Computer Decision Aiding under Soft Boundaries with Application to Freight Train Dispatching

by

JIE REN

B.S. in Mechanical Engineering
Shanghai Jiao Tong University
(1982)

M.S. in Mechanical Engineering
Massachusetts Institute of Technology
(1987)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

December 1993

© Massachusetts Institute of Technology 1993. All rights reserved.

Author

Department of Mechanical Engineering
December 7, 1993

Certified by

Thomas B. Sheridan
Thesis Supervisor

Accepted by

Ain A. Sonin
Chairman, Departmental Committee

ARCHIVES
MASSACHUSETTS INSTITUTE
OF TECHNOLOGY
MAR 08 1994
Computer Decision Aiding under Soft Boundaries
with Application to Freight Train Dispatching

by

JIE REN

Submitted to the Department of Mechanical Engineering

on December 7, 1993, in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN MECHANICAL ENGINEERING

Abstract

Deterministic and probabilistic mathematical approaches are used to model purely
physical processes, which obey well-defined physical laws. However these methods often
fail when being applied to human-in-the-loop systems where both measurements and
decision-making processes are affected by human subjectivity. In a real environment in
which deterministic models and human expertise coexist, it usually would be too strict and
non-responsive to use only an algorithmic mathematical model, but it would be also far
too unstructured to try to solve the problems by heuristic rules alone.

This thesis addresses such an environment and sets a formal framework of decision
aiding to handle quantitative and qualitative information simultaneously for better
decision making. Specifically, this thesis investigates various methods of dealing with the
uncertainty involved in human knowledge and the corresponding data fusion. It employs
fuzzy set theory as a main vehicle to handle imprecise human expertise and soft
boundaries within a resource-constrained mathematical model.

A novel decision framework - fuzzy linear programming coordinated with a fuzzy
linguistic knowledge base - is proposed as a new approach. In this approach, fuzzy linear
programming is used to model quantitatively the basic resource constraints and objective
functions, while a linguistic knowledge base is used to model imprecise human expertise.

The proposed technique is used in aiding of railroad freight train dispatching as a case
study, where subtle factors require human yardmasters to play an active decision-making
role and optimization in terms of the rail track network and the published schedule is too
complex for a human to do by himself. Simulation suggests that using the new decision
aid can improve the overall performance of freight train dispatching. The success of this
case study indicates that the proposed decision-making framework can be applied to other
decision environments where some mathematical model is available and human
involvement is important.

Thesis Supervisor : Dr. Thomas B. Sheridan
Title : Professor of Engineering and Applied Psychology
and Professor of Aeronautics and Astronautics
Thesis Committee: Professor Thomas B. Sheridan
Professor Haruhiko Asada
Professor Harilaos N. Koutsopoulos
For my son, *Steven Y. Ren*
Acknowledgments

I would like to express my deepest gratitude to Professor Thomas B. Sheridan, my thesis supervisor, for his inspiring advice and constant encouragement both academically and mentally through all my years in MIT. Without him, this thesis would never have been written.

A special thank to my committee members: Professor Haruhiko Asada and Professor Harilaos N. Koutsopoulos for their sound advice and keen insight that helped me in many ways to complete this thesis.

I would like to thank Dr. Carl Martland, Dr. Patrick Little and their rail group at MIT Center for Transportation Studies for the support and valuable advice.

Many thanks to the current members of the Human-Machine Systems Laboratory for the companionship and support: Shih-Ken Chen, with him I shared the same culture as well as the computer code; Nicholas Patrick, the real airplane pilot in the lab.; David Schloerb, with whom, I missed registering for pistol class together; Shumei Yin, the heroine who inspired the lab.'s name revolution; James Thompson, the genuine Harvard M.D.; Thomas Chao, the new blood for the lab.; and Suyeong Kim, Ed Lanzilotta, Jianjuen Hu and Sheit Sheng Chen; as well as the graduates of the former Man-Machine Systems Laboratory members: G. Jagannath Raju who always gave me a hand when needed; James B. Roseborough who taught me the first lesson on C programming; Max B. Mendel, the aggressive Baysian; Chi Cheng Cheng (C³), with whom I spent endless evenings together in the lab., Kan-Ping Chin who set up the Latex in Macintosh which processed this thesis, Forrest Buzan, with whom, I designed robot controllers in Systemantics; and Hari Das, Sam Landsberger, Mike Massimino, Leon Charny, Jong Park and Wael Yared.

I am grateful to my parents who endured hardship in order to make my junior high school education in China possible.

The NASA Johnson Space Center was very generous in supporting the research in its early stage. The major work presented herein was supported through the generosity of the Association of American Railroads. I wish to thank both of these sponsors for their financial supports.
Contents

1 Introduction .......................................................... 14
   1.1 A Real Complex Decision Environment ....................... 15
       1.1.1 Resource Constrained Decision-making .................. 18
       1.1.2 Human Knowledge Imprecision and Soft Boundaries ...... 20
   1.2 Rail Freight Operations ......................................... 23
       1.2.1 A Brief View of Railroad History ......................... 23
       1.2.2 Structure of Rail Freight Transportation ................. 25
       1.2.3 A Typical Car Journey ..................................... 26
   1.3 Literature Survey ................................................. 30
       1.3.1 Methods of Dealing with Uncertainty ..................... 30
       1.3.2 Fuzzy Mathematical Programming .......................... 31
       1.3.3 Fuzzy Knowledge Systems .................................. 33
       1.3.4 Freight Train Operational Models ......................... 34
   1.4 Summary of this Thesis ......................................... 36
       1.4.1 Research Objectives ........................................ 36
       1.4.2 Thesis Contributions ....................................... 38
       1.4.3 Thesis Outline ............................................. 40

2 Research Methodology ............................................. 41
   2.1 Methods of Dealing with Uncertainty ........................ 41
       2.1.1 Probability Theory ......................................... 42
       2.1.2 Evidence Theory ............................................ 45
       2.1.3 Certainty Factors .......................................... 46
4.3.2 Fuzzy Rule Network (FRN) ........................................ 106
4.3.3 Recursive Learning of an FRN ................................. 111
4.3.4 Discussion of the Weight Matrix W ......................... 116
4.3.5 Example ....................................................... 117

4.4 Adaptation for a Human-in-the-Loop System .................. 121

4.5 Summary ....................................................... 123

5 Decision Aid for Freight Train Dispatching ........................ 125

5.1 Yard Operations ................................................. 128
  5.1.1 Yard Delays ............................................... 128
  5.1.2 Dispatching Policies ..................................... 129

5.2 Application of the ADAUS in Freight Train Dispatching .......... 136
  5.2.1 Linguistic Rule Base Module ............................... 139
  5.2.2 Fuzzy Linear Programming Module ....................... 147

5.3 Simulation and Experiment ..................................... 152
  5.3.1 One-Train Example ..................................... 152
  5.3.2 Railroad Network Simulation ............................ 157
  5.3.3 Results with Human Subject ............................ 159

5.4 Summary ....................................................... 161

6 Conclusions and Future Research ................................ 162

6.1 Conclusions .................................................... 162

6.2 Direction for Future Research .................................. 164

A Fuzzy Logic: What, Why and How ................................. 166

A.1 What is Fuzzy Logic ........................................ 167

A.2 Why Use Fuzzy Logic ......................................... 181

A.3 How to Use Fuzzy Logic .................................... 184
  A.3.1 Fuzzy Rule Base ......................................... 185
  A.3.2 Fuzzy Mathematical Programming ....................... 192

A.4 Computer Software ........................................... 196
List of Figures

1-1 A Real Environment and its Control .............................................. 16
1-2 Typical Journey of a Loaded Car .................................................... 26
1-3 Hump Classification Yard ............................................................... 28
1-4 Flat Classification Yard ................................................................. 29

2-1 Proposed General Framework of Decision Making ............................ 53

3-1 Degree of Satisfaction of the Goal $\mu_G$ ........................................ 67
3-2 Trapezoidal Fuzzy Membership Function .......................................... 68
3-3 The Complete Solution Set .............................................................. 73
3-4 No Solutions (Maximization Problems) ............................................ 76
3-5 Solutions Existence Criteria 1 .......................................................... 77
3-6 Solutions Existence Criteria 2 .......................................................... 77
3-7 Autogoal Setting .................................................................................. 83
3-8 Results of Nutrition Problem ............................................................. 84

4-1 Human-in-the-loop Process .............................................................. 89
4-2 A Fuzzy Knowledge System and Its Functions .................................... 93
4-3 Histogram from the Compiler-supplied Random Generator .................. 96
4-4 Histogram from the RSWOR Method .................................................. 97
4-5 Membership Elicitation Experiment .................................................... 98
4-6 Histogram of Driving Speed ............................................................... 100
4-7 Membership Function of Driving Speed .............................................. 100
4-8 Fuzzy Rule Network (FRN) .............................................................. 107
List of Tables

3.1 Comparison of $CMP$ and $FMP$ ............................................. 86

4.1 Linguistic Rules ............................................................... 103
4.2 Two-input, Single-output Example ....................................... 105

5.1 Fuzzy Variable $\tilde{G}_{policy}$ in Table Form ......................... 139
5.2 Fuzzy Variable $\tilde{T}_{past}$ in Table Form .............................. 141
5.3 Fuzzy Variable $\tilde{L}_{min}$ in Table Form .............................. 143
5.4 Comparison of $FLP$ and $CLP$ Approaches ............................ 156
5.5 Comparison of Human Test Result ....................................... 160

A.1 Fuzzy Rule Structure ....................................................... 185
A.2 Supplies ............................................................................. 204
A.3 Demands ............................................................................. 205
Chapter 1

Introduction

Advances in technology have helped scientists explore deeper into the universe. With these advances, more and more elusive and unstructured human activities need to be taken into account in decision making; as a result, decision processes are becoming more complex and difficult to analyze.

In the field of decision-making, deterministic and probabilistic mathematical models can be used to represent purely physical processes which obey well-defined physical laws such as Newton's laws of motion. However, these deterministic and probabilistic approaches fail to apply to human-in-the-loop systems which are adaptive in nature. In these systems, both the measurements and decision-making are subject to human subjectivity.

Although probabilistic or statistical inference is pervasive in the analysis for most situations, the knowledge or the meaning of certain empirical facts or data is never only probabilistic. In general, decision-making involves some form of approximate
and vague reasoning which occupies a central role in all human thinking as well as in the inquiries and meaning of the various soft or empirical sciences. It is not surprising that in many areas of cognition and complex decision-making processes, a human's performance far exceeds a computer's performance.

In daily decision making, humans are capable of coping efficiently and successfully with highly subjective partial beliefs and of drawing reasonable conclusions from partially distorted information of restricted reliability. On the other hand, computers, which is characterized by boolean operations, can help scientists and engineers in many ways with decision-making. However, the practical application of computing devices is restricted to those clearly defined problems that require the handling of great amounts of data or the performance of repeatable calculations. It is an important research area to adapt human problem-solving strategies to models suitable for application in the fields where exact theories do not exist but where human expertise is widely available. In these kinds of real environment applications, human expertise has to be adequately represented, and thus requires us to take into account having to deal with uncertainty, vagueness and soft boundaries.

1.1 A Real Complex Decision Environment

A real decision environment is usually very complex. Often, there is no full mathematical model available to describe it. If such a model does exist, it is usually nonlinear to such a degree that not enough analysis tools are currently available to design controllers or make decisions [91].
Figure 1-1: A Real Environment and its Control

Figure (1-1) illustrates such a real environment where a human is in the control loop (we define a system where a human takes part in the decision loop as a human-in-the-loop system). The shaded object on the left is the environment, and the human’s task is to control the environment based on its measurements (states). Without any decision aid, the only information available to the human for decision-making is a set of physical measurements such as speed, temperature or sound. The human relies on experience to make on-line control decisions by observing the states of the environment and performing qualitative reasoning. The problem’s difficulty increases as the system becomes more complex. A decision aid is highly desirable when the complexity reaches a certain level.

Taking a closer look at the environment, we find that a certain part of the envi-
ronment obeys well-defined physical laws and can be called a *mathematically modelable part*, from which one can perform certain calculations and make predictions at certain level. The model can also provide partial decision information to the human. In later chapters of this thesis, we will discuss a case study regarding freight train dispatching in which the mathematically modelable part will be the physical rail track network and the published schedule.

Having a mathematical model may still not utilize all available information, since part of the information related to human expertise has been neglected. The larger the human expertise part is, the larger the information neglected – in effect, an information loss. The other part of the environment, which can only be described by human expertise, is called *mathematically unmodelable part*.

In order to prevent such information loss, we have to consider two decision-making aspects: a mathematically modelable aspect and a mathematically unmodelable aspect. Unfortunately, two schools of researchers tend to work in these two approaches separately: one group works on a quantitative mathematical modeling approach such as operations research while the other group works on a qualitative reasoning approach such as expert systems. Either approach by itself neglects the other part of the available information.

From the above discussions, we understand that it would be too strict and non-responsive to use only an algorithmic mathematical model to make decisions, and it would be also too loose and unstructured to try to solve the problem by heuristic rules alone.
With the fast-growing power and availability of various computers, it is now possible to design a decision aid that is capable of processing complicate qualitative logic reasoning and fast quantitative calculation together.

This thesis addresses the issues of how to set a formal framework of decision aiding to handle quantitative and qualitative information simultaneously. Since the scope of designing a decision-aid for all real environments is too broad to be addressed at any depth in a single thesis, we will only investigate decision aiding with resource constraints for a class of human-in-the-loop systems. This research was supported by the American Association of Railroads (AAR), therefore, the freight train dispatching process serves as a case study.

1.1.1 Resource Constrained Decision-making

In most decision making a decision maker must achieve his/her goal with limited resources that are used as input to produce some kind of output. These resources generally have alternative uses. Also, a cost must be sustained in using the resource. Similarly, the output, or the benefit, from using the resources generally has a value. Efficient decision-making tries to minimize the costs associated with a given benefit, or to maximize the benefit associated with a given use of resources. These opportunities invariably generate decision-making in some kind of cost-benefit analysis. A decision will consist of the specific utilization of particular controllable resources, which are selected from among all resources that are available. If there is only one course of action available, we do not speak of a decision problem.
The field of decision-making is partitioned according to whether it is effected under conditions of *certainty* with which each action is known to lead invariably to a specific outcome, or under conditions of *uncertainty* in which each action has as its consequence, a set of possible specific outcomes, but the probabilities of these outcomes are completely unknown.

Mathematically, decision-making under certainty boils down to this: given a set of possible acts, one (or all) of those actions which maximize (or minimize) some given index will be chosen. Symbolically, let $x$ be a generic act in a given set $F$ of feasible acts and let $f(x)$ be an index associated to an appraising $x$; then find those $x^*$ in $F$ which yield the maximum (or minimum) index, i.e., $f(x^*) \geq f(x)$ for all $x$ in $F$.

On the other hand, decision-making processes under uncertainty are situations where the set of possible acts, the index and the resources available are not precisely defined. We denote a variable that is not precisely defined by a tilde over its name and call it a *fuzzy* variable that will be discussed in Section (2.1.6). Similar to decision-making under certainty, in decision-making under uncertainty, $\tilde{x}$ is a generic act in a given set $F$ of feasible acts and $f(\tilde{x})$ is an index associated to or appraising $\tilde{x}$; then find those $\tilde{x}^*$ in $F$ which yield the maximum (or minimum) index, i.e., $f(\tilde{x}^*) \geq f(\tilde{x})$ for all $\tilde{x}$ in $F$.

The following is the general sequence in the operation of decision-making in resource constrained problems [5]:

1. Select the objectives and specify their dimensions and values (the objectives may be a single goal or multiple goals).
2. Isolate all of the variables that are pertinent to the attainment of the objective value, i.e., the relevant independent variables.

3. Develop the relationships that exist between the independent variables.

4. Distinguish the controllable variables from the uncontrollable variables.

5. Develop predictions for the uncontrollable variables, which should be treated as states of nature.

6. Determine whether or not the predictions are based on stable processes.

7. Develop the functions that relate the independent variables to the dependent objective variables.

8. State the restrictions that limit the possible values of controllable variables.

9. Choose those values of the controllable variables which promise to maximize the degree of attainment of the objective, within the limits set by the restrictions.

1.1.2 Human Knowledge Imprecision and Soft Boundaries

Imprecision is one form of uncertainty embodied in human knowledge. It arises from a variety of sources, such as incomplete knowledge, inexact language, ambiguous definitions, inherent stochastic characteristics, measurement errors, etc. In short, uncertainty characterizes a human being’s valuation of some datum, reflecting his/her degree of faith or doubt in its truth of sources.
Can we build up a fully automatic system and bypass humans? For a simple system, such as a dc-motor, the answer is "yes", whereas for a complex system, such as an airplane, the answer is "no". Roseborough [71] concluded that the situations where human-in-the-loop control is considered necessary are: (1) Models of controlled process are inadequate. The human is included because of his extraordinary judgment and reasoning powers when the process attains some unmodeled states; (2) Best alternatives cannot be computed. This includes the situations in which different human experts have different opinions about the best solution; (3) The unique sensory abilities of humans cannot be replaced; and (4) Political reasons prevent replacement of human.

Railroad train dispatching is a system where human-in-the-loop control is considered essential; other examples include power plant, aircraft and automobile control. For these human-in-the-loop systems, a common feature is that they have processes or variables with a certain degree of uncertainty. The uncertainty may arise from inaccurate or incomplete data, linguistic imprecision, difficulties in prediction, or disagreement between information sources.

It is always fascinating that a human is capable of making decisions under these uncertainties based on imprecise, incomplete or even distorted information. Szolovits and Pauker [83] found that humans often refuse to give precise numerical estimates of outcomes and tend to make verbal judgments in the knowledge elicitation process. Kouchen [45] further concluded that precision and clarity beyond a certain point may be as ineffective in representing the truth as vagueness and confusion.
The need for reasoning under uncertainty is discussed in reference [58]. There is
general consensus among researchers that some form of uncertain reasoning is needed
for general purpose intelligence, as well as a variety of more specific applications in
perception and learning.

Before we discuss methods of dealing with uncertainty, we need to distinguish two
different kinds of uncertainty:

1. The degree of *expectation*, before the fact, that an event (e.g., a coin flip) will
occur in one of at least two mutually exclusive and collectively exhaustive cat-
egories (e.g., heads or tails). The event finally occurs in only one.

2. The degree to which an event actually does occur simultaneously to different degrees
in two or more categories. For example the existence of a *tall* man may map to
a modest degree in a 70-75 inch height category, and to a greater degree in
a 75-80 inch category. In this case there need be no *crisp* height (threshold)
below or above which a man is *tall*: the boundary is said to be *soft* or *fuzzy.*
The categories appropriate to a *very tall* man may overlap those for a *tall* man,
though the relative *occurrence* in this sense (or weighting or truth or *membership*)
for *tall* might differ from that for *very tall.*

The first definition is related to the frequency of an event and has been well studied
in probability and statistical theory. The latter definition is about the imprecision of
a state and is only touched on in probability theory.

In this thesis, we will concentrate on the second definition of uncertainty – impre-
cision or vagueness. For a variable whose boundary is characterized by imprecision or vagueness, we call this variable a soft variable.

Mathematically, a precise value is called a crisp value and sometimes is referred to as a variable with a hard boundary. In a similar way, an imprecise value is called a fuzzy value and this variable is referred to as a variable with a soft boundary. Crisp variables can be dealt with by means of traditional mathematics, while fuzzy variables should be coped with by other methods which are discussed in Chapter 2.

1.2 Rail Freight Operations

This section briefly views railroad history, particularly freight operations, as well as the current needs for technology upgrade. A typical journey of a loaded freight car is illustrated, and the corresponding classification yard operation is discussed.

1.2.1 A Brief View of Railroad History

Railroads usually transport large quantities of freight over relatively long distances and are perceived as the most energy-efficient transportation mode. With increasingly competition from trucks and waterways in the United States, the once glorious old industry needs new technology to compete.

For many years railroads dominated the transportation business in the United States. Construction began in the eastern states in the 1830's, particularly in New York, Pennsylvania and New England. A through connection between the East Coast
and Chicago was finished in 1853. Prior to the Civil War, about 30,000 miles of railroad line were constructed in the eastern part of the country. Construction was interrupted during the Civil War and was speeded up when the Civil War ended. Total railroad mileage in the States reached its peak of 254,037 miles in 1916. The reason for such rapid expansion was that the railroad transportation was ten times cheaper than its major competitor, the horse-drawn cart.

In The Depression years of the 1930's, the railroad suffered not only along with everyone else from a lack of business, but also from the increasing competition of truckers. As the interstate highway system grew in the 1960's, the railroads slowly collapsed under the combined weights of truck competition, loss of passenger service, overbuilt facilities and an unwieldy organization structure. In 1979 total mileage was reduced to 184,500 miles. The declining share of freight traffic led to consequent track abandonment.

The railroad industry, however, still remains in the transportation business today, especially to transport grain, mine ores and automobiles over relatively long distances. Currently it is operating profitably within its selected market domain.

As the competition increases, and the structure of the railroad industry changes, there is a growing awareness of the need to develop useful analytical tools. With the development of computers, various mathematical network models have introduced into railroad management starting from the early 1980's, however, such models did not include human expertise.

In this thesis, we proposed a new technique which combines human experience with
mathematical modeling and modern computer technology. Rail freight dispatching is a case study (Chapter 6), where we investigate techniques for combining human expertise (that has been accumulating over more than one hundred years) with modern computer technology for data fusion (which is defined as the process of integrating raw and processed data into some meaningful form for intelligent interface to improve system performance) to improve railroad operations.

1.2.2 Structure of Rail Freight Transportation

The physical structure of any rail system consists of a complex network made up of a number of large terminals (yards), numerous small stations and junction points interconnected by main and secondary lines. At any given moment, a large number of rail cars are moving between specific origin-destinations in the network, or waiting in the yards.

Rail cars move in blocks which are groups of cars with different possible final destinations. Blocks are assembled (classified) in origin yards and reclassified or reassembled into new blocks at intermediate yards or destination yards. One or more blocks and the necessary locomotive engines make up a train.

Rail trains are captive in nature – cars and engines move exclusively on rail tracks. Trains travel following predefined routes and normal schedules over a planned period of time. The route, frequency and other operation characteristics, such as train type, speed, priority, etc., define the train service. We will explain rail freight operation and train service using an example of a typical car journey in the following section.
1.2.3 A Typical Car Journey

A typical freight shipment is an individual car that is loaded for a particular consignee hundreds or thousands of miles away. Figure (1-2) shows the typical journey of a loaded freight car.

![Diagram of a typical car journey]

**Figure 1-2: Typical Journey of a Loaded Car**

The process starts when the shipper places an order for a car with the railroad agent by specifying the destination and the type of car, such as box car, flat car, et cetera. After checking the train schedule and routing book of his own railroad and possibly other railroads when the destination is out of his own territory, the railroad office suggests a delivery date for the load. The car service division then picks out an appropriate car, inspected by the mechanical department, for placement
at the shipper's railroad siding by a local engine. After the shipper loads the car and submits a straight bill of lading, the railroad office makes up a waybill to help keep track of the car and to inform the rest of the railroad that the shipment is moving on the network. A local engine is dispatched to the shipper's siding to pick up the car and take it to the classification yard where the car will be switched (classified) into a proper outbound train with other cars sharing a similar destination. All the waybills for the particular train are gathered together and a wheel report, which shows destination, weight, load or empty status, et cetera for each car in the train, is generated.

In its journey to the destination, the loaded car may pass through one or more intermediate yards and be reclassified to other trains. When the car and the waybill arrive at the destination yard, the consignee is notified. The consignee will either order the car to be sent to their sidings as soon as feasible or will order it by number, i.e., a specific sequence of cars. The car is then delivered by a local engine, unloaded and returned to the railroad service.

**Classification**

Classification, which rearranges cars into different trains according to their destinations, is a unique process of freight train operation at rail terminals (yards). In a terminal, incoming trains stop in the receiving or arrival yard, which has enough tracks to accommodate trains coming from all principal lines within a period of a few hours. The road locomotives and cabooses are removed, and car inspectors examine
the set of cars for possible repairs and lubrication, et cetera. The cars then wait for classification – cars are assembled from various sources into blocks headed for individual destinations. After the classification, the car blocks are pulled together to form trains for line haul.

Classification is a key procedure that directly affects reliability and delivery time in freight transportation. One can imagine that at a yard with thousands of cars in hundreds of acres, even to find a particular car is not an easy job.

In general, there are two major switching methods of performing classifications: the hump (gravity) switching method shown in Figure (1-3), and the flat switching method shown in Figure (1-4).

![Figure 1-3: Hump Classification Yard](image)

In a hump yard that uses the hump switching method, a hump engine pushes the cars over an artificial hill or hump at a speed of about three miles per hour. These hills or humps are elevated some ten to thirty feet above the classification track level. The cars then roll down the hump freely by gravity and are directed onto the proper classification track by throwing switches manually or automatically. This is the most efficient classification method when the traffic pattern is such that most
of the passing-through cars are heading in different directions.

In a flat yard that uses the flat switching method, instead of moving a car from the inclined hump, a yard engine literally kicks the uncoupled cars on one end of a block by starting forward quickly and then braking sharply to give the block an impact. When the block begins to move, a switchman throws a switch manually to direct (or cut) the rolling cars onto the proper departure tracks.

A typical hump yard may handle about 4000 cars and 60 trains daily at a humping rate of 3–8 cars per minute. A typical flat yard has 10–20 class tracks and may handle 1000 cars daily. Humps are usually found in large modern yards and flat yards are usually found in small yards.

The cars are now on one or more classification tracks. They are then assembled according to a specified order into an outbound train using a yard engine on the departure tracks. Once the complete blocks of cars are assembled, one or more locomotives are attached to the train. A caboose is put on and the air in the brake hoses is pumped up. The train then departs after routine outbound inspection.
It is obvious that not all inbound trains in a yard undergo complete reclassification. A train may stop at a yard only for refueling or crew change, or dropping off a block of cars, or picking up a new block of cars that has been classified in the yard, and it is possible that a train may go through several intermediate yards without having to be classified at each yard.

The details of dispatching policy and models are discussed in Chapter 5.

1.3 Literature Survey

As discussed in the previous section, this research focuses on resource constrained decision-making problems with considerations of human knowledge imprecision and soft boundaries. We conducted a literature review related to our research in the following topics:

1. Various methods of dealing with uncertainty.
2. Fuzzy mathematical programming.
3. Fuzzy knowledge systems.
4. Railroad freight train operational models.

1.3.1 Methods of Dealing with Uncertainty

Various methods have been developed to formalize the notion of uncertainty, to quantify degrees of belief and to set mechanisms for reasoning under uncertainty. The
oldest theory for uncertain reasoning is the classical probability theory. Other approaches have been developed because of the difficulty in applying probability theory to all forms of uncertainty: there are nonmonotonic logics in which a conclusion derivable from one set of premises fails to be derivable when a new premise is added [57] [67], defaults in inheritance networks that allow one to express a property that is shared by most members of a class [30] [41] [86], belief functions that associate two strengths of belief with every proposition – a strength of belief that the proposition is true and a strength of belief that the proposition is false [25] [38] [72], and fuzzy set theory that allows degrees of truth as well as degrees of likelihood [102]. In some sense, all of these theories compete as formal models of subjective degrees of belief by human reasoning. A discussion of the relation between probability theory and other schemes can be found in Cheeseman [19] while a broad presentation of the issues surrounding uncertain reasoning and a survey of various approaches can be found in Pearl [64].

The relative merits of the four major methods: probability theory, evidence (belief) theory, certainty factors in heuristic rules, and possibility (fuzzy) theory, will be discussed in Section 2.1.5.

1.3.2 Fuzzy Mathematical Programming

Mathematical programming is developed as a structured tool to solve decision-making problems in resource allocation and management. However, its implementation in practical applications is limited because the crisp data requirement is not always
satisfied in reality. It has been realized that a deterministic model for real problems is quite inadequate. Fuzzy programming and stochastic programming are two cousins derived from mathematical programming to accommodate uncertainties. Stochastic programming [92] is built on the assumption that some of the parameters used in describing the problem are random variables, which are difficult to determine until after the fact.

In 1970’s, Bellman and Zadeh [6] proposed the concept of decision making in a fuzzy environment. Their framework has been developed into choosing among fuzzy alternatives [4] [16] combined with mathematical programming, particularly in the form of linear programming. This was formulated as fuzzy linear programming by Zimmermann [105]. In his fuzzy linear programming formulation, it is assumed that fuzzy goals and fuzzy constraints are explicitly given as fuzzy sets and one optimal solution is obtained. Chanas [17] and Verdegay [89] independently worked out a parametric approach to solve fuzzy linear programming. By using parametric programming it is possible to analyze overall changes in the constraints and the objective function. Since fuzzy linear programming treats both goals and constraints as fuzzy numbers, multi-objective fuzzy linear programming [106] catches researchers’ attention. Some interactive multiple objective fuzzy linear programming applications [75] have been developed.

In fuzzy linear programming, an aspiration level (Chapter 3) – a fuzzy variable associated with the goal – is a subjectively predetermined value. When the aspiration level is too aggressive, the solution may not exist. All the mentioned references imply that a feasible aspiration level is always available. For practical application, however,
an explicit criterion is needed to determine the appropriate aspiration level.

1.3.3 Fuzzy Knowledge Systems

Fuzzy knowledge systems, also known as fuzzy rule bases, have been used successfully in many applications, from subway control [61] to smart air conditioners [14]. They are formulated as If – Then rules. This formulation has been studied extensively. One of the recent developments in fuzzy knowledge systems is the conception and design of systems capable of learning.

The self organizing controller (SOC) is at the earliest stage of the development of fuzzy rule learning. It evaluates the overall performance of a small set of system states on a global criterion, and makes use of a correction procedure resembling a human supervisor who determines the reward or penalty. In SOC, reward-penalty decision tables have to be generated and stored before the learning process starts. But the generation of these decision tables itself is not an easy job [56].

Lee and Berenji [53] combined the Adaptive Heuristic Critic (AHC) model of Barto, Sutton and Anderson [3] to learn the membership functions of the conclusions of the rules in the knowledge base. It is based on Barto et. al's pioneering work of applying neural networks to control. Two neural-like elements are used in this model. The AHC learns by updating the predictions of the system's failure over time, and has been tested in a cart-pole balancing example. However, this structure suffers from a lack of generality and is difficult to apply for larger scale systems due to the fact that developing mathematical forms for the trace function and the credit assignment
are not trivial.

Takagi and Hayashi [84] presented an three-step algorithm for combining neural networks with fuzzy logic. First, a suitable partition of the training data is found by using a clustering algorithm. Secondly, the memberships of the left-hand sides of rules are identified. And finally, the amount of control for each rule is determined. A neural network is used to identify the membership functions in the second step. This approach changes the shapes of the input membership functions. The resulting membership functions might be distorted compared to the human experts' perceptions in reality. Other methods such as changing the scales of input data as well as scales of the output variables are also discussed in the literature [40]. However these methods violate the $\text{If} - \text{Then}$ rules by artificially offsetting the input-output variables, and work only in certain specified contexts.

It has been shown that existing methods are suitable only to certain applications and still are inadequate in rule base calibration and interpretation. A simple but effective fuzzy rule network capable of learning is desired.

1.3.4 Freight Train Operational Models

Freight train dispatching is the case study in our research. This section reviews its operational models.

Rail companies have established a number of rules and policies at two levels of decision-making – operational decisions and strategic decisions – to plan and con-
trol their operations. Strategic decisions are concerned with long term developments, such as major investment, important modifications, or operation policies. Operational decisions deal with local operations, such as time tables, track priority rules, yard dispatching policies, power and empty car scheduling and so forth. Assad [1] and Crainic [21] [22] present reviews and model formulations that address both operations.

At the yard-level of operation, the Harvard Model [49] of the 1970's is probably the first published predictive freight network model with a fairly simple direct-link representation of the physical network. In the 1980's the automatic blocking model (ABM) [88] was developed using heuristics to determine the blocks built at each yard. This approach tries to improve the existing plan by forming bypass blocks to avoid delays at intermediate yards, which are subject to upper bounds on the number of blocks. But the train connections to move the proposed blocks have not been considered in this model.

For a medium-term planning horizon, Keaton [44] used an all-integer linear programming method to determine train connections, frequencies and routing problems for freight cars in single-carload general commodity service with all deterministic parameters (which is not realistic). Koskosidis [48] first proposed to deal with uncertainty in scheduling problems by using the ideas of soft time windows. But other soft variables such as train length and estimated incoming cars were not considered in his model.

Currently, freight train dispatching processes are dominated by the yardmaster's
rule of thumb and no decision aid is available at a network level. In practice, the human yardmaster’s expertise is so important that some railroad customers count on a particular yardmaster’s working schedule for their own shipping plan.

A decision aid which takes consideration of both the physical track network and human expertise is certainly capable of improving a railroad company’s performance.

1.4 Summary of this Thesis

In the previous sections, we have presented a decision-making environment with both deterministic and undeterministic parts, and reviewed the corresponding literature. In this section we will state the research objectives, summarize this thesis’ contributions and provide its outline.

1.4.1 Research Objectives

The objectives of this thesis are to investigate decision supports for real environments, and find techniques of decision-aiding capable of dealing with both quantitative and qualitative information in resource-constrained management processes. For this purpose, we will investigate the following issues:

1. Human subjectivity and linguistic imprecision in complex decision-making or data fusion environments, as well as the methodology of dealing with them.
2. Interfaces among: (a) human and computer via fuzzy variables; (b) fuzzy mathematical programming with soft boundaries; and (c) linguistic knowledge systems.

3. Fuzzy variable membership elicitation from human experts.

4. Adaptive learning of linguistic knowledge systems.

5. Evaluation of this approach relative to the currently used approach.

This research is implemented based on the following tasks:

1. **Proposal of a Framework of Decision-making under Uncertainties**

   A better framework of decision aiding by making use of both crisp (deterministic) and fuzzy (undeterministic) information to deal with uncertainties or soft boundaries involved in a real environment is proposed; and techniques of fuzzy linear programming, linguistic knowledge bases and their coordination into one single system to cope with complex environments is investigated.

2. **Analysis of Railroad Dispatching Model**

   Railroad dispatching processes are analyzed and the corresponding crisp and soft variables as well as linguistic variables are identified. A freight train dispatching model is established.

3. **Development of Fuzzy Mathematical Programming**

   The dispatching model is extended into a fuzzy mathematical programming module to accommodate both crisp and soft variables. A general purpose fuzzy
linear programming software package with objective oriented concepts is developed.

4. Development of Linguistic Knowledge Base

A linguistic knowledge base module, in which all the rules are described in natural language for easy interaction between computer and humans (railroad manager and yardmasters, for freight train dispatching, for example) is built. A suitable structure of knowledge representation is proposed, and techniques of membership elicitation (which relate to knowledge acquisition) and adaptive learning (which relates to knowledge maintenance) are investigated.

1.4.2 Thesis Contributions

Based on the research objectives and the implementations, this thesis makes the following major contributions:

1. A novel framework of decision aids to accommodate both quantitative and qualitative data simultaneously is proposed. The new decision framework uses a fuzzy mathematical programming module and a linguistic knowledge module. In this decision aid, the fuzzy linear programming module is used to model quantitatively the basic resource constraints and objective functions. It allows for soft constraints and soft variables in the right-hand-side and the technology matrix to accommodate uncertainties, while the linguistic knowledge base module is used to model imprecise human expertise. The two modules are linked together by a parameter vector which may take the form of
either fuzzy or crisp variables. Human operators in the context of active human-in-the-loop control are able to choose a satisfactory level for the goal and the constraints. The goal and the constraint specification in the fuzzy mathematical programming module makes use of well-structured information, while the rule specification in the fuzzy knowledge base module permits expertise and intuition to be taken into account.

2. A criterion to check solution existence for fuzzy mathematical programming is devised. A complete solution set is obtained by a parametric approach based on the ideas of fuzzy extension theory. The upper and lower bounds of the complete solution set suggest the feasible aspiration ranges.

3. A calibratable fuzzy rule network to store linguistic rules is implemented. This thesis discusses the imprecision of human knowledge in the process of establishing linguistic knowledge bases. The fuzzy rule network can not only represent linguistic knowledge but can also be calibrated recursively. Different from a neural network, the weights in the fuzzy rule network have physical meanings which give an indication of the convergence level.

4. The proposed new decision framework is applied to railroad freight train dispatching. It is the first an approach of such in this field. A simulation shows that using the new decision aid can improve the overall performance of freight train dispatching.
1.4.3 Thesis Outline

This thesis is arranged as follows:

Chapter 1, thesis introduction, describes general decision-making problems in a complex environment, discusses the application to real freight train dispatching, surveys the literature, introduces uncertainty in human expertise, and states the research objectives. Chapter 2 discusses various methods of dealing with uncertainties and imprecise human knowledge, and summarizes the research methodology used in this thesis. Chapter 3 describes fuzzy linear programming and discusses its solution existence. Chapter 4 discusses linguistic knowledge bases, the implementation of a fuzzy rule network and the corresponding learning procedure. Chapter 5 describes railroad freight train operations and yard dispatching models, reviews the straight-line dispatching policy and proposes a new soft dispatching policy. The general framework proposed in Chapter 2 (detailed in Chapter 3 and Chapter 4) is applied to the freight train dispatching process. Chapter 6 states the conclusions and suggests the directions for future research.

Readers who are interested primarily in rail freight dispatching may prefer to read only Chapters 1, 2, 5 and 6. For readers interested in decision-making theory, it is more appropriate to read all chapters in order.
Chapter 2

Research Methodology

Many methodologies, such as operations research, cybernetic and system control are concerned with the decision process. This chapter discusses the methods of dealing with uncertainty, summarizes the research approach used to solve decision-making problems in an environment which consists of both objective and subjective information, and proposes a novel decision framework for general resource-constrained decision problems.

2.1 Methods of Dealing with Uncertainty

Decision-making is a root process and is intertwined with all human activities and various uncertainties. Any one who has never made a decision in the face of uncertainty is almost certain from outer space. The everyday conversation strongly support this statement. "It will probably work out," "The chance for the train to
get there before noon is very good”, ..., all-too-familiar sentences are the phrases of someone who is faced to make decisions under uncertainty. One may be confident on making personal decisions such as where to go for a vacation, but may be perplexed facing very complicated decision environments. Our research is to attempt to make clear and precise nature of the difficulties that arise in decision-making under uncertainty, to investigate what assumptions and knowledge are required to overcome these difficulties, and to follow through the analysis to the stage of actually making a decision.

Basically, there are four methods to deal with uncertainty: probability theory, evidence (belief) theory, certainty factor in heuristic rules and possibility (fuzzy) theory as shown in the literature survey in Section 1.3. In this chapter, we will investigate the nature of resource constrained decision making, human uncertainty and the methods of dealing with uncertainty.

2.1.1 Probability Theory

The best known method is probability theory, including both classical and Bayesian. Bayesian probability is a function not only of the event, but also of the state of information. It is a function of two arguments, denoted as $P(h|s)$, where $h$ (hypothesis) is the uncertain event, and $s$ (state) is the person’s state of information on which it is conditioned. Classical probability theory can be formulated as a logic --- a formalism with well defined syntactic statements about situations and well defined syntactic statements about probabilities. Treating probability theory as a formal logic allows us
to construct automated inference procedures that reason about probabilities. Many analysts and theorists in the mathematical community advocate the Bayesian philosophy of analysis. In the decision analysis and reliability assessment fields, this philosophy has come to mean the evaluation of gathered data as they are conditioned on other events or circumstances. Bayes’ rule was founded on insights made in the early 1700’s by Thomas Bayes who is credited as being the first to write the equation:

\[ P(h|s) = \frac{P(h \land s)}{P(s)} \]  

(2.1)

as an analytic definition of “the likelihood of h given that we know s”. Equation (2.1) immediately leads to other useful equations such as:

\[ P(h \land s) = P(h|s) \cdot P(s) = P(s|h) \cdot P(h) \]  

(2.2)

and the posterior probability

\[ P(h|s) = \frac{P(s|h) \cdot P(h)}{P(s)}. \]  

(2.3)

Equation (2.3) is called Bayes’ rule. The usefulness of Bayes’ theory derives from the fact that it is often easier to assess the prior probability \( P(h) \) and the likelihood \( P(s|h) \) and compute the posterior probability rather than assess the posterior probability \( P(h|s) \) directly.

A conditional probability is analogous to a logical implication. Consider the statement \( P(h/s) \geq 0.99 \). The proposition can be viewed as the statement “if s is true, then h is very likely to be true”.

43
Some problems occur when applying the Bayesian philosophy to expert judgment. Both laboratory settings and actual applications have demonstrated that human cognitive processes do not seem to naturally follow Bayes’ rule [42]. First, most probabilistic models can be applied to our first definition of uncertainty (Section 1.1.2), but are left out of the considerations in the second definition. Consider for example [104] the following problem, in which the italicized words have the uncertainty in the second definition: An urn contains approximately 100 balls of various sizes, of which several are large. What is the probability that a ball drawn at random is not large? Second, the experts seem to fail to change or adjust their estimates in view of new information. Mathematically, the failure to update estimates means that \( P(h|s_1) = P(h|s_2) \); that is, the probability of \( h \) is not altered when the conditions governing \( h \) are changed from \( s_1 \) to \( s_2 \). This equation would only be true if \( P(h) \) was independent of any condition; that is \( P(h|s_1) = P(h) \) and \( P(h|s_2) = P(h) \). In estimating probabilities, however, it is unlikely that any event would be totally independent of conditions. Third, a person’s state of information \( s \) can not be isolated. It is the summation of many facts. Knowing \( P(h|s_1) \) does not allow one to conclude anything about \( P(h|s_1 \land s_2) \). In practice, one is generally interested in deriving conditional probabilities of the form \( P(h|\Sigma) \), where \( \Sigma \) is a large conjunction of facts about a situation. Even with a small number of propositions \( (s) \) there is a very large number (exponentially many) of different conditions and one can not know in advance which of these conditions will arise in practice\(^1\).

\(^1\text{In the multiple hypothesis (n) and multiple pieces of evidence (m), the complete prior requires specification of } 2^n \cdot (2^m - 1) \text{ parameters. This generality would be quite impractical when } m \text{ and } n \text{ are large.}\)
Other characteristics of situations which are difficult for human cognition to access and therefore prevent experts from being Bayesian are: the frequencies of truly rare events, the meaning of randomness, and the effects of variability [59].

2.1.2 Evidence Theory

Apparent practical difficulties in applying probabilistic schemes to model complex bodies of uncertain knowledge have given rise to the development of a variety of alternatives which includes interval representations such as Dempster-Shafer evidence (belief) functions, heuristic approximations to probability used in rule-based expert systems and fuzzy set theory designed to handle linguistic imprecision.

The theory of evidence is based on Shafer's belief functions and Dempster's rule of combination [72]. Like Bayesian approaches, the Dempster-Shafer theory of evidence aims to model and quantify by degrees of belief. In contrast to Bayesian approaches, it permits the assignment of degrees of belief to a set of hypotheses rather than to hypotheses in isolation. In assigning evidential support to the hypotheses according to Shafer's rules, the belief in $h$ and the belief in non-$h$ need not sum to 1.0. It means that if we have no evidence supporting $h$ at this moment, it does not mean that we necessarily disbelieve $h$; we simply lack any basis for believing $h$. Within the system of belief function, the algebraic rule for combining evidential support across items of evidence is Dempster's rule:

Suppose an uncertain variable contains three mutually exclusive and exhaustive
values $X = \{x_1, x_2, x_3\}$. Instead of distributing the unity probability mass\(^2\) among each of these three possible values, the belief function $Bel(X)$, allows mass to be distributed among all the subsets, including singletons, doublets and the entire set of $X$, that is:

$$\{\{x_1\}, \{x_2\}, \{x_3\}, \{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\}, \{x_1, x_2, x_3\}\} \quad (2.4)$$

This set is called the frame of discernment. The support for a hypothesis $h$, denoted as $Sup(h)$, is defined as the sum of the masses assigned to $h$ and all its subsets:

$$Sup(h) = \sum_{X \in h} Bel(h) \quad (2.5)$$

The underlining idea is that the process of narrowing the hypothesis set with the collection of evidence is better represented in terms of this theory than in terms of Bayesian approaches.

However, Dempster’s rule contains an adjustment process which simply discards whatever support is given to inconsistent possibilities, and this method can be viewed as a special case of Bayesian inference.

### 2.1.3 Certainty Factors

Another response to the apparent dilemma caused by restrictive assumptions or intractability is certainty factors in heuristic reasoning. These are the developments

\(^2\)probability mass distribution is the probability density function.
of numerical representations of uncertainty, intended to be more tractable approximations to probability. The best known of these, such as certainty factors (CFs) of Mycin Programming Project. (an expert system used for medical diagnosis) [11] and Prospector (another expert system used for the identification of mineral deposits) [35], are combined with production-rule knowledge representations. The production-rule representation was originally developed for logical reasoning without uncertainty. A major part of its appeal lies in its modularity; that is, individual rules may be added or removed without having to modify others to maintain consistency. Unfortunately it has become apparent that uncertain knowledge is intrinsically less modular than certain knowledge, and the assumption of modularity in rule-based schemes has certain restrictive implications. Analysis of these schemes in probabilistic terms demonstrates that they, too, unavoidably make various dependence assumptions which may not always be appropriate [42].

2.1.4 Fuzzy Set Theory

An alternative is proposed by advocates of fuzzy set theory which represents the imprecision explicitly. Fuzzy techniques have been extensively developed and the theory has provided the only consistent way of treating all the types of uncertainty that arise in expert system operations [103]. Since imprecision is intrinsic in human language, it is suggested that knowledge engineering and other aspects of human-computer interaction will be greatly eased if the imprecision is represented explicitly.

Fuzzy sets were originally intended to model the second kind of uncertainty (Sec-
tion 1.1.2), but they turned out to be also useful for the description of first kind uncertainty defined previously [99].

A fuzzy set is a good interface between linguistic variables seemingly preferred by humans and quantitative characterizations appropriate for computers. It emphasizes the possibility distribution interpretation of the concept of fuzziness, and provides a meaningful interpretation for some distributions that we believe are useful but might find difficult to justify on the basis of objective probabilities.

2.1.5 Relative Merits

Each of the methods mentioned above can handle uncertainties one way or another under different scenarios. Lots of active research have gone into working on all the approaches. There is very little consensus, however, about their relative merits and the tasks for which each is appropriate.

The key questions about the probabilistic approach are the interface and tractability for knowledge engineering, as well as inference in large-scale systems. The use of influence diagrams and belief nets appears to greatly facilitate the encoding and explanation of probability knowledge. However, large-scale applications remain to be demonstrated and good user-interface techniques must be evolved.

Heuristic numeric approaches or rule-based representations have been most popular for large knowledge bases of expert systems. In the past there was a belief among artificial intelligence researchers that the details of the uncertainty calculus generally
make little practical difference to system performance. However, recent experimental comparisons of heuristic approximations to coherent probabilistic approaches have found disturbing discrepancies [39]. The active performance depends on the class of inference task, and the differences between techniques are most significant when evidence is weak or conflicting.

Explicit handling of linguistic imprecision and uncertainties about degrees of belief seems desirable for ease of communication between knowledge support systems and users, and to facilitate sensitivity analysis. So far, fuzzy set theory is the most developed approach, although interval probabilities, Dempster-Shafer belief functions and second-order probability distributions (a probability distribution over another probability distribution) may also play a role. Computational tractability is currently an issue for all these approaches.

In selecting a technique for a particular application it is important to consider several criteria. Among them, two critical issues are: (1) the ease of encoding human expert knowledge into the scheme; and (2) the comprehensibility of the scheme to users.

It is noticed that the probabilistic approach is suitable for solving the first kind of uncertainty (Section 1.1.2), and fuzzy set theory is better when linguistic terms are used to describe the expertise where both the first and second kind of uncertainty are involved.

This research concerns with human expertise in a complicated system, such as a yardmaster in a freight train dispatching process, in which expertise is usually
described in terms of natural language that has three kinds of imprecision:

1. **generality** in which a word applies to a multiplicity of objects in the field of reference

2. **ambiguity** which occurs when a finite number of alternative meanings have the same phonetic form (e.g. blank)

3. **vagueness** in which there are no precise boundaries to the meaning of a word (e.g. young, large).

A membership function in a fuzzy set can be used to overcome such linguistic inexactness. *Generality* occurs when the portion of the universe of discourse where the membership values equal one is not just one point; *ambiguity* occurs when there is more than one local maximum of the membership function; and *vagueness* occurs when the function takes values other than just 0 and 1.

Since it seems that a fuzzy set with its membership function is the most general way of describing uncertainty, we will use fuzzy sets to deal with the imprecision of human linguistic knowledge (fuzzy knowledge).
2.1.6 What is a Fuzzy Set

For any crisp set $A$, a characteristic function which determines, for any element of the universe, whether that element is a member of $A$, is defined as:

$$
\mu_{A(x)} = \begin{cases} 
1 & \text{iff } x \in A \\
0 & \text{otherwise}
\end{cases}
$$

(2.6)

It is seen that the characteristic function $\mu_{A(x)}$ of a classical or crisp set takes a unique value in the two element set $\{0, 1\}$. That is, classical set theory is governed by a logic that permits a proposition to possess only one of two values: true or false. This logic does not accord well with the need to represent vague concepts. We see things in shades of gray, not in black and white. In everyday life, the observations and thoughts of most people most of the time may be said to be mentally modeled and/or communicated to other persons in terms of sets of imprecisely defined natural language words (such as short, medium, long, or very long as applied, for example, to a train's length, or urgent, soon, later, or much later as applied to the time available for a train's departure). The key elements in human thinking are not crisply defined but are more approximately defined. In other words, classes of objects in which the transition from non-membership to membership in set theory is gradual rather than abrupt as in that of a crisp set. It appears that much of the logic behind human reasoning is not the traditional two-valued or even multivalued logic, but logic with fuzzy truth, fuzzy connectives and fuzzy rules of inference [80].

Fuzzy mathematics was introduced by Zadeh in the mid-1960s when he was trying to find a method of modeling the human thought process. When first exposed
to fuzzy logic, humans sometimes associate membership functions with probability density functions. That is incorrect, since probability density is an abstraction from empirical frequency, and thus is an aggregate property, indicating how often events occur in different ways – ways that are quite crisp and mutually exclusive after the occurrence [80]. Fuzzy relations, by contrast, are properties of single events that are always present, and not different from occurrence to occurrence. However, modern fuzzy set theory has made provisions to accommodate both the notions of partial set membership (sometimes called possibility) and likelihood of occurring in any given state (probability).

2.2 Proposed Decision Aid

As we have pointed out in Chapter 1, a real decision environment contains both quantitative and qualitative information. Mathematically, we say that there exist two types of data: one is crisp data and the other is fuzzy data. The crisp data are usually related with the mathematically modelable part (which obeys well-defined physical laws) of the environment. The fuzzy data are most often associated with mathematically unmodelable part (in which human expertise plays an important role) of the same environment. Two schools of researchers tend to work in two different approaches by using either quantitative data or qualitative data. Either approach by itself neglects the other part of the available information. It is our goal to make good use of both types of information and coordinate them in one decision aid.
2.2.1 General Decision Framework

We propose a general decision framework, called *Adaptive Decision Aiding Under Soft Boundaries (ADAUS)*, as shown in Figure (2-1). Here *Soft Boundaries* implies uncertainty and *Adaptive* means that the decision aid has the ability to adjust itself for a better performance according certain criteria. In the *ADAUS*, four modules cooperate each other to handle two different types of information as well as to interface between the computer and the human decision-maker. Fuzzy set theory is used as a common vehicle to move information back and forth among these four modules.

![Figure 2-1: Proposed General Framework of Decision Making](image)

The four modules are:

1. fuzzy mathematical programming module,

2. linguistic knowledge base module,
3. satisfaction level analysis module, and

4. adaptive calibration module.

The main idea is the combination of fuzzy mathematical programming and linguistic knowledge base. It is based on the fact that in decision making it would be too strict and non-responsive to use a formal mathematical module only, and it would be too loose and unstructured to try to solve the problem by heuristic rule module alone; complex problems are better solved with both kinds of information. The fuzzy mathematical module makes use of well-structured and logically-strict information, while the linguistic knowledge base module permits expertise and intuition to be taken into account. The satisfaction analysis module shows the decision aid to the human decision-maker and receives his/her responses. Hence it is the interface between the computer and the human decision-maker. The adaptive calibration module recursively calibrates the linguistic knowledge module based on the discrepancies between the optimal solution from the decision aid and the decision-maker's response. We are going to briefly explain the four modules in the next sections. They will be discussed in detail by the following chapters.

2.2.2 Fuzzy Mathematical Programming (FMP)

There has been a strong tendency to view decision systems in the context of quantifiable problem structures and objectives, and this has led to the development of numerous quantitative models and analysis tools in the field of operations research, in which mathematical programming acts as a backbone. Crisp mathematical pro-
gramming maximizes or minimizes a crisp goal while simultaneously subjecting the objective to a set of crisp constraints.

Crisp mathematical programming is in the form:

$$\max_{x \in \mathbb{R}^n} f(x)$$  \hspace{1cm} (2.7)

subject to crisp constraints.

Equation (2.7) is to find an $x^*$ which makes $f(x^*) \geq f(x)$ for all the feasible alternatives $x$. It has been used successfully for years in problems related to crisp systems where all variables have clearly defined boundaries, and it has helped managers and engineers to understand the nature of those problems. Unfortunately, it can not be applied to systems where variables are ill-defined, or soft-boundaried. A key role there is played by human judgments, preferences, etc. which are subjective, imprecise and not easily quantifiable.

Following from the fact that classical mathematical programming (CMP) is often insufficient in practical situations, fuzzy mathematical programming (FMP) allows for its structure with imprecise coefficients that are described by fuzzy variables [18] [107].

Fuzzy mathematical programming has the form of:

$$\max_{x \in \mathbb{R}^n} f(x) \leq \tilde{g}$$  \hspace{1cm} (2.8)

subject to fuzzy constraints.
Compared with classical mathematical programming, there are fuzzy constraints which represent soft boundaries, and the aspiration variable \( \bar{g} \) which indicates the range of acceptable objective values.

In classical mathematical programming, any violation of a single constraint by even a small amount renders the solution infeasible. In fuzzy mathematical programming environments, however, decision makers can accept small violations of constraints and attach different degrees of importance to the violations of different constraints. The constraints which can accept certain violations illustrate the idea of soft boundaries. For example, one four-passenger-automobile and five persons are an infeasible solution for one single trip under classical mathematical programming because of the violation of the passenger-capacity constraint. In reality, however, it is often a satisfactory solution to crowd all five persons together for one trip. The overall performance is greatly improved (in terms of cost and time) by sacrificing some comfort of each passenger. FMP is designed to deal with this kind of situation. By solving the FMP problem (2.8), a solution is found which maximizes the degree of satisfaction on both the objective and the constraints.

For details, readers can refer to the following books and papers [12] [15] [18] [23] [70] [75] [81] [85] [107], as well as Chapter 4.

2.2.3 Linguistic Knowledge Base (LKB)

The attraction of a linguistic knowledge base (fuzzy logic) is that it captures an inherent and obvious property of most human communication which is not crisply
defined. People communicate orally and in writing by means of words, and the meanings of many (even most) words with respect to the associated physical variables are fuzzy. This is not to say that one cannot use numbers in written or oral discourse, or add enough qualifiers so that the meaning of a set of alphanumeric symbols becomes quite crisp. It is rather to say that the great majority of ordinary communication by ordinary people is through fuzzy terms. Nor is it to recommend that action be based on fuzzy terms and rules, when there are trustworthy crisp measures and rules available. Rather, fuzziness is a reality to be coped with.

The $LKB$ module is assembled by a series of linguistic rules from experts (detailed in Chapter 4). It exhibits the following characteristics: (1) it operates in domains in which the precise effect of an action is not always predictable; (2) it operates in incomplete domains in which crucial data may be missing or may be uncertain for a decision; (3) it operates in domains so large that it may not be possible to predict the precise effect of an action, although in principle the effect should be determinable; and (4) it has a learning component that enables it to extend its knowledge base in operation.

The fuzzy rule structure we are going to use is called fuzzy rule network ($FRN$) (Chapter 4). It has all four characteristics. The last characteristic, learning capability, is especially considered. The $FRN$ has the capability of adaptive supervisor learning based on the input-output teaching pairs.

The $FMP$ module is to model the basic resource constraints and objective functions. It takes consideration of goals to be achieved and resources available with un-
certainty involved. Most crisp data, such as physical structures and configurations, are considered by the \textit{FMP} module. The fuzzy coefficients in the technology matrix and the fuzzy constraints in the right-hand-side of the equations accommodate the uncertainties and soft boundaries. Some fuzzy coefficients are adjustable parameters that change their values as the mathematically unmodelable (usually fuzzy) environment changes. These fuzzy coefficients are outputs from the \textit{FKB} module. They provide a connection between the fuzzy mathematical programming module and the linguistic knowledge base module. These parameters change the resources available or the constraint structure in the \textit{FMP} module. Since fuzzy variables are the common building blocks of both modules, the interface between them is seamless.

Fuzzy rules and fuzzy logic have been applied in many and varied kinds of automatic and human-machine systems. Reference \cite{94} contains a collection of the most important papers including the first paper on fuzzy sets. The mathematical description of fuzzy set theory is in reference \cite{29}. Recent research and applications on fuzzy reasoning are contained in reference \cite{61}.

\subsection{Satisfaction Level}

Simplifications allow humans to deal with complex systems using limited resources for decision making. It remains fascinating and unclear how a human selects a decision algorithm which simplifies data while retaining the essential characteristics to achieve moderately satisfactory decisions.

In the \textit{ADAUS}, the fuzzy mathematical programming module synthesizes all the
desired goals, the uncertain resources and human expertise into two simple decision
curves: a goal satisfaction curve and a constraint satisfaction curve. These two curves
are analogous to the return/risk curves in investment decision-making situations: the
goal satisfaction curve is the analog to the return curve and the constraint satisfaction
curve is the analog to the inverse of the risk curve (non-risk curve). Synthesizing
various uncertain variables into two decision curves (Section 3.2) is one step toward
human cognition.

With the satisfaction level analysis, the decision-maker's workload can be reduced.
S/he can be freed from tedious and simple logic reasoning, and can concentrate on
more general and important decision making.

2.2.5 Adaptive Calibration

Another important module is the rule adaptive mechanism. It recursively calibrates
the linguistic knowledge base from the discrepancies between the optimal decision point
in the satisfaction analysis and the decision-maker's preference.

The optimal solution from the FMP module is the intersection of the goal satisfac-
tion curve and the constraint satisfaction curve. When a decision-maker is prompted
with these satisfaction curves, s/he may select a satisfaction level other than the
computer-suggested optimal level. The discrepancy indicates that the ADAUS does
not fully agree with the human decision-maker. The ADAUS is then adaptively
calibrated according to the discrepancies by adjusting the linguistic knowledge base.
Please refer to Section 4.3.2 for details.
The ADAUS can be regarded as an expert system under human supervisory control [80]. The decision-making process is supervised by human supervisors who are able to interrupt the computer, make changes and perform rule calibration. The freight train dispatching example will further illustrate this new idea (see Chapter 5).

2.3 Summary

Numerous methods have been developed to formalize the notion of uncertainty and we have discussed the most commonly used methods. Among them fuzzy set theory is justified to be the most suitable formulation to deal with the imprecision and vagueness in human knowledge.

In human-in-the-loop decision-making environments, the available information can be generalized into two categories: hard data and soft data. Hard data is the data which are precisely defined such as the physical rail track, while soft data is the data which are not clearly defined such as linguistic variables.

We proposed a general framework of decision aid to deal with these two different kinds of information simultaneously by using fuzzy mathematical programming and linguistic knowledge systems. This approach is based on the fact that in decision making it would be too strict and non-responsive to solve a real world problem by using a formal mathematical module only, and it would be too loose and unstructured to try to solve the problem by heuristic rule module alone. Furthermore, this approach allows for a human to actively interact with the decision aid by selecting a satisfaction
level which can be used as one of the indications to calibrate the linguistic rule recursively.

Though freight train dispatching is taken as a case study in this thesis, the proposed decision-making methodology can be used for any resource-constrained decision-making situation with imprecise data and vague human expertise.
Chapter 3

Fuzzy Mathematical Programming Module

The essential feature of a mathematical model in operations research is its involvement in a set of mathematical relationships (such as equations, inequalities and logical dependencies, et cetera) which corresponds to down-to-earth relationships in the real world (such as technological relationships, physical laws and marketing constraints, et cetera). Mathematical programming\(^1\) is one of the most widely used models in operations research and management science.

This chapter starts with classical mathematical programming (\(CMP\)), then develops into fuzzy mathematical programming (\(FMP\)) with soft boundaries and solutions.

\(^1\)It should be pointed out that mathematical programming is very different from computer programming. Mathematical programming is programming in the sense of planning. It has nothing to do with computers. Inevitably, mathematical programming does become involved with computing science, since practical problems almost always involve large quantities of data and arithmetic which can only reasonably be tackled by the calculating power of a computer.
It discusses a complete solution and the criterion of solution existence. Finally, the major features of CMP and FMP are compared in the summary.

3.1 Fuzzy Mathematical Programming

All mathematical programming is involved with optimization, either maximization or minimization. It tends to be divided into two broad classes of problems: deterministic problems and undeterministic problems. Deterministic problems are those in which uncertainty is either negligible or entirely absent, so that for any policy, that a decision-maker chooses, the resources required and the eventual outcome can be predicted with complete certainty. That is, classical mathematical programming can be used for solving deterministic problems, but not for undeterministic problems; whereas fuzzy mathematical programming can be used for undeterministic problems.

3.1.1 CMP Form

A classical mathematical programming problem is an algebraic formulation of a deterministic optimization problem. It is generally in the format of:
\[
\begin{align*}
\underset{x \in \mathcal{R}^n}{\text{Max or Min}} & \quad f(x) \\
\text{subject to:} & \\
\left\{ \begin{array}{c}
w_i(x) \leq b_i \\
  \geq \\
\end{array} \right. & \quad i = 1, 2, \cdots, m \\
x \geq 0
\end{align*}
\] (3.1)

where \( x = [x_1, x_2, \cdots x_n] \in \mathcal{R}^n \) is a vector of decision variables;

\( f(x) : \mathcal{R}^n \rightarrow \mathcal{R} \) is an objective function; and

\( w_i(x) : \mathcal{R}^n \rightarrow \mathcal{R} \) is the \( i \)-th constraint.

Any particular assignment of numerical values to \( x \) is known as a solution to the problem and may be thought of as a point in the real Euclidean space of \( \mathcal{R}^n \). A solution is feasible if it meets all the constraints; otherwise, the solution is infeasible. The collection of all feasible solutions or points in the \( n \)-space \( \mathcal{R}^n \) is known as feasible region. In deterministic models, all variables are crisply defined; small errors and uncertainties are neglected or deliberately ignored.

### 3.1.2 FMP Form

In the real world, truly deterministic problems are quite rare: operation costs are usually estimated rather than known, measurement instruments with 99.5% accuracy are treated as perfect, completion dates for construction projects are established with normal allowance for bad weather, and so forth. In such circumstances the use of a
Deterministic model is justified only if the deviations in practice are expected to be small.

Fuzzy mathematical programming follows from the fact that classical mathematical programming is often insufficient in practical situations. In this sense FMP is an extension of classical mathematical programming and deals with imprecise coefficients by using fuzzy variables [18] [107]. For instance, the objectives and the resource constraints in Figure (1-1) may be ill-defined, either because their values depend on other parameters, or, because they can not be precisely assessed and thereby only qualitative estimates are available.

Fuzzy mathematical programming is of the form

$$\max_{x \in \mathcal{R}^n} f(x) \equiv \tilde{G}$$

subject to:

$$\tilde{w}_i(x) \leq \tilde{b}_i \quad i = 1, 2, \ldots m$$

$$x \geq 0$$

where

$$x = [x_1, x_2, \ldots x_n] \in \mathcal{R}^n$$ is a vector of decision variables;

$$f : \mathcal{R}^n \rightarrow \mathcal{R}$$ is an objective function;

$$\tilde{G}$$ : is an aspiration level of the objective function;

$$\tilde{w}_i : \mathcal{R}^n \rightarrow \mathcal{R}$$ are constraints;

---

Note that maximization can replaced by minimum; ‘\(\leq\)’ is easily be replaced by ‘\(\geq\)’ or ‘\(\approx\)’ (approximate equal). In this thesis, the tilde sign above a variable means this variable is a fuzzy variable.
\( \bar{b}_i \in \mathcal{R} \) are the right-hand-side vector or resources available.

Compared to classical programming, there are fuzzy constraints \( \bar{b}_i \) and an aspiration level \( \bar{C} \) which indicates the range of acceptable objective values.

In classical mathematical programming, any violation of a single constraint \( b_i \) by even a small amount renders the solution infeasible. We say this kind of constraint has a hard boundary, corresponding to the binary characteristic functions of a crisp set.

In the fuzzy mathematical programming environment, however, the decision maker could accept small violations (in terms of a degree of \( \theta_i \)) of the constraints \( b_i \) and attach different degrees of importance (\( \mu_{\bar{b}_i} \)) to violations of different constraints as he would do in the real world. We say this kind of constraint has a soft boundary, corresponding to the fuzzy membership functions of a fuzzy set. The degree of satisfaction described by fuzzy memberships \( \mu_{\bar{b}_i} \) ranges between one (which means no violation) to zero (which means the largest allowable violation). The relation between the constraint violation \( (\theta_i) \) and the degree of constraint satisfaction \( (\mu_{\bar{b}_i})^3 \) is

\[
\theta_i = 1 - \mu_{\bar{b}_i}, \quad i = 1, 2, \ldots, m. \tag{3.3}
\]

The larger the violation, the smaller the degree of constraint satisfaction.

For example, one four-passenger-automobile and five persons are an infeasible solution for one single run under classical mathematical programming because of the

\[3\text{Later in this chapter, } \mu_{\bar{b}_i} \text{ is equal to } \alpha.\]
violation of the passenger-capacity constraint. But in reality it is often a satisfactory solution to crowd all five persons together.

The aspiration level $\tilde{G}$ is a goal expectation. It is defined as a degree of goal satisfaction denoted by $\mu_G(f(x))$. It is the measure of closeness of $f(x)$ with the predefined goal $\tilde{G}$. The more closeness, the higher the degree of goal satisfaction. The aspiration level $\mu_G$, specified by an expert, has a membership function (in maximization problems)$^4$ of:

$$
\mu_G(x) = \begin{cases} 
1.0 & \text{if } f(x) \geq g_{\text{max}} \\
g(x) & \text{if } f(x) \in [g_{\text{min}}, g_{\text{max}}] \\
0.0 & \text{if } f(x) \leq g_{\text{min}} 
\end{cases} 
$$

(3.4)

where $g_{\text{max}}$ is an aspiration level, and $g_{\text{min}}$ is a lowest acceptable level as shown in Figure (3-1)$^5$.

![Figure 3-1: Degree of Satisfaction of the Goal $\mu_G$](image)

Equation (3.4) reads as: the goal achievement is fully satisfied ($\mu_G = 1$) with the $x$'s for which $f(x)$ attains a value not below an aspiration level $g_{\text{max}}$; the goal achievement is less satisfied to a degree of $\mu_G \in (0,1)$ for which $f(x)$ attains a value between $g_{\text{min}}$ and $g_{\text{max}}$; and it is fully dissatisfied when $f(x)$ attains a value below

---

$^4$For a minimization problem, the aspiration function shown in Figure (3-1) flips horizontally and Equation (3.4) changes accordingly.

$^5$Here $g(x)$ is assumed a linear function.
The membership function of a fuzzy constraint $\tilde{b}_i$ in Equation (3.2) is usually defined as a trapezoidal fuzzy subset:

$$
\mu_{\tilde{b}_i}(x) = \begin{cases} 
0.0 & \text{if } w_i(x) \leq A_i^l - \delta_{Ai} \\
1 - \frac{A_i^l - w_i(x)}{\delta_{Ai}} & \text{if } w_i(x) \in [A_i^l - \delta_{Ai}, A_i^l] \\
1 & \text{if } w_i(x) \in [A_i^l, A_i^r] \\
1 - \frac{w_i(x) - A_i^r}{\gamma_{Ai}} & \text{if } w_i(x) \in [A_i^r, A_i^r + \gamma_{Ai}] \\
0.0 & \text{if } w_i(x) \geq A_i^r + \gamma_{Ai} 
\end{cases}
$$ 

(3.5)

where $A_i^l$ and $A_i^r$ are lower and upper bounds (inside the bounds the memberships take full value 1); $\delta_{Ai}$ and $\gamma_{Ai}$ bracket the intervals of confidence for the left and right sides of $\tilde{A}_i^l$ and $\tilde{A}_i^r$ respectively. $\tilde{b}_i$ in trapezoidal form is shown in Figure (3-2) and symbolically written as:

$$
\tilde{b}_i = (\delta_{Ai}, A_i^l, A_i^r, \gamma_{Ai})
$$ 

(3.6)

Figure 3-2: Trapezoidal Fuzzy Membership Function

With different values of $\delta_{Ai}, A_i^l, A_i^r$ and $\gamma_{Ai}$, we can use $\tilde{b}_i$ to represent all different constraint types ('=', '<=' or '>=' ) in Equation (3.2):

1. When a constraint has an equal sign '=', $\tilde{b}_i$ is defined as in Equation (3.6). All
\( \delta_{Ai}, A^l_i, A^r_i, \gamma_{Ai} \) have finite values with a membership function as shown in Figure (3.2).

2. When a constraint has a '\( \leq \) ' sign, Equation (3.6) becomes \( \bar{b}_i = (0, \infty, A^r_i, \gamma_{Ai}) \).

3. When the constraint has a '\( \geq \) ' sign, Equation (3.6) becomes \( \bar{b}_i = (\delta_{Ai}, A^l_i, \infty, 0) \).

4. When no violation of constraint is allowed in the \( i \)-th constraint, that is, a hard constraint, Equation (3.6) becomes \( b_i = (0, a, a, 0) \), where \( a \) is a real value.

### 3.1.3 A Solution of FMP

To solve the fuzzy mathematical programming problem (3.2) is to find a solution which maximizes the degree of satisfaction on both the objective and the constraints. The solution of the problem (3.2) is equivalent to the solution of the following equation in classical mathematical programming:

\[
\text{Max } \mu_D(x) \\
\text{subject to:} \\
\mu_G(x) \geq \mu_D(x) \\
\mu_C(x) \geq \mu_D(x) \\
\mu_D(x) \in [0, 1] \\
x \geq 0
\]

(3.7)

where \( \mu_D(x) \) is the overall degree of satisfaction, \( \mu_G(x) \) is the degree of goal satisfaction and \( \mu_C(x) \) is the degree of constraint satisfaction.

It is noticed that the equal sign constraint \( w_i(x) = (\delta_{Ai}, A^l_i, A^r_i, \gamma_{Ai}) \) in Equation
(3.2) can be split into two constraints:

\[ w_i(x) \leq (0, \infty, A_i^r, \gamma_{Ai}) \quad \text{and} \quad w_i(x) \geq (\delta_{Ai}, A_i^l, \infty, 0). \quad (3.8) \]

Equation (3.2) is then written as

\[
\begin{align*}
\max & \text{ or } \min \\ x \in \mathbb{R}^n
\end{align*}
\]

subject to:

(1) \hspace{1cm} w_i(x) \leq (0, \infty, A_i^r, \gamma_{Ai}) \\
(2) \hspace{1cm} w_i(x) \geq (\delta_{Ai}, A_i^l, \infty, 0) \\
x \geq 0
\quad (3.9)

The degree of constraint satisfaction \( \mu_{Ci} \) for the \( i \)-th constraint is:

\[ \mu_{Ci}(x) = \mu_{bi}(x) \quad (3.10) \]

and the total degree of constraint satisfaction \( \mu_C(x) \) is:

\[ \mu_C(x) = \bigwedge_{i=1}^{m} \mu_{Ci}(x) \quad (3.11) \]

where \( \bigwedge \) is defined as a fuzzy minimum\(^6 \) operator.

According to the Bellman-Zadeh concept of decision-making in a fuzzy environment \cite{99}, this is a problem of looking for a fuzzy subset \( \tilde{D} \) in the solution space \( \chi \) with membership function:

\[ \mu_D(x) = \mu_G(x) \bigwedge \mu_C(x) \quad x \in \chi \quad (3.12) \]

\(^6\)In general case, it is a \( T_{\text{norm}} \) operator discussed in Appendix A.
When a solution vector $x^*$ maximizes the membership function, it is called the \textit{optimal} solution, that is,

$$\mu_D(x^*) = \max \mu_D(x).$$  \hspace{1cm} (3.13)

\subsection*{3.2 Decision Curves}

The optimal $x^*$ in (3.13) is only one solution of the $FMP$ problem (3.2). Since the degree of goal satisfaction is a subjective value, the decision-maker will like to see a complete solution set, which contains all possible solutions, for a subjective choice.

\subsection*{3.2.1 Complete Solution Set}

A complete solution set contains all possible solutions. It is a set of $x$'s with which the overall degree of satisfaction $\mu_D(x) > 0$ in Equation (3.7). The reasons for studying a complete solution set are:

1. Since the degree of goal satisfaction is a subjective measure, only with the complete solution set, which provides the complete information, can a decision-maker perform the best subjective decision-making.

2. The solution $x$ with which the overall degree of satisfaction $\mu_D(x) > 0$ may not exist. That is, the solution set $\hat{D}$ is an empty set. The complete solution set shows this situation clearly.
3. The optimal solution \( \mu_D(x^*) = \max \mu_D(x) \) will be certainly included in the complete solution set.

Chanas [17] and Verdegay [89] proposed independently to use the so-called extension theorem which says that a fuzzy set can be uniquely represented by all its \( \alpha \)-cuts (see Appendix A). This approach may be summarized as follows:

If membership functions of the fuzzy constraints in Equation (3.2) are strictly monotone and continuous, which is often natural, the \( \alpha \)-cuts of the set of constraints can be expressed as:

\[
B_\alpha = \{ x \in \mathbb{R}^n | w_i^{-1}(\alpha), \; i = 1, 2, \cdots, m. \} \tag{3.14}
\]

where the \( w_i^{-1}(\cdot) \) is the inverse function of \( w_i(\cdot) \) in Equation (3.2).

According to the extension theorem, the fuzzy constraint set \( \tilde{B} = (\tilde{b}_1, \tilde{b}_2, \cdots, \tilde{b}_m) \) is

\[
\tilde{B} = \sum_{\alpha \in [0,1]} \alpha \cdot B_\alpha \tag{3.15}
\]
With linear membership functions of $\tilde{b}_i$'s, Equation (3.9) becomes a classical parametric mathematical programming

$$\begin{align*}
\max_{x \in \mathbb{R}^n} \text{ or } \min_{x \in \mathbb{R}^n} \quad & f(x) \\
\text{subject to:} \\
(1) \quad & w_i(x) \leq A^+_i + \gamma_{Ai} \cdot (1 - \alpha) \\
(2) \quad & w_i(x) \geq A^-_i - \delta_{Ai} \cdot (1 - \alpha) \\
(3) \quad & \alpha \in [0, 1] \\
& x \geq 0
\end{align*}$$

(3.16)

By solving the classical parametric mathematical programming, we obtain a complete solution set:

$$z(\alpha) = f(x) \quad \forall \quad \alpha \in [0, 1]$$

(3.17)

as shown in Figure (3-3)$^7$.

$^7$This figure is for maximization problems. For minimization problems, Figure (3-3) flips horizontally.
3.2.2 The Existence Criterion of the Decision Set

Since the aspiration level $\bar{G}$ is a subjective value, when a decision-maker is too aggressive to set an unreasonable level $\bar{G}$, there might be no solution at all. In the literature of $FMP$, it is usually assumed that a solution is always available. Few papers in the field discuss the existence criterion of a decision set. We are going to investigate the existence criterion of the decision set in the following.

In the previous section, we have shown that the complete solution set $\hat{D}$ can be written to an explicit objective function $z(\alpha) = f(x)$, where $\alpha \in [0, 1]$. It says that when $\alpha$ changes from 0 to 1, $z$ changes from $z_{\alpha 0}$ to $z_{\alpha 1}$. In order to have a solution, the aspiration level $\bar{G}$ has to have an intersection with $z(\alpha)$. Then the overall degree of satisfaction becomes:

$$
\mu_D(z) = \text{Min} ( \mu_G(z), \mu_C(z) )
$$

and

$$
\mu_C(z) = \bigwedge_{i=1}^{m} \mu_{C_i}(z). \tag{3.18}
$$

It is denoted that $\alpha$ is the degree of satisfaction for all fuzzy constraints

$$
\alpha = \bigwedge_{i=1}^{m} \mu_{C_i}(z) \tag{3.19}
$$

Equation (3.19) is equivalent to

$$
\mu_{C_1}(x) \geq \alpha \quad \mu_{C_2}(x) \geq \alpha \quad \cdots \quad \mu_{C_m}(x) \geq \alpha \tag{3.20}
$$
The $FMP$ problem in Equation (3.2) is then mapped to the following programming problem:

$$\begin{align*}
\text{Max or Min} & \quad z(\alpha) \\
\text{subject to:} & \\
\mu_{Ci}(z) & \geq \alpha \quad i = 1, 2, \ldots, m \\
x & \geq 0 \\
\alpha & \in [0, 1]
\end{align*}$$

(3.21)

The obtained $z(\alpha)$ is the mapping of all constraints with their common (minimize) satisfaction degree $\alpha$ defined in Equation (3.19). Compared to $\mu_C(z)$, we see that $\alpha = \mu_C(z)$.

In order to derive $\mu_C(z)$, the inequality $\mu_{Ci}(x) \geq \alpha$ in Equation (3.21) must be further studied. Its explicit form is $\mu_{Ci}(w_i) \geq \alpha$.

In a maximization\textsuperscript{8} problem, $\mu_G(z)$ must be an increasing function of $z$ and $\mu_C(z)$ must be a decreasing function of $z$. Since $\mu_D(z)$ is defined as the intersection of $\mu_G(z)$ and $\mu_C(z)$, if $z_{co} \leq z_{g_{\min}}$ at $\mu = \mu_G = \mu_C = 0$, no intersections can occur between $\mu_G(z)$ and $\mu_C(z)$ at $\mu > 0$ levels. This means that no decision set $x$ can result in $\mu_D(z) > 0$. Therefore, the existence criterion of the decision set is\textsuperscript{9}:

$$z_{g_{\min}} \leq z_{co} \quad \text{at} \quad \mu = \mu_G = \mu_C = 0,$$

(3.22)

\textsuperscript{8}For a minimization problem, $z$ is replaced by $-z$ and the inequality signs have to be changed accordingly in Equation (3.22).

\textsuperscript{9}$z_{g_{\min}} = g_{\min}$ and $z_{g_{\max}} = g_{\max}$ in Equation (3-1).
otherwise, no solution exists as shown in Figure (3-4).

![Figure 3-4: No Solutions (Maximization Problems)](image)

When criterion (3.22) is satisfied, there is a feasible decision region (shaded area) as shown in Figure (3-5). The decision curves synthesize all the soft constraints into two curves which are similar to the risk-return curves in an investment decision problem. In this situation, a decision-makers has the option of selecting an operating point where s/he feels comfortable. There are two cases in such situations:

1. $z_{co0} \geq z_{g_{min}}$ and $z_{co1} \leq z_{g_{max}}$

   In this case, the complete decision set is composed of those solution vectors $x$ which result in

   $$z_{g_{min}} \leq z \leq z_{co0}.$$ 

   It shows that there exists one and only one intersection point between $\mu_G(z)$ and $\mu_C(z)$ for $\mu_D \in (0, 1]$. The intersection point $\mu_D(x)^*$ is the optimal solution.

2. $z_{co1} \geq z_{g_{max}}$

---

10This figure is for a maximization problem.
Figure 3-5: Solutions Existence Criteria 1

In this case, the complete decision set $x$ with its membership functions can also be obtained by the formulas in the first case. Only in this case, $\mu_D$ will be covered from zero to one as shown in Figure (3-6).

Figure 3-6: Solutions Existence Criteria 2
3.3 Nutrition Example

We will illustrate the procedure of solving an FMP problem by using a nutrition example\textsuperscript{11}.

A farmer who specializes in breeding beef is concerned about the quality of the beef and the cost of maintaining the level of quality. Certain kinds of vitamins (A, B and C) affect cow growth rate and beef flavor. For a predefined quality, cows must consume at least 100 milligrams of vitamin A, 200 milligrams of vitamin B and exactly three hundred grams of vitamin C. The farmer is able to purchase several different kinds of bags of grain, which have different amounts of the mentioned three vitamins. The first and second kinds of bags cost $12 each, the third costs $13 each, and the fourth costs $14 each. The objective is to minimize the cost and provide the cows with enough vitamins without ruining his budget.

This example shows different approaches to solve the above problem, depending on the softness of the constraints. Soft constraints will give the farmer more leeway in satisfying his cows’ nutritional needs. For instance, instead of getting exactly 100 milligrams of vitamin C, the cows will thrive on some amount between 95 and 105. The problem was varied according to different preferences that the farmer might have. FLP123 has been run with each of these preferences, showing how low cost can be brought by certain nutritional cuts.

\textsuperscript{11}A computer software, FLP123, has been developed by the author for solving FMP problems. See Appendix A.4.2.
Crisp Example:

The first case run was a classical linear program. In this case, each constraint must be met absolutely. This case would correspond to a nutrition problem where the farmer has no leeway in the amount of each grain needed.

Problem Formulation:

Min $Cost = 12x_1 + 12x_2 + 13x_3 + 14x_4$

subject to:

$$2x_1 + 3x_2 + 7x_3 + 5x_4 \geq 100$$

$$x_1 + 6x_2 + 2x_3 + x_4 \geq 200$$

$$3x_1 + 2x_2 + x_3 + 4x_4 = 300$$

$$x_1 \geq 0, \ x_2 \geq 0, \ x_3 \geq 0, \ x_4 \geq 0$$

(3.23)

The variables $x_1, x_2, x_3,$ and $x_4$ stand for the numbers of each type of bag of grain the farmer needs to buy. The coefficients of the variables on the top line signify the different costs of each bag; $x_1$ costs $12 each, $x_2$ costs $12, $x_3$ costs $13$, et cetera. The coefficients on the second line of the equation are the milligrams of vitamin $A$ contained in each bag of grain; $x_1$ has 2 milligrams each, $x_2$ has 3 milligrams, $x_3$ has 7 milligrams, and $x_4$ has 5 milligrams. The 100 on the right-hand-side shows that whatever combination of grains the farmer does buy, the combination must have at least 100 milligrams of vitamin $A$. The third line means the same thing as the line above it, except that the coefficients and the right-hand-side are demanding that the farmer buy some combination of bags that contains at least 200 milligrams of vitamin $B$. The fourth equation signifies that the farmer must buy a combination of grains that has exactly 100 milligrams of vitamin $C$. The last four equations,
\( x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, \) just mean that the farmer cannot sell bags; literally, he cannot buy negative amounts of grain.

Since each of the constraints must be precisely met, the user cannot require the total cost to be less than any given number. Thus, by allowing no freedom in the constraints, the objective function must be totally satisfied.

After running \( FLP123 \), the following results were obtained:

Cost = $1163

\( x_{11} = 0.0, \quad x_{22} = 22.7, \quad x_{33} = 0.0, \quad x_{44} = 63.6. \)

Fuzzy Example:

If the farmer has no leeway in the amount of food he must buy, then the crisp solution above would be his optimal buying plan. However, life is often not so clear-cut that a classical linear program can give a satisfactory answer. All the vitamins needed have a tolerance range. It is more realistic to say that \( about \) 300 grams of vitamin \( C \) is needed. For a case like this, a linear program with \( soft \) constraints can be useful for determining how much more the costs can be minimized if the farmer skims on purchases of different amount of grains. The following fuzzy linear program deals with the case where the farmer allows for variations in the optimal amounts of each grain in order to bring costs down further.
Min \( C^\text{cost} = 12x_{11} + 12x_{22} + 13x_{33} + 14x_{44} \leq (0, 0, 1000, 117) \)
subject to:

\begin{align*}
(1) & \quad 2x_{11} + 3x_{22} + 7x_{33} + 5x_{44} \geq (20, 100, \infty, 0) \\
(2) & \quad x_{11} + 6x_{22} + 2x_{33} + x_{44} \geq (40, 200, \infty, 0) \\
(3) & \quad 3x_{11} + 2x_{22} + x_{33} + 4x_{44} = (30, 300, 300, 30) \\
(4) & \quad x_{11} \geq 0, \quad x_{22} \geq 0; \quad x_{33} \geq 0, \quad x_{44} \geq 0
\end{align*}

where the four elements of the right hand term characterize the trapezoidal fuzzy membership function (see Equation 3.6).

Notice that this problem formulation is almost the same as the crisp formulation and the only difference now is that each of the constraints has a fuzzy variable.

Now the objective function is no longer allowed to vary from zero to infinity. We want to make costs zero if possible, but by definition they cannot go lower than that. The middle two elements require cost to be between 0 and 1000 to be perfectly satisfied. If absolutely necessary, cost can go as high as 1117, but no further.

The first two constraints are similar. In the crisp case, \( 2x_{11} + 3x_{22} + 7x_{33} + 5x_{44} \) can sum to any number between 100 and infinity. Now \( 2x_{11} + 3x_{22} + 7x_{33} + 5x_{44} \) can sum to 80 (20 under 100). If the equation does sum to some number between 80 and 100, the satisfaction index \( \alpha \) (which is a measure of the satisfaction level of a constraint: 1.0 means full satisfaction and 0.0 means no satisfaction) will fall below 1.0 to signify some dissatisfaction with the results. The second constraint will be perfectly satisfied at any number over 200. However, the sum of the variables can fall as low as 160 if
necessary.

The third constraint reflects the same philosophy, but its right-hand-side structure is slightly different because it is an equality constraint rather than a greater-than constraint. The equation can only fully be satisfied if $3x_1 + 2x_2 + x_3 + 4x_4$ sums to exactly 300. However, the sum can be as low as 270 or as high as 330, and still be acceptable.

**Goal Satisfaction: Autogoal**

When fuzzifying a problem, it is often difficult to determine how severely the objective function should be constrained. The farmer might like to see a cost that is under $500. Yet he will not know if that is possible with the constraints as tightly bound as they are. He may need to allow his vitamin A quota to fall below 50 to get such a low cost. Yet the way his problem is currently formulated, vitamin A can only fall to 80. Autogoal, a suggested goal setting procedure, is a feature of our software FLPI23 that will determine how far the objective function can vary under the current fuzzified constraints. Section 3.2.2 shows that in order to have a solution $\mu_G(z(\alpha))$ has to intersect with $\mu_C(z(\alpha))$ (in this example, $z=$cost). Autogoal determines the range of $[z_{co0} = \mu_C^{-1}(\alpha = 0), z_{ca1} = \mu_C^{-1}(\alpha = 1)]$ as shown in Figure (3-7).

The existence solution for this example is

$$Cost_{max} \geq z_{co0} \quad (3.25)$$

---

12 A feature of FLPI23.
In this way, the Autogoal feature suggests a possible goal setting range.

For this nutrition example, Autogoal will show a solution graph like Figure (3-8). Clicking with the computer’s mouse on the right side of the graph and the left side of the graph shows the numbers between which the objective function will vary. In this problem, Autogoal gave $z_{c_0} = 1000$ and $z_{c_1} = 1117$.

Final Discussion of Fuzzy Example

After running this problem in FLP123, we get the result shown in Figure (3-8). Notice that there are two lines, demonstrating that the objective function has variable satisfaction levels. The line which has a decreasing value of $\mu(\alpha)$ from left to right is the constraint degree of satisfaction $\mu_C(\alpha)$. The line which has an increasing value of $\mu(\alpha)$ from left to right, is the degree of goal satisfaction $\mu_G(\alpha)$. The overall degree of satisfaction $\mu(\alpha)$ is the minimization of the two values.

\[ 13^0 = 1.0 - \alpha. \]
When we maximize the overall degree of satisfaction \( \mu(\alpha) \), the following results are obtained:

\[
\text{cost} = 1,094, \quad \alpha = 0.5
\]

\[
x_{11} = 0.0, \ x_{22} = 19.7, \ x_{33} = 0.0, \ x_{44} = 61.3.
\]

![Graph](image)

**Figure 3-8: Results of Nutrition Problem**

Constraint (1) in Equation (3.24) is fully satisfied (summation is 365.5); constraint (2) and constraint (3) are binding constraints with the same degree of constraint satisfaction of 0.5 (summations are 180 and 285 respectively). Also notice that total cost is 6.3% lower; the farmer can indeed shrink his costs by $69.
3.4 Summary

Fuzzy mathematical programming extends classical mathematical programming by allowing acceptable constraint violations. The $FMP$ approach has two advantages: (1) it allows uncertainty in the equation; (2) it has a larger feasible region than that in a $CMP$, so the solution will be better in the Pareto sense; (3) it can locate the critical boundaries for further improvement by checking the binding constraints.

Table (3.1) compares the classical mathematical programming and fuzzy mathematical programming approaches. It is important to notice that though the objectives, the constraints and the technology matrix coefficients (coefficients of the left side of the constraints) can be soft boundaries in $FMP$, the final decisions (solutions) are always crisp.
<table>
<thead>
<tr>
<th></th>
<th>CMP</th>
<th>FMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goals</td>
<td>Maximize/minimize</td>
<td>Maximize satisfaction of goals</td>
</tr>
<tr>
<td></td>
<td>objective function</td>
<td>AND constraints taken together</td>
</tr>
<tr>
<td>Human</td>
<td>Not Allowed</td>
<td>Can allow fuzzy constraints to be</td>
</tr>
<tr>
<td>Preference</td>
<td></td>
<td>very soft, i.e., variables go to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>extremes, or very hard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(likewise for goals).</td>
</tr>
<tr>
<td>Objectives</td>
<td>Crisp</td>
<td>soft</td>
</tr>
<tr>
<td>Constraints</td>
<td>Crisp</td>
<td>Soft or Crisp</td>
</tr>
<tr>
<td>Technology Matrix</td>
<td>Crisp</td>
<td>Soft or Crisp</td>
</tr>
<tr>
<td>Coefficients</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final Decisions</td>
<td>Crisp</td>
<td>Crisp</td>
</tr>
</tbody>
</table>

Table 3.1: Comparison of CMP and FMP
Chapter 4

Linguistic Rule Base Module

Linguistic rule bases provide an alternative or complement to analysis where traditional mathematical model does not exist, but where human expertise is widely available. This chapter discusses the imprecision of human knowledge in the process of establishing linguistic knowledge bases, implements a fuzzy rule network (FRN) structure for representing linguistic knowledge and a method for adaptive rule calibrations to maximize overall performance. A Random-Sampling-Without-Replacement (RSWOR) technique for membership function elicitation on experimental designs is also investigated.
4.1 Human Expertise and Knowledge Imprecision

Approximate reasoning plays a central role in human thinking as well as in the inquiries and meaning of various soft or empirical sciences. All human-in-the-loop systems are more or less affected by subjective expertise which is not a simple boolean-type logic. Human logic theory seeks to classify the varieties of subjective reasoning and the codification of rules, to assess degrees of belief of conclusions and to investigate rule rationality.

In the knowledge elicitation process, Szolovits and Pauker [83] found that experts often refuse to give precise numerical estimates of outcomes and tend to make verbal judgments. Kouchen [45] further concluded that precision at clarity beyond certain point may be as ineffective in representing the truth as vagueness.

Imprecision arises from a variety of sources, such as incomplete knowledge, inexact language, ambiguous definitions, inherent stochastic characteristics, measurement errors, et cetera, and has been quantified primarily by means of probability theory. Several authors have emphasized the need for differentiating among the sources of imprecision underlying particular assumptions or items of evidence. As discussed in Chapter 1, the linguistic knowledge base, or the fuzzy knowledge system, is a promising approach. It draws conclusions by measuring linguistic rules and the consistency of facts in terms of possibility, and thereby forms a special kind of knowledge system.
4.1.1 Knowledge Systems

The study of knowledge systems investigates methods and techniques for constructing human-machine systems with specialized problem-solving expertise. A knowledge system simulates the behavior of a human expert in a specialized problem domain, offers intelligent advice and justifies its inference. Figure (4-1) shows a typical human-in-the-loop system. The process, which may be a dynamic process, economic process or social process, is controlled, diagnosed or evaluated by a human expert based on input-output state observations.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{human_in_loop.png}
\caption{Human-in-the-loop Process}
\end{figure}

Babbage and Boole, who lived in early nineteenth century England, are the intellectual grandfathers of knowledge systems. Babbage designed an analytical engine to play championship chess and Boole developed a calculus of logic. The modern history of knowledge systems begins with digital electronic computers. The 1980s witnessed an enormous burgeoning of academic and industrial interest worldwide in the development of expert systems. Among them, Mycin and Prospector are two typical examples [11] [35].

Mycin addresses the problem of diagnosing and treating infectious blood diseases.
Its knowledge comprises approximately 400 rules relating possible conditions to associated interpretations. *Mycin* tests a rule's conditions against available data or requests data from the physician. Its use of independent rules with the simple *If-Then* form stimulated a variety of related systems. *Prospector* was developed to use an analogous form of knowledge representation for mineral deposit relationships. It includes about a dozen knowledge bases for different kinds of deposits.

*Mycin* was originally thought by many to incorporate knowledge in a form similar to that used by physicians; however, it was subsequently seen as too arbitrary and difficult to extend. *Neomycin* was heralded as the *cognitively realistic* successor to *Mycin*. *Neomycin* separates strategy and domain knowledge. The *Mycin* research group has directed much of it efforts of the past few years to work on an oncology ward information-management system called *Oncocin*. The oncologist would not want to seek advice from the computer, if s/he had to first type into the computer large amounts of previously collected and recorded data about the patient. Accordingly, *Oncocin* includes software for maintenance and an attractive display of patient records. When the physician comes to the terminal, the information available is similar to that in the standard paper medical record.

These two examples exhibit the characteristics of most knowledge systems which include the following characteristics: (a) operate in domains in which the precise effect of an action is not always predictable; (b) operate in incomplete domains in which crucial data may be missing or may be uncertain for a decision; (c) operate in domains so large that it may not be possible to predict the precise effect of an action, although in principle the effect should be determinable; and (d) have learning
components that enable them to extend their knowledge base while they operate.

The above major characteristics lead to an ideal architecture of a knowledge system which contains the following basic components: (1) a knowledge base which contains facts, rules, concepts and skills; (2) an inference engine, the reasoning system in which new knowledge is deduced from the already existing knowledge; (3) a learning component which acquires knowledge from users, learns the domain of the application and generates decision rules; (4) a user-friendly dialogue interface by which the user interacts with the system; (5) planning or control knowledge consisting of procedures to control the evaluation process and the communication with the external interface; and (6) an explanation component for explaining why a certain conclusion has been made.

It is seen that even the simplest expert systems have at least two basic parts: a knowledge base with a set of rules, and an inference engine. The knowledge base consists of facts and rules, while the inference engine consists of processes applied to the knowledge base.

A rule in the knowledge base is a pattern-invoked sentence. Of particular interest is the production rule that is in the form:

\[ \text{Condition } \Rightarrow \text{Action} \quad (4.1) \]

Meaning: \( \text{If ( condition ), then ( action )} \)

The condition (observation) is usually one or more predicates that test properties of
the current state of facts. The action is the output decision which changes the current state of the facts based on the condition.

4.1.2 Fuzzy Knowledge Systems

For the expert rules discussed above, both the observations and the actions are assumed to be crisply defined. In the real world, facts and rules from expertise are neither totally certain nor totally consistent. The rules need to be modified to have computational capability for imprecise or vague evidence from domain experts.

For this purpose, researchers have augmented their inference procedures with mechanisms which combine degrees of evidence according to the rules of plausible reasoning. As justified in Chapter 1, a promising approach is based on fuzzy set theory, or approximate reasoning, which means drawing conclusions by taking the consistency of the facts into account. The treatment of fuzziness is a new idea in knowledge representation. In all expert systems based on symbolic manipulation and plausible reasoning, uncertainty is supposed to reside in the state of the knowledge. In knowledge systems based on approximate reasoning, the emphasis is on fuzziness viewed as an intrinsic property of natural language. An evident advantage of the fuzzy set approach is the possibility of representing numeric and linguistic variables in a uniform way and of using a sound formalism to handle them.

A fuzzy knowledge system, which replaces the human in Figure (4-1), is shown in Figure (4-2). The system has a linguistic rule base to store human expertise and includes fuzzification, rule inference, defuzzification, as well as recursive rule
adaptation procedures.

Figure 4-2: A Fuzzy Knowledge System and Its Functions

A general fuzzy knowledge system consists of four major functions. The first function is the fuzzification procedure through which the physical measurements are mapped into fuzzy variables defined by membership functions corresponding to a human’s subjective degrees of belief. The second function is the linguistic rule base and its inference engine. The third function is the defuzzification procedure which maps the linguistic variables to crisp control signals. The fourth function, which is crucial in knowledge representation, is a recursive rule adaptation through which the linguistic rules are calibrated based on the teaching data from the input-output pairs.

The first function, fuzzification procedure, which maps the crisp measurements to fuzzy linguistic variables, is an important human-machine interface reflecting part of the subjective and imprecise knowledge. The other three procedures are related to
each other, and aim to achieve the desired input-output relations. In the following sections, we are going to discuss membership elicitation and adaptive rulebase learning in detail.\footnote{The basic ideas of fuzzification and defuzzification procedures are explained in Appendix A.}

4.2 Fuzzy Membership Elicitation

4.2.1 Inexactness of Linguistic Terms and Fuzzy Memberships

As discussed in the previous section, any output measurement from a process with a crisp value has to be fuzzified according to a predefined membership function which represents linguistic terms. For a linguistic term, there exist three kinds of inexactness: generality, in which a word applies to a multiplicity of objects in the field of reference; ambiguity, which occurs when a finite number of alternative meanings have the same phonetic form; and vagueness, in which there are no precise boundaries to the meaning of a word.

This inexactness in linguistic rules can lead to disaster. For example [5], in 1961, President Kennedy ordered a review of the CIA's plan for an invasion of Cuba by expatriates. The general in charge concluded that the chances of overall success were fair, by which allegedly he meant that they were thirty percent. But the report to the president did not mention the percentage; instead the report stated: "The plan has a fair chance of ultimate success". The rest is history. Later the general recalled:
“We thought other people would think that a fair chance would mean not too good”.

However, there is no consensus on how best to represent linguistic variables (such as cold, warm, short, tall, et cetera) as membership functions. It sometimes is concluded that there are two attributes that generally describe fuzzy membership functions: the crossover point and the support [31]. The crossover point is a point whose degree of membership is 0.5, and the support is the set of points having a degree of membership greater than 0. The crossover point in turn is affected by the shape of the membership function. This means that determining the shape and the support of a membership function is very important.

4.2.2 Membership Elicitation

Since a membership function is subjective, membership elicitation is an essential procedure for defining linguistic terms. The process of generating fuzzy membership functions can be done by using clustering algorithms [8] [84]. With this method, a user defines the number and relative order of linguistic terms as they apply to a particular attribute. The clustering algorithm determines membership functions based on sample data.

In the clustering process, it is necessary to generate enough sample data for analysis. Few papers have touched on the issue of how to design an experiment with human subjects (experts), minimizing the number of samples or maximizing the credibility of the resulting membership function.
Random Sampling Without Replacement (RSWOR)

In the past, experiments have been designed in such a way that the subjects were prompted with random inputs generated by a computer-supplied random generator. While it is true that the computer-supplied random generator theoretically has the property of uniform distribution, the assumption is based on infinitely large samples. For any real experiment with subjects, however, the number of samples is limited and therefore the property of uniform distribution is not guaranteed.

![Histogram from the Compiler-supplied Random Generator]

**Figure 4-3:** Histogram from the Compiler-supplied Random Generator

For instance, we need a series of random values from 30 to 90 and set the total number of samples as 1220. For a desired uniform distribution, each value within the range should appear exactly 20 times. However, the histogram of limited samples from a compiler-supplied random generator shows that the actual frequency varies from 13 to 27 as shown in Figure (4-3) where the ordinate is each random number and the abscissa is the frequency. It is seen that if we use this random data set in experiments without a complex compensation algorithm, the resulting membership function will certainly be distorted.
To solve this problem, a method called \textit{random sampling without replacement} (\textit{RSWOR}) is proposed. The main idea is to create a number pool. First, a fixed number of samples for each discrete value in the range are thrown into the pool. Then one number is picked at random from the pool until all numbers are exhausted. In this way, the whole sampling will be a guaranteed strictly uniform distribution and the resulting membership will not be distorted due to uneven sample distributions.

Figure (4-4) shows the resulting histogram of a random generator using the \textit{RSWOR} method. The distribution is strictly uniform.

![Histogram](image)

\textit{Figure 4-4: Histogram from the RSWOR Method}

**Linguistic Term Evaluator**

A computer program, \textit{Linguistic Term Evaluator}, has been developed to elicit fuzzy membership functions using the \textit{RSWOR} method. Figure (4-5) shows an example screen display for evaluating the linguistic variable \textit{Driving-Speed-on-the-Highway}. Driving speed was chosen for demonstration because of the easy availability of experts...
(other linguistic variables could be evaluated using the same method).

![Linguistic Term Evaluator](image)

**Figure 4-5: Membership Elicitation Experiment**

The speed ranges from 30 mph to 90 mph, which is shown by a grey rectangle. It has 61 discrete possible values. For each value, the sampling frequency is set to 5 and hence there are total $61 \times 5 = 305$ numbers in the pool. Five linguistic terms, *Very Slow, Slow, Normal, Fast and Very Fast*, are evaluated.

The subject is prompted with a random *RSWOR* speed, and is asked to evaluate each speed by clicking a mouse on the appropriate linguistic buttons on the right side of the screen. Upon clicking the mouse, a row of degree of belief from 0 to 100
(percentage) appears on the left of these buttons. As a default, the degree of belief for the button clicked is 100 and the others are 0. However, the subject has the option to modify the numbers according his/her judgment (the degrees of belief for five terms are constrained to add up 100 for consistency). For instance, in Figure (4-5) the subject was prompted with a random speed of 54 mph. He evaluated this speed as Normal with a degree of belief at 0.9 and as Slow with a degree of belief at 0.1.

A weighted histogram of all the degrees of belief is constructed after the RSWOR samplings are completed. It turns out that all the memberships for five linguistic terms have reasonable ranges and move to the left for more conservative drivers in the other experiments.

Figure (4-6) shows a typical resulting histogram without any modifications by using the methods discussed above. The ripples partially reflect the imprecision of human judgment.

To smooth the curves in Figure (4-6), we use a trapezoidal membership form (with four unknown parameters), because of its simplicity of being linear. Figure (4-7) shows the membership function for Driving-Speed-On-the-Highway after data fitting.

After the membership elicitation (which is an important step to implement a linguistic rule base), we will discuss the structure of a linguistic knowledge base and its adaptive learning mechanism in the next section.
Figure 4-6: Histogram of Driving Speed

Figure 4-7: Membership Function of Driving Speed
4.3 Structure of a Linguistic Rule Base

At the center of a fuzzy knowledge system is a linguistic rule base with a desired input-output fuzzy relation that needs to be calibrated to make the fuzzy knowledge system useful.

4.3.1 Fuzzy Relation

Any decision-making process involves input states and output actions. The function between input and output is called a relation. A crisp relation shows the presence or absence of the association and interaction between elements of two or more sets. Each relation has a strength of connection which is either one or zero, corresponding to whether the relation is connected or not. A fuzzy relation is an extension of a crisp relation. In a fuzzy relation, the strength of connection is no longer only one or zero; it can take any value between them.

To show the property of a fuzzy relation, we assume a binary relation between two variables ($x$ and $y$) in two sets ($\tilde{X}$ and $\tilde{Y}$), where $\tilde{X}$ is the input and $\tilde{Y}$ is the output. The binary fuzzy relation, defined in the Cartesian set $\tilde{X} \times \tilde{Y}$, is a mapping from $\tilde{X}(x)$ to $\tilde{Y}(y)$ through $\tilde{R}(x,y)$ that is defined as:

$$\tilde{R} : \tilde{X} \times \tilde{Y} \Rightarrow [0, 1]$$

$$\forall x \in \tilde{X}(x), \quad y \in \tilde{Y}(x).$$
For each pair of elements \( (x_i, y_i) \in (\tilde{X}, \tilde{Y}) \), there exists an element of \( \tilde{R}, \tilde{r}_{ij} \in [0, 1] \), which expresses the strength of connection between the pair \( x_i \) and \( y_j \). \( \tilde{R} \) is called the relation between input \( \tilde{X} \) and output \( \tilde{Y} \).

When the universe of discourse is infinite, the relation takes a continuous form. For instance, the word *similar* can be mathematically expressed in the form

\[
\tilde{R}(x, y) = \begin{cases} 
\frac{1}{1+(x-y)^5} & \text{if } |x - y| \leq 5 \\
0 & \text{otherwise}
\end{cases}
\]  

(4.2)

When the universe of discourse is finite, the relation is in a discrete form and is treated as a fuzzy relation matrix.

\[
\tilde{R}(\tilde{X}, \tilde{Y}) = \begin{bmatrix}
\tilde{r}_{11} & \tilde{r}_{12} & \cdots & \tilde{r}_{1m} \\
\tilde{r}_{21} & \tilde{r}_{22} & \cdots & \tilde{r}_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
\tilde{r}_{i1} & \tilde{r}_{i2} & \cdots & \tilde{r}_{ij} \\
\vdots & \vdots & \ddots & \vdots \\
\tilde{r}_{n1} & \tilde{r}_{n2} & \cdots & \tilde{r}_{nm}
\end{bmatrix}
\]  

(4.3)

where \( \tilde{r}_{ij} \in [0, 1] \) \( i = 1, 2, \ldots m; \) \( j = 1, 2, \ldots n. \)

For instance, Equation (4.4) is an example of a fuzzy relation matrix \( \tilde{R} \) defined in \( \tilde{X} = \{x_1, x_2\} \) and \( \tilde{Y} = \{y_1, y_2, y_3\} \). Here \( \tilde{r}_{ij} \in [0, 1] \) is a kind of correlation between the variables \( \tilde{X} \) and \( \tilde{Y} \).

\[
\tilde{R}(\tilde{X}, \tilde{Y}) = \begin{bmatrix}
0.3 & 1.0 & 0.0 \\
0.9 & 0.5 & 0.1
\end{bmatrix}.
\]  

(4.4)
In general, suppose we have a linguistic rule base:\(^2\):

\[
\begin{align*}
&IF \ X_1^1 \text{ and } X_2^1 \text{ and } \cdots \text{ and } X_n^1 \text{ THEN do } Y_1 \\
&IF \ X_1^2 \text{ and } X_2^2 \text{ and } \cdots \text{ and } X_n^2 \text{ THEN do } Y_2 \\
&IF \ \cdots \ \cdots \ \cdots \ \cdots \ \text{ do } \cdots \\
&IF \ \cdots \ \cdots \ \cdots \ \cdots \ \text{ do } \cdots \\
&IF \ X_1^i \text{ or } X_2^i \text{ and } \cdots \text{ and } X_n^i \text{ THEN do } Y_i \\
&IF \ \cdots \ \cdots \ \cdots \ \cdots \ \text{ do } \cdots \\
&IF \ \cdots \ \cdots \ \cdots \ \cdots \ \text{ do } \cdots \\
&IF \ X_1^m \text{ and } X_2^m \text{ and } \cdots \text{ and } X_n^m \text{ THEN do } Y_m
\end{align*}
\]

Table 4.1: Linguistic Rules

The example below is a linguistic rule concerning train length given the volume of incoming traffic and the time-past-schedule:

"IF the incoming traffic is small

and the time-past-schedule is long,

THEN a short train can be dispatched".  

(4.5)

This one rule forms one fuzzy relation \(\hat{R}_i(\hat{X}_1, \hat{X}_2, \hat{Y}) = \hat{X}_{1i} \times \hat{X}_{2i} \times \hat{Y} \). In the same way, the \(m\) rules in Table (4.1) constitute \(m\) fuzzy relations, each of them has the form:

\[
\hat{R}_i(\hat{X}_1, \hat{X}_2, \cdots \hat{X}_n, \hat{Y}_i) = \hat{X}_{1i} \times \hat{X}_{2i} \times \cdots \times \hat{X}_{ni} \times \hat{Y}_i \quad i = 1, 2, \cdots m. \tag{4.6}
\]

\(^2\)Where the logical or can also be used in the place of, or in combination with logical and.
The overall relation matrix is

\[ \tilde{R} = \bigcup_{i=1}^{m} \tilde{R}_i \]

where \( \tilde{R}_i \) represents the \( i \)-th rule.

The antecedents of a \( \tilde{R}_i \) form a fuzzy set

\[ \tilde{X}_1 \times \tilde{X}_i \times \cdots \times \tilde{X}_n \]

in the Cartesian space of

\[ U_1 \times U_i \times \cdots \times U_n. \]

Hence, the \( i \)-th rule \( \tilde{R}_i \) can be regarded as a fuzzy mapping:

\[ \tilde{R}_i : \tilde{X}_1 \times \tilde{X}_i \times \cdots \times \tilde{X}_n \rightarrow (\tilde{Y}_i). \]
To further understand the fuzzy relation, let us look at a two-input single-output example:

\[
\begin{align*}
\text{Input:} & \quad x \text{ is } \tilde{A}' \quad \text{and} \quad y \text{ is } \tilde{B}' \\
\text{Rule } \tilde{R}_1: & \quad \text{if } x \text{ is } \tilde{A}_1 \quad \text{and} \quad y \text{ is } \tilde{B}_1 \quad \text{then} \quad z \text{ is } \tilde{C}_1 \\
\text{Rule } \tilde{R}_2: & \quad \text{if } x \text{ is } \tilde{A}_2 \quad \text{and} \quad y \text{ is } \tilde{B}_2 \quad \text{then} \quad z \text{ is } \tilde{C}_2 \\
& \vdots \\
\text{Rule } \tilde{R}_i: & \quad \text{if } x \text{ is } \tilde{A}_i \quad \text{and} \quad y \text{ is } \tilde{B}_i \quad \text{then} \quad z \text{ is } \tilde{C}_i \\
& \vdots \\
\text{Rule } \tilde{R}_m: & \quad \text{if } x \text{ is } \tilde{A}_m \quad \text{and} \quad y \text{ is } \tilde{B}_m \quad \text{then} \quad z \text{ is } \tilde{C}_m \\
\text{Output:} & \quad z \text{ is } \tilde{C}'
\end{align*}
\]

Table 4.2: Two-input, Single-output Example

where $\tilde{A}_i$, $\tilde{B}_i$, $\tilde{C}_i$ are linguistic terms of the linguistic variable $x, y, z$ in the universes of discourse $U, V$ and $W$, respectively. The fuzzy rule if \( (x \text{ is } \tilde{A}_i \text{ and } y \text{ is } \tilde{B}_i) \) then \( (z \text{ is } \tilde{C}_i) \) is implemented as a fuzzy relation $\tilde{R}_i$ that is defined as

\[
\mu_{R_i} = \mu_{A_i \text{ and } B_i} \rightarrow \tilde{C}_i(u, v, w) = [\mu_{A_i}(u) \text{ and } \mu_{B_i}(v)] \rightarrow \mu_{C_i}(w)
\]

where "$\tilde{A}_i \text{ and } \tilde{B}_i$" is a fuzzy set in $U \times V$; $\tilde{R}_i = (\tilde{A}_i \text{ and } \tilde{B}_i) \rightarrow \tilde{C}_i$ is a fuzzy relation in $U \times V \times W$; and \( \rightarrow \) denotes a fuzzy mapping function. The consequence $\tilde{C}'$ is deduced from the compositional rule of inference employing the definitions of a fuzzy relation function and the connectives $\text{and}$ and $\text{or}$.

$\tilde{R}$ is a key element in a linguistic rule base, since it determines the relation between the input and the output. We use a fuzzy rule network structure to represent the input-output relation including $\tilde{R}$.
4.3.2 Fuzzy Rule Network (FRN)

The essential idea of fuzzy rule base is to make use of human expertise with linguistic if-then rules. The rule base emulates the behavior of human experts to derive proper actions. In contrast to modern control theory which has a design scheme such as pole-placement or state-space design approach, the design of a fuzzy rule base depends entirely on the knowledge and experience of experts, or the physical sense as well as intuitions that are not well structured.

The issue of recursive learning is therefore of particular interest for a fuzzy rule base. Instead of designing a perfect rule base, the strategy is to allow it to learn proper input-output relationships. Mamdani [56] proposed a performance index table for updating rules with some success, but the generation of the performance index table itself is not an easy job. Barto, Sutton and Anderson [3] used two neuron-like elements to solve the cart-pole problem. Lee and Berenji [53] extended further to incorporate neural network architecture in their work. However, these structures suffer from a lack of generality and may be difficult to apply for larger scale systems due to the fact that developing mathematical functions for the trace function and the credit assignment are not trivial.

Later, various of fuzzy-neural-networks type knowledge bases have been proposed, such as the associative memory neural networks which controls an experimental helicopter by Yamaguchi, Goto and Takagi [97]. These approaches adapt the neural networks' learning algorithm as well as the weight structures. There is no indication of convergence of learning and the weights have no physical meanings.
We are implementing a fuzzy rule structure – Fuzzy Rule Network (FRN) which has the capability of adaptive supervisor learning based on teaching of input-output pairs. The structure of an FRN is shown by example in Figure (4-8). For clarity of explanation, the example has only two inputs \((x_1, x_2)\) and one output \((y)\).

![Fuzzy Rule Network (FRN)](image)

**Figure 4-8: Fuzzy Rule Network (FRN)**

An FRN has three layers. The input layer, or fuzzification layer, includes a fuzzifier whose task is to match the values of the input variables \(i\) to the linguistic terms \(k\) used in fuzzy rules. \(\mu_{ik}\) is the degree of belief for the corresponding partitioned linguistic terms. The hidden layer, or fuzzy logic layer in this network corresponds to the rules and the decision-making logic which is characterized by the triangular
T-norms and T-conorms. \( D(\mu_{ik}) \) is a rule's final left-hand-side confidence (LHSC) or membership over all the antecedents of a rule (see Table 4.1). \( D(\mu) \) is expressed as

\[
D(\mu_{.}) = LHSC \ (IF \ \tilde{X}_1^i \ and \ \tilde{X}_2^i \ and \ \ldots \ and \ \tilde{X}_n^i ) \quad (4.7)
\]

In our two-input, one-output example in Figure (4-8), \( D(\mu_{ik}) \) is

\[
D(\mu_{ik}) = \mu_i(x_1) \ AND \ \mu_i(x_2) = T_{\text{norm}}(\mu_i(x_1), \mu_i(x_2)). \quad (4.8)
\]

When the logic operator in the rule is or instead of and, equation (4.8) becomes

\[
D(\mu_{ik}) = \mu_i(x_1) \ OR \ \mu_i(x_2) = T_{\text{conorm}}(\mu_i(x_1), \mu_i(x_2)). \quad (4.9)
\]

Since in many practical applications a crisp control action is required, the defuzzification which maps from a space of fuzzy actions, defined over an output universe of discourse into a space of nonfuzzy value, is necessary. Unfortunately there is no unique systematic procedure to do this. Zadeh [99] first pointed out this problem and made the tentative suggestions for dealing with it. At present, the commonly used strategy (see Appendix A) is the center-of-area defuzzification method which states:

\[
\hat{y} = \frac{\sum_{i=1,k=1}^{m,n} [D(\mu_{ik}) \cdot w_{ik}]}{\sum_{i=1,k=1}^{m,n} D(\mu_{ik})} \quad (4.10)
\]

\(^3T_{\text{norm}}\) and \(T_{\text{conorm}}\) are fuzzy logic operators explained in Appendix A.
where $w_{ik}$ is the right-hand value of a rule, such as

$$\text{IF } \tilde{x}_1 \text{ and } \tilde{x}_2^k \text{ THEN do } w_{ik}$$

(4.11)

Integrating the inference engine and the defuzzification procedures into one single output layer is the main feature of the FRN approach.

Equation (4.10) can be rewritten as:

$$\hat{y} = \sum_{i=1, k=1}^{m, n} (r_{ik} \cdot w_{ik})$$

(4.12)

where

$$r_{ik} = r_{ik}(x_1, x_2) = D(\mu_{ik})/ \sum_{i=1, k=1}^{m, n} D(\mu_{ik})$$

(4.13)

The above equations show that the output layer in the FRN combines fuzzy inference and defuzzification into one single procedure. This approach enables us to figure out what really is in a human subject’s mind by identifying the $w_{ik}$ from the input-output pairs in a way similar to neural network learning.

Different from ε neural network, which has no systematic way to determine the number of neurons and in which the weights have no physical meanings, the FRN has a meaningful structure and the identified parameters have physical meanings. In the FRN, the number of nodes and their distributions are closely related to the fuzzy variables and their partitions. The number of nodes in the fuzzification layer depends on the number of input variables, as well as the number of partitions in their
corresponding universes of discourse. The fuzzy logic layer represents the antecedent of the linguistic rule base $\tilde{R}$ and forms fuzzy sets $\tilde{X}_1 \times \tilde{X}_2$ in the Cartesian space of $U \times V$. The number of nodes in the fuzzy logic layer equals the number of elements in the Cartesian space. Each element has the value of $\mu_{ik} = \mu(x_{1i} \text{ and } x_{2k})$. The number of nodes in the defuzzification layer equals the number of outputs. Each learned parameter between the fuzzy logic layer and the defuzzification layer is the decision value for the corresponding rule.

If we rewrite Equation (4.12) in a matrix form, it is:

$$\dot{y} = \tilde{R} \times \tilde{W}^T$$

(4.14)

where

$$\tilde{R} = \begin{bmatrix}
    r_1^1 \\
    r_2^2 \\
    \vdots \\
    r_i^i \\
    \vdots \\
    r_n^n
\end{bmatrix}, \quad \tilde{W} = \begin{bmatrix}
    w_1^1 \\
    w_2^2 \\
    \vdots \\
    w_i^i \\
    \vdots \\
    w_n^n
\end{bmatrix}$$

(4.15)

and

$$\tilde{r}_i = \begin{bmatrix}
    r_{1i} \\
    r_{2i} \\
    \vdots \\
    r_{ki} \\
    \vdots \\
    r_{mi}
\end{bmatrix}, \quad \tilde{w}_i = \begin{bmatrix}
    w_{1i} \\
    w_{2i} \\
    \vdots \\
    w_{ki} \\
    \vdots \\
    w_{mi}
\end{bmatrix}$$

(4.16)
To make the formula clearer, we define

\[
R = \begin{bmatrix}
    r_1 & r_2 & \cdots & r_n \\
    r_{11} & r_{12} & \cdots & r_{1n} \\
    r_{21} & r_{22} & \cdots & r_{25} \\
    \vdots & \vdots & \ddots & \vdots \\
    r_{m1} & r_{m2} & \cdots & r_{mn}
\end{bmatrix}
\]

and

\[
W = \begin{bmatrix}
    w_{11} & w_{12} & \cdots & w_{1n} \\
    w_{21} & w_{22} & \cdots & w_{25} \\
    \vdots & \vdots & \ddots & \vdots \\
    w_{m1} & w_{m2} & \cdots & w_{mn}
\end{bmatrix}
\]

Therefore, Equation (4.14) is symbolically written as:

\[
\hat{y} = R \times W
\]

4.3.3 Recursive Learning of an FRN

We have shown that a linguistic rule base can be interpreted as a fuzzy relation in which fuzzy variables are used to describe linguistic rules in a fuzzy rule network. After the network is established, it needs to be calibrated.
The manual calibration of a linguistic rule base takes a long time, if not an impossible time for large scale systems. In practice, engineers spend days and days tuning up rules and membership functions by trial and error to make a fuzzy rule base work as desired. We are going to show that the FRN has the capability of recursive learning.

To illustrate the recursive learning mechanism, the FRN is redrawn in the form of a block diagram as shown in Figure (4-9).

![Block Diagram of Fuzzy Recursive Learning](image)

**Figure 4-9: Fuzzy Recursive Learning**

We notice that $R$ (tilde is omitted for clarity), which is a non-linear function, is the relation matrix and that $W$, which is a linear function, is the fuzzy network weighting matrix. For the given input ($x_1$ and $x_2$ in Figure 4-8), matrix $R$ is fixed when the membership functions are fixed. $W$ is an adjustable matrix, in which each element can be identified in such a way that the output $\hat{y}_i$ (computed from Equation 4.19 with the input values $x_i$) agree as closely as possible with the measured output $y_i$. That is, the cost function

$$J(W) = \frac{1}{2} \sum_{n=1}^{N} c_k^2,$$

(4.20)
is minimized, where

$$
\epsilon_i = y_i - \hat{y} = y_i - \sum_{i=1,k=1}^{m,n} (r_{ik} \cdot w_{ik}) \quad i = 1, 2, ..., N
$$

Because the relation between the input and output of $W$ is linear, we can recursively update the weighting matrix $W$ based on the teaching data using the least-squares method which states as follows.

Suppose we have $N$ sample pairs $(x_{1i}, x_{2i}, y_i, i = 1, 2, \cdots, N)$. The following vector notations are introduced:

$$
Y = \begin{bmatrix}
    y_1 - (\bar{R}_0)_1 \\
    y_2 - (\bar{R}_0)_2 \\
    \vdots & \vdots \\
    y_N - (\bar{R}_0)_N
\end{bmatrix} \quad \Phi = \begin{bmatrix}
    \bar{R}^T(x_1) \\
    \bar{R}^T(x_2) \\
    \vdots \\
    \bar{R}^T(x_N)
\end{bmatrix}
$$

$$
\bar{\epsilon} = \begin{bmatrix}
    \epsilon_1 \\
    \epsilon_2 \\
    \vdots \\
    \epsilon_n
\end{bmatrix}^T
$$

The least squares problem can now be formulated in a compact form. The cost function $J$ is written as

$$
J(\bar{W}) = \frac{1}{2} \bar{\epsilon}^T \bar{\epsilon} = \frac{1}{2} ||\bar{\epsilon}||^2
$$
where
\[
\hat{\epsilon} = Y - \hat{Y} \quad \text{and} \quad \hat{Y} = \Phi \hat{W}
\]

The solution to the above least-squares problem is given by

\[
\Phi^T \Phi \hat{W} = \Phi^T Y
\]  \hspace{1cm} (4.23)

and

\[
\hat{W} = (\Phi^T \Phi)^{-1} \Phi^T Y
\]  \hspace{1cm} (4.24)

In this way, the initial weight matrix \( W \) is identified with the \( N \) sample pairs.

In many cases, observations are obtained sequentially. It may then be desirable to compute the weights for different values of \( N \). If the least-squares problem has been solved for \( N \) observations, it is a waste of computational resources to start from scratch when a new observation is obtained. Hence, it is desirable to arrange the computations in such a way that the results obtained from \( N \) observations can be used in order to get the estimates for \( N + 1 \) observations.

Recursive equations can be derived for the case when the observations are obtained sequentially. We denote \( \hat{W}(N) \) as the least-squares estimate based on \( N \) measurements. To derive the equations, \( N \) is introduced as a formal parameter in
the functions:

\[
\Phi(N) := \begin{bmatrix}
\tilde{R}^T(x_1) \\
\tilde{R}^T(x_2) \\
\vdots \\
\tilde{R}^T(x_N)
\end{bmatrix}, \quad \quad \quad \quad Y(N) = \begin{bmatrix}
y_1 - (\tilde{R}_0)_1 \\
y_2 - (\tilde{R}_0)_2 \\
\vdots \\
y_N - (\tilde{R}_0)_N
\end{bmatrix}
\]  

(4.25)

It is assumed that the matrix \( \Phi^T \Phi \) is regular for all \( N \). The least-squares estimate \( \hat{\tilde{W}}(N) \) is then given by Equation (4.24)

\[
\hat{\tilde{W}}(N) = [\Phi^T(N)\Phi(N)]^{-1}\Phi^T(N)Y(N)
\]  

(4.26)

When an additional measurement is obtained, a row is added to the matrix \( \Phi \) and an element is added to the vector \( Y \). Hence

\[
\Phi(N+1) = \begin{bmatrix}
\Phi(N) \\
\tilde{R}^T(x_{N+1})
\end{bmatrix}, \quad \quad \quad \quad Y(N+1) = \begin{bmatrix}
Y(N) \\
y_{N+1}
\end{bmatrix}
\]  

(4.27)

The estimate \( \hat{\tilde{W}}(N + 1) \) given by (4.26) can be written as

\[
\hat{\tilde{W}}(N + 1) = [\Phi^T(N + 1)\Phi(N + 1)]^{-1}\Phi^T(N + 1)Y(N + 1)
\]

\[= [\Phi^T(N)\Phi(N) + \tilde{R}(N + 1)\tilde{R}^T(N + 1)]^{-1} \times [\Phi^T(N)Y(N) + \tilde{R}(N + 1)y_{N+1}]
\]

The recursive solution can be obtained by least squares algorithm described in Appendix B.
4.3.4 Discussion of the Weight Matrix \( W \)

An \( FRN \) has many features that attract attention: its linearity, its clear paradigmatic simplicity as a kind of parallel computation and its clear physical meaning in the weight matrix \( W \), which is the most attractive quality. In summary, it possesses the following features:

1. One unique feature of the weight matrix \( W \) is that the learned parameters in the weight matrix \( W \) are the decision values for the corresponding rules. That is, they are the right-hand-side values (\( \bar{Y}_i \)) of Equation (4.1).

2. When the input-output relation is not fully excited (i.e., the sample pairs do not cover all universe of discourse), accurate identification is impossible. In the \( FRN \), this can be isolated by identifying the null cells of the partitions which are subspaces defined by the antecedent variables. The corresponding element \( r_{ik} \) in the weight matrix \( W \) is set to zero.

3. The learned weight matrix is transparent to human experts. The experts know the physical meanings of the weight matrix and therefore the weight matrix can be checked and verified. In a neural network structure, learning process does not guarantee to find a solution (of all solvable problems) and there is no way of knowing the level of convergence, because the weights learned are meaningless. In an \( FRN \) structure, however, we can look at the learned weight matrix \( W \) and compare with the original elicited rules. When the patterns are not well matched, it means the learning has not converged, either because the
rule network is not well designed, the teaching data are conflicting, or more teaching data are needed.

4. The weight matrix can also be updated by gradient descent or the combination of gradient descent and least-squares method. The selection should reflect the trade-off between computational complexity and resulting performance.

4.3.5 Example

This example considers a two-input \((x\) and \(y\)), one-output \((z)\) system. The nonlinear control surface is determined by:

\[
z = 2.4 \times \sqrt{100 \left(1 - \frac{(x - 50)^2}{25}\right) - \frac{(y - 80)^2}{64}}
\]

where \(x \in [0, 100]\) and \(y \in [0, 160]\).

\(x\), \(y\) are partitioned into five linguistic terms with membership function represented by

\[
\tilde{X} = \{ \text{very small, small, medium, large, very large} \} \text{ and}
\]

\[
\tilde{Y} = \{ \text{very small, small, medium, large, very large} \}.
\]

The universal of discourse for \(\tilde{X}\) is \([0, 100]\) and the universal of discourse for \(\tilde{Y}\) is \([0, 160]\). Figures (4-10) shows the membership functions.

The relation matrix is defined in a Cartesian space of
Figure 4-10: X and Y Partitions

\[ R(X, Y) = \begin{bmatrix}
    r_{11} & r_{12} & \cdots & r_{15} \\
    r_{21} & r_{22} & \cdots & r_{25} \\
    \vdots & \vdots & \ddots & \vdots \\
    r_{51} & r_{52} & \cdots & r_{55}
\end{bmatrix} \]  \hspace{1cm} (4.28)

where

\[ r_{1,1} = r_{\text{very small, very small}}; \quad r_{1,2} = r_{\text{very small, small}}; \]
\[ r_{1,3} = r_{\text{very small, medium}}; \quad r_{1,4} = r_{\text{very small, large}}; \]
\[ r_{1,5} = r_{\text{very small, very large}}; \quad r_{2,1} = r_{\text{small, very small}}; \]
\[ r_{2,2} = r_{\text{small, small}}; \quad r_{2,2} = r_{\text{small, medium}}; \quad \cdots \text{and so on.} \]

Each element in the relation matrix is an antecedent. For example, \( r_{1,2} \) is the linguistic antecedent of "if \( x \) is very small, and \( y \) is small" and is a node in the fuzzy logic layer. Notice that the logic \textit{and} is a general logic sign. It could be \textit{and} or \textit{or}. The corresponding operators are defined by T-norm and T-conorms (see Appendix A), a pair of non-linear operators. The objective is to find the weight matrix \( W \) which

118
minimizes

\[ J_W = \frac{1}{2} \sum_{n=1}^{N} c_k^2, \]  

(4.29)

\( W_R \) has 25 elements and is in the form of

\[
W = \begin{bmatrix}
w_{11} & w_{12} & \cdots & w_{15} \\
w_{21} & w_{22} & \cdots & w_{25} \\
. & . & \cdots & . \\
. & . & \cdots & . \\
w_{51} & w_{52} & \cdots & w_{55}
\end{bmatrix}
\]  

(4.30)

Using the recursive learning discussed, \( W \) is converged into

\[
W = \begin{bmatrix}
-2.3 & 5.5 & -35.9 & 18.4 & 59.7 \\
5.6 & -15.3 & 113.7 & 174.6 & 182.4 \\
-36.0 & 113.6 & 198.4 & 217.7 & 232.7 \\
18.9 & 175 & 218.6 & 246.5 & 258.9 \\
57.6 & 180.2 & 229.4 & 254.6 & 265.2
\end{bmatrix}
\]  

(4.31)

Figure (4-11) and Figure (4-12) show the resulting and the original control surface.
Figure 4-11: Resulted Control Surface

Figure 4-12: Original Control Surface
4.4 Adaptation for a Human-in-the-Loop System

In the previous sections, we have discussed recursive learning in a fuzzy rule network (FRN) by using teaching data. In order to result in a good fuzzy rule network, selecting good teaching data is necessary. In this section, we are going to show how an ADAUS system can help a decision-maker to select good teaching data.

As discussed in Chapter 2, the ADAUS has a module that displays two satisfaction curves to the human decision-maker. One curve is the degree of goal satisfaction and the other is the degree of constraint satisfaction. The overall degree of satisfaction is the minimum of these two, and therefore the optimal operational point is the intersection of these two curves (the intersection of the solid lines in Figure 4-13).

![Diagram showing the intersection of the solid lines representing degree of goal satisfaction and degree of constraint satisfaction.]

Figure 4-13: Discrepancy Between Human and the ADAUS

If the ADAUS system is fully calibrated, the decision-maker will agree with the optimal operational point as the ADAUS suggests (the overall degree of satisfaction is 0.5 in this case). Otherwise the decision-maker would select another operational
point $z^*$ where the overall degree of satisfaction may be reduced (to 0.43 in this example). The discrepancy between the ADAUS's optimal and the decision-maker's selection indicates that the ADAUS has not fully calibrated yet. It is desirable to modify the ADAUS in such a way that the satisfaction curves shift to a new optimal operational point that matches the human decision-maker's selection, as shown in Figure (4-13) with the twisted line. Using sensitivity analysis, we can find the appropriate parameter vector (input to the FMP module) that will move the satisfaction curves accordingly. Here we assume that the structure of the FMP module has been correctly set up but the parameters have not fully calibrated.

The current measurements and the resulting parameter vector can then be used as the teaching data to calibrate the linguistic knowledge base, i.e., fuzzy rule network (FRN). The whole process of the adaptation for a human-in-the-loop system is shown in Figure (4-14).

When a discrepancy $d$ is detected in the satisfaction analysis module and the decision-maker wants the ADAUS to learn something from this discrepancy, s/he turns on a learning switch to start a calibration procedure that consists of two steps:

1. **Find the teaching pattern.** This is done in the FMP-Satisfaction-Sensitivity loop. An appropriate parameter vector $P^k$ is found.

2. **Calibrate the weight matrix in the fuzzy net work.** This is done by using the teaching parameters $R^k$ and $P^k$ as discussed in the previous sections.
Figure 4-14: Adaptation for a Human-in-the-Loop System

4.5 Summary

In contrast to classical mathematics and classical logic which demand precise and quantitative data, a human expert possesses the ability to reach reasonably good decisions based on imprecise, qualitative data. Researchers as well as industry applications have shown that a linguistic rule base could generate a design that achieves satisfactory results much faster than the classical approach. Since it is found that experts often refuse to give precise numerical estimates of outcomes, and that excessive precision and clarity may be as ineffective as excessive vagueness, a fuzzy rule base is a good candidate to represent these fuzzy rules.

A membership function in fuzzy set theory can be used to overcome linguistic
inexactness. The shapes as well as the support of a membership function are critical to a linguistic rule. To maximize the efficiency of knowledge elicitation from human experts, a Random-Sampling-Without-Replacement technique is proposed. An example of Driving-Speed-on-the-Highway is discussed and the result shows that it agrees well with a human driver's expertise. Based on the fuzzy relation and fuzzy defuzzification, this chapter implements a network-type fuzzy knowledge structure with a weight matrix subject to calibration from teaching data. Each element in the weight matrix has its physical meaning in the rules and can be used as an indication of convergence. The adaptation mechanism for a human-in-the-loop is also discussed. This adaptive approach reduces design time, and improves performance, accuracy and reliability at lower cost.
Chapter 5

Decision Aid for Freight Train Dispatching

Freight train dispatching can be thought of as a network consisting of yards which send freight cars to other yards in the network. Certain pairs of yards are connected by lines of track. Trains move on the lines among yards and each train consists of a number of cars. In railroad management decision-making processes, operation cost and average trip time are two essential considerations for performance evaluation. Operation costs include train cost, car time cost, car handling cost and total logistics cost; while average trip time is characterized by car mean trip time and trip time reliability.

The objective of the decision making in freight train dispatching is to minimize operation cost and time delay. This objective is achieved by railroad operational planning which consists of two levels: operational planning and tactical planning.
Operational planning includes train scheduling, car routing/scheduling, locomotive scheduling, crew scheduling and empty car scheduling. Tactical planning involves train routing, blocking, makeup and aggregate trip routing. Research shows that yard delays are key factors in average trip time and reliability. Dispatching policy controls yard operations and hence controls yard delays.

Among the varieties of operation costs we will consider only the cost associated directly with operating a train, which involves a fixed cost and a variable cost. The fixed cost consists mostly of crew labor and a single unit of motive power (locomotive) independent of the number of cars attached to the train. The variable cost consists of fuel and additional units of motive power as the number of cars in the train increases.

The delivery time required to move rail cars from the origin to the destination is the sum of the following four time components: 1) waiting time for departure in the origin yard; 2) classifying time for connections in intermediate yards; 3) moving time on lines; and 4) waiting time for delivery to customers in the destination yard.

The fixed cost in rail freight transportation is sizable (about $9/mile verses $0.35/mile for truck transportation). Because of the level of fixed train costs and the limitation of railroad resources such as motive power, it is generally not economical to provide all pairs of yards with direct train connections. Many cars have to be reclassified to other trains in intermediate yards. This process causes a considerable yard delay (often around 24 hours) and is the most important factor affecting car delivery time.

The decision-making environment in railroad operations has two sets of information as described in Chapter 1. One set of information is the physical rail track
network and the published schedules; and another set of information is the expertise from rail managers. As discussed in Chapter 2, for a complex system like railroads, the overall operational performance will benefit from a computer decision aid.

In rail operations, the decision variables which cannot be crisply defined can be described as fuzzy variables, such as human expertise that is imprecise and vague in nature but plays an important role in railroad operation. Take train length for an example. Railroad managers have a powerful incentive to operate long trains, because of the sizable fixed cost involved in operating a train. The term long is an imprecise linguistic word. A manager would undoubtedly agree that it is physically impossible to move a 300-car-train with a single locomotive, but there is no precise dividing line between feasible and infeasible train length. If a 120-car-train can be moved over a given section of line, a 121-car-train almost certainly can be moved also. In resource-constrained decision problems, these imprecise variables produce soft feasible regions – or soft boundaries. Some other soft variables are weather, seasons and train priority, et cetera.

In the previous chapters, we have proposed and studied a general framework (ADAUS) to deal with a decision-making environment having both soft-boundaries and linguistic rules. In this chapter, we will investigate its application in rail freight train dispatching.
5.1 Yard Operations

Classification is one of the key tasks operated in a yard. The costs, delays and workloads of a yard depend on: 1) the number of trains received, dismantled and sorted; 2) the number of cars and groups of cars handled; and 3) the number of blocks and trains formed for travel to other yards in the network.

5.1.1 Yard Delays

Yard delay time is one measure of a yard performance. It is sensitive to traffic intensity, to specific characteristics of a given yard and to operating policies. Car time in yards is spent in: 1) classification and makeup operations; 2) waiting for the departure of the appropriate outbound train after classification has been completed; and 3) congestion-induced yard delays.

The classification process sorts the cars of incoming trains onto groups of parallel yard tracks according to their next downstream destinations, and this may take several hours. The makeup operation forms outbound trains by consolidating groups of cars into a single series (a train) and this takes several more hours.

The waiting time for departure is typically the major delay in intermediate yards. A car might have spent only 17 hours moving with a train that traveled at 30 mph for 500 miles, but the total delivery time could easily be up to six days or even longer. Each inbound train will be connected with a number of outbound trains. Some inbound-outbound train pairs are scheduled close each other, while others may
have almost twenty-four hours between arriving time and departure time.

Petersen [65] presents a classical description of classification yard configurations, operations and performance measures, as well as a series of models for the main yard operations. Several other authors have built upon this work. Assad [1] and Crainic [21], among others, present reviews of this subject. These delay models are network models of either the movements of, or the sequence of, operations performed on the engines and cars passing through the yard.

Actual arriving and departure times as well as traffic volumes will vary considerably on a day-to-day basis. As a result, trains might depart so close together that cars can frequently miss their connections and be forced to wait in the yard for next available departures. This reduces the terminal reliability and increases the delivery time. Dispatching policy therefore is very important to make good use of railroad resources (locomotives, tracks and yards, etc.) to reduce yard delays.

5.1.2 Dispatching Policies

A freight train timetable is scheduled at the operational planning level (that could be another research subject and will not be discussed here) and advertised to potential customers. These scheduled trains are regarded as the backbone of a railroad’s service. The connections at each yard are closely watched as part of dispatching policies.

In the timetable, some trains may stop at every yard along the line to pick up all the traffic, while other trains will skip major stops and accept only high-rated traffic at
their connections in order to reach principal destinations at the desired time. Traffic
tends to be concentrated at certain times of the day or in different seasons or during a
certain kind of weather. Extra trains will be dispatched in the case of traffic overflow
and non-economical trains will be canceled if there is not enough traffic.

In contrast to a passenger train whose first priority is on schedule, a freight train
(in the United States) has a one to one-a-half-hour average delay relative to the sched-
ule. A yardmaster, whose job is to move blocks of cars quickly and efficiently using
well-organized classification and dispatching processes, is the key person responsible
for reducing such delays. The yardmaster oversees all the incoming and outgoing
traffic, anticipates how long the makeup and departure of the outbound train will
take, manages his crew, power and caboose resources, and makes decisions on when
and how to makeup cars and when to dispatch trains accordingly.

There are many tradeoffs in this kind of decision-making. One of them is the
tradeoffs between the accumulated delays and trainload economics. The decision-
making reflecting these tradeoffs involves determining dispatching policies based on
objective functions and resource constraints. Unfortunately, the nature of trainload
economics has never been made very explicit. Instead, yardmasters acquire their
expertise on-the-job.

A simple example of a commonly used dispatching rule is described as follows:
Train No. 23 will leave if –

At 8:00 AM at least 70 cars are waiting in the yard, or
At 9:00 AM at least 60 cars are waiting in the yard, or
At 10:00 AM at least 50 cars are waiting in the yard. (5.1)

... The train may be canceled altogether if no more traffic
for this outbound train is expected to arrive within a reasonable
time and the available cars in the yard are considered uneconomical to run; or, if too many cars have been accumulated in the
yard, additional trains will be added to alleviate traffic buildups.

These decisions are made in the local yard, but they will affect the operations of
the downstream yards and influence the overall performance of the rail network.

**Straight-line Dispatching Policy**

The rule described by yardmasters in Equation (5.1) was suggested by Beckman, et
al. as a straight-line dispatching policy [9], which is a function between the available
traffic in the yard for an outbound train and the time past the schedule.

Figure (5-1) shows a typical straight-line policy. The abscissa is the time past the
scheduled makeup of an outbound train, and the ordinate is the train length. The
policy line is characterized by a maximum train length, a minimum train length and
a maximum holding time. The maximum train length, limited by engine horsepower
and track conditions, is the maximum number of cars that a train can take. The

131
Figure 5-1: Straight-line Dispatching Policy

minimum train length is the least number of cars with which a train is considered economically efficient to run. No train will be dispatched if the number of available cars in the yard is below this minimum train length. The maximum holding time is the time window for dispatching this train. If there is not enough traffic in the yard within the time window, this scheduled outbound train will be canceled. Otherwise, when the accumulated traffic reaches the policy line and the time is within the time window, the train will be made up.

The major benefits for holding up an outbound train are: 1) those cars which would have missed their scheduled outbound connection, may yet make it; and 2) the outbound train is assembled to a normal (longer) length and hence is more economical, and will not cause cancellations in the downstream yards. The major negative effects of holding up an outbound train are: 1) the train tends to arrive late at the next yard, and this could cause disconnection and even cancellation; and 2) the shipments will be delayed.
The cancellation of an outbound train has the same effect as holding it up, except the consequences may be more severe. The main benefits of cancellations are the dollar savings by not running uneconomical trains.

An extreme policy, called a *schedule-adherence* policy, is one with a vertical line at the scheduled time. Under this policy, makeup for outbound trains starts at the scheduled time. The general policy of ConRail, a large railroad company in the United States, is moving in this direction as a result of some high priority and highly profitable contracts. Another extreme policy called a *tonnage-train policy* is one with a horizontal line at maximum train length. Under this policy, makeup begins only after enough traffic has built up.

The straight-line policy has been used as a tool to analyze terminal delay and reliability issues. Folk [34] investigated the straight-line policy by simulating traffic moving across a simple network. Martland and Kwon [60] further enriched the network to study reliability versus different straight-line dispatching policies.

**Proposed Dispatching Policy with Soft Boundaries**

Numerous operational and tactical models of rail transportation have been presented to analyze railroad operations so that operation costs and yard delays can be reduced [1] [24] [32]. All of these models are based on crisp variables. None of them has addressed the issues of soft variables and human imprecise knowledge explicitly. Koskosidis and Powell [48] recently presented a formalism of vehicle routing and scheduling using the ideas of soft time window constraints.
In practice, the imprecise information, the soft constraints and the nature of inherent approximate reasoning in decision-making prevent us from talking about *optimal* solutions. An "optimal" solution, found in the models with crisp parameters and constraints, may not be necessarily actually *optimal*, and satisfactory in practice. In the train length example, we assume the maximum train length in a decision model is crisply defined as 120 cars, and there are 121 cars available in the yard headed for the same destination. The next scheduled time is far off and no more incoming cars are foreseen. The *optimal* solution from a crisp model would be to dispatch only 120 cars and leave the extra one in the yard, possibly for another 24 hours. In practice, a yardmaster will probably have a 121-car train dispatched to reduce yard delays.

To overcome such shortcomings in crisp models, we propose a dispatching policy with soft boundaries as shown in Figure (5-2). Here we use the idea of *satisfaction* (Chapter 2) instead of *optimization*. In this soft dispatching policy, the maximum train length is no longer a crisp line as in Figure (5-1). Instead, it is a grey-layered rectangular area characterized by a fuzzy variable with a lower and upper bound of $L_{lower}$ and $L_{upper}$. The corresponding membership (see Chapter 2) function is:

$$
\mu_{L_{max}}(L) = \begin{cases} 
1.0 & \text{if } L \leq L_{lower} \\
1.0 - \frac{L - L_{lower}}{L_{upper} - L_{lower}} & \text{if } L_{max} \in [L_{lower}, L_{upper}] \\
0.0 & \text{otherwise}
\end{cases}
$$  \hspace{1cm} (5.2)

which means:

1. when the train length is below $L_{lower}$, the constraint is fully satisfied with a membership of 1.0;
Figure 5-2: Dispatching Policy with Soft Boundaries

2. When the train length falls between $L_{lower}$ and $L_{upper}$, the constraint is partially satisfied with memberships from 1.0 to 0.0;

3. When the train length is above $L_{upper}$, the constraint is totally dissatisfied with a membership of 0.0, which means an infeasible solution.

The second case is of particular interest, because it provides rooms for making trade-offs between the constraint satisfaction and the objective satisfaction in a structured way.

Furthermore, a policy line in the real world is not only a function of time, but also a function of some other qualitative variables such as the day of the week, the season.
and the weather, etc. Hence, the policy line can be described by an irregular line surrounded by a soft area representing the effects of multiple qualitative dimensions. The relation between the policy line and the multiple qualitative dimensions is usually only available in terms of linguistic rules and hence is mathematically unmodelable (in traditional mathematics). A linguistic rule base, which has been developed into a fuzzy rule network (proposed in Chapter 4), will be used to model such relations.

5.2 Application of the ADAUS in Freight Train Dispatching

With the considerations of soft boundaries and qualitative reasoning in the proposed soft dispatching line, the model is more realistic but also more complex. The complexity requires us to seek new methodologies for problem-solving.

In Chapter 2, we proposed a general framework – ADAUS (Adaptive Decision Aiding Under Soft Boundaries) technique to deal simultaneously with soft data (which include imprecise variables and linguistic variables) and hard data (which are crisp variables). We will use the ADAUS technology to model rail freight train dispatching.

In freight train dispatching application, hard data are the physical railroad network and the published schedules while soft data are countable variables such as the maximum allowable length of a train, the forecast number of incoming cars, and linguistic (uncountable) variables such as the season and the weather. Figure (5-3) shows the block diagram of the ADAUS application for rail freight train dispatching.
Figure 5-3: ADAUS Application for Freight Train Dispatching

There are three blocks or modules in the diagram. The first module is a linguistic rule base, in which a rule base takes linguistic variables (weather, season, priority and time past schedule) as inputs and gives out a fuzzy variable (policy line) as an output. Some other factors which affect the policy line may be put in another rule base when further considerations are needed. The second module is fuzzy linear programming (FLP) which is a linear version of fuzzy mathematical programming (FMP). FLP takes in the crisp data, such as the physical railroad network and the published schedules, and soft variables, such as the estimated number of incoming cars and the maximum train length. The third module is a pair of complete solution curves (see Chapter 3) from the FLP module and serves as a satisfaction analysis, through which a human operator can select an appropriate working point to maximize
the satisfaction level of the constraints and the objectives at the same time.

The policy line from the linguistic rule base module works interactively between the linguistic rule base module and the fuzzy linear programming module, and makes them cooperate with each other. The *FLP* guarantees that the problem-solving has a solid mathematical structure with certain flexibility, while the *LKB* provides a means to bring in linguistic variables from human expertise for an overall consideration.

In the fuzzy linear programming module, the objective has an aspiration level described by a fuzzy variable whose membership represents degree of goal satisfaction; and the constraints have soft boundaries to allow for a certain degree of constraint violation which is an indication of negative constraint satisfaction. The smaller the degree of constraint satisfaction, the larger the feasible regions (see Chapter 3) and therefore the larger the degree of goal satisfaction. These two satisfaction degrees are displayed (shown in the satisfaction analysis module) to the human operator. After the human determines a comfortably satisfactory level, the system suggests the final crisp decision to the human operator. The difference between the *optimal* solution from the *FLP* module and the human selected *comfortably satisfactory level* is used to calibrate the rules in the linguistic knowledge base module as discussed in Chapter 4.

In this way, the proposed general decision-making framework synthesizes numerous *hard* and *soft* data into two simple risk-return-like curves, which should greatly reduce the human operator's workload and improve overall system performance. The three modules: the linguistic rule base, the fuzzy linear programming and satisfaction analysis, will be discussed separately in the following three sections.
5.2.1 Linguistic Rule Base Module

There are many linguistic variables associated with dispatching policy lines: a company's general policy, the weather, the season, the days of week, the reliability of the motive power, the train length, the train priority, et cetera. We will show how these linguistic variables are represented by fuzzy variables.

General Policy

General policy is the highest operational level for decision making as to whether the company's dispatching policy should be of schedule-adherence (first concern is on schedule), tonnage-train (first concern is the running cost) or something between (trade-offs).

It can be represented by a fuzzy variable $\tilde{G}_{policy}$ with a policy reference scale and three linguistic terms: Schedule-adherence, Between, and Tonnage-train as shown in Figure (5-4) or in Table (5.1). The policy reference scale is normalized to 10. A scale of ten means a fully Tonnage-train policy while zero means a fully schedule-adherence policy.

<table>
<thead>
<tr>
<th>Terms</th>
<th>General Policy Preference Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0</td>
</tr>
<tr>
<td>Schedule-Adherence</td>
<td>1.0</td>
</tr>
<tr>
<td>Between</td>
<td>0.0</td>
</tr>
<tr>
<td>Tonnage-Train</td>
<td>0.0</td>
</tr>
</tbody>
</table>

*Table 5.1: Fuzzy Variable $\tilde{G}_{policy}$ in Table Form*
Figure 5-4: Fuzzy Variable $\tilde{G}_{policy}$

An example of the fuzzy variable $\tilde{G}_{policy}$ shown by $x$ in Figure (5-4) is represented in the form of

$$\tilde{G}_{policy}(x) = \{0.0, 0.4, 0.6\}, \quad (5.3)$$

which means the policy is in favor of Tonnage-train policy (see Figure 5-4).

Weather and Season

Similar to the fuzzy variable $\tilde{G}_{policy}$, the fuzzy variables $\tilde{W}_{weather}$ and $\tilde{S}_{season}$ also have three linguistic terms. They are defined as:

$$\tilde{W}_{weather} \rightarrow \{good, so-so, bad\}$$

and

$$\tilde{S}_{season} \rightarrow \{busy, medium, notbusy\}.$$
Time Past Scheduled Makeup

The fuzzy variable of time past schedule $\tilde{T}_{past}$ has six linguistic terms: Very short, Short, Medium, Long, Very long and Extra long. Its graphic form and table form is in Figure (5-5) and Table (5.2) respectively. Time past schedule in this variable is normalized to 0–12.5, with 0 corresponding to the scheduled time, and 12.5 to the allowed maximum holding time.

![Figure 5-5: Fuzzy Variable $\tilde{T}_{past}$](image)

<table>
<thead>
<tr>
<th>Terms</th>
<th>Time Past Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0</td>
</tr>
<tr>
<td>Very Short</td>
<td>1.0</td>
</tr>
<tr>
<td>Short</td>
<td>0.0</td>
</tr>
<tr>
<td>Medium</td>
<td>0.0</td>
</tr>
<tr>
<td>Long</td>
<td>0.0</td>
</tr>
<tr>
<td>Very Long</td>
<td>0.0</td>
</tr>
<tr>
<td>Extra Long</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 5.2: Fuzzy Variable $\tilde{T}_{past}$ in Table Form
An example of $\tilde{T}_{past}$ shown by $x$ in Figure (5-5) is:

$$\tilde{T}_{past}(x) = \{0.0, 0.0, 0.5, 0.5, 0.0, 0.0\}, \quad (5.4)$$

which means the current time is between medium and long.

**Policy Train Length at Policy Line**

The policy train length $\tilde{L}$ is the minimum train length required to dispatch a train economically. It controls the dispatching policy by giving the inbound traffic and time past the scheduled makeup; and is usually a decreasing function with the variable of time past the schedule. $\tilde{L}$ is discretized as

$$\tilde{L} = \{\tilde{L}_1, \tilde{L}_2, \ldots, \tilde{L}_i, \ldots, \tilde{L}_N\}.$$  

The fuzzy variable $\tilde{L}_i$ has six linguistic terms: Very short, Short, Medium, Long, Very long and Extra long as shown in Figure (5-6) or Table (5.3) respectively. The policy train length is normalized to 0 – 125, with 0 corresponding to the predefined shortest train and 125 corresponding to the longest train allowed. An example of the fuzzy variable $\tilde{L}_i$ shown by $x$ in Figure (5-6) is:

$$\tilde{L}_i(x) = \{0.0, 0.0, 0.0, 0.44, 0.56, 0.0\}. \quad (5.5)$$
Figure 5-6: Length Fuzzy Variable $\tilde{L}_i$

<table>
<thead>
<tr>
<th>Terms</th>
<th>Minimum Train Length (in number of cars)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Very Short</td>
<td>1.0</td>
</tr>
<tr>
<td>Short</td>
<td>0.0</td>
</tr>
<tr>
<td>Medium</td>
<td>0.0</td>
</tr>
<tr>
<td>Long</td>
<td>0.0</td>
</tr>
<tr>
<td>Very Long</td>
<td>0.0</td>
</tr>
<tr>
<td>Extra Long</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 5.3: Fuzzy Variable $\tilde{L}_{min}$ in Table Form
Linguistic Rule Acquisition and Representation

This research, aimed at finding new technologies for railroad operation, will consider only the following five linguistic variables for simplicity:

1. Weather (input);
2. Season (input);
3. Train priority (input);
4. Time past the schedule (input);
5. Policy train length (output).

The fuzzy knowledge base is thereby a four-input, one-output control system as shown in Figure (5-7). The relation between the inputs and the output is based on the expertise of human railroad managers in terms of linguistic rules.
The linguistic rules for dispatching are described in the form of

$$\begin{align*}
IF & \quad \text{weather IS good AND season IS busy} \\
& \quad \text{AND Priority IS high AND time past IS short} \\
THEN & \quad \text{policy train length IS long.}
\end{align*}$$

(5.6)

**Figure 5-8: Fuzzy Knowledge Acquisition Software**

A computer software program has been designed for the rule acquisition from human rail managers. Its screen plot is shown in Figure (5-8). The software prompts subjects with combinations of the four input variables in a RSWOR manner (Chapter 4). The subjects are then asked to pick their decisions for the policy on the train length from the five linguistic terms on the left-side of the screen with the option to assign weights (greater than the default which is zero) from above or below. For
instance, in Figure (5-8), the corresponding linguistic rule is read as:

\[
\begin{align*}
IF & \quad \text{weather IS good} \quad \text{AND} \quad \text{season IS busy} \\
& \quad \text{AND} \quad \text{Priority IS very high} \quad \text{AND} \quad \text{time past IS medium} \\
\text{THEN} & \quad \text{policy train length IS long with 80\% confidence,} \\
& \quad \text{and medium with 20\% confidence.}
\end{align*}
\] (5.7)

As discussed in Chapter 4, this linguistic knowledge base can be represented in a fuzzy rule network (FRN) form as shown in Figure (5-9) with the capability of adaptive learning.

![FRN of the Dispatching Model](image)

**Figure 5-9**: FRN of the Dispatching Model
5.2.2 Fuzzy Linear Programming Module

The fuzzy linear programming module having a soft dispatching policy line is shown in Figure (5-10). The time interval \([t_1, t_N]\) is a discrete dispatching window. Time \(t_1\) is the scheduled time and \(t_N\) is the maximum holding time. A train will be dispatched at \(t_i\) when the number of cars in the yard exceeds the dispatching line. The dispatching line is represented by discrete points, determined from a linguistic rule base (discussed in the previous section). The maximum train length \(\bar{L}_{\text{max}}\) is a soft variable determined by the horsepower of a locomotive as well as line conditions. \(\bar{A}_i\), another soft variable, is the forecast number of cars arriving between time interval \(t_{i-1}\) and \(t_i\). \(S_i\) is the number of cars left in the yard after time \(t_i\), and \(y_i\) is a boolean decision variable (a train is dispatched at time \(t_i\) when \(y_i = 1\)). \(F_i\) is the extra number of cars above the dispatch line. The actual train length of a dispatched train at time \(t_i\) is thereby \(P_i = L_i + F_i\). The soft variables \(\bar{L}_{\text{max}}\) and \(\bar{A}_i\) are described by two fuzzy variables, both defined in trapezoidal form

\[
\mu_{A_i}(x) = \begin{cases} 
0.0 & \text{if } x \leq A_i^l - \delta_{A_i} \\
1.0 - \frac{A_i^l - x}{\delta_{A_i}} & \text{if } x \in [A_i^l - \delta_{A_i}, A_i^l] \\
1.0 & \text{if } x \in [A_i^l, A_i^u] \\
1.0 - \frac{x - A_i^u}{\gamma_{A_i}} & \text{if } x \in [A_i^u, A_i^u + \gamma_{A_i}] \\
0.0 & \text{if } x \geq A_i^u + \gamma_{A_i} 
\end{cases} \tag{5.8}
\]

and

\[
\mu_{L_{\text{max}}}(x) = \begin{cases} 
1.0 & \text{if } x \leq L_i^l \\
1.0 - \frac{x - L_i^l}{\delta_{L_i}} & \text{if } x \in [L_i^l, L_i^l + \gamma_{L_i}] \\
0.0 & \text{if } x \geq L_i^l + \gamma_{L_i} 
\end{cases} \tag{5.9}
\]

where \(A_i^l\) and \(A_i^u\) are lower and upper bounds (inside the bounds the memberships
Figure 5-10: Dispatching Model of One Freight Train

taking full value of 1.0). \( \delta_{Ai} \) and \( \gamma_{Ai} \) bracket the intervals of confidence for the left and right sides of \( \bar{A}_i \) and \( \bar{A}_i \), respectively. Similarly, \( L^r_i \) and \( \gamma_{L1} \) are the upper bound and the right interval of confidence for \( \bar{L}_{\text{max}} \).

The soft variables \( \bar{A}_i \) and \( \bar{L}_{\text{max}} \) in trapezoidal form are symbolically written as

\[
\bar{A}_i = (\delta_{Ai}, A^l_i, A^r_i, \gamma_{Ai}) \\
\bar{L}_{\text{max}} = (0, 0, L^r_i, \gamma_{L1})
\]  

(5.10)

The simple interpretation of the membership function \( \bar{A}_i \) is that between time \( t_{i-1} \) and \( t_i \), the number of forecast incoming cars is very likely to be between \( A^l_i \) and \( A^r_i \) with linear decreasing possibility back to \( A^l_i - \delta_{Ai} \) and up to \( A^r_i + \gamma_{Ai} \). The
interpretation of the membership function $\tilde{L}_{\text{max}}$ is that the locomotive can certainly take $L_i$ cars, but could still move up to $L_i' + \gamma L_1$ cars.

The fuzzy linear program for dispatching one scheduled train is the solution of the following equation:

$$\text{Min } \text{Cost} = \sum_{i=1}^{N} (C_{si} \cdot S_i + C_{pi} \cdot P_i + C_{yi} \cdot y_i)$$

subject to the following constraints:

1) $F_i + S_i + L_i \cdot y_i - S_{i-1} = A_i$

2) $F_i - M_{inf} \cdot y_i \leq 0$ \hspace{1cm} (5.11)

3) $F_i + L_i \cdot y_i \leq \tilde{L}_{\text{max}}$

4) $\sum_{i=1}^{N} y_i \leq 1$

5) $y_i = 0, 1$

6) $F_i, S_i \geq 0 \quad i = 1, 2, \ldots N$

where

1. $C_{si}$ is the variable cost of one car left over in the yard between time $t_{i-1}$ and $t_i$.

2. $C_{pi}$ is the variable cost for one car to move from one yard to another.

3. $C_{yi}$ is the fixed cost for one train to move from one yard to another.

4. $M_{inf}$ is a positive infinitive number (a large number in the computer).

The first constraint of Equation (5.11) is a car flow balance equation after time $t_i$. The second constraint implies a fact: when $y_i = 0$ (no train dispatched), $F_i = 0$. The third constraint ensures that the resulting train length will not exceed the capacity
of a locomotive in the sense of its soft boundary. The fourth constraint says that at most only one train could be dispatched in the time interval. Zero summation value of $y_i$ means train cancellation.

The objective function (goal) $C^\text{cost}$ is a function of $P_i, S_i$, and $y_i$ which in turn are functions of $\bar{L}_{\max}$ and $\bar{A}_i$. $C^\text{cost}$ is a fuzzy variable having a membership function:

$$
\mu_G(\text{cost}) = \begin{cases} 
1.0 & \text{if cost } \leq c_{\min} \\
g(\text{cost}) & \text{if cost } \in [c_{\min}, c_{\max}] \\
0.0 & \text{if cost } \geq c_{\max}
\end{cases}
$$

where $c_{\min}$ is an aspiration level, and $c_{\max}$ is the lowest acceptable level. It will be fully satisfied with a solution $x = x(F_i, S_i, y_i)$ for which cost takes a value smaller than $c_{\min}$. It will be partially satisfied to the degree $0 < g(x) < 1$ with a solution $x$ for which cost falls between $c_{\max}$ and $c_{\min}$. It will be fully dissatisfied with a solution $x$ for which cost takes the value larger than $c_{\max}$. Figure (5-11) is the graphic form of $\mu_G(\text{cost})$, where $g(\text{cost})$ is determined by the human yardmaster based on daily operations and can be checked by the Autogoal software (Chapter 3).

![Figure 5-11: Degree of Satisfaction of Goal $\mu_G$](image)

150
Parametric Approach of Solving FLP

The parametric approach (see Chapter 3) obtains not only one optimal solution, but the complete set of solutions. Since it is based on the $\alpha$-cut of fuzzy set theory¹, for every satisfaction level (parameter) $\alpha \in [0, 1]$, a fuzzy subset changes its confidence interval. Figure (??) is an example of parameterization of fuzzy subset $\tilde{A}_i$.

![Figure 5-12: Parameterization of Fuzzy Subset $\tilde{A}_i$](image)

For every $\alpha$, constraint (1) and (3) in Equation (5.11) can be rewritten as²:

\[
F_i + S_i + L_i \cdot y_i - S_{i-1} \in [A_i^l - \delta_{A_i} \cdot (1 - \alpha), A_i^r + \gamma_{A_i} \cdot (1 - \alpha)]
\]

\[
F_i + L_i \cdot y_i \in [0, L_i^r + \gamma_{L_i} \cdot (1 - \alpha)]
\]  

(5.13)

With $\alpha \in [0, 1]$, the solution of Equation (5.11) minimizes the cost function as analytically dependent on the parameter $\alpha$. That is, for every $\alpha$, there is a solution that satisfies $\mu G$ with a degree of $\mu G = \alpha$ and simultaneously attains the goal satisfaction $\mu G$ with the possible highest degree. The minimum value of the cost function can then be represented as analytically dependent on the parameter $\alpha$, and is a continuous, piece-wise linear and concave function on $\alpha$.

¹A confidence interval (of a fuzzy subset) determined as a confidence base by cutting the membership function horizontally with a given value $\mu$.
²$\alpha = 1 - \theta$ or $\theta = 1 - \alpha$. 

151
The solution is in the form of
\[ x(\alpha) = \{F_i(\alpha), S_i(\alpha), y_i(\alpha)\} \]

with
\[ \mu_C(x(\alpha)), \mu_G(x(\alpha)) \in [0, 1] \]

5.3 Simulation and Experiment

5.3.1 One-Train Example

Assume that the scheduled time for a train is 12:00 (noon), the maximum holding
time is 16:00 (4:00 pm) and the number \(N\) of points determining the policy line is set
to five. Hence,

\[ T = \{t_1, t_2, t_3, t_4, t_5, \} = \{12:00, 13:00, 14:00, 15:00, 16:00\}. \]

For simplicity in this example, the policy line is assumed to be linear

\[ L = \{L_1, L_2, L_3, L_4, L_5, \} = \{120, 95, 70, 45, 20\}. \]
The fuzzy variables $\tilde{L}_{max}$ and $\tilde{A}_i$ ($i = 1, 2, \ldots, 5$) are in trapezoidal form:

\begin{align*}
\tilde{L}_{max} &= (0, 0, 120, 10) \\
\tilde{A}_1 &= (0, 100, 100, 0) \\
\tilde{A}_2 &= (6, 28, 28, 6) \\
\tilde{A}_3 &= (2, 8, 8, 2) \\
\tilde{A}_4 &= (8, 35, 35, 8) \\
\tilde{A}_5 &= (9, 29, 29, 9)
\end{align*} \hfill (5.14)

The costs are assumed: $C_{si} = 10$, $C_{pi} = 1$, $C_{yi} = 1000$. 

153
The resulting FLP equations are:

\[
\min \text{Cost} = \sum_{i=1}^{5} (10 \cdot S_i + P_i + 1000 \cdot y_i)
\]

subject to:

\[
\begin{align*}
F_1 + S_1 + L_1 \cdot y_1 - S_0 &= (0, 100, 100, 0) \\
F_1 - M_{inf} \cdot y_1 &\leq 0 \\
F_1 + L_1 \cdot y_1 &\leq (0, 0, 120, 10) \\
F_2 + S_2 + L_2 \cdot y_2 - S_1 &= (6, 28, 28, 6) \\
F_2 - M_{inf} \cdot y_2 &\leq 0 \\
F_2 + L_2 \cdot y_2 &\leq (0, 0, 120, 10) \\
F_3 + S_3 + L_3 \cdot y_3 - S_2 &= (2, 8, 8, 2) \\
F_3 - M_{inf} \cdot y_3 &\leq 0 \\
F_3 + L_3 \cdot y_3 &\leq (0, 0, 120, 10) \\
F_4 + S_4 + L_4 \cdot y_4 - S_3 &= (8, 35, 35, 8) \\
F_4 - M_{inf} \cdot y_4 &\leq 0 \\
F_4 + L_4 \cdot y_4 &\leq (0, 0, 120, 10) \\
F_5 + S_5 + L_5 \cdot y_5 - S_4 &= (9, 29, 29, 9) \\
F_5 - M_{inf} \cdot y_5 &\leq 0 \\
F_5 + L_5 \cdot y_5 &\leq (0, 0, 120, 10) \tag{5.15} \\
\sum_{i=1}^{5} y_i &\leq 1 \\
y_i &= 0, 1 \\
F_i, S_i &\geq 0 \quad i = 1, 2, \ldots, 5
\end{align*}
\]

Figure (5-13) shows the solution of Equation (5.15) with parameter \(\alpha\), where the curve \(\mu_C\) is related to the degree of satisfaction of all constraints. The larger \(\alpha\), the smaller the undesired possible violation of constraints. The curve \(\mu_C\) is related to the degree of goal satisfaction; the larger \(\alpha\), the smaller the degree of satisfaction of the goal.
Using this technique, the human managers can obtain an exact (non-fuzzy) solution vector \( x \) in terms of degree of satisfaction of the goal and degree of satisfaction of constraints for any given parameter \( \alpha \). The intersection point \( (\alpha = \alpha^*) \) of the curves in Figure (5-13) is of particular interest. The corresponding solution \( x(\alpha^*) \) maximizes the overall degree of satisfaction \( \mu_D \).

![Graph showing DSOC and DSOG curves intersecting at point \( \mu^* \) and \( \alpha^* \)]

Figure 5-13: Fuzzy Decision on Parameter \( \alpha \)

In this example, \( \alpha^* = 0.62 \) and \( \mu^*_D = 0.62 \).

The boolean decision variable: \( y_3 = 1 \). \( F_2 = 30 \) and \( S_2 = 0 \).

This means the best decision is to dispatch a train at time 2:00PM with a train length of 95+30 = 125 cars. No cars are left in the yard when the train is dispatched \( (S_2 = 0) \).

Notice that if the traditional crisp mathematical method is used, a train might be dispatched at 1:00PM with a train length of 120 and 5 cars will be left in the yard.
until the next scheduled decision.

Comparison of FLP and CLP Methods in One-train Case

Figure (5.4) shows a comparison of dispatching policy making on one freight train at one yard by using the fuzzy linear programming (FLP) approach (Section 3) and the classical linear programming (CLP) method. Because of the introduction of fuzzy constraints or soft boundaries in train length, the decision from the FLP approach is closer to the decision made by a human yard master. The train length will be extended to 125 in order to move away all the cars that have already arrived at the yard.

<table>
<thead>
<tr>
<th></th>
<th>FLP</th>
<th>CLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha^*$</td>
<td>0.62</td>
<td>N/A</td>
</tr>
<tr>
<td>$\mu_B$</td>
<td>0.62</td>
<td>N/A</td>
</tr>
<tr>
<td>Dispatch At Time</td>
<td>$t_3$</td>
<td>$t_2$</td>
</tr>
<tr>
<td>Train Length</td>
<td>125</td>
<td>120</td>
</tr>
<tr>
<td>No. of Cars Left</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Cost</td>
<td>$3860</td>
<td>$4120</td>
</tr>
</tbody>
</table>

**Table 5.4:** Comparison of FLP and CLP Approaches
5.3.2 Railroad Network Simulation

A simple railroad network simulation is used to test the idea of a fuzzy dispatching policy more extensively. The configuration of an assumed simulation network is shown in Figure (5-14).

![Railroad Network Diagram]

**Figure 5-14: Railroad Network**

Yards $C_0$, $C_1$ and $C_2$ are classification yards. Yards $N_0$ through $N_9$ are local yards. Yard $C_0$ receives traffic from local yards $N_0$, $N_1$, $N_2$, $N_3$ and $N_4$. Yard $C_1$ receives traffic from local yards $N_5$, $N_6$ and upstream classification yard $C_0$. Yard $C_2$ receives traffic from $C_1$ and distributes it to local yards $N_7$, $N_8$ and $N_9$. The distance from local yards $N_0$-$N_4$ to classification yard $C_0$ is 50 miles each; from local yards $N_5$ and $N_6$ to classification yard $C_1$ is 60 miles each; from classification yard $C_2$ to local yards $N_7$-$N_9$ is 80 miles each, and the distances between classified yards $C_0$-$C_1$ and $C_1$-$C_2$ are 150 miles each. All traffic in the current simulation is unidirectional with the destination in one of the local yards $N_7$, $N_8$ or $N_9$. The mean value of train
speed is 30 MPH, with standard deviation 10 MPH.

All traffic from local yards is of normal (Gaussian) distribution with mean values and standard deviations. Traffic from local yards \( N_0, N_1, N_2, N_3 \) and \( N_4 \) is scheduled twice a day (at 0:00 and 12:00) with mean traffic values of 7 cars and standard deviation 3 cars, for each destination yard \( N_7, N_8 \) and \( N_9 \).

Traffic from local yards \( N_5 - N_6 \) is scheduled four times a day at 0:30, 6:30, 12:30 and 18:30 o'clock, with mean traffic values of 9 cars and standard deviation 4 cars, towards each destination yard \( N_7, N_8 \) and \( N_9 \).

The mean traffic volume from classification yard \( C_0 \) to \( C_1 \) is 210 cars/day. Train makeup at classification yard \( C_0 \) is twice a day at 3:30 and 15:30 o'clock.

The mean traffic volume from classification yard \( C_1 \) to \( C_2 \) is 426 cars/day. Train makeup at classification yard \( C_1 \) is four times a day at 4:00; 10:00, 16:00 and 22:00 o'clock.

Train makeup at classification yard \( C_2 \) is four times a day at 4:30, 10:30, 16:30 and 22:30 o'clock. It departs to one of the local yards \( N_7, N_8 \) and \( N_9 \), dependent on whether the number of cars toward that local yard (with priority order from \( N_7, N_8 \) and \( N_9 \)) has reached the dispatching policy line.
5.3.3 Results with Human Subject

A simulation of freight train dispatching decision aid has been set up. Figure (5-15) shows the screen plot. In this simulation, it has three classification yards represented

![Diagram of Freight Train Dispatching Decision Aid]

**Figure 5-15: Experimental Setup**

by three rectangulars on the bottom of the screen. The shaded rectangular indicates the yard which is active for decision making. The policy lines for that yard are shown on the left side of the screen, and each policy line is determined by a linguistic knowledge base. The inputs to the linguistic knowledge base and the estimated number of coming cars are displayed on the right side of the screen. The windows on the middle of the screen show the goal satisfaction curve (in the upper window)
and the constraint satisfaction curve (in the lower window). The subject is able to select the degrees of satisfaction by clicking the mouse on these two windows. After the subject selects the degrees of satisfaction, the final crisp decisions are shown on the right-bottom corner of the screen.

The rail network and the decision aiding system have been tested with a human subject, with the cost and other parameters set to what the author regarded as reasonable values. Running the example in the computer simulation, the results are summarized in Table (5.5). It is seen that the total cost has reduced by using the ADAUS.

<table>
<thead>
<tr>
<th></th>
<th>Without Aid</th>
<th>With ADAUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satisfaction of Goal $\mu^*(g)$</td>
<td>N/A</td>
<td>0.54</td>
</tr>
<tr>
<td>Satisfaction of Constraints $\mu^*(c)$</td>
<td>N/A</td>
<td>0.52</td>
</tr>
<tr>
<td>Overall Satisfaction of $\mu^*(d)$</td>
<td>N/A</td>
<td>0.52</td>
</tr>
<tr>
<td>Total Cost</td>
<td>$23,000</td>
<td>$19,456</td>
</tr>
</tbody>
</table>

**Table 5.5: Comparison of Human Test Result**
5.4 Summary

Railroad freight train dispatching provides a case study of the proposed ADAUS technology which takes both soft data and hard data into consideration. Soft data, represented by fuzzy variables with memberships, includes soft variables and linguistic variables while hard data contains only crisp variables. A soft line dispatching policy is proposed to model the flexibility of maximum train length and a linguistic rule base is used to model the expertise about the policy line itself. According to ADAUS technology discussed in Chapter 2, a fuzzy linear program models the dispatching process in a structured mathematical way with all the crisp and soft variables. The linguistic rule base is modeled as a fuzzy rule network and can be continuously calibrated. Computer software for rule acquisition has been written for this special purpose. With the cost and other parameters set to what the author regarded as reasonable values, the one-train simulation and multiple yards simulation show the improvement of railroad operational performance for the hypothetical example.
Chapter 6

Conclusions and Future Research

6.1 Conclusions

Most decision-making processes in the real environment consist of two parts: the mathematical modelable part and the heuristic rule part. Two schools of researchers currently work on two separate approaches: one group works on a quantitative mathematical modeling approach, as is typical in operations research, while the other group works on a qualitative reasoning approach, such as an expert system. Either approach by itself loses part of the information available.

This thesis investigates various methods of dealing with the uncertainty involved in human knowledge and selects fuzzy set theory as a main vehicle to coordinate imprecise human expertise and soft boundaries within a resource-constrained decision-making environment. It proposes a new approach, *Adaptive Decision Aiding Under Soft Boundaries* (*ADAUS*), which combines quantitative and qualitative reasoning to deal
with objective crisp information and subjective fuzzy information simultaneously for a better decision support.

In this approach, fuzzy mathematical programming (FMP) is used to model quantitatively basic resource constraints and objective functions. In contrast to classical mathematical programming, FMP allows for soft constraints on the right-hand side of equations and in its technology matrix to accommodate uncertainties. A linguistic knowledge base (LKB) is used to model imprecise human expertise. FMP and LKB interact via a parameter vector which is a link between the two modules. A human decision-maker is able to choose a satisfactory level between the goal and the constraints for active human-in-the-loop control.

Subjective information, which is usually fuzzy and described by human experts by means of linguistic rules, is mainly modeled and calibrated by a fuzzy knowledge system, whereas objective information is systematically structured by fuzzy mathematical programming – an extension of classical mathematical programming that introduces soft constraints and soft parameters. The FMP makes use of well-structured information, while the LKB permits expertise and intuition to be taken into account. The feature of FMPs allowing fuzzy variables in its structured formula makes it easy to have the outputs from the fuzzy knowledge base serve as links between the two modules.

This thesis also discusses the imprecision of human knowledge in the process of establishing linguistic knowledge bases, implemented a fuzzy rule network (FRN), which can not only represent linguistic knowledge but also can be calibrated re-
cursively. Different from a neural network, the weights in the FRN have physical meanings which give an indication of the convergence level. A new Random-Sampling-Without-Replacement (RSWOR) experiment design in membership function elicitation is also suggested. ADAUS has another feature that can adaptively calibrate itself from the human decision-maker's response.

The proposed ADAUS framework is applied to railroad freight train dispatching as a case study. In the dispatching process, a human yardmaster plays an active decision-making role, but the mathematical model for the rail track network and the published schedule is too complex for a human to handle. A simulation example shows that using the new decision aid can improve the overall performance of freight train dispatching.

The success of this case study suggests that the proposed decision-making framework (ADAUS) can be applied to other decision environments where some mathematical model is available and human involvement is important.

6.2 Direction for Future Research

Though in this thesis the proposed framework of decision support was applied to freight train dispatching, other applications can be examined. Some possible areas for which models can be constructed are transportation planning, intelligent highway/vehicle systems (IVHS), manufacturing operation management and system diagnosis. Some less easily modeled systems include economic systems, political systems.
and military systems. In the direction of practical application, a more general software program can be developed for industrial use. Further simulation of freight train dispatching on a real classification yard needs to be done, and if justified, tests on a real rail network should be tried.

We have worked with single goal decision-making problems in which the objective evaluation in inherent. Multiobjective decision-making problems can be reduced into single goal problems in the $FMP$ formula as discussed in Appendix A. Interactive dialogues with human beings need further study.

Experts are not just static repositories of knowledge and skills but rather managers of the knowledge acquisition process itself, arbitrating the inductive process whereby knowledge is created and updated through experience. The linguistic knowledge base network representations and calibrations need to be further developed, especially in the direction of considering the effects of $FMP$ on the $FRN$.

In decision making, simplifications, or data fusion, allow humans to deal with complex systems using limited resources. It remains fascinating and unclear how a human selects a decision algorithm which simplifies data while retaining the essential characteristics to achieve moderately satisfactory decisions. The $ADAUS$ is one step toward this kind of simplification. With more study on human behavior, fuzzy mathematical programming and linguistic knowledge bases, a more powerful and reliable decision aid can be produced.
Appendix A

Fuzzy Logic: What, Why and How

In the real world, truly deterministic problems are quite rare: operation costs are usually estimated rather than known, the capital available is estimated, completion dates for construction projects are established with normal allowance for bad weather, and so on. In such circumstances the use of a deterministic model is justified only if the deviations in practice are expected to be small.

To deal with uncertainty, one well known method is probability theory, which works well in some cases and not in others. However, both laboratory settings and actual applications have demonstrated that human cognitive processes do not seem to naturally follow probability models [42]. Consider for example the following problem [104]: An urn contains approximately 100 balls of various sizes, of which several are large. What is the probability that a ball drawn at random is not large? To address uncertainty and imprecision simultaneously, fuzzy logic is a new approach which is a complement to probability theory.
A.1 What is Fuzzy Logic

Fuzzy logic, or fuzzy set theory, was introduced by Zadeh [99] in the mid-1960s as a means of representing and manipulating imprecise data. Conventional crisp sets allow for only yes/no responses, which in mathematics means an element is either 0 or 1. On the other hand, fuzzy sets allow for more imprecise categorizing between the yes/no answers which mathematically means that an element can be graded anywhere between 0 to 1. Fuzzy set theory is hence an extension of classical set theory and provides a mechanism for representing linguistic constructs such as “many”, “few”, “good”, “bad” and “most possible”, et cetera.

Though fuzzy set theory was introduced in the United States, it has been more extensively developed in Asian and European nations than in the United States, where it is controversial. Culture and language perhaps have contributed part of the reasons: in European and Asian languages, the word “fuzzy” is either a neutral word or a fashionable one implying “new technology”. The word fuzzy helps sales of appliances in Japan. There is no contradiction in fuzzy lenses that improve the focus of camcorders. But in the U.S. customers would hesitate when they see a new appliance that is fuzzy-logic-controlled. Historically the word “fuzzy” in English means not clear and implies not so good. Therefore the words “fuzzy logic” sound strange at least initially to some people in the United States: “How could the word fuzzy, meaning not clear, combine with the word logic, meaning clear and distinct?”

In fact, the basic idea of fuzzy sets is easy to communicate and grasp. For instance, when you watch your friend driving too close to the car in front of him, it is more
natural and meaningful to tell him to "brake a little" than to instruct him to "brake with a force of 2 pounds and 7 ounces".

Sets and Subsets

A set is defined as a collection or aggregate of objects. The objects that belong to the set are termed the elements of the set. The term universal set, or universe of discourse is applied to the set that contains all the elements which one wishes to consider. The symbol $\mathcal{U}$ represents a general universe. A subset is the set that contains only certain elements from the universal set. For example, letting a set $A$ represent the letters in the English alphabet and a subset $B$ represent the letters in the word \textit{train}, $A = \{ a, b, c, \ldots, x, y, z \}$ is a set and $B = \{ t, r, a, i, n \}$ is a subset. It is clear that whether a collection of objects is called set or subset is determined by the definition of the universe. A subset is sometimes called set under known context.

Characteristic Functions

A crisp set refers to the traditional classification by feature where an object either belongs to or does not belong to a given set. The characteristic function of a crisp set is accordingly binary valued: one, if some element belongs to the set; zero, if it does not. A fuzzy set is distinguished from a crisp set by another characteristic function which takes on any values in the interval $[0, 1]$. This special characteristic function is called a fuzzy membership function or simplified as membership function. For example, given that the normal driving speed on a highway takes its value from the interval $[50,
the linguistic term “normal speed” may be described either as a crisp set $A(x)$ or as a fuzzy set $\tilde{A}(x)$ of some appropriate universe of discourse (the interval $[50, 70]$ in this case) as shown in Figure (A-1).

Figure A-1: Crisp Characteristic Functions v.s. Fuzzy Membership Function

The characteristic function for this crisp set $A(x)$ is defined as

$$
\mu_{A(x)} = \begin{cases} 
1 & \text{iff } x \in [50, 70] \\
0 & \text{otherwise} 
\end{cases} \quad (A.1)
$$

It says that any speed between 50 mph and 70 mph is perfectly normal which is of course not true.

The fuzzy membership function of $\tilde{A}(x)$ improves the definition. It is defined as

$$
\mu_{\tilde{A}(x)} = \begin{cases} 
0 & \text{if } x \leq 50 \\
1 - \frac{55-x}{5} & \text{if } x \in (50, 55) \\
1 & \text{if } x \in [55, 65] \\
1 - \frac{x-65}{5} & \text{if } x \in (65, 70) \\
0 & \text{if } x \geq 70
\end{cases} \quad (A.2)
$$

In the fuzzy membership function, it is noticed that the transition between the
member and non-member is gradual rather than abrupt. It says that only the speeds which fall in the interval between 55 mph and 65 mph are perfectly normal (membership of 1). The speeds in [50, 55] or [65, 70] are marginally normal with a certain degree of membership.

In a mathematical world, a fuzzy set $\tilde{A}$ of some universe of discourse $\mathcal{U}$ is a collection of objects from $\mathcal{U}$ such that the characteristic function $\mu_{\tilde{A}}$ takes any value in the interval $[0,1]$. A crisp set is a special case of a fuzzy set, when the membership is restricted to either of two values, 0 and 1. The fuzzy subset $\tilde{A}(x)$ with a trapezoidal fuzzy membership function in Equation (A.2) is symbolically denoted as

$$\tilde{A}(x) = (5, 55, 65, 5)$$  (A.3)

Convex and Normal

The notion of convexity can be generalized to fuzzy subsets of a universe $R$, which is assumed to be a real Euclidean N-dimension space. $\tilde{A}$ is convex iff:

$$\mu_{\tilde{A}}(\alpha \cdot x_1 + (1 - \alpha) \cdot x_2) \geq \min(\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2))$$

$$\forall \ x_1, x_2 \in R \ and \ \alpha \in [0,1]$$  (A.4)

If at least one element of a fuzzy subset in the universe of discourse has the membership one, that fuzzy subset is said to be normal. That is:

$$\exists \ x \in R \ such \ that \ \mu_{\tilde{A}}(x) = 1.$$
Partitions

In the previous section, the characteristic function associates with a subset which covers only part of the universe of discourse. In order to cover the whole space of the universe of discourse, more than one fuzzy subset is defined. We denote each subset by a symbol of $\tilde{A}_i$, and refer to the collection of these subsets as a set of $\tilde{A} = \{\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_i, \ldots, \tilde{A}_n\}$. This collection consists of $n$ fuzzy subsets covering the whole universe of discourse.

For instance, the driving speed on a highway may be partitioned into five categories ($n=5$) very slow, slow, normal, fast and very fast as shown in Figure (A-2).

![Figure A-2: Membership Function of Driving Speed](image)

Fuzzy set $\tilde{A}$ is symbolically denoted $\tilde{A} = \{\mu_{\tilde{A}(x)} | x \in \mathcal{U}\}$ or $\tilde{A} = \int_{x \in \mathcal{U}} \mu_{\tilde{A}(x)} \# x$ when $\mathcal{U}$ is a continuum. When $\mathcal{U}$ has $n$ discrete elements, the fuzzy set is written as

$$\tilde{A} = \{\mu_{\tilde{A}(x_1)} \# x_1, \mu_{\tilde{A}(x_2)} \# x_2, \ldots, \mu_{\tilde{A}(x_n)} \# x_n\}.$$  \hspace{1cm} (A.5)
The symbol \# is employed to link the elements of the support (speed in the above example) with their grade of membership, and the support of a fuzzy set is the crisp set that contains all the elements that have nonzero membership grade. In the known context, the symbol \# is usually omitted, so that Equation (A.5) becomes:

$$\tilde{A} = \{\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2), \cdots, \mu_{\tilde{A}}(x_n)\}$$  \hspace{1cm} (A.6)

Some restrictions may be imposed on the definition of the individual elements, such as

$$\sum_{i=1}^{n} \mu_{\tilde{A}_i}(x) = 1 \quad \forall \ x \in U,$$

which means that fuzzy classification must be compatible with a feature-based classification in terms of crisp sets where each element is categorized under one and only one class.

**Interval of confidence**

The interval of confidence is a closed range of an uncertain variable with a confident level assigned to this range. For instance, one may say that a person's weight is \textit{maybe} between 140 and 160 pounds. Here [140, 160] is the interval with a confidence of \textit{maybe}.

For a convex fuzzy set (which is natural) \(\tilde{A}\) in the universe \(U\), the interval of confidence can be described by a method called \(\alpha\)-cut (discussed later). The \(\alpha\)-cut transforms \(\tilde{A}\) into a crisp subset \(A_\alpha\) as shown in Figure (A-3). The smallest value of
x for which \( \mu_\tilde{A}(x) \geq \alpha \) is called lower support \( a_1^\alpha \), and the largest value of x for which \( \mu_\tilde{A}(x) \geq \alpha \) is called the upper support \( a_2^\alpha \). The interval \([a_1^\alpha, a_2^\alpha]\) and the corresponding membership \( \alpha \) is an interval of confidence:

\[
A_\alpha = [a_1^{(\alpha)}, a_2^{(\alpha)}]
\]  

(A.7)

This means that it is certain that the event would happen in \([a_1, a_2]\) under possibility level \( \alpha \).

![Figure A-3: \( \alpha \)-cuts](image)

The fuzzy set \( \tilde{A} \) is the integration of \( A_\alpha \), i.e.,

\[
\tilde{A}(x) = \int_{\alpha \in [0,1]} A_\alpha \cdot d\alpha
\]  

(A.8)

\( \alpha \)-cuts

The \( \alpha \)-cut \( A_\alpha \) of a fuzzy set \( \tilde{A} \) is the classical or crisp set for all \( x \in U \) such that the membership value is greater than some threshold \( \alpha \in (0,1] \).

It is denoted as
\[ A_\alpha = \{ x \in U, \mu_A(x) \geq \alpha \} \quad \alpha \in (0,1] \quad (A.9) \]

If the \( \alpha \)-cut sets (\( \alpha \in [0,1] \)) are unioned together, it returns to the original fuzzy set \( \tilde{A} \)

\[ \tilde{A} = \bigcup_{\alpha \in [0,1]} \alpha \cdot A_\alpha \quad (A.10) \]

Notice that \( A_\alpha \) itself is a crisp set, which links fuzzy sets and crisp sets. The idea of \( \alpha \)-cuts is often used in fuzzy decision making and fuzzy algorithmic operations.

**Fuzzy Union and Intersection**

The definitions of basic operations on sets must be modified for use in fuzzy set theory. Several notable structures can be defined on \( \mu_A \) in the interval \([0,1]\), each of which introduces the union and intersection operations, and which coincide with the classical ones. The widely accepted max/min definition is given by Zadeh \[100]\:

**Union**

\[ \mu_{\tilde{A} \cup \tilde{B}} = \max(\mu_{\tilde{A}(x)}, \mu_{\tilde{B}(x)}) \quad (A.11) \]

**Intersection**

\[ \mu_{\tilde{A} \cap \tilde{B}} = \min(\mu_{\tilde{A}(x)}, \mu_{\tilde{B}(x)}) \quad (A.12) \]
The more general definition of fuzzy logic operator is $T_{\text{norm}}$ and $T_{\text{conorm}}$.

**Fuzzy Logic Operator: T-Norms and T-Conorms**

The most general logic operators are T-Norms and T-Conorms, which include the most frequently used fuzzy operators $\text{max}$ and $\text{min}$.

T-norms, or triangular norms, and T-conorms, or triangular conorms, are general operators used to deal with data which fall in the interval $[0,1]$. Statisticians have used this concept for a long time and now it has been adapted to fuzzy set theory as fuzzy logic operators [93].

**Definition**

A T-norm is a mapping from two arguments $L \in [0,1]$ to $T \in [0,1]$. That is, $L \times L \Rightarrow T$ and T-conorm is a mapping from two argument $L \in [0,1]$ to $T_{\text{co}} \in [0,1]$. That is, $L \times L \Rightarrow T_{\text{co}}$, such that it has the following characteristics:

1. Monotonic

   \[
   \text{if } x \leq y, \ w \leq z, \ \text{then } \ T(x, w) \leq T(y, z) \\
   T_{\text{co}}(x, w) \leq T_{\text{co}}(y, z)
   \]  

   (A.13)

2. Commutative

   \[
   T(x, y) = T(y, x) \quad T_{\text{co}}(x, y) = T_{\text{co}}(y, x)
   \]  

   (A.14)
3. Associative

\[ T(T(x, y), z) = T(x, T(y, z)) \quad T_{co}(T_{co}(x, y), z) = T_{co}(x, T_{co}(y, z)) \]  \hspace{1cm} (A.15)

4. Boundary conditions

\[ T(x, 0) = 0 \quad \text{and} \quad T(x, 1) = x \quad \text{for} \ T - \text{norm} \]
\[ T_{co}(x, 0) = x \quad \text{and} \quad T_{co}(x, 1) = 1 \quad \text{for} \ T - \text{conorm} \]  \hspace{1cm} (A.16)

\[ \forall \ x, y, z, w \in [0, 1] \]

A method of generating a T-norm and a T-conorm is summarized as follows:

Suppose \( g(s) \) and \( h(s) \) are strictly monotonic in a segment of \( \mathbb{R} \), and

\[ G(t) = g^{-1}(s), \hspace{1cm} H(t) = h^{-1}(s) \]  \hspace{1cm} (A.17)

If \( F(a, b) \) is generated by \( g(s) \), where \( g(0) = 0 \) and \( g(1) = 1 \), then

\[ F(a, b) = G[1 \wedge g(a) + g(b)] \]  \hspace{1cm} (A.18)

is a T-conorm and

\[ F(a, b) = G[0 \vee (g(a) + g(b) - 1)] \]  \hspace{1cm} (A.19)

is a T-norm.
Some T-norms and T-conorms

\[ T(x, y) = \min(x, y) = x \land y \hspace{1cm} T_{co}(x, y) = \max(x, y) = x \lor y \]

\[ T(x, y) = x \cdot y \hspace{1cm} T_{co} = x + y - x \cdot y \]

\[ T(x, y) = \frac{x y}{1 + (1-x)(1-y)} \hspace{1cm} T_{co}(x, y) = \frac{x + y}{1 + x y} \]

\[ T(x, y) = \frac{2 x y}{x + (1-x)(x+y-x y)} \hspace{1cm} T_{co}(x, y) = \frac{x + y - x y(1-v)x y}{v + (1-v)(1-x y)} \]

\[ v \in [0, \infty] \]

\[ T(x, y) = 1 - \{1 \land [(1 - x)^p + (1 - y)^p]^{1/p}\} \hspace{1cm} T_{co}(x, y) = (x^p + y^p)^{1/p} \]

\[ p \in [1, \infty] \]

(A.20)

All T-norms and T-conorms satisfy DeMorgan's law, but only the first set of T-norm and T-conorm (max-min) satisfies the following four equations:

\[ T(x, x) = x \]

\[ T_{co}(x, x) = x \]

\[ T(x, T_{co}(y, z)) = T_{co}(T(x, y), T(x, z)) \]

\[ T_{co}(T(x, T(y, z)), T_{co}(x, z)) = T(T_{co}(x, y), T_{co}(x, z)) \]

(A.21)

Therefore, the max-min operators are widely used in fuzzy set theory as fuzzy operators. But in the fuzzy relation operation, other T-norms and T-conorms can be used and compared under certain circumstances.

Fuzzy Logical not, and, or

The negation not and the connective and and or are logical operations. They are defined as follows:

Let \( \tilde{A} \) and \( \tilde{B} \) be fuzzy subsets of \( U_1 \) and \( U_2 \), respectively. Then
\[ \tilde{A} = \{ \mu(x) \# x \mid \mu(x) = 1 - \mu_A(x) \text{ and } \mu_A(x) \# x \in A \} \quad (A.22) \]

\[ \tilde{A} \text{ and } \tilde{B} = \tilde{A} \cap \tilde{B} \]
\[ \tilde{A} \text{ or } \tilde{B} = \tilde{A} \cup \tilde{B} \quad (A.23) \]

Complementation

In a crisp set, the complement of \( A \) is the subset of elements of \( U \) that are not members of \( A \). For a fuzzy set \( \tilde{A} \), the complement of \( \tilde{A} \) is denoted \( \tilde{A}' \) and is defined as follows:

\[ \tilde{A}' = \{ \mu(x) \# x \mid \mu(x) = 1 - \mu_{\tilde{A}}(x) \quad \text{if } x \in U; \mu(x) = 1 \text{ otherwise} \} \quad (A.24) \]

Distance Between Two Fuzzy Sets

Let \( \tilde{A}, \tilde{B} \in U \), then

\[ D(\tilde{A}, \tilde{B}) = \left\{ \sum_{i=1}^{n} \| \tilde{A}(x_i) - \tilde{B}(x_i) \|^p \right\}^{1/p} \quad (A.25) \]

is called the Minkowski [78] distance between the fuzzy sets \( \tilde{A} \) and \( \tilde{B} \). When \( p = 1 \), the Minkowski distance becomes the Hamming distance. When \( p = 2 \), the Minkowski distance becomes the Euclidean distance. The idea of distance between two fuzzy sets is used to determine if they resemble each other.
Addition

Let $\tilde{A}$ and $\tilde{B}$ be two fuzzy subsets in the universe $\mathbb{R}$, and $\tilde{A}_\alpha$ and $\tilde{B}_\alpha$ their intervals of confidence for the level of assumed $\alpha$, $\alpha \in [0,1]$. The addition of the two fuzzy subsets is defined

$$\tilde{A}_\alpha(+)\tilde{B}_\alpha = [a_1^{(\alpha)} + b_1^{(\alpha)}, a_2^{(\alpha)} + b_2^{(\alpha)}]$$ (A.26)

In the representation of the Cartesian product, this can be rewritten as

$$\mu_{\mathcal{C}}(z) = \mu_{\tilde{A}}(+)\mu_{\tilde{B}} = \bigvee_{z=x+y} [\mu_{\tilde{A}}(x) \land \mu_{\tilde{B}}(y)]$$ (A.27)

Subtraction

The definition of addition can be extended to the definition of subtraction:

$$\tilde{A}_\alpha(-)\tilde{B}_\alpha = [a_1^{(\alpha)} - b_2^{(\alpha)}, a_2^{(\alpha)} - b_1^{(\alpha)}]$$ (A.28)

In the representation of the Cartesian product, this can be rewritten as

$$\mu_{\mathcal{C}}(z) = \mu_{\tilde{A}}(-)\mu_{\tilde{B}} = \bigvee_{z=x-y} [\mu_{\tilde{A}}(x) \land \mu_{\tilde{B}}(y)]$$ (A.29)
Multiplication

Let $\tilde{A}$ and $\tilde{B}$ be two fuzzy subsets in the universe $R^*$, where $R^*$ represents the universe as real and positive numbers, and $\tilde{A}_\alpha$ and $\tilde{B}_\alpha$ represent their intervals of confidence for the level of assumed $\alpha$, $\alpha \in [0, 1]$. The multiplication of the two fuzzy subsets is defined

$$\tilde{A}_\alpha(\cdot) \tilde{B}_\alpha = [a_1^{(\alpha)} \cdot b_1^{(\alpha)}, a_2^{(\alpha)} \cdot b_2^{(\alpha)}]$$ (A.30)

Multiplication can also be given in the form of the Cartesian product

$$\mu_C(z) = \mu_A(\cdot) \mu_B = \bigvee_{z=x \cdot y} [\mu_A(x) \wedge \mu_B(y)]$$ (A.31)

Division

The definition of multiplication can be extended to the definition of division:

$$\tilde{A}_\alpha(\cdot) \tilde{B}_\alpha = [a_1^{(\alpha)} / b_2^{(\alpha)}, a_2^{(\alpha)} / b_1^{(\alpha)}] \quad b_2^{(\alpha)} > 0$$ (A.32)

In the Cartesian product form, it is

$$\mu_C(z) = \mu_A(\cdot) \mu_B = \bigvee_{z=x / y} [\mu_A(x) \wedge \mu_B(y)]$$ (A.33)
A.2 Why Use Fuzzy Logic

It has been shown that the characteristic function \( \mu_{A(x)} \) of a classical or crisp set takes a unique value in the two element set \( \{0, 1\} \). That is, classical set theory is governed by a logic that permits a proposition to possess only one of two values: true or false. This logic does not accord well with the need to represent vague concepts. We see things in shades of gray, not in black and white. In everyday life, the observations and thoughts of most people most of the time may be said to be mentally modeled and/or communicated to other persons in terms of sets of imprecisely defined natural language words (such as short, medium, long, or very long when applied, for example, to a train’s length, or urgent, soon, later, much later, when applied to the time available for a train’s departure). Key elements in human thinking are not crisply defined but are more approximate. It appears that much of the logic behind human reasoning is not the traditional two-valued or even multivalued logic, but logic with fuzzy truth, fuzzy connectives and fuzzy rules of inference.

Fuzzy logic’s ability to deal with imprecise or ambiguous information allows it to cope effectively with noisy data and variation in system components. Fuzzy logic is most appropriate when a process meets the following three criteria:

1. adequate mathematical models either do not exist or are too complex to solve in available time

2. the process involves nonlinearities or multiple parameters

3. the process parameters’ basic interrelationships are understood by human ex-
perts.

In the past five to ten years, fuzzy logic has successfully supplanted more conventional technologies in many scientific applications and engineering systems. An article\(^1\) in *Business Week* indicates that in 1988 Japan started a Laboratory of Industrial Fuzzy Engineering (LIFE) with an annual budget of around 24 millions US dollars for seven years. The Japanese now hold thousands of patents on fuzzy devices from subway trains, camcorders, air conditioners, washing machines to ship navigators and automobile transmissions.

![Figure A-4: Evolution of Fuzzy Systems](image)

Bezdek [7] pointed out an interesting evolution curve of fuzzy logic. The evolution is summarized in Figure (A-4). The abscissa in Figure (A-4) is time of the year and the ordinate is the investment or expectation on fuzzy logic development. It shows that fuzzy system development had its peak of hype in the early 1970's followed by an overreaction which led to a crash of sorts and a period of wallowing in the depths of cynicism, and finally in 1991 began a more realistic development. The trend is similar to most other new technologies that eventually become useful.

\(^{1}\)U.S. Companies Are Trying to Catch up To Japan’s Ability to Get Machines Working More Like People, April, 1992.
Recently, the research and application of fuzzy logic in academia, industry and government have been manifested by the rapid growth of national and international conferences which include:


With the new Transaction by IEEE in January, 1993, there are now several major journals devoted to fuzzy logic:


2. Fuzzy Sets and Systems.


4. SOFT Journal.

In the U.S., fuzzy logic has recently attracted lots of interest from industry. Motorola Inc. has begun selling a software development system to help engineers design fuzzy logic products. Ford Motor Co. challenged one of its engineering teams to apply
fuzzy logic controls that keep a car's engine idling smoothly under changing conditions, such as when the air-conditioning kicks in at a traffic light. Otis Elevator tried to match Hitachi, Mitsubishi and Toshiba which already have elevators that reduce waiting time by using fuzzy-logic controls to respond on the fly to changing demands. General Electric is pushing to use fuzzy logic to regulate jet-aircraft engines, save water in home applications, and drive 200-ton rollers in steel mills.

The inability of crisp sets to deal with imprecise, ambiguous information (which is the reality in daily life) makes it necessary to seek more efficient solutions. Fuzzy logic has been proven itself as a very promising approach both in academia and industry.

All three criteria (mentioned in this section) of using fuzzy logic are applicable to railroad operations. With the development of fuzzy logic theory and the wide availability of powerful computers as well as many successful industrial researches and applications, it seems time for railroad companies to investigate the possibilities of using fuzzy logic to improve service and reduce costs.

### A.3 How to Use Fuzzy Logic

One common use of fuzzy logic is to form a fuzzy rule base consisting of various common linguistic if-then statements. Most fuzzy logic applications mentioned above are working in this way. Another recent development uses fuzzy variables to represent uncertain values in mathematical programming to maximize or minimize objective functions. We will use two simple examples to illustrate the two approaches.
A.3.1 Fuzzy Rule Base

A typical fuzzy rule base consists of many linguistic if-then rules as shown in Table (A.1). This rule base has \( m \) rules in a \( n \)-input/\( k \)-output system as shown in Figure (A-5). In the \( i \)-th rule, \( X^n_i \) is the \( n \)-th observation and \( Y^k_i \) is the \( k \)-th output action.

\[
\begin{align*}
\text{If } X^n_1 \text{ and } X^n_2 \text{ and } \cdots \text{ and } X^n_n & \text{ THEN } Y^k_1 \text{ and } \cdots \text{ and } Y^k_k \\
\vdots & \vdots \quad \vdots \quad \vdots \\
\text{If } X^n_1 \text{ or } X^n_2 \text{ and } \cdots \text{ and } X^n_n & \text{ THEN } Y^k_2 \text{ and } \cdots \text{ and } Y^k_k \\
\vdots & \vdots \quad \vdots \quad \vdots \\
\text{If } X^n_1 \text{ and } X^n_2 \text{ or } \cdots \text{ or } X^n_n & \text{ THEN } Y^k_m \text{ and } \cdots \text{ and } Y^k_k
\end{align*}
\]

Table A.1: Fuzzy Rule Structure

![Fuzzy Rules Diagram](image)

Figure A-5: Fuzzy Rule Base

What a fuzzy rule base does is first to fuzzify input data \( X \) and then test it with the if-then rules (which can be processed in parallel) to produce one or more responses, depending on how many rules are applicable. Each rule's response is later weighted into \( Y \) in proportion to the degree of membership of its inputs.

In Table (A.1), the \( n \) observations in the \( i \)-th rule are connected by logic operators and or or. The \( n \) observations and the logic operators are called the condition part, or the left-hand side of a linguistic rule. In a similar way, the right-hand side of a linguistic rule is called the action part.
A fuzzy rule base usually consists of three stages: fuzzification, fuzzy reasoning and defuzzification.

**Fuzzification**

Fuzzification is a procedure to code a crisp variable into a fuzzy variable by a measure of membership. The membership determines the degree of belief that a measurement $x$ falls in a special category called a set.

In a crisp set, the membership of $x$ in set $A$ is described by a characteristic function $\mu_A(x)$. For example, the *normal* driving speed on the highway may be a crisp variable which would have a membership function of:

$$
\mu_A(x) = \begin{cases} 
1.0 & \text{if } x \in [60, 65] \\
0.0 & \text{otherwise}
\end{cases} \quad (A.34)
$$

The membership is a boolean type function. If $x$ is within the set $A$ (60 – 65), the membership is one. Otherwise, it is zero. When the driving speed $x$ is 58 $MPH$, the membership of *normal speed* is zero which is not reasonable.

In a fuzzy set, the membership of $x$ is continuous from one to zero. For example, the *normal* driving speed on the highway is a fuzzy variable which would have a
membership function of:

\[
\mu_A(x) = \begin{cases} 
0.0 & \text{if } x \leq 50 \\
1.0 - \frac{60-x}{10} & \text{if } x \in (50, 60) \\
1.0 & \text{if } x \in [60, 65] \\
1.0 - \frac{x-65}{5} & \text{if } x \in (65, 70) \\
0.0 & \text{if } x \geq 70 
\end{cases} \tag{A.35}
\]

Defuzzification is a procedure to determine the fuzzy membership. When the driving speed \( x \) is 58 MPH, the membership of \textit{normal speed} is 0.8 as shown in Figure A-6.

![Figure A-6: Fuzzification](image)

**Fuzzy Reasoning**

Much of human reasoning is approximate rather than exact as in the statement:

\[
\begin{align*}
\text{IF} & \quad \text{road IS wet} & \text{THEN} & \quad \text{drive slowly} \\
\text{THIS} & \quad \text{road IS sort of wet} & \text{drive} & \quad \text{sort of slowly.} \tag{A.36}
\end{align*}
\]

Such reasoning can not be made sufficient by the inference rules of classical two-
valued logic. To reason in this way with fuzzy set theory, Zadeh [100] first suggested an inference rule — *compositional rule of inference*. He proposed translation rules for translating a fuzzy conditional proposition "If $x$ is $A$, then $y$ is $B$" into a fuzzy relation. Several alternative approaches were presented later, such as Baldwin [2], Tsukamoto [87] and Yager [95] by introducing fuzzy logic with fuzzy true values.

**Defuzzification**

Since in many practical applications, a crisp control is required, a defuzzification which maps from a space of fuzzy variable, defined over an output universe of discourse, into a space of nonfuzzy values is necessary. Unfortunately there is no unique systematic procedure for defuzzification. Zadeh [99] first pointed out this problem and made the tentative suggestions for dealing with it. At present, three strategies are commonly used. They are described as the *max criterion method*, the *mean of maximum method* and the *center of area method*.

1. The max criterion method

The max criterion produces the point at which the possibility distribution of the control action reaches a maximum value

$$z_0 = \max_{j=1}^l w_j$$

where $w_j$ is the support value at which the membership function reaches the maximum value $\mu_z(w_j)$, and $l$ is the number of supports (if discrete)
or range of support (if continuous).

2. The mean of maximum method

The mean of maximum method, sometimes called \( MOM \), generates a control action which represents the mean value of all local control actions whose membership functions reach the maximum. More specifically, in the case of discrete universe, the control action may be expressed by

\[
z_0 = \frac{1}{l} \sum_{j=1}^{l} w_j
\]

where \( w_j \) is the support value at which the membership function reaches the maximum value \( \mu_z(w_j) \), and \( l \) is the number of supports.

3. The center of area method

The center of area method, sometime called \( COA \), is a widely used strategy which generates the center of gravity of the possibility distribution of the control action. In the case of a discrete universe, a \( COA \) defuzzification is in the form of:

\[
z_0 = \frac{\sum_{j=1}^{n} (\mu_z(w_j) \cdot w_j)}{\sum_{j=1}^{n} \mu_z(w_j)}
\]

where \( n \) is the number of quantization levels of the output.
Braae and Rutherford [10] presented a detailed analysis of various defuzzification strategies (COA, MOM, etc.) and concluded that the COA strategy yields superior results.

It is noted that fuzzy defuzzification is parallel inherited. That means, in contrast to the inference engine of backward or forward reasoning in a symbolic rule-base expert system, this method allows all the rules be activated in parallel. This feature makes much faster computation possible (for example, using fuzzy chips [61] at the hardware level).

To show how a fuzzy knowledge base works, let's consider a simple example with three rules regarding freight train dispatching and explain the procedures in graphics.

**Graphical Explanation of Fuzzy Rules**

This fuzzy rule base has three fuzzy variables: inbound traffic $\bar{P}$, time past schedule $\bar{T}$ and the minimum train length $\bar{L}$ that is regarded as being marginally economic-efficient. The three rules say:

1. Rule 1: *if predicted inbound traffic is medium and the time past schedule is very long, then the minimum train length is medium;*

2. Rule 2: *if predicted inbound traffic is medium and the time past schedule is long, then the minimum train length is long;*

3. Rule 3: *if predicted inbound traffic is small and the time past schedule is very long, then the minimum train length is short.*
Figure (A-7) illustrates the rules and rule inference in graphics. \( \hat{P} \) is the fuzzy value of inbound traffic and \( \hat{T} \) is the fuzzy value of time past schedule. \( \hat{L} \) is the minimum train length.

![Figure A-7: Graphic Explanation of Fuzzy Rules](image)

The rule structure is:

<table>
<thead>
<tr>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Rule 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>If ( \hat{P} ) is medium and ( \hat{T} ) is very long Then ( \hat{L} ) is medium</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If ( \hat{P} ) is medium and ( \hat{T} ) is long Then ( \hat{L} ) is long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If ( \hat{P} ) is small and ( \hat{T} ) is very long Then ( \hat{L} ) is short</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In figure (A-7), hypothetical states \( \hat{P} \) and \( \hat{T} \) are evaluated by fuzzification from \( p \) and \( t \), as shown by vertical dashed lines. The memberships for inbound traffic \( \hat{P}_{\text{cars}} \) (in the first column) are 0.2, 0.2 and 0.8; and the memberships for the time past \( \hat{T}_{\text{past}} \) (second column) are 0.8, 0.25 and 0.8.
Here we use *max* and *min* fuzzy logic operators as representation for *or* and *and* respectively in the rule statements. According to the fuzzy logic, for each rule the minimum train length is determined by the three paired memberships, that is, \( \text{min}(0.2, 0.8), \text{min}(0.2, 0.25) \) and \( \text{min}(0.8, 0.8) \).

In the defuzzification procedure, the membership functions representing the minimum train length are weighted by the contributions from the three rules. The Centroid of Area (COA) method is used here, which says:

\[
\text{Minimum Train Length} = \frac{\sum_{i=1}^{3}(\mu_i \cdot L_i)}{\sum_{i=1}^{3}\mu_i} \quad (A.37)
\]

where \( \mu_i \) is the membership function of the minimum train length corresponding to the \( i \)-th rule, and \( L_i \) is the minimum train length corresponding to the \( i \)-th rule.

In this example, the resulting minimum train length will be:

\[
\text{Minimum Train Length} = \frac{(0.2 \times 50) + (0.2 \times 75) + (0.8 \times 25)}{0.2 + 0.2 + 0.8} = 38 \quad (A.38)
\]

It means that for a train to run economically, it must take at least 38 cars, i.e.,

\[
\text{Train Length} \geq 38.
\]

**A.3.2 Fuzzy Mathematical Programming**

Another usage of fuzzy logic is *fuzzy mathematical programming (FMP)*.
The fuzzy mathematical programming (FMP) module is to model the basic resource constraints and objective functions. It takes consideration of goals to be achieved and resources available with uncertainty involved. Most crisp data, such as physical structures and configurations, are considered by the FMP module. The fuzzy coefficients in the technology matrix and the fuzzy constraints in the right-hand-side of the equations accommodate the uncertainties and soft boundaries. Some fuzzy coefficients are adjustable parameters that change their values as the mathematically unmodelable (usually fuzzy) environment changes. These parameters provide a connection between the fuzzy mathematical programming module and the linguistic knowledge base module. These parameters change the resources available or the constrain structure in the FMP module. Since fuzzy variables are the common building blocks of both modules, the interface between them is seamless. Chapter 3 discusses a single-objective FMP in detail.

Fuzzy Multiple Objective Programming

In this section, we will show that a fuzzy multiple objective programming problem can be transformed into a single objective FMP form.

Decision problems involving multiple and conflicting objectives, goals or attributes are generally referred to as multicriteria decision problems. In the literature on multicriteria decision problems, the terms objectives, goals and attributes are used to convey some distinct meanings, although they are often used interchangeably.

Since the distinguishing feature of the multicriteria decision problem is the ex-
istence of conflicting and incommensurable objectives, there is no solution which optimizes all the goals simultaneously. Instead of optimization, the concept of Pareto solutions is basic to multicriteria decision problems. A Pareto solution, sometimes called an efficient solution, is a noninferior or a nondominated solution. A set of feasible solutions is reducible to a set of efficient solutions without any information about the preferences of the decision maker.

Multicriteria decision problems are often represented as discrete problems which have a limited number of predetermined alternatives characterized by multiple attributes. The primary concern is with choosing the best one for the decision maker, ranking the alternatives in order of importance or reducing their number for making a final decision. In decisions under certainty, the multiattribute value function is a popular model for the preferences of the decision maker. In decisions under uncertainty, the approaches that admit the information of preferences including inconsistency is a fuzzy ranking relation model that does not require the assumption of transitivity nor the complete comparability in the underlining preference relation [36].

Fuzzy multiobject mathematical programming is in the form of:

\[
\begin{align*}
\sup_{x \in \mathbb{R}^n} & f_1(x) \Rightarrow \tilde{G}_1, \cdots, f_k(x) \Rightarrow \tilde{G}_k, \cdots, f_m(x) \Rightarrow \tilde{G}_m \\
\text{subject to:} & \\
& \begin{cases} 
\leq & i = 1, 2, \cdots, n \\
= & \tilde{b}_i \\
\geq & i = 1, 2, \cdots, n 
\end{cases} \\
x & \geq 0
\end{align*}
\] (A.39)
where \( x = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^n \) is a vector of decision variables;

\[
f_k(x), k = 1, 2, \ldots, m : \mathbb{R}^n \rightarrow \mathbb{R} \text{ are objective functions;}
\]

\[
\tilde{G}_k, k = 1, 2, \ldots, m : \text{are aspiration levels;}
\]

\[
\tilde{w}_i(x), i = 1, 2, \ldots, n : \mathbb{R}^n \rightarrow \mathbb{R} \text{ are constraints;}
\]

\[
\tilde{b}_i, i = 1, 2, \ldots, n : \in \mathbb{R} \text{ are the right-hand-side values or estimated available resources.}
\]

Since fuzzy multiobjective mathematical programming is a satisfactory approach under the assumption of predetermined aspiration levels and the total degree of satisfaction is the intersection of the degree of goal satisfaction and the degree of constraint satisfaction, this approach makes no distinction between objective functions and constraints. Thus a fuzzy approach to the multiobjective mathematical programming problem is stated as:

\[
\begin{align*}
\text{Max or Min} \quad & f_p(x) \quad \Rightarrow \quad \tilde{G}_p \\
\text{subject to:} & \\
& \begin{cases} 
  f_k(x) & \leq & \tilde{G}_k \\
  \tilde{w}_i(x) & = & \tilde{b}_i \\
  \end{cases} \\
& x \geq 0
\end{align*}
\]

(A.40)

In this way, equation (A.39) becomes equation (A.40) which is a single objective fuzzy programming with argumented constraints. Because aspiration levels are mixed
with constraint satisfactions in fuzzy multiple mathematical programming, they have
to be assigned carefully. A *Pareto* solution is usually obtained through interactive
dialog with decision-makers [76].

### A.4 Computer Software

A computer software package has been developed during the research. It includes the
software for linguistic knowledge base (*FOLOC*) and the software for fuzzy mathemat-
ical programming (*FLP123*).

#### A.4.1 FOGIC – Software for LKB

The linguistic knowledge base described in Chapter 4 is reasonably general and the
easy of implementation process can be further improved by the use of a computer-
aided development system. The software package *FOGIC* (*Fuzzy LOGIC*), which
works in Microsoft Windows for a more user-friendly interface, has been developed
as a tool for linguistic knowledge base applications.

**Linguistic Knowledge Bases Development**

*FOGIC* has the following features:

1. Graphical design of fuzzy variables.

2. Fuzzy membership elicitations.
3. Form-free design of linguistic rulebase structure.

4. Simulation of the constructed linguistic knowledge base.

5. Graphic display of the results.


To simplify the rule representation in Table (A.1), we define an observation (antecedent) plus a logic operator to be a logic cell, or simplified as a cell. In the same way, one action and one logic operator form a cell.

Figure (A-8) shows such a rule structure. The cell \( X_m \) is an observation and \( Y_n \) is an action. The lines between cells are logic operators that determine the tightness of the connection.

The \( i \)-th rule in table (A.1) then has the structure

<table>
<thead>
<tr>
<th>If</th>
<th>( X_1^i ) and</th>
<th>( X_2^i ) and</th>
<th>( \cdots ), ( X_m^i ) THEN</th>
<th>( Y_1^i ) and</th>
<th>( \cdots ), ( Y_n^i ) end</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell</td>
<td>Cell</td>
<td>Cell</td>
<td>Cell</td>
<td>Cell</td>
<td>Cell</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Condition</th>
<th>Action</th>
</tr>
</thead>
</table>

- 197
Some Definitions

1. **Logic Operator** is the connection method between two observations or two actions. The available logic operators are: \textit{AND}, \textit{OR}, \textit{NOT} and the pseudo logic operations are \textit{THEN}, \textit{END}. Mathematically, \textit{AND} and \textit{OR} logic operators are defined as a generalized pair of operators, \textit{T-norms} and \textit{T-conorms} which will be described in Section (A.1). Among them, the most widely used forms are \textit{max} and \textit{min}.

2. **Logic Cell** or **Cell** is a small logic unit consisting of an observation (action) and a logic operator. This is an example:

\[
\begin{array}{c|c|c}
\text{Season} & \text{IS} & \text{Normal} \\
\text{observation} & & \text{logic operator}
\end{array}
\]

3. **Condition** is all the preceding cells including the key word \textit{THEN} in the rule, sometimes called the \textit{left-hand-side} terms. A condition consists of several Cells:

\[Cell_1 \ Cell_2 \ \cdots \ Cell_m\]

4. **Action** is all the cells after the key word \textit{THEN} in the rule, sometimes called the \textit{right-hand-side} terms. An action consists of one or more Cells:

\[Cell_1 \ Cell_2 \ \cdots \ Cell_n\]

In the single-input single-output system, both condition and action parts have one cell. For a multiple-input, multiple-output system, the number of condition cells equals the number of inputs, and the number of action cells equals the number of outputs.
Object-Oriented LKB Programming

Due to the massive number of logic cells, fuzzy variables and rules, as well as the interface dialogues, the software package is relatively large. For such a large software package, we used the object-oriented concept for development, because of its cleanliness in structure and ease of maintenance.

Figure (A-9) shows the screen plot of the linguistic rule base construction. It has one rule base (R00), two inputs (Priority and Length) and one output (TrainLen). By clicking the mouse on the corresponding variable icons, another screen will be popped up for membership function adjustment as shown in Figure (A-10).

![Figure A-9: Linguistic Rulebase Construction](image)

Figure (A-11) shows the overall flow chart of the software.
Figure A-10: Membership Function Editor
Figure A-11: Overall Flow Chart of the Software FOGIC
A.4.2 FLP123 – Software for FLP

**FLP123** is a general purpose computer software for fuzzy linear programming, which is an extension of linear programming by introducing uncertainty of resource constraints, is written in the object-oriented language C++ and runs under Microsoft Windows. It includes a text editor which takes in FLP equations, and a graphic editor which edits fuzzy membership functions.

**Fuzzy Linear Programming**

Fuzzy linear programming is a simple form of fuzzy mathematical programming and is in the form of:

\[
\begin{align*}
\max_x & \quad \forall \quad \tilde{C}x \cong \tilde{g} \\
\text{subject to:} & \quad (\tilde{A}x)_i \leq \tilde{b}_i \quad i = 1, 2, \cdots m
\end{align*}
\] (A.41)

where

\[
\tilde{C}x = \sum_{j=1}^{n} \tilde{c}_j x_j
\]

and

\[
(\tilde{A}x)_i = \sum_{j=1}^{n} \tilde{a}_{ij} x_j.
\]

Degrees of constraint violations and goal satisfactions are measured by fuzzy vari-
ables which commonly have trapezoidal membership function:

\[
\mu_{\tilde{A}}(x) = \begin{cases} 
0.0 & \text{if } \tilde{A}X \leq A^l - \delta_A \\
1.0 - \frac{A^l - \tilde{A}X}{\delta_A} & \text{if } \tilde{A}X \in [A^l - \delta_A, A^l] \\
1.0 & \text{if } \tilde{A}X \in [A^l, A^r] \\
1.0 - \frac{\tilde{A}X - A^r}{\gamma_A} & \text{if } \tilde{A}X \in [A^r, A^r + \gamma_A] \\
0.0 & \text{if } \tilde{A}X \geq A^r + \gamma_A
\end{cases}
\] (A.42)

where \(A^l\) and \(A^r\) are lower and upper bounds. \(\delta_A\) and \(\gamma_A\) bracket the intervals of confidence for the left and right sides of \(\tilde{A}^l\) and \(\tilde{A}^r\) respectively.

\(\tilde{A}\) is symbolically written as \(\tilde{A} = (\delta_A, A^l, A^r, \gamma_A)\). For example, the linguistic term *train length around 60* is represented by a trapezoidal fuzzy variable as \(\tilde{A}\) may be symbolically written as \(\tilde{L}_{\text{train}} = (5, 60, 60, 5)\).

*FLP123* has the following features:

1. **Fuzzy Linear Programming Editor**

   Input form of fuzzy linear equations. Each constraint or objective function is terminated by a semicolon. Any text after the semicolon is regarded as a comment.

2. **Complete Solution of Fuzzy Linear Programming**

   Solution in terms of different degrees of satisfaction\(^2\). \(\mu_G(\alpha)\) is the degree of satisfaction of all constraints, and \(\mu_G(\alpha)\) is the degree of satisfaction of the goal.

   The overall degree of satisfaction takes the minimum of \(\mu_G(\alpha)\) and \(\mu_G(\alpha)\). The

\(^2\alpha = 1 - \theta.\)
rectangle at the bottom of Figure (A-14) shows the current $\alpha$ and objective value. The user has the opportunity to select a desired value of $\alpha$. Clicking the mouse button updates the solution vector at the right side of the window.

3. Membership Function Editor

Editor of fuzzy membership functions in graphics. Clicking and moving the mouse changes the shape of the membership function.

Example of a Transportation Problem

Since $FLP123$ is a general purpose fuzzy linear programming package, we will discuss an example of a transportation problem.

One company makes a forecast for the following season that it will purchase a certain amount of corn at three locations: Muncie, Ind., Brazil, Ind. and Xenia, Ohio. It also expects potential buyers at four different locations: Macon, Ga., Greenwood, S.C., Concord, S.C., and Chatham, N.C.

The supplies and demands can be estimated with uncertainties as characterized as fuzzy numbers in the table below.
All shipments must be routed through either Louisville or Cincinnati.

The transportation network is shown in Figure (A-12).

![Transportation Network Diagram]

Figure A-12: Transportation Network

The objective function for the fuzzy linear programming is related with a fuzzy number \( \tilde{G} \) determining the *admissible* total transportation cost. The membership function of the goal \( \tilde{G} \) is assumed to be in the form:

\[
\mu_G(x) = \begin{cases} 
1.0 & \text{if } f(x) \leq 40050 \\
\frac{45550-f(x)}{5500} & \text{if } f(x) \in [40050, 45550] \\
0.0 & \text{otherwise}
\end{cases}
\]  \hspace{1cm} (A.43)

<table>
<thead>
<tr>
<th>Location</th>
<th>Number of Rail Car Loads</th>
<th>Linguistic Terms</th>
<th>Forms in Fuzzy Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macon, Ga</td>
<td>Around 200</td>
<td>( b_1 ) = (50, 200, 200, 50)</td>
<td></td>
</tr>
<tr>
<td>Greenwood, S.C.</td>
<td>Around 400</td>
<td>( b_2 ) = (100, 400, 400, 100)</td>
<td></td>
</tr>
<tr>
<td>Concord, S.C.</td>
<td>Around 350</td>
<td>( b_3 ) = (90, 350, 350, 90)</td>
<td></td>
</tr>
<tr>
<td>Chatham, N.C.</td>
<td>Around 300</td>
<td>( b_4 ) = (80, 300, 300, 80)</td>
<td></td>
</tr>
</tbody>
</table>

Table A.3: Demands

205
The soft constraints are represented by a non-negative fuzzy membership function in trapezoidal form. To solve the above equation is equal to maximizing the degree of satisfaction $\mu_D$, with consideration of all the soft boundaries (all the goals and constraints). Figure (A-13) shows the editor input of the fuzzy linear equation and Figure (A-14) shows the solution.

![FLP Editor Input](image)

**Figure A-13: FLP Editor Input**

Figure (A-14) shows when $\alpha = 0.5$ (degree of overall satisfaction is 0.5) is selected, the solution is:
Figure A-14: FLP Solution

\[ x_{14} = 0 \quad x_{46} = 175 \]
\[ x_{24} = 500 \quad x_{47} = 325 \]
\[ x_{34} = 0 \quad x_{48} = 0 \]
\[ x_{15} = 250 \quad x_{49} = 0 \]
\[ x_{25} = 0 \quad x_{56} = 0 \]
\[ x_{35} = 450 \quad x_{57} = 25 \]
\[ x_{58} = 335 \]
\[ x_{59} = 340 \]
Appendix B

Recursive Least-squares Estimation

The least squares problem can be formulated in compact form. The cost function $J$ can be written as

$$J(\hat{W}) = \frac{1}{2} \hat{\varepsilon}^T \hat{\varepsilon} = \frac{1}{2} \| \hat{\varepsilon} \|^2$$

where

$$\hat{\varepsilon} = Y - \hat{Y} \quad \text{and} \quad \hat{Y} = \Phi \hat{W}$$

The solution to the above least-squares problem is given by

$$\Phi^T \Phi \hat{W} = \Phi^T Y \quad \text{(B.1)}$$

and

$$\hat{W} = (\Phi^T \Phi)^{-1} \Phi^T Y \quad \text{(B.2)}$$
In this way, the initial weight matrix $W$ is identified.

In many cases, observations are obtained sequentially. It may then be desirable to compute the weights for different values of $N$. If the least-squares problem has been solved for $N$ observations, it is a waste of computational resources to start from scratch when a new observation is obtained. Hence, it is desirable to arrange the computations in such a way that the results obtained from $N$ observations can be used in order to get the estimates for $N + 1$ observations.

Recursive equations can be derived for the case when the observations are obtained sequentially. Let $\hat{W}(N)$ denote the least-squares estimate based on $N$ measurements. To derive the equations, $N$ is introduced as a formal parameter in the functions:

$$
\Phi(N) = \begin{bmatrix}
\bar{R}^T(x_1) \\
\bar{R}^T(x_2) \\
. \\
. \\
. \\
\bar{R}^T(x_N)
\end{bmatrix} \quad Y(N') := \begin{bmatrix}
y_1 - (\bar{R}_0)_1 \\
y_2 - (\bar{R}_0)_2 \\
. \\
. \\
. \\
y_N - (\bar{R}_0)_N
\end{bmatrix} \quad (B.3)
$$

It is assumed that the matrix $\Phi^T\Phi$ is regular for all $N$. The least-squares estimate $\hat{W}(N)$ is then given by equation (B.2)

$$
\hat{W}(N) = [\Phi^T(N)\Phi(N)]^{-1}\Phi^T(N)Y(N) \quad (B.4)
$$

When an additional measurement is obtained, a row is added to the matrix $\Phi$ and an element is added to the vector $Y$. Hence
\[
\Phi(N + 1) = \begin{bmatrix} \Phi(N) \\ \bar{R}^T(x_{N+1}) \end{bmatrix} \quad Y(N + 1) = \begin{bmatrix} Y(N) \\ y_{N+1} \end{bmatrix} \quad (B.5)
\]

The estimate \( \hat{W}(N + 1) \) given by (B.4) can be written as

\[
\hat{W}(N + 1) = [\Phi^T(N + 1)\Phi(N + 1)]^{-1}\Phi^T(N + 1)Y(N + 1)
\]

\[
= [\Phi^T(N)\Phi(N) + \bar{R}(N + 1)\bar{R}^T(N + 1)]^{-1} \times [\Phi^T(N)Y(N) + \bar{R}(N + 1)y_{N+1}]
\]

The recursive solution is given by the following theorem.

**Theorem - Recursive least-square estimation.** When the matrix \( \Phi^T(N)\Phi(N) \) is positive definite. The least-squares estimate \( \hat{W} \) is then given by the recursive equation

\[
\hat{W}(N + 1) = \hat{W}(N) + K(N)[y_{N+1} - \bar{R}^T(N + 1)\hat{W}(N)] \quad (B.6)
\]

where

\[
K(N) = P(N + 1)\bar{R}(N + 1) =
\]

\[
= P(N)\bar{R}(N + 1)[1 + \bar{R}^T(N + 1)P(N)\bar{R}(N + 1)]^{-1} \quad (B.7)
\]

and

\[
P(N + 1) = [I - K(N)\bar{R}^T(N + 1)]P(N) \quad (B.8)
\]

To prove the theorem, The following lemma is useful

**lemma - Matrix inversion lemma.**
Let $A, C$ and $C^{-1} + DA^{-1}B$ be nonsingular square matrices; then

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \quad (B.9)$$

**Proof.** By direct substitution,

$$[A + BCD][A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}]$$

$$= I + BCDA^{-1} - B[C^{-1} + DA^{-1}B]^{-1}DA^{-1} - BCD A^{-1}B[C^{-1} + DA^{-1}B]^{-1}DA^{-1}$$

$$= I + BCDA^{-1} - BC[C^{-1} - DA^{-1}B][C^{-1} + DA^{-1}B]^{-1}DA^{-1}$$

$$= I + BCDA^{-1} - BCD A^{-1} = I$$

**Proof of Theorem.** To simplify the notation in the manipulations that follow, the argument $N$ of $\Phi(N)$ and $Y(N)$ and the argument $N + 1$ of $\tilde{R}T(N + 1)$ will be suppressed. Equation (B.4) can then be written as

$$\hat{W}(N + 1) = [\Phi^T \Phi + \tilde{R}\tilde{R}^T]^{-1}[\Phi^TY + \tilde{R}y_{N+1}]$$

$$= (\Phi^T \Phi)^{-1}\Phi^TY + [(\Phi^T \Phi + \tilde{R}\tilde{R}^T)^{-1} - (\Phi^T \Phi)^{-1}]\Phi^TY +$$

$$(\Phi^T \Phi + \tilde{R}\tilde{R}^T)^{-1}\tilde{R}y_{N+1}. \quad (B.10)$$

Observe that

$$\hat{W}(N) = (\Phi^T \Phi)^{-1}\Phi^TY$$

211
and

$$[(\Phi^T \Phi + \bar{R} \bar{R}^T)^{-1} - (\Phi^T \Phi)^{-1}] \Phi^T Y = (\Phi^T \Phi + \bar{R} \bar{R}^T)^{-1}(\Phi^T \Phi - \Phi^T \Phi - \bar{R} \bar{R}^T)(\Phi^T \Phi)^{-1} \Phi^T Y$$

$$= - (\Phi^T \Phi + \bar{R} \bar{R}^T)^{-1} \bar{R} \bar{R}^T (\Phi^T \Phi)^{-1} \Phi^T Y = - (\Phi^T \Phi + \bar{R} \bar{R}^T)^{-1} \bar{R} \bar{R}^T \hat{W}.$$ 

Equation (B.10) can be written as

$$\hat{W}(N + 1) = \hat{W}(N) + K(N)[y_{N+1} - \bar{R}^T(N + 1)\hat{W}(N)] \quad (B.11)$$

where

$$K(N) = [\Phi^T(N) \Phi(N) + \bar{R}(N+1) \bar{R}^T(N+1)]^{-1} \bar{R}(N+1) = [\Phi^T(N+1) \Phi(N+1)]^{-1} \bar{R}(N+1).$$

In order to obtain a recursive equation for the weighting factor $K(N)$, it is convenient to introduce the quantity $P$ defined by

$$P(N) = [\Phi^T(N) \Phi(N)]^{-1}$$

$P$ is proportional to the variance of the estimates. Applying Lemma (B.9) to the matrix $P(N + 1)$ gives

$$P(N + 1) = [\Phi^T(N + 1) \Phi(N + 1)]^{-1} = [\Phi^T \Phi + \bar{R} \bar{R}^T]^{-1}$$

$$= (\Phi^T \Phi)^{-1} - (\Phi^T \Phi)^{-1} \bar{R}[I + \bar{R}^T(\Phi^T \Phi \bar{R})^{-1} \bar{R}]^{-1} \bar{R}^T(\Phi^T \Phi)^{-1}$$

212
Hence

\[ P(N + 1) = P(N) - P(N)\bar{R}(N + 1) \times \]

\[ [I + \bar{R}^T(N + 1)P(N)\bar{R}(N + 1)]^{-1}\bar{R}^T(N + 1)P(N) \]  \hspace{1cm} (B.12)

A simple calculation now gives

\[ K(N) = P(N + 1)\bar{R}(N + 1) = P(N)\bar{R}(N + 1)[I + \bar{R}^T(N + 1)P(N)\bar{R}(N + 1)]^{-1} \]

Notice that a matrix inversion is necessary to compute \( P \). However, the matrix to be inverted is of the same dimension as the number of measurements, i.e., for a single-output system, it is a scalar.

Equation (B.6) has a strong intuitive appeal. The estimate \( \hat{W}(N + 1) \) is obtained by adding a correction to the previous estimate \( \hat{W}(N) \). The correction is proportional to \( y_{N+1} - \bar{R}^T(N + 1)\hat{W}(N) \), where the last term can be interpreted as the value of \( y \) at step \( N + 1 \) predicted by the model (4.19). The correction term is thus proportional to the difference between the measured value of \( y_{N+1} \) and the predication of \( y_{N+1} \) based on the previous estimates of the parameters. The components of the vector \( K(N) \) are weighting factors that tell how the correction and the previous estimate should be combined. Notice that the component \( K_i(N) \) is proportional to \( \bar{R}_i^T(N + 1) \).

The least-squares estimate can be interpreted as a Kalman filter for the process

\[ \hat{W}(k + 1) = \hat{W}(k) \]

\[ y(k) = \bar{R}^T(k)\hat{W}(k) + \epsilon(k) \]
Notice that the matrix $P(N)$ is defined only when the matrix $\Phi^T(N)\Phi(N)$ is nonsingular. Because

$$\Phi^T(N)\Phi(N) = \sum_{k=1}^{N} \tilde{R}(k)\tilde{R}^T(k)$$

it follows that $\Phi^T(N)\Phi(N)$ is always singular if $N$ is sufficiently small. In order to obtain an initial condition for $P$, it is necessary to choose an $N = N_0$ such that $\Phi^T(N_0)\Phi(N_0)$ is nonsingular and determine

$$P(N_0) = [\Phi^T(N_0)\Phi(N_0)]^{-1} \quad (B.13)$$

$$\hat{W}(N_0) = P(N_0)\Phi^T(N_0)y(N_0) \quad (B.14)$$

The recursive equations can then be used from $N \leq N_0$. It is, however, often convenient to use the recursive equations in all steps. If the recursive equations begin with the initial condition

$$P(0) = P_0$$

where $P_0$ is positive definite, then

$$P(N) = [P_0^{-1} + \Phi^T(N)\Phi(N)]^{-1} \quad (B.15)$$

This can be made arbitrarily close to $[\Phi^T(N)\Phi(N)]^{-1}$ by choosing a $P_0$ which sufficiently large.
Bibliography


decide. In Proceedings of 1982 IEEE International Conference on Cybernetics and Society,
82-CH-1840-8, Seattle, WA. 1982.

[14] Chammas, E. A Fuzzy Logic Approach to the Active Control of Room Air-Temperature


[16] Chang, S. S. L. On risk and decision making in a fuzzy environment. In Fuzzy
Sets and Their Applications to Cognitive and Decision Process (Zadeh, L. A. and Fu, K. S.

[17] Chanias, S. Parametric programming in fuzzy linear programming. In Fuzzy Sets

[18] Chanias, S. Fuzzy programming in multiobjective linear programming – a para-

[19] Cheeseman, P.C. An inquiry into computer understanding. In Proc. of Compu-

[20] Chih, K.C., and Hornung, M.A. Implementation of a real time locomotive dis-
tribution system. In Computer Applications in Railway Planning and Management,

[21] Crainic, T. G. Operation research models of intercity freight transportation:
The current state and future research issues. In Logist. Trans. Rev., V23, p189-

[22] Crainic, T. G. Rail tactical planning: issues, models and tools. In Freight Trans-
port Planning and Logistics, L. Bianco and A. La Bella (eds.), p463-509, Springer–Verlag,

[23] Delgado, M., Verdegay, J.L. and Vila, M.A. Imprecise costs in mathematical


[26] Dinola, A., Sessa, S., Pedrycz, W. and Sanchez, E. Fuzzy Relation Equations and

[27] Drainkon, D. Inference with a single fuzzy conditional propositions. In Fuzzy


[101] Zadeh, L. The concept of a linguistic variable and its application to approximate reasoning (I); (II); (III). In Infromation Sciences, 8, p199-249, 1975.


