prove that it is in accordance with the Strong Law of Large Numbers or provide an alternate limiting theorem.

As for the area of Quality of Service Networking, this dissertation concentrated on the solutions that use Fuzzy Computing and Probabilistic Computing. The future plans include the integration of Neural and Evolutionary Computing to provide the network with self-learning and self-tuning capabilities. The researcher also intends to work more on the theory of Optimal Control and its application to Quality of Service Networking. The expected path is also bi-directional, that is, to work on developments in the Multi-Variable Optimal Control Theory to construct a distributed computationally feasible model of such theories as Pontryagin Maximum Principle, and apply the developed model to the area of Quality of Service Distributed Computing.

It is also the hope and vision of the researcher that a comprehensive network protocol/architecture for intelligent adaptive networks could be designed and tested on a large physical network. The goal will be to develop a new AI-based networking technology where the network may be modeled and built as an intelligent adaptive autonomous system and the nodes subscribed to the network act as intelligent agents with distributed control capabilities, optimization goals and mechanisms, and apply neural-fuzzy-probabilistic computing to capture the uncertainty and chaos of the dynamical network in order to offer intelligent Quality of Service, Fault Tolerant, networking. The author believes such a technology would have a high chance of being accepted as a standard and adopted by the industry.

Speaking of developing and testing the new network technology, it is important here to mention that a new network simulator has been under development by the author of the current dissertation. The development of the new simulator, called NetSim, was necessitated by the desire and vision of the researcher to adapt a radically new approach to the field of networking. New networking modeling, control, routing, and communication techniques have been under development. Current network simulators do not support such technology. The new simulator is designed to study, model, and simulate the network as a discrete, dynamical, and multi-variable system with chaotic, possibly biotic, behavior. Although the new simulator has been developed as a matter of course under the current research, the study and utilization of the simulator is beyond the scope of the current dissertation. However, the future plans include the completion and expansion of NetSim, conducting experimental comparative studies to evaluate it against other simulators, and using it for the development and testing of the proposed AI-based networking techniques.

REFERENCES

[1] Abdelzaher, T. and K. Shin: *"End-host Architecture for QoS-Adaptive Communication*" IEEE Real-time Technology and Applications Symposium. Denver, Colorado, USA, June 1998.

[2] Aleksander, Igor and H. Morton: An Introduction to Neural Computing. Chapman and Hall, London, UK 1990.

[3] Anderson, J. and E. Rosenfeld (editors): Neurocomputing: Foundations of Research. MIT Press, Cambridge, MA, USA 1988.

[4] Apostolopoulos, G.; R. Guérin, S. Kamat, and S. Tripathi: *"Quality of Service Based Routing: A Performance Perspective"* ACM SIGCOMM, 1998.

[5] Apostolopoulos, G.; R. Guérin; S. Kamat; S. Tripathi: *"Improving QoS Routing Performance Under Inaccurate Link State Information"* 16th International Teletraffic Congress, June 1999.

[6] Arabshahi, P.; J. Choi, R. Marks, T. Caudell: *"Fuzzy Control of Backpropagation"* Proceedings of First IEEE International Conference on Fuzzy Systems. pp. 967-972, San Diego, CA, USA 1992.

[7] Armitage, Grenville: Quality of Service in IP Networks: Foundations for a Multi-Service Internet. MTP, Indianapolis, IN, USA 2000.

[8] Bandler, W. and L. J. Kohout: *"Relations, mathematical"* In M.G. Sigh, editor, Systems and Control Encyclopedia. Pergamon Press, Oxford, pp 4000-4008, 1987.

[9] Bandler, W. and L. J. Kohout: *"Special properties, closures and interiors of crisp and fuzzy relations"* Fuzzy sets and Systems 26(3): 317-332, June 1988.

[10] Bandler, W. and L. Kohout: *"Fuzzy Relational Products as a Tool for Analysis and Synthesis of the Behaviour of Complex Natural and Artificial Systems"* Fuzzy Sets: Theory and Applications to Ploicy Analysis and Information Systems. P. Wang and S. Chang editors, pp. 341-367, Plenum Press, NY, USA 1980.

[11] Bandler, W. and L. Kohout: *"Mathematical Relations, Their Products and Generalized Morphisms"* Technical Report, EES-MMS-REL 77-3, Man-Machine Systems Laboratory, Department of Electrical Engineering, University of Essex, Colchester, Essex, UK 1977.

[12] Bayes, Thomas.: *"An Essay Towards Solving a Problem in the Doctrine of Chances"* Philosophical Transactions of the Royal Society of London. 53 pp. 370-418, London, UK 1763. (Facsimile reproduction with commentary by E. Molina in [Deming 1940, 1963])

[13] Bernet, Yoram: Networking Quality of Service and Windows Operating Systems. New Riders Publishing, USA 2001.

[14] Bersini, H.; J. Nordvik, and A. Bonarini: *"A Simple Direct Adaptive Fuzzy Controller Derived from its Neural Equivalent"* IEEE International Confernece on Neural Networks. pp. 345-350, San Francisco, CA, USA 1993.

[15] Bersini, H.; J. Nordvik, and A. Bonarini: *"Comparing RBF and Fuzzy Inference Systems on Theoretical and Practical Basis"* International Conference on Artificial Neural Networks (ICANN95), pp. 169-174, Paris, France 1995.

[16] Bonissone, Piero: *"Soft Computing: The Convergence of Emerging Reasoning Technologies"* Soft Computing. Springer-Verlag, Germany/USA 1997.

[17] Bossomajer, Terry and David Green (editors): Complex Systems. Cambridge University Press, Cambridge, UK 2000.

[18] Cardano, Girolamo: The Book of My Life. trans. Jean Stoner. Toronto: J. M. Dent and Sons, Ltd., 1931.

[19] Chen, S. and K. Nahrstedt: *"An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions"* IEEE Network. Special Issue on Transmission and Distribution of Digital Video. November/December, 1998.

[20] Chen, S. and K. Nahrstedt: *"Distributed QoS Routing with Imprecise State Information"* International Conference on Computer and Communication Networks. October 1998.

[21] Chen, S. and K. Nahrstedt: *"Distributed Quality of Service Routing in High Speed Networks Based on Selective Probing"* 23rd IEEE/LCN Technical Program. Lowell, MA, USA, October 1998.

[22] Chen, S. and K. Nahrstedt: *"On Finding Multi-constrained paths"* IEEE International Conference on Communications. June 1998.

[23] Cheng, B. and D. Titterington: *"Neural Networks: a Review from a Statistical Perspective"* Statistical Science. Volume 9, 1994. pp.2-54.

[24] Cidon, I.; R. Rom, and Y. Shavitt: *"Multi-Path Routing combined with Resource Reservation*" IEEE INFOCOM. 1997.

[25] Cisco Systems, et al.: Internetworking Technologies Handbook, 3rd edition. Cisco Press, Indianapolis, IN, USA 2001.

[26] Cordon, O.; H. Herrera, and M. Lozano: A Classified Review on the Combination Fuzzy Logic-Genetic Algorithms Bibliography" Technical Report DECSAI95129, http://decsai.ugr.es/~herrera/fl-ga.html. Department of Computer Science and AI, Universidad de Granada, Granada, Spain, December 1996.

[27] Costa Branco, P. J. and J. A. Dente: *"Design of an Electro-Hydraulic System Using Neuro-Fuzzy Techniques"* Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms: Industrial Applications. CRC Press LLC, CRC Press LLC, Boca Raton, FL, USA 1999.

[28] Deasington, Richard: A Practical Guide to Computer Communications and Networking, 2nd edition. Ellis Horwood, England, UK 1984.

[29] DeFinetti, B.: La prevision: ses lois logiques, see sources subjectives. Annales de l'Institut H. Poincaré, 7 (1937) 1-68, sited by [Bonissone 1997].

[30] Deming, E.: Facsimile of Two Papers by Bayes. Washington, D.C. 1940. (Reprinted with commentary by G. Barnard in Biometrika, 25 pp. 293-315, 1970).

[31] Dempster, Arthur: *"Upper and Lower Probabilities Induced by Multivalued Mapping"* Annals of Mathematical Statistics. Volume 38, No. 2, 1967. pp. 325-339.

[32] Dodd, Annabel: The Essential Guide to Telecommunications, 3rd edition. Prentice Hall, Inc., NJ, USA 2002.

[33] Dossey, John A.; Albert D. Otto, Lawrence E. Spence, and Charles V. Eyden: Discrete Mathematics. 2nd edition. HarperCollins College Publishers, New York, NY, USA 1993.

[34] Dote, Yasuhiko; S. Taniguchi, and T. Nakane: *"Intelligent Control Using Soft Computing"* Proceedings of the 5th Online World Conference on Soft Computing in Industrial Applications (WSC5). 2000.

[35] Douligeris, C. and S. Palazzo: *"Fuzzy Expert Systems in ATM Networks"* Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms: Industrial Applications. CRC Press LLC, CRC Press LLC, Boca Raton, FL, USA 1999.

[36] Doyle, Jeff: CCIE Professional Development: Routing TCP/IP. Volume I. Cisco Systems, Macmillan Technical Publishing, Indianapolis, IN, USA 1998.

[37] Fausett, Laurene: Fundamentals of Neural Networks: Architectures, Algorithms, and Applications. Prentice Hall, NJ, USA 1994.

[38] Fogel, L.: *“Autonomous Automata”* Industrial Research. 4 (1962) pp. 14-19.

[39] Fogel, L.: Evolutionary Computation. IEEE Press, New York, NY, USA 1995.

[40] Freedman, Alan: The Computer Glossary: The Complete Illustrated Dictionary, 9th edition. AMACOM, New York, NY, USA 2001.

[41] Gallivan, Kyle: Introduction to Computational Science and Engineering. Lecture notes, Department of Computer Science, Florida State University, Tallahassee, FL, USA 1997.

[42] Gallivan, Kyle: Introduction to Parallel Programming. Lecture notes, Department of Computer Science, Florida State University, Tallahassee, FL, USA 1998.

[43] Garey, M. and D. Johnosn: Computers and Intractability: A Guide to the Theory of NP-Completeness Freeman and Co., 1979.

[44] Gerstel, Ori: *“On the Future of Wavelength Routing Networks”* IEEE Network Magazine, November 1996.

[45] Gerstel, Ori: On The Future of Wavelength Routing Networks. IBM, Watson Research Center, New York, NY, USA (date unavailable).

[46] Ghosh, Donna, Raj Acharya: *“A Probabilistic Approach to Hierarchical QoS Routing”*, 9th IEEE International Conference on Networks ICON2001, Bangkok, October 2001.

[47] Guérin, R. and A. Orda: *“QoS-based Routing in Networks with Inaccurate Information: Theory and Algorithm”* IEEE INFOCOM. 1997.

[48] Gurney, Kevin: An Introduction to Neural Networks. UCL Press, London, UK 1999.

[49] Halabi, Sam and Danny McPherson: Internet Routing Architectures, 2nd edition. Cisco Press, Indianapolis, IN, USA 2000.

[50] Hebb, Donald: The Organization of Behavior. John Wiley & Sons. NY, USA 1949. (Partly reprinted in [Anderson and Rosenfeld 1988]).

[51] Henrion, Max: *“Practical Issues in Constructing a Bayes’ Belief Network”* Proceedings of the Third Annual Conference on Uncertainty in Artificial Intelligence. Holland, (Kanal et al. editors) pp. 161-173, 1989.

[52] Herrera, F. and M. Lozano: *“Adaptive Genetic Algorithms Based on Fuzzy Techniques”* Proceedings of IPMU96. pp. 775-780, Granada, Spain 1996.

[53] Herrera, F.; M. Lozano, and J. L. Verdegay: *"Tackling Fuzzy Genetic Algorithms"* Genetic Algorithms in Engineering and Computer Science. (winter et al. editors) pp. 167-189, Wyley and Sons, 1995.

[54] Herrera, F.; M. Lozano, and J. L. Verdegay: *"Tuning Fuzzy Logic Control by Genetic Algorithms"* International Journal of Approximate Reasoning (IJAR) Volume 12, pp. 299-315, 1995.

[55] Holland, John: Adaptation in Natural and Artificial Systems. MIT Press, Cambridge, MA, USA 1992. (cited by [Mitchell 1995] and [Bonissone 1997])

[56] Holland, John: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI, USA1975. (cited by [Klir 1995])

[57] Horvitz, E.; J. Suermondt, and G. Cooper: *"Bounded Conditioning Flexible Inference for Decisions Under Scarce Resources"* Proceedings of the Fifth Workshop on Uncertainty in Artificial Intelligence. pp. 182-193, 1989.

[58] Huitema, Christian: Routing in the Internet, 2nd edition. Prentice Hall, Inc., Upper Saddle River, NJ, USA 2000.

[59] Jain, Lakhmi and N. Martin (editors): Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms: Industrial Applications. CRC Press LLC, Boca Raton, FL, USA 1998.

[60] Jang, Jyh-Shing: *"ANFIS: Adaptive Neural Fuzzy Inference Systems"* IEEE Transactions on Systems, Man, and Cybernetics. Volume 23, No. 3, pp. 665-685, May/June, 1993.

[61] Kacpzyk, Janusz (Editor): Advances in Soft Computing. Springer-Verlag, Heidelberg, Germany, 2001.

[62] Kacpzyk, Janusz (Editor): Studies in Fuzziness and Soft Computing. Springer-Verlag, Heidelberg, Germany, 2001.

[63] Kalman, R. E.; P. L. Falb; M. A. Arbib: Topics in Mathematical System Theory. McGraw-Hill, USA, 1969.

[64] Karr, C. L. and E. J. Gentry: *"Fuzzy control of pH using genetic algorithms"* IEEE Transactions on Fuzzy Systems. Volume 1, No. 1, pp. 46-53, February 1993.

[65] Karr, C. L.: *"Design of an Adaptive Fuzzy Logic Controller Using Genetic Algorithms"* Proceedings of International Conference on Genetic Algorithms (ICGA91) pp. 450-456, San Diego, CA, USA 1991.

[66] Karr, C. L.: *"Genetic Algorithms for Fuzzy Controllers"* AI Expert. Volume 6, No. 2, pp. 27-33, 1991.

[67] Kartalopoulos, Stamatios: Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications. IEEE Press, NJ, USA 1996.

[68] Kawamura, A.; N. Watanabe, H. Okada, K. Asakawa: *"A Prototype of Neuro-Fuzzy Cooperation Systems"* First IEEE International Conference on Fuzzy Systems. pp. 1275-1282, San Diego, CA, USA 1992.

[69] Khan, Emdad: *"Neural Fuzzy Based Intelligent Systems and Applications"* Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms: Industrial Applications. CRC Press LLC, CRC Press LLC, Boca Raton, FL, USA 1999.

[70] Kinzel, J.; F. Klawoon, and R. Kruse: *"Modifications of Genetic Algorithms for Designing and Optimizing Fuzzy Controllers"* Proceedings of First IEEE Conference on Evolutionary Computing (ICEC94) pp. 28-33, Orlando, FL, USA 1994.

[71] Kitano, H.: *"Empirical Studies on the Speed of Convergence of Neural Networks Training Using Genetic Algorithms"* 8th National Conference in Artificial Intelligence (AAAI90) pp. 789-796, Boston, MA, USA 1990.

[72] Klir, G. and B. Yuan: Fuzzy Sets and Fuzzy Logic: Theory: and Applications. Prentice Hall, NJ, USA 1995.

[73] Klir, George: Fuzzy Logic: Its meaning and Significance. Department of Systems Science and Industrial Engineering, Watson School of Engineering and Applied Science, State University of New York, Binghamton, New York, USA.

[74] Klir, George; U. St.Clair, and B. Yuan: Fuzzy Set Theory: Foundation and Applications. Prentice Hall, NJ, USA 1997.

[75] Kohout, Ladislav and W. Bandler: *"Fuzzy Interval Inference Utilizing The Checklist Paradigm and BK-Relational Products"* Applications of Interval Computations. R. Kearfott and V. Kreinovich editors, pp. 291-335, Kluwer Academic Publishers, Netherlands 1996.

[76] Kohout, Ladislav and W. Bandler: *"Fuzzy relational products in knowledge engineering"* In V. Novak et al., editor, Fuzzy Approach to Reasoning and Decision Making. pp 51-66, Academia and Kluwer, Prague and Dordrecht, Zchec 1992.

[77] Kohout, Ladislav: *"Boolean and fuzzy relations"* In P.M. Pardalos and C.A. Floudas, editors, The Encyclopedia of Optimization. Kluwer, Boston, MA, USA 2001.

[78] Kohout, Ladislav: *"Checklist Paradigm Semantics for Fuzzy Logics"* The Encyclopedia of Optimization. Kluwer, Boston, 1999. In press.

[79] Kohout, Ladislav: *"Foundations of Knowledge Engineering for Systems with Distributed Intelligence: A relational Approach"* Encyclopedia of

Microcomputers. A. Kent and J. Williams editors, Vol. 24, Supplement 3, Marcel Dekker Inc., NY, USA 2000.

[80] Kohout, Ladislav: *"Notes on Fuzzy Logics"* Class Notes, Fuzzy Systems and Soft Computing. Department of Computer Science, Florida State University, Tallahassee, FL, USA 1999.

[81] Kohout, Ladislav: A Perspective on Intelligent Systems: A Framework for Analysis and Design. Chapman and Hall Computing, London, UK 1990.

[82] Kohout, Ladislav; B. Granville, and E. Kim: *"Granular Relational Computing with Semiotic Descriptors Using BK-Products of Fuzzy Relations"* Computing with Words. John Wiley & Sons, Inc., New York, NY, USA 2001.

[83] Kohout, Ladislav; J. Anderson, and W. Bandler (editors): Knowledge-Based Systems for Multiple Environments. Ashgate Publications, Aldershot, UK 1992

[84] Köppen-Seliger, B. and P. M. Frank: *"Fuzzy Logic and Neural Networks in Fault Detection"* Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms: Industrial Applications. CRC Press LLC, CRC Press LLC, Boca Raton, FL, USA 1999.

[85] Koza, J.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA, USA 1992.

[86] Lauritzen, S. and D. Spiegelhalter: *"Local computations with probabilities on graphical structures and their applications to expert systems"* Readings in Uncertain Reasoning. (G. Shafer and J. Pearl Morgan, editors) Kaufmann, San Mateo, California. pp. 415-448, 1990.

[87] Lavagetto, F.: *"Multimedia Telephone for Hearing-Impaired People"* Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms: Industrial Applications. CRC Press LLC, CRC Press LLC, Boca Raton, FL, USA 1999.

[88] Lee, M. A. and H. Takagi: *"Dynamic Control of Genetic Algorithm Using Fuzzy Logic Techniques"* Proceedings of the Fifth International Conference on Genetic Algorithms. (S. Forrest editor) pp. 76-83, Morgan Kaufmann 1993.

[89] Lee, M. A. and H. Takagi: *"Integrating Design Stages of Fuzzy Systems Using Genetic Algorithms"* Proceedings of Second IEEE Conference on Fuzzy Systems (FUZZ-IEEE93). pp. 612-617, San Francisco, CA, USA 1993.

[90] Lee, S. and E. Lee: *"Fuzzy Sets and Neural Networks"* Journal of Cybernetics. Volume 4, No. 2, pp. 83-013, 1974.

[91] Leitmann, George: The Calculus of Variations and Optimal Control: An Introduction. Plenum Press, New York, N.Y., USA 1981.

[92] Lorenz, D. H. and A. Orda: *"QoS Routing in Networks with Uncertain Parameters"* IEEE, INFOCOM 98. March 1998.

[93] Ma, Q. and P. Steenkiste: *"Quality of Service Routing with Performance Guarantees"* Proceedings of the 4th International IFIP Workshop on Quality of Service. May 1997.

[94] Mamdani, E. and S. Assillian: *"An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller"* International Journal of Man Machine Studies. 7(1), pp. 1-13, 1975. (cited by [Bonissone 1997])

[95] Maniezzo, V.: *"Genetic Evolution of the Topology and Weight Distribution of Neural Networks"* IEEE Transactions on Neural Networks. Volume 5, No. 1, pp. 39-53, 1994

[96] Marti, P. and M. Rueher: A Distributed Cooperating Constraints Solving System. Université de Nice, France 1995.

[97] McCulloch, Warren and W. Pitts: *"A Logical Calculus of the Ideas Immanent in Nervous Activity"* Bulletin of Mathematical Biophysics. Volume 5, 1943. pp. 115-133. (Reprinted in [Anderson and Rosenfeld 1988])

[98] Mendel, Jerry: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice Hall PTR, NJ, USA 2001.

[99] Mendenhall, W.: Introduction to Probability and Statistics, 2nd ed., Wadsworth Publishing Company, Inc. Belmont, CA, USA, 1968.

[100] Mitchell, Melanie: An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA, USA 1999.

[101] Moussa, Ahmed and L. Kohout: *"A New Network State Information Updating Policy Using Fuzzy Tolerance Classes and Fuzzified Precision"* Proceedings of NAFIPS/FLINT-2002, pp. 541-545, IEEE, New Orleans, LA, USA, June 2002.

[102] Moussa, Ahmed and L. Kohout: *"Fuzzy Probabilistic Quality of Service Routing for Uncertain Networks"* Proceedings of JCIS/FT&T-2003 (to be published) Cary, NC, USA, September 2003.

[103] Moussa, Ahmed and L. Kohout: *"Using BK-Products of Fuzzy Relations in Quality of Service Adaptive Communication"* Proceedings of IFSA/NAFIPS-2001, pp. 681-686, IEEE, Vancouver, Canada, July 2001.

[104] Moussa, Ahmed: *"Questions on Probability Theory and its Effect on Probabilistic Reasoning"* Proceedings of JCIS-2003 (to be published) Cary, NC, USA, September 2003.

[105] Moussa, Ahmed: *“Soft Probability: Motivation and Foundation*” Proceedings of IPSI-2003 (to be published) Sveti Stefan, Montenegro, October 2003.

[106] Moussa, Ahmed: The Fuzziness of Truth and Reality: Application to the concept of color. Term paper for the course PHI-6935 Realism and Objectivity, Department of Philosophy, Florida State University, Tallahassee, FL, USA, Summer 2001.

[107] Nielsen, Michael and Isaac Chuang: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge, UK 2001.

[108] NREN: *“Research/Quality of Service”* NASA Research & Education Network (NREN) http://www.nren.nasa.gov/research/index.html. Last updated, July 2001.

[109] Ore, Oystein: Cardano: The Gambling Scholar. Princeton: University Press, 1956.

[110] Pacheco, Peter: Parallel Programming with MPI. Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA 1997.

[111] Patel, M. J. and V. Maniezzo: *“NN's and GA's: Evolving Co-operative Behaviour in Adaptive Learning Agents”* Proceedings of the First IEEE World Congress on Evolutionary Computation. pp. 290-295, Orlando, FL, USA 1994.

[112] Patterson, David and John Hennessy: Computer Architecture: A Quantitative Approach. Second edition. Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA 1996.

[113] Pearl, J.: *“Fusion, Propagation, and Structuring in Belief Networks”* Artificial Intelligence. Volume 29, pp. 241-288, 1986.

[114] Pearl, J.: *“On evidential Reasoning on a Hierarchy of Hypothesis”* Artificial Intelligence. Volume 38, pp. 9-15, 1988.

[115] Pearl, J.: *“Reverend Bayes on Inference Engines: a Distributed Hierarchical Approach”* Proceedings of Second National Conference on Artificial Intelligence. pp. 133-136, 1982.

[116] Pearl, J.: *Evidential Reasoning Under Uncertainty”* Exploring Artificial Intelligence. pp. 381-418, [Shrobe editor, 1988]

[117] Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan-Kaufmann, San Mateo, CA, USA 1988.

[118] Perkins, Charles (editor): Ad Hoc Networking. Addison-Wesley, USA 2001.

[119] Perkins, Charles: *“Ad Hoc Networking: An Introduction”* Ad Hoc Networking [Perkins (editor) 2001]. Addison-Wesley, USA 2001.

[120] Petrov, Iu. P.: Variational Methods In Optimum Control Theory. Academic Press Inc., New York, N.Y., USA 1968.

[121] Pfaffenberger, Bryan: Webster's New World Dictionary of Computer Terms, 8th edition. IDG Books Worldwide, Inc., Foster City, CA, USA 2000.

[122] Pierre, S.; S. Hassane, and W. G. Probst: *"An artificial Neural Network Approach for Routing in Distributed Computer Networks"* Engineering Applications of Artificial Intelligence. International Journal of Intelligent Real-Time Automation. Vol. 14, No. 1, February 2001.

[123] Plevyak, Thomas (editor): IEEE Communications Magazine. (Special Issue: Fuzzy and Neural Networks) Volume 30, No. 9, September 1992.

[124] Potvin, J-Y and S. R. Thangiah: *"Vehicle Routing Through Simulation of Natural Processes"* Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms: Industrial Applications. CRC Press LLC, CRC Press LLC, Boca Raton, FL, USA 1999.

[125] Qiao, C. and M. Yoo: Optical Burst Switching (OBS) – A New Paradigm for an Optical Internet. LANDER, University at Buffalo, Buffalo, New York, USA.

[126] Ramsay, Allan: *"Dempster-Shafer Theory of Evidence"* Lecture Notes on Logic. Department of Language and Linguistics, UMIST, UK 1999.

[127] Rechenberg, I.: Cybernetic Solution Path of an Experimental Problem. Royal Aircraft Establishment, Library translation No. 1122, 1965.

[128] Renders, J-M. and H. Bersini: *"Hybridizing Genetic Algorithms with Hill-Climbing Methods for Global Optimization: Two Possible Ways"* Proceedings of the First IEEE Conference On Evolutionary Programming. pp. 312-317, Orlando, FL, USA 1994.

[129] Rosenbaltt, F.: *"Two Theorems of Statistical Separability in the Perceptron"* Mechanization of Thought Processes. pp. 421-456, Symposium at National Physical Laboratory, London, UK 1959.

[130] Ross, Sheldon: A First Course in Probability. 4th Edition, Prentice Hall, Engleweood Cliffs, NJ, USA, 1994.

[131] Ross, Timothy J.; J. M. Booker, and W. J. Parkinson (editors): Fuzzy Logic and Probability Applications: Bridging the Gap. ASA-SIAM, VA-PA, USA, 2002.

[132] Rudin, Walter: Principles of Mathematical Analysis, 2nd ed., McGraw-Hill, New York, USA, 1964.

[133] Saadawi, Tarek; M. Ammar; A. El Hakeem: Fundamentals of Telecommunication Networks. John Wiley & Sons, Inc. New York, NY, USA 1994.

[134] Sackett, George: Cisco Router Handbook. McGraw-Hill, New York, NY, USA 2000.

[135] Salama, H. F.; D. S. Reeves, and Y. Viniotis: *"A Distributed Algorithm for Delay-constrained Unicast Routing"* INFOCOM 97. Japan, April 1997.

[136] Samuel, Pierre; H. Said, and W. G. Probst: *"An Artificial Neural Network Approach for Routing in Distributed Computer Networks"* Engineering Applications of Artificial Intelligence: The International Journal of Intelligent Reat-Time Automation. Volume 14, No. 1, R. A. Vingerhoeds (editor), February 2001.

[137] Schwefel, H.: Kybernetische Evolution als Strategie dir Experimentellen Forschung in der Strömungstechnik. Ph.D. Dissertation. Technical University of Berlin, Germany 1965.

[138] Shafer, Glenn: *"Perspectives on the Theory and Practice of Belief Functions"* International Journal of Approximate Reasoning. Volume 4, 1990. pp. 323-362.

[139] Shafer, Glenn: A Mathematical Theory of Evidence. Princeton University Press, Princeton, NJ, USA 1976.

[140] Shaikh, A.; J. Rexford; K. Shin: *"Evaluating the Impact of Stale Link State on Quality of Service Routing"*. IEEE/ACM Transactions on Networking. April 2001.

[141] Shaikh, Anees; J. Rexford, and K. Shin: *"Evaluating the Overheads of Source-Directed Quality 0f Service Routing"* International Conference on Network Protocols (ICNP), 1998.

[142] Shaikh, Anees; J. Rexford, and K. Shin: Dynamics of Quality of Service Routing with Inaccurate Link State Information. Technical Report CSE-TR-350-97, Computer Science and Engineering Division, Dept. of Electrical Engineering and Computer Science, University of Michigan, November 1997.

[143] Shankar, M.; M. De Miguel, and J. Liu: *"An End-to-End QoS Management Architecture"* Proceedings of the Fifth Real-Time Technology and Applications. Symposium. Vancouver, Canada, IEEE June 1999.

[144] Shimojima, Koji and T. Fukuda: *"A New Fuzzy-Neural Controller"* Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms:

Industrial Applications. CRC Press LLC, CRC Press LLC, Boca Raton, FL, USA 1999.

[145] Shin, K. G. and C. Chou: *"A Distributed Route Selection Scheme for Establishing Real-Time Channel"* Sixth IFIP International Conference on High Performance Networking (HPN95), pp. 319-329, September 1995.

[146] Shrobe, Howard: Exploring Artificial Intelligence. Morgan-Kaufmann, San Mateo, CA, USA 1988.

[147] Slattery, Terry and Bill Burton: Advanced IP Routing in Cisco Networks, 2nd edition. McGraw-Hill, Inc., USA 2000.

[148] Smets, P.: *"The Degree of Belief in a Fuzzy Event"*, Information Sciences 25, pp. 1-19, 1981.

[149] Stark, L.; M. Okajima, and G. Whipple: *"Computer Pattern Recognition Techniques: Electrocardiographic Diagnosis"* Communications of the ACM. 5, 1962. pp. 527-532.

[150] Suarez, Tiki: Intelligent Wireless Management: Improving TCP Performance Over Wireless Networks. Ph.D. Dissertation, Department of Computer Science, Florida State University, Tallahassee, FL, USA 2001.

[151] Suermondt, J.; G. Copper, and D. Heckerman: *"A Combination of Cutset Conditioning with Clique-Tree Propagation in the Pathfinder System"* Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence. Amsterdam, Holland, (Bonissone *et al.* editors) pp. 245-253, 1991.

[152] Sun, Qiang and H. Langendorfer: *"A Distributed Delay-Constrained Dynamic Multicast Routing Algorithm"* The European Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS'97), 1997.

[153] Takagi, H.: *"Fusion Technology of Fuzzy Theory and Neural Networks: Survey and Future Directions"* IIZUKA90: International Conference on Fuzzy Logic and Neural Networks. pp. 13-26, Iizuka, Japan 1990.

[154] Tanaka, Makoto: *"Application of The Neural Network and Fuzzy Logic to The Rotating Machine Diagnosis"* Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms: Industrial Applications. CRC Press LLC, CRC Press LLC, Boca Raton, FL, USA 1999.

[155] Tomsu, Peter and Christian Schmutzer: Next Generation Optical Networks: The Convergence of IP Intelligence and Optical Technologies. Prentice Hall, Inc., Upper Saddle River, NJ, USA 2002.

[156] Vegesna, Srinivas: IP Quality of Service. Cisco Press, Indianapolis, IN, USA 2001.

[157] Verma, Pramode: Performance Estimation of Computer Communication Networks: A structured Approach. Computer Science Press, Rockville, MD, USA 1989.

[158] Wang, Paul (editor): Computing with Words. John Wiley & Sons, Inc., New York, NY, USA 2001.

[159] Wang, Z. and J. Crowcroft: *"QoS Routing for Supporting Resource Reservation"* JSAC. September 1996.

[160] Wang, Z. and J. Crowcroft: *"QoS Routing for Supporting Resource Reservation"* IEEE Journal on Selected Areas in Communications, September 1996.

[161] Widrow, B. and M. Hoff: *"Adaptive Switching Circuits"* IRE WESCON Convention Record. Part 4, pp. 96-104, IRE, NY, USA 1960.

[162] Widrow, Bernard: *"ADALINE: An Engineering Model for Neurocomputing"* IIZUKA'90: International Conference on Fuzzy Logic and Neural Networks. Iizuka, pp. 5-9, Japan 1990.

[163] Wierzchoń, Sławomir and Mieczysław Kłopotek: Data and Knowledge Mining in Extended Bayesian Networks. Springer-Verlag, Still under preparation.

[164] Yuan, Xin and G. Yang: "*Empirical Probability Based QoS Routing"*. IEEE International Conference on Communications *(ICC 2003)*, Pages 1713-1717, Anchorage, Alaska, May 11-15, 2003.

[165] Yuan, Xin, W. Zheng, and S. Ding: "*A Comparative Study of Quality of Service Routing Schemes That Tolerate Imprecise State Information"*. the 11th IEEE International Conference on Computer Communications and Networks (IC3N'02), pages 230-235, Miami, Florida, October, 14-16, 2002.

[166] Yuan, Xin: On the Extended Bellman-Ford Algorithm to Solve Two-Constrained Quality of Service Routing Problems. Deparment of Computer Science, Florida State University, Tallahassee, FL, USA.

[167] Zadeh, Lotfi: *"From Computing with Numbers to Computing with Words – from Manipulation of Measurements to Manipulation of Perceptions"* Plenary Speaker, Proceedings of IEEE International Workshop on Soft Computing in Industry. Muroran, Japan 1999.

[168] Zadeh, Lotfi: *"Fuzzy Logic and Softcomputing"* Plenary Speaker, Proceedings of IEEE International Workshop on Neuro Fuzzy Control. Muroran, Japan 1993.

[169] Zadeh, Lotfi: *"Fuzzy Sets"* Information and Control. 8(3), pp.338-353, 1965. (Cited by [Klir 1995, 1997], [Bonissone 1997])

[170] Zadeh, Lotfi: *"Probability Measures of Fuzzy Events"*, Journal of Mathematical Analysis and Applications, pp. 421 – 427, vol. 23, 1968.

[171] Zadeh, Lotfi: *"The Role of Soft Computing and Fuzzy Logic in the Conception, Design, Development of Intelligent Systems"* Plenary Speaker, Proceedings of the International Workshop on soft Computing Industry. Muroran, Japan, 1996.

[172] Zadeh, Lotfi: *"What is Soft Computing"* Soft Computing. Springer-Verlag Germany/USA 1997.

BIOGRAPHICAL SKETCH

Ahmed Shawky Moussa was born and educated in Cairo, Egypt and at the time of writing this manuscript is a PhD candidate at the Department of Computer Science, Florida State University (FSU), Tallahassee, Florida, USA, where he also works as a research assistant and a lecture instructor. He obtained his MS in computer science also at FSU. FSU filed and sponsored a patent application on his Master's degree work.

Prior to the PhD work in computer science, he worked as an assistant faculty at the Department of Music Theory and Composition, Helwan University, Cairo, Egypt. He received the Fulbright post-doctorate scholarship in 1994 after ranking number one nation-wide – all fields – in the annual competition for post-doctoral research, before he was even registered in any doctoral program.

He also holds degrees and certificates in music, computational science, and cognitive sciences. In music, he has accomplishments in the fields of education, composition, and performance – as a classical guitarist. He also has accomplishments in sports and was a member of the national team of Judo in Egypt. More details are in the following Curriculum Vitae.

1. Education

- **PhD** Computer Science, Florida State University, Tallahassee, FL, USA Fall 2003. Dissertation title: *"The Implementation of Intelligent QoS Networking by the Development and Utilization of Novel Cross-Disciplinary Soft Computing Theories and Techniques",* successfully defended in September 2003.
- **MS** Computer Science, Florida State University, Tallahassee, FL, USA Spring 2001. Thesis title: *"A Dynamic Microtunable MIDI System: Problems and Solutions"*, successfully defended in December 2000.

- **MA** Music Education, Helwan University, Cairo, Egypt, 1993. Thesis title: "*The Role of the Computer in the Fields of Music*", successfully defended in October 1993.
- **BA** Music Education, Helwan University, Cairo, Egypt, 1984.
- **Certificate** in Computational Science and Information Technology, Florida State University, Tallahassee, FL, USA, completed, Fall 2002.
- **Certificate** in Cognitive Sciences, Florida State University, Tallahassee, FL, USA, completed, Spring 2003.
- **Secondary Diploma**, Music Theory and Composition, Academy of Arts, Cairo, Egypt, 1983.
- **Non-degree Studies**, Departments of Computer Science and Physics, American University in Cairo (1993 - 1996) and College of Physical Education, Helwan University, Cairo, Egypt (1978).

2. Professional Employment History

Full-time/Main Jobs:

- Tenure-track Assistant, Department of Music Theory and Composition, College of Music Education, Helwan University, Cairo, Egypt 1985 - 1993.
- Promoted to Assistant Faculty 1993 – 1996.
- (Sound & Recording) Engineer, Studio of Ishaa, Helwan University, Cairo, Egypt 1992 - 1996.
- Fulbright Post-Doctoral Visiting Scholar, Center for Music Research, Florida State University, 1994.
- Teaching Assistant, Department of Computer Science, Florida State University, Fall 2000 – Spring 2001.
- Lecture Instructor, Department of Computer Science, Florida State University, Summer 2001 – Summer 2003.
- Cross-departmental Researcher and Research Assistant at Florida State University, 1997 – present.

Part-time/Side Jobs:

- MIDI/Sound technology specialist for the Arabic Music Conservation project, Information and Decision Support Center (IDSC), Cabinet of Egypt 1995 – 1996.
- Visiting Instructor, College of Qualitative Education, Qena, Egypt 1995 – 1996.
- Classical Guitar teacher/performer 1983 – 1996.

3. Research Experience

- **Theoretical and Applied Artificial Intelligence (Intelligent Systems, Soft Computing, and Optimal/Adaptive Control):** The Solutions I developed for QoS networking in my PhD research are based on new methods for the integration of fuzzy and probabilistic computing with theories of complex systems and optimal control. The goal is to develop a model for complex dynamical systems with autonomous AI and distributed control capabilities. The techniques are applied on large dynamical networks as an example of such a complex chaotic system with the goal of achieving dependable computing under uncertainty. The dissertation included innovations in both Fuzzy Set Theory and Probability Theory. (2001 – present)
- **QoS Networking:** QoS Networking is the area of application of my PhD work for which I developed a new approach for solving QoS networking problems under uncertainty as part of my PhD research by developing integrated cross-disciplinary AI solutions for the modeling, global state maintenance, and routing. (2001 – present)

- **Multimedia and MIDI Systems:** Developed solutions for the problem of pervasive, dynamic micro-tuning in MIDI networks as my MS research.
- **U-Tune, DMB Hardware Prototyping (co-developer):** This is the hardware prototyping project of my patented invention, which is the outcome of my Master's thesis in Computer Science at Florida State University, 2002 – present.
- **Ada POSIX Testing:** Research Assistant, Dept. of Computer Science, Florida State University, Summer 1998.
- **Acoustical Analysis and Design:** Cross-departmental Research Assistant with Dr. Michael Kasha, Acoustical Analysis and Design Research Laboratory, Florida State University, 1997 – present.
- **Computers in Music and Recording Arts:** A number of research and development projects on applying computer technology in music education, composition, and performance, College of Music Education, Helwan University, 1990 – 1996 and Center for Music Research, Florida State University, 1994.

4. Teaching Experience

- At FSU, I taught Introductory, Intermediate, and Advanced Computer Programming in C and C++ as follows:
  - Teaching assistant/Grader, recitation/demonstration, average classes size 20 students, Fall 2000 – Spring 2001.
  - Lecture instructor, average size 120 students, Summer 2001- Summer 2003.
  - FSU Instructor for the Young Scholars Program (YSP), Introductory Computer Programming in C/C++, 2000, 2001, and Advanced Computer Programming in C/C++, 2002, 2003 (Summer courses).
    - As a faculty assistant and assistant faculty at Helwan University and as a visiting instructor at the College of Qualitative Education, I have taught, as a main instructor, a variety of music related subjects, some of which were computer music and Introduction to Computers. Other courses I taught were in sound and MIDI technology (recording & processing). Non-technology related courses included Music Theory, Harmony, Analysis and Perception, History, and Instrumentation. (class sizes range from 20 to 60 students)
    - As a Fulbright Scholar at FSU, I participated in teaching a summer course in MIDI and Music Technology for Educators, Summer 1994.
    - As a classical guitar teacher, I initiated the program and taught in a number of places among which are Spanish Cultural Center, Claudio Monteverdi Conservatory, and British International School (all in Egypt).

5. Other, Relevant Professional Experience and Skills

- My job as a studio engineer also included computerizing, and periodically upgrading, the recording studio, and expanding it by including a computer multimedia unit and a computer laboratory for music and related applications. This involved designing policies, writing and defending proposals, receiving and managing funds, and executive supervision of process development.
- Designing and building anechoic chambers with experience in Dr. Kasha's patented portable anechoic chambers.
- Building Linux-based computer clusters for high performance, parallel, and distributed computing.

6. Publications

Projected Refereed Journal Publications:

[1] Ahmed Shawky Moussa: *"The Design of Intelligent Adaptive Network State Updating Mechanism for Netowrk performance Optimization under Uncertainty"*. IEEE/ACM Transactions on Networking (Submitted, a supplement is required, in revision for reconsideration).

[2] Ahmed Shawky Moussa: *"Pervasive Micro-tuning Over MIDI Networks: A Perception-Based Technique"*. IEEE Pervasive Computing (Submitted, passed initial reviewing, currently in refereeing).

[3] Ahmed S. Moussa and Ladislav Kohout: *"Building Communication Networks as Intelligent Multi-Body self-adaptive Systems"*. Journal of Intelligent Systems (In Submission).

[4] Ahmed Shawky Moussa: *"Multi-constraints QoS Routing in Large-scale Dynamical Networks: an Intelligent Fuzzy Probabilistic Approach"*. IEEE/ACM Transactions on Networking (In Submission).

[5] Ahmed Shawky Moussa: *"Soft Probability: Computing Probabilities for the Real World"*. Nature (In Submission).

Refereed Conferences Proceedings and Presentations:

[1] Ahmed S. Moussa and Ladislav Kohout: *"Using BK-Products of Fuzzy Relations in Quality of Service Adaptive Communications"*. IFSA/NAFIPS-2001, Vancouver, Canada, pp. 681-686, July 2001 (published and presented).

[2] Ahmed S. Moussa and Theodore Baker: *"A Dynamic Microtunable MIDI System: Problems and Solutions"*. WAC-2002, Orlando, FL, USA, June 2002 (published and presented).

[3] Ahmed S. Moussa and Ladislav Kohout: *"A New Network State Updating Policy Using Fuzzy Tolerance Classes and Fuzzified Precision"*. NAFIPS/FLINT-2002, New Orleans, LA, USA, pp. 541-545, June 2002 (published and presented).

[4] Ahmed S. Moussa and Ladislav J. Kohout: *"Fuzzy Probabilistic QoS Routing for Uncertain Networks"* JCIS-2003, Research Triangle Park, North Carolina, USA, September 2003 (published and presented).

[5] Ahmed Shawky Moussa: *"Questions on Probability Theory and Its Effect on Probabilistic Reasoning"*. JCIS-2003, Research Triangle Park, North Carolina, USA, September 2003 (published and presented).

[6] Ahmed Shawky Moussa: *"Justification of the Logical Implication of Propositional Calculus: Proof by Incremental Constructive Reasoning"*. JCIS-2003, Research Triangle Park, North Carolina, USA, September 2003 (published and presented).

[7] Ahmed Shawky Moussa: *"Soft Probability: Motivation and Foundation"*. IPSI-2003, Montenegro, October 2003 (published and presented).

[8] Michael Kasha, Thomas Phipps, Dapo Sanu, and Ahmed S. Moussa: *"Development of New String-Instruments of the Violin-Family and Guitar-Family via Engineering Physics and Computer-Resolved Harmonic Spectral Intensities"*. IPSI-2003, Montenegro, October 2003 (published and presented).

In Preparation:

[1] Ahmed Shawky Moussa: *"Network Modeling and Simulation as a Biotic Discrete Dynamical System"*.

[2] Ahmed Shawky Moussa: *"Automatable Reasoning in Propositional Logic"*.

[3] Ahmed Shawky Moussa: *"The Introduction of a New Fuzzy Set Operation to Preserve the Law of Excluded Middle and the Law of Contradiction"*. Or *"The Missing Concept Necessary for the Preservation of the Law of Excluded Middle and the Law of Contradiction in Fuzzy Set Theory"*.

[4] Ahmed Shawky Moussa: *"The Mutual Exclusion and its Effect on the Accuracy and Validity of Probabilistic Computing"*.

[5] Ahmed Shawky Moussa: "*Harmonics: Altering the Harmonic Spectrum of String Instruments by the Performer*".

[6] Kasha *et. al.*: Series of papers on acoustical analysis and the utilization of computational physics in the design of musical instruments.

Non-Refereed, Non-Computer Science Publications:

[1] When working as a MIDI/Sound technology specialist for the Arabic Music Conservation project, IDSC, Cabinet of Egypt 1995 – 1996, I participated with a team in the production of a CD-ROM.

[2] *"Klassiche Guitarre hat keine tradition in Agypten"*, interview with the German magazine Gitarre and Laute. March/April issue, 1994.

[3] Mr. Guitar Vol. 10, Ahmed Shawky classical guitar solo performance album released as a CD in Europe and as a cassette tape in Egypt, 1996.

7. Patents

- "Dynamic Microtunable MIDI Interface Process and Device". FSU Research Foundation has patented the techniques and ideas I developed in my MS thesis. A Hardware prototyping project was funded by the FSU-RF. The prototype was successfully completed in June 2003. The project is in the attempted commercialization phase now by support from FSU-Technology Transfer.
- A second patent is hopefully going to be filed soon by FSU-RF (or my next university affiliation), for a new intelligent insulin pump for diabetes treatments.

8. Grants

- Patent prototyping project, FSU grant number 1327-583-45. ($14,250.90).

9. Honors and Awards

- Fulbright Postdoctoral Scholarship after winning the first rank order in Egypt for the post-doctoral research nationwide in the 1992 annual competition.
- FSU dissertation grant award, Fall 2002.
- Outstanding Teaching Assistant of The Year award in Computer Science at FSU, academic year 2002 – 2003.

10. Memberships

- NAFIPS – North American Fuzzy Information Processing Society.
- ACM – Association of Computing Machinery.
- Fulbright Alumni of Egypt
- IMA-International MIDI Association.

- Jeunesse Musicale of Egypt.

11. Major Interests.

My hobbies are my professions. I do what I enjoy and enjoy what I do. I take all the things I do equally seriously and find them equally enjoyable. However, for scheduling purposes, I shifted my main concentration along the way, which now seem to remain, for the rest of my life, on Computing and Computational Science.

Science: While working on obtaining a Ph.D. in Computer Science, I have taken classes in Electrical Engineering, Physics, Sound Engineering, Mathematics, Philosophy, and Computational Science in addition to core and special topics Computer Science classes. I'm also interested in reading in the Arts, philosophy, and History. Searching for knowledge is an enjoyable hobby as well as a career for me.

Music: Studied Music Education, Composition, and Classical Guitar. Have worked in all three fields on a professional and semi-professional level. Have one published recording, as indicated above, and have performed many concerts in Egypt, and some in Europe. I practiced, and still enjoy, other arts too but never as deeply as I did music.

Sports: Was a member of the Egyptian national team of Judo in addition to reaching high levels in other sports, mostly martial arts, but I have done many other sports such as soccer, motorcycles, yoga, and chess. Also studied in the College of Physical Education for one year and am still interested in the scientific aspects of sports. In general, I enjoy practicing and learning all types of sports and outdoor activities.

12. Languages and International Experience

Arabic, English, French, and Japanese. I'm not as fluent in French and Japanese anymore as I am in Arabic and English due to the lack of practicing. I lived in, and visited, Egypt, USA, France, England, Spain, Hungary, Canada, and Montenegro.