TRAFFIC CONTROL AND TRANSPORT PLANNING: A Fuzzy Sets and Neural Networks Approach

by

Dušan Teodorović
Katarina Vukadinović

INTERNATIONAL SERIES IN INTELLIGENT TECHNOLOGIES

TRAFFIC CONTROL AND TRANSPORT PLANNING:

A Fuzzy Sets and Neural Networks Approach

INTERNATIONAL SERIES IN INTELLIGENT TECHNOLOGIES

Prof. Dr. Dr. h.c. Hans-Jürgen Zimmermann, Editor
European Laboratory for Intelligent
Techniques Engineering
Aachen, Germany

Other books in the series:

Fuzzy Logic and Intelligent Systems
edited by Hua Li and Madan Gupta

Fuzzy Set Theory and Advanced Mathematical Applications
edited by Da Ruan

Fuzzy Databases: Principles and Applications
by Frederick E. Petry with Patrick Bose

Distributed Fuzzy Control of Multivariable Systems
by Alexander Gegov

Fuzzy Modelling: Paradigms and Practices
by Witold Pedrycz

Fuzzy Logic Foundations and Industrial Applications
by Da Ruan

Fuzzy Sets in Engineering Design and Configuration
by Hans-Juergen Sebastian and Erik K. Antonsson

Consensus Under Fuzziness
by Mario Fedrizzi, Janusz Kacprzyk, and Hannu Nurmi

Uncertainty Analysis in Enginerring Sciences: Fuzzy Logic, Statistices, and Neural Network Approach
by Bilal M. Ayyub and Madan M. Gupta

Fuzzy Modeling for Control
Robert Babuška

TRAFFIC CONTROL AND TRANSPORT PLANNING:

A Fuzzy Sets and Neural Networks Approach

Dušan Teodorović, Ph.D.
Katarina Vukadinović

Faculty of Transport and Traffic Engineering
University of Belgrade
Belgrade, Yugoslavia

SPRINGER SCIENCE+BUSINESS MEDIA, LLC

ISBN 978-94-010-5892-6 ISBN 978-94-011-4403-2 (eBook)
DOI 10.1007/978-94-011-4403-2

Library of Congress Cataloging-in-Publication Data

Printed on acid-free paper.

CONTENTS

Preface

When solving real-life engineering problems, linguistic information is often encountered that is frequently hard to quantify using "classical" mathematical techniques. This linguistic information represents subjective knowledge. Through the assumptions made by the analyst when forming the mathematical model, the linguistic information is often ignored. On the other hand, a wide range of traffic and transportation engineering parameters are characterized by uncertainty, subjectivity, imprecision, and ambiguity. Human operators, dispatchers, drivers, and passengers use this subjective knowledge or linguistic information on a daily basis when making decisions. Decisions about route choice, mode of transportation, most suitable departure time, or dispatching trucks are made by drivers, passengers, or dispatchers. In each case the decision maker is a human. The environment in which a human expert (human controller) makes decisions is most often complex, making it difficult to formulate a suitable mathematical model. Thus, the development of fuzzy logic systems seems justified in such situations. In certain situations we accept linguistic information much more easily than numerical information. In the same vein, we are perfectly capable of accepting approximate numerical values and making decisions based on them. In a great number of cases we use approximate numerical values exclusively. It should be emphasized that the subjective estimates of different traffic parameters differs from dispatcher to dispatcher, driver to driver, and passenger to passenger. When solving real-life traffic and transportation problems we should not use only objective knowledge (formulas and equations) or only subjective knowledge (linguistic information). We simply cannot and should not ignore the existence of linguistic information that is, subjective knowledge. Fuzzy logic is an extremely suitable concept that combines subjective knowledge and objective knowledge.

Basic results linked to the development of fuzzy sets date from Zadeh (1965). Many sets encountered in reality do not have precisely defined bounds that separate the elements within the set from those outside the set. Thus, it might be said that a certain stoplight has a "long" waiting time. If we denote by A the set of "long waiting time at a stoplight," the question logically arises as to the bounds of such a defined set. Does a waiting time of 25 seconds belong to this set? What about 15 seconds or 60 seconds? The answers to these questions are always positive. All these times belong to the set called "long waiting time at a stoplight." Other members of this set are waiting times equal to 22 seconds, 37 seconds, 18 seconds, and so on. On the other hand, we intuitively feel that a waiting time of 60 seconds belongs to the set called "long waiting time at a stoplight" "more" or "stronger" than a waiting time of 22 seconds. In other words, there is more truth in the statement that a waiting time of 60 seconds is a "long waiting time at a stoplight" than in the statement that a waiting time of 22 seconds is a "long waiting time at a stoplight." Within the context of this simple example we can fully appreciate Bart Kosko's observation that "everything is a matter of degree." All waiting times at a stoplight can be treated as long. If we introduce a set called "short waiting time at a stoplight," we note that all waiting times at a stoplight can also be treated as short. Finally, we can ask ourselves whether a waiting time of 15 seconds is long, short, or perhaps medium. The answer is very simple. A waiting time of 15 seconds is long, short, and medium, all at the same time. In other words, a waiting time of 15 seconds belongs to the sets "long waiting time," "medium waiting time," and "short waiting time" with different grades of membership.

Basic results linked to the development of fuzzy logic date from Zadeh (1973) and Mamdani and Assilian (1975). Introducing a concept he called *approximate reasoning*, Zadeh successfully showed that vague logical statements enable the formation of algorithms that can use vague data to derive vague inferences. Zadeh assumed his approach would be beneficial above all in the study of complex humanistic systems. Realizing that Zadeh's approach could be successfully applied to traffic controllers Pappis and Mamdani (1977) were first to use the principles of fuzzy logic to control the isolated intersection of two one-way streets.

In the mid and late 1980s, a group of authors made a significant contribution to fuzzy set theory applications in traffic and transportation. Nakatsuyama et al. (1983), Sugeno and Murakami (1985), Sugeno and Nishida (1985), and particularly Sasaki and Akiyama (1986, 1987, 1988) solved complex traffic and transportation problems indicating the great potential of using fuzzy set theory techniques. At the end of the 1980s and beginning of the 1990s, the fuzzy set theory in traffic and transportation became extensively used. The pioneer work of the research team of the University of Delaware deserves special attention, headed by Professor Shinya Kikuchi (Chakroborthy, 1990; Chakroborthy and Kikuchi, 1990; Perincherry, 1990; Perincherry and Kikuchi, 1990; Teodorovic and Kikuchi, 1990, 1991a; Kikuchi et al., 1991). At the beginning and mid-1990s, interest in fuzzy logic applications in traffic and transportation increased in other world universities. Different traffic and transportation problems successfully solved using fuzzy set theory techniques were presented in the works of Chen et al. (1990), Tzeng and Teng (1993), Lotan and Koutsopoulos (1993a, 1993b), Xu and Chan (1993a, 1993b), Teodorovic and Babic (1993), Akiyama and Shao (1993), Chang and Shyu (1993), Chanas et al. (1993), Vukadinovic and Teodorovic (1994), Teodorovic et al. (1994), Teodorovic (1994), Teodorovic and Kalic (1995), Milosavljevic et al. (1996), Teodorovic and Pavkovic (1996), Teodorovic and Lucic (1998) and Tzeng et al. (1996).

People relatively easily perform a variety of complex tasks that are highly difficult to solve by computational techniques of traditional algorithms. The brain architecture largely differs from common serial computers, and the researchers of artificial neural networks seek to endow these machines with the abilities of data processing similar to those of a human. Artificial neural networks are inspired by biology that is; they are composed of the elements that function similarly to a biological neuron. These elements are organized in a way that is reminiscent of the anatomy of a brain. In addition to this superficial similarity, artificial neural networks display a striking number of the brain's properties. For example, they are able to learn from experience, to apply to new cases generalizations derived from previous instances, to abstract essential characteristics of input data that often contain irrelevant information. Despite these functional similarities, even the most optimistic advocates could not claim that artificial neural networks will soon completely mimic the functions of a human brain. However, it is equally wrong to neglect the performance of certain artificial neural networks that staggeringly resembles that of a brain. These abilities, however limited they may be, indicate that a deeper understanding of human intelligence may be at hand accompanied by a number of revolutionary changes.

In the 1990s over fifty papers were published applying neural network models to transportation problems (Dougherty, 1995). Most of these papers are concerned with road transportation. Yang et al. (1992) and Dougherty and Joint (1992) modeled the driver's behavior when making strategic and instinctive decisions. When the developed networks were applied to new data, they produced good results more quickly and accurately than alternative techniques like logit models. Many authors study the simulation of the driver's behavior while driving a vehicle. For example, Hunt and Lyons (1994) use neural networks to model the driver's behavior while changing the speed and lane on a highway. Ritchie and Cheu (1993) use neural networks to detect long-lasting traffic jams (traffic accidents, broken vehicles, dropped cargo, maintenance, constructions). Neural networks are used to analyze congestions of urban travel networks and seasonal fluctuations in the vehicle flow (Hua and Faghri, 1993), to estimate O-D matrices (Kikuchi et al., 1993), and to evaluate travel network improvement (Wei and Schonfeld, 1993). In predicting the vehicle flow (over a long period or immediately after the observed moment) a number of authors present very good results (Chin et al., 1992; Doughety et al., 1994). The applications of neural networks in transportation have reached the stage in which all these important investigations should be applied to real transportation systems.

The primary goal of this book is to acquaint the reader with the basic elements of the fuzzy set theory, fuzzy logic, fuzzy logic systems, artificial neural networks, neurofuzzy modeling, and applications of fuzzy logic and neural networks to date in traffic and transportation engineering, and to indicate the directions for future research in this area.

Belgrade

Dušan Teodorović
Katarina Vukadinović

Acknowledgments

This book presents a fuzzy and neuro approach to modeling complex problems of traffic control and transport planning. We started to be interested in the amazing world of fuzzy sets and systems and neural networks when the first author was a visiting professor in the United States at the University of Delaware in 1989. At that time, the researchers from the University of Delaware headed by Professor Shinya Kikuchi began to study the possibilities of fuzzy set theory applications in traffic and transportation. A few years later, the second author also spent two years working with the University of Delaware investigators. Both of us greatly benefited from our collaboration with the University of Delaware, professionally as well as personally. We want to use this opportunity to give our thanks to Professor Shinya Kikuchi, Dr. Vijay Perincherry, Dr. Partha Chakroborthy, and many others for valuable discussions, essential critiques, unselfish cooperation, and the enthusiastic spirit they shared with us. We are also very thankful to Dr. David Gillingwater, the editor-in-chief of the *Transportation Planning and Technology Journal*, who gave the opportunity to Professors Shinya Kikuchi and Dusan Teodorovic to be the guest editors of the special issue of this journal devoted to the fuzzy set theory applications in traffic and transportation published in 1993. Our greatest thanks go to Professor Hans-Jurgen Zimmermann, who believed in us and who encouraged and enabled us to become the authors in this prestigious series. We also give our gratitude to Zachary Rolnik, vice president of Kluwer Academic Publishers, for his continuos support. The greatest part of this book is the result of investigation conducted by the members of a small and impassioned research unit within the Faculty of Transport and Traffic Engineering, University of Belgrade. We would like to thank them all for making the significant contributions to this book: Ph.D. student Goran Pavkovic, who coauthored a few papers used in the book and was an indispensable computer whiz in our group, and Miomir Segovic, our graduate student who was of great help in providing figures.

And last, but not the least, our deepest thanks we give to members of our families Smilja, Jelena, and Ljubica Teodorovic and Mihailo and Zoran Ristic, who helped us with continuous support, encouragement, and understanding for the time we took from them and invested in creating this book.

About the Authors

Dušan Teodorović received a B.S. degree in transportation engineering at the University of Belgrade in 1973. He joined Yugoslav Airlines and worked as a traffic engineer. In 1974 he joined University of Belgrade and worked in the areas of transportation planning, transportation networks, transportation modeling, air transportation, operations research, and fuzzy and neuro modeling. Dr. Teodorović received M.S. and Ph.D. degrees in transportation engineering in 1976 and 1982, respectively. He is currently professor of transportation engineering and operations research in the Faculty of Transport and Traffic engineering at the University of Belgrade. He was a member of the Program Committees of several conferences. Dr. Teodorović served as a cochair of the Ninth Mini EURO Conference on Fuzzy Sets in Traffic and Transport Systems held in Budva in 1997. He has published widely in international journals. Dr. Teodorović is the author of the books *Transportation Networks* (Gordon and Breach Science Publishers, 1986) and *Airline Operations Research* (Gordon and Breach Science Publishers, 1988). He has served as an associate editor of *Yugoslav Journal of Operations Research* and editorial board member of *Journal of Transportation Engineering* (ASCE). Dr. Teodorović was visiting professor at the Technical University of Denmark, Department of Civil Engineering; Operations Research Program at the University of Delaware (U.S.A); and National Chiao Tung University, Taiwan.

Katarina Vukadinović received a B.S. degree in transportation engineering at the University of Belgrade in 1985. In 1986 she joined the civil engineering firm that performed inland water transport of bulk freight. In 1988 she joined the Faculty of Transport and Traffic Engineering at the University of Belgrade and received M.S. and Ph.D. degrees in transportation engineering in 1993 and 1997, respectively. Dr. Vukadinović worked in the areas of port planning and development, inland water transportation, traffic engineering, operations research, fuzzy sets, and neural networks applications in transportation. Dr. Vukadinović spent a few months with Road Data Laboratory and the Institute for Roads, Transport, and Town planning at the Technical University of Denmark and two years with the University of Delaware. She is currently assistant professor of Transportation Engineering and Operations Research in the Faculty of Transport and Traffic engineering at the University of Belgrade. She was a member of the Program Committee of the Ninth Mini EURO Conference on Fuzzy Sets in Traffic and Transport Systems held in Budva in 1997.

Chapter 1.

Basic Definitions of the Fuzzy Sets Theory

1.1. THE CONCEPT OF FUZZY SETS

In the classic theory of sets, very precise bounds separate the elements that belong to a certain set from the elements outside the set. In other words, it is quite easy to determine whether an element belongs to a set or not. For example, if we denote by A the set of signalized intersections in a city, we conclude that every intersection under observation belongs to set A if it has a signal. Element x's membership in set A is described in the classic theory of sets by the membership function $\mu_A(x)$, as follows:

$$\mu_A(x) = \begin{cases} 1, & \text{if and only if } x \text{ is member of A} \\ 0, & \text{if and only if } x \text{ is not member of A} \end{cases} \tag{1.1}$$

Figure 1.1 presents set A and elements x, y, and z.

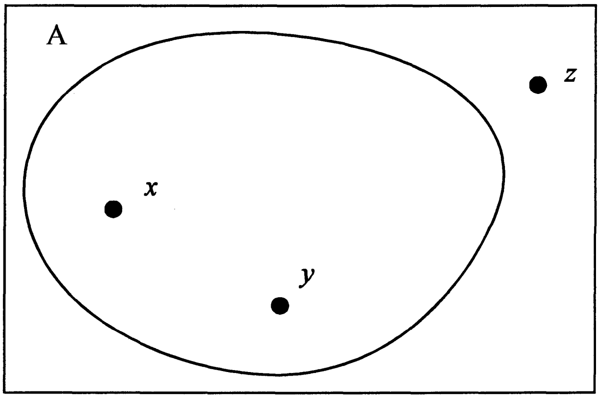

Figure 1.1. Set A and elements x, y, and z

It is clear from *Figure 1.1* that $\mu_A(x) = 1$, $\mu_A(y) = 1$, and $\mu_A(z) = 0$.

D. Teodorovič et al., *Traffic Control and Transport Planning:*

Many sets encountered in reality do not have precisely defined bounds that separate the elements in the set from those outside the set. Thus, it might be said that waiting time of a vehicle at a certain signal is "long." If we denote by A the set of "long waiting time at a signal," the question logically arises as to the bounds of such a defined set. In other words, we must establish which element belongs to this set. Does a waiting time of 25 seconds belong to this set? What about 15 seconds or 90 seconds? The air traffic between two cities can be described as having "high flight frequency." Do flight frequencies of five flights a day, eight flights a day, three flights a day belong to "high flight frequency" category? Travel time between origin and destination is usually subjectively estimated as "short," "not too long," "long," "medium," "about twenty minutes," "around half an hour," and so on. Does a travel time of 40 minutes, 25 minutes, or 8 minutes belong to the set called "travel time of around half an hour"? We intuitively know that a travel time of 25 minutes belongs to the set called "travel time of around half an hour" "more" or "stronger" than a travel time of 8 minutes. In other words, there is more truth in the statement that travel time of 25 minutes is "travel time of around half an hour" than in the statement that travel time of 8 minutes is "travel time of around half an hour."

The membership function for fuzzy sets can take any value from the closed interval [0,1]. Fuzzy set **A** is defined as the set of ordered pairs $\mathbf{A} = \{x, \mu_{\mathbf{A}}(x)\}$, where $\mu_{\mathbf{A}}(x)$ is the grade of membership of element x in set **A**. The greater $\mu_{\mathbf{A}}(x)$, the greater the truth of the statement that element x belongs to set **A**.

Let us denote by $X = \{x_1, x_2, ..., x_3\}$ the finite discrete set of elements x_i, $i = 1, 2, ..., n$. Set X can also be shown in the form

$$X = x_1 + x_2 + \ldots + x_n = \sum_{i=1}^{n} x_i \tag{1.2}$$

where the sign + denotes the union of the elements. Set X is referred to as the "universe of discourse," and it may contain either discrete or continuous values (elements).

Fuzzy set **A** defined over set X is most often shown in the form

$$\mathbf{A} = \frac{\mu_{\mathrm{A}}(x_1)}{x_1} + \frac{\mu_{\mathrm{A}}(x_2)}{x_2} + \ldots + \frac{\mu_{\mathrm{A}}(x_n)}{x_n} = \sum_{i=1}^{n} \frac{\mu_{\mathrm{A}}(x_i)}{x_i} \tag{1.3}$$

When X is a continuous and not a finite set, fuzzy set **A** defined over set X is expressed as

$$\mathbf{A} = \int_{x} \frac{\mu_{\mathrm{A}}(x)}{x} \tag{1.4}$$

where the integration sign represents the union of the elements.

Example 1.1. Let us note set $\mathrm{X} = \{2, 5, 9, 18, 21, 25\}$, whose elements denote the number of vehicles waiting in line at a signal. Set **B** consists of the fuzzy set "small number of vehicles in line." Fuzzy set **B** can be shown as

$$\mathbf{B} = \frac{0.95}{2} + \frac{0.55}{5} + \frac{0.20}{9} + \frac{0.10}{18} + \frac{0.05}{21} + \frac{0.01}{25} \tag{1.5}$$

The grades of membership $0.95, 0.55, \ldots, 0.01$ are subjectively determined and indicate the "strength" of membership of individual elements in fuzzy set **B**. For example, 2 with a grade of membership of 0.95 belongs to fuzzy set **B**, which comprises a "small number of vehicles in line" at the signal.

Fuzzy sets are often defined through membership functions to the effect that every element is allotted a corresponding grade of membership in the fuzzy set. Let us note fuzzy set **C**. The membership function that determines the grades of membership of individual elements x in fuzzy set **C** must satisfy the following inequality:

$$0 \leq \mu_{\mathrm{C}}(x) \leq 1 \qquad \forall x \in \mathrm{X} \tag{1.6}$$

Example 1.2. Let us note fuzzy set **A**, which is defined as "travel time is approximately 30 minutes." Membership function $\mu_{\mathrm{A}}(t)$, which is subjectively determined is shown in *Figure 1.2.*

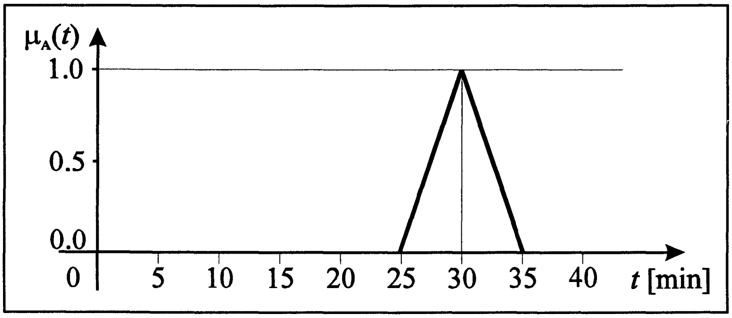

Figure 1.2. Membership function $\mu_{\mathrm{A}}(t)$ of fuzzy set **A**

In this case, we have subjectively estimated that travel time between the two points can be within the limits of 25 to 35 minutes. A travel time of 30 minutes has a grade of membership of 1 and belongs to the set "travel time is approximately 30 minutes." All travel times within the interval of 25 to 35 minutes are also members of this set because their grades of membership are greater than zero. Travel times outside this interval have grades of membership equal to zero.

1.2. THE EQUALITY OF FUZZY SETS

Let us note fuzzy sets **A** and **B** defined over set X. Fuzzy sets **A** and **B** are equal (**A** = **B**) if and only if $\mu_{\mathbf{A}}(x) = \mu_{\mathbf{B}}(x)$ for all elements of set X.

1.3. SUBSETS OF FUZZY SETS

Fuzzy set **A** is a subset of fuzzy set **B** if and only if $\mu_{\mathbf{A}}(x) \le \mu_{\mathbf{B}}(x)$ for all elements x of set X. In other words, $\mathbf{A} \subset \mathbf{B}$ if, for every x, the grade of membership in fuzzy set **A** is less than or equal to the grade of membership in fuzzy set **B**.

Example 1.3. Figure 1.3 presents the membership functions of the fuzzy sets "long" and "very long" travel times.

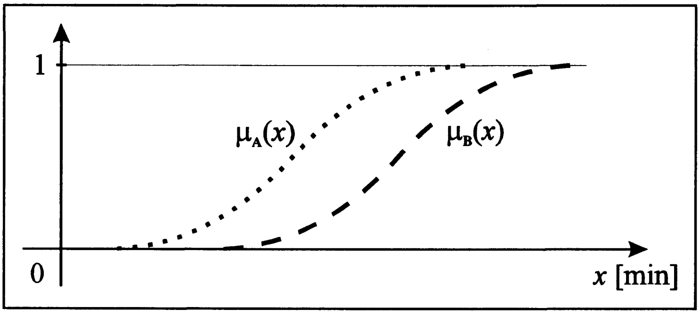

Figure 1.3. Membership functions of fuzzy sets "long" and "very long" travel times

We denote by **A** and **B**, respectively, the sets of "long" and "very long" travel times. The fuzzy set "very long" travel time is a subset of the fuzzy set "long" travel time since the following relation is satisfied for every x:

$$\mu_{\mathrm{B}}(x) \le \mu_{\mathrm{A}}(x) \tag{1.7}$$

1.4. THE INTERSECTION OF FUZZY SETS

The intersection of fuzzy sets **A** and **B** is denoted by **A** ∩ **B** and is defined as the largest fuzzy set contained in both fuzzy sets **A** and **B**. The intersection corresponds to the operation "and."

Membership function $\mu_{\mathbf{A}\cap\mathbf{B}}(x)$ of the intersection **A** ∩ **B** is defined as follows:

$$\mu_{\mathbf{A}\cap\mathbf{B}}(x) = \min\left\{\mu_{\mathbf{A}}(x), \mu_{\mathbf{B}}(x)\right\} \tag{1.8}$$

The symbol $\wedge$ is often used instead of the symbol *min*. *Figure 1.4* presents the membership functions of sets **A, B,** and **A** ∩ **B**.

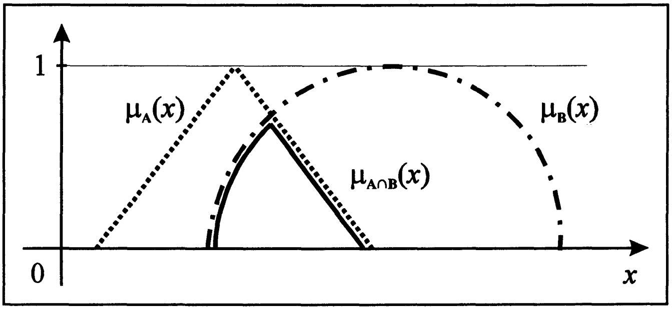

Figure 1.4. Membership functions of fuzzy sets **A**, **B**, and **A** ∩ **B**

Example 1.4. Visibility on airport runways and the height of the cloud base are measured in meters. Visibility can be "good," "medium," or "poor." The cloud base can be "low" or "high."

We note fuzzy sets **A** and **B**, which are defined as follows: **A** defines "poor visibility on airport runways," and **B** defines "high cloud base." Fuzzy set **A** ∩ **B** denotes "poor visibility on airport runways and high cloud base." The membership functions of fuzzy sets **A**, **B**, and **A** ∩ **B** are shown in *Figure 1.5*.

In addition to min operator, there are other operators, under the name *T-norm* operators, frequently used in describing the intersection of two fuzzy sets, such as the *algebraic product* and the *bounded product*.

The algebraic product of fuzzy set **A** and fuzzy set **B** is denoted as **A·B**. The membership function $\mu_{\mathbf{A}\cdot\mathbf{B}}(x)$ of the algebraic product is calculated as

$$\mu_{\mathbf{A}\cdot\mathbf{B}}(x) = \mu_{\mathbf{A}}(x)\cdot\mu_{\mathbf{B}}(x) \tag{1.9}$$

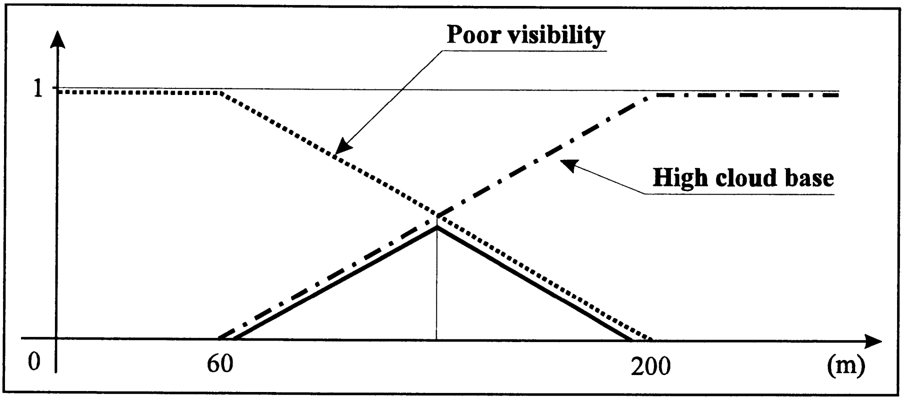

Figure 1.5. Membership functions of the fuzzy sets "poor visibility," "high cloud base," and "poor visibility and high cloud base"

The bounded product of fuzzy sets **A** and **B** is denoted as **A(·)B**. The corresponding membership function is given by

$$\mu_{A(\cdot)B}(x) = \max\{0, \mu_A(x) + \mu_B(x) - 1\}. \tag{1.10}$$

Besides these, there are operators used in describing the intersection of two fuzzy sets characterized by a particular parameter, the task of which is to indicate the strength of "and." These operators, which represent a parameterized intersection, include Hamacher's min operator, Yager's min operator, and Dubois and Prade's min operator. Let us clarify the concept of a parameterized intersection by the example of Dubois and Prade's min operator.

Dubois and Prade (1984) defined their min operator in the following way:

$$\mu_{A\lambda B}(x) = \frac{\mu_A(x) \cdot \mu_B(x)}{\max(\mu_A(x), \mu_B(x), \lambda)}, \quad \lambda \in [0, 1] \tag{1.11}$$

It is easily shown that for λ=0 and λ=1, respectively,

$$\mu_{A\lambda B}(x) = \min\{\mu_A(x), \mu_B(x)\} \tag{1.12}$$

$$\mu_{A\lambda B}(x) = \mu_A(x) \cdot \mu_B(x) \tag{1.13}$$

Different values of parameter λ are corresponded by different strengths of demand for fuzzy set **A** "and" fuzzy set **B**. Thus, for example, the choice of route in urban traffic between a number of alternative routes is influenced by a great variety of factors, of which travel time and total length of route are certainly among the most important. Different values of λ are corresponded by different strengths of a driver's (decision maker's) desire to select the route that has a low travel time "and" that is short.

1.5. THE UNION OF FUZZY SETS

The union of fuzzy sets **A** and **B** is denoted by $\mathbf{A} \cup \mathbf{B}$ and is defined as the smallest fuzzy set that contains both fuzzy set **A** and fuzzy set **B**. The membership function $\mu_{\mathbf{A}\cup\mathbf{B}}(x)$ of the union $\mathbf{A} \cup \mathbf{B}$ of fuzzy sets **A** and **B** is defined as follows:

$$\mu_{\mathbf{A}\cup\mathbf{B}}(x) = \max\left\{\mu_{\mathbf{A}}(x), \mu_{\mathbf{B}}(x)\right\} \tag{1.14}$$

The symbol $\vee$ is often used instead of the symbol *max*. The union corresponds to the operation "or." *Figure 1.6* presents the membership functions of sets **A**, **B**, and $\mathbf{A} \cup \mathbf{B}$.

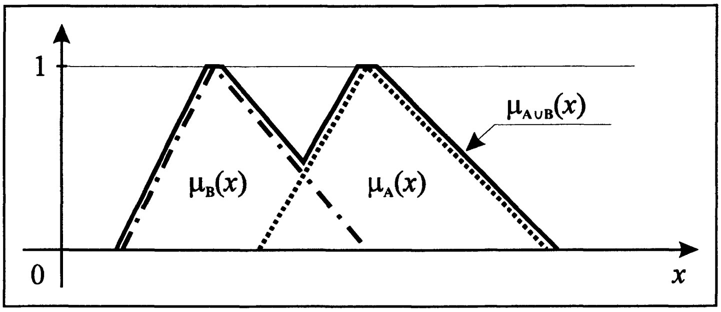

Figure 1.6. Membership functions of fuzzy sets **A**, **B**, and $\mathbf{A} \cup \mathbf{B}$

Example 1.5. Delays in air transportation can be due to technical reasons, meteorological conditions, late or nonappearing flight crews, and so on. We assume that out of total aircraft delays, technical reasons make "approximately 30%" of the reasons for delays. We also assume that meteorological conditions cause delays in "approximately 25%" of the cases. We will define fuzzy sets **A** and **B** as follows: **A** defines "approximately 30% of the cause of delay," and **B** defines "approximately 25% of the cause of delay."

The fuzzy set $\mathbf{A} \cup \mathbf{B}$ denotes "approximately 25% or approximately 30% of the cause of delay" and refers to aircraft delays due to technical reasons or meteorological conditions. The membership functions of fuzzy sets **A**, **B**, and $\mathbf{A} \cup \mathbf{B}$ are shown in *Figure 1.7.*

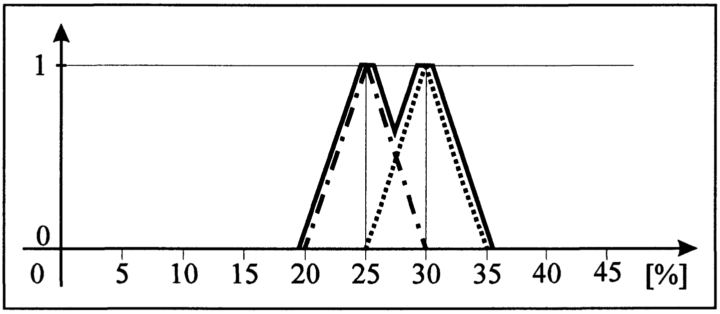

Figure 1.7. Membership functions of fuzzy sets "approximately 25%," "approximately 30%," and "approximately 25% or approximately 30%" of the cause of delay

In addition to max operator, under the name *T - conorm* operators, the *algebraic sum* and the *bounded sum* can also be used to describe the union of two fuzzy sets.

The membership function of the algebraic sum of fuzzy sets **A** and **B** is defined as

$$\mu_{\mathbf{A}+\mathbf{B}}(x) = \mu_{\mathbf{A}}(x) + \mu_{\mathbf{B}}(x) - \mu_{\mathbf{A}}(x) \cdot \mu_{\mathbf{B}}(x) \tag{1.15}$$

The membership function of the bounded sum of fuzzy sets **A** and **B** is given by

$$\mu_{\mathbf{A}(+)\mathbf{B}}(x) = \min\left\{1, \mu_{\mathbf{A}}(x) + \mu_{\mathbf{B}}(x)\right\} \tag{1.16}$$

As in the case of intersection of two fuzzy sets, in representing the union of two fuzzy sets there also exist operators representing a parameterized union. These include Hamacher's max operator, Yager's max operator, and Dubois and Prade's max operator. Let us illustrate the concept of a parameterized union by the example of Yager's max operator.

Yager (1982) defined his max operator in the following manner:

$$\mu_{\mathbf{A}\beta\mathbf{B}}(x) = \min\left[1, \left(\mu_{\mathbf{A}}^{\beta}(x) + \mu_{\mathbf{B}}^{\beta}(x)\right)^{\frac{1}{\beta}}\right] \quad \beta \geq 1 \tag{1.17}$$

In the case of $\beta = 1$, we obtain

$$\mu_{A\beta B}(x) = \min\{1, \mu_A(x) + \mu_B(x)\} \tag{1.18}$$

For $\beta = \infty$, we have

$$\mu_{A\beta B}(x) = \max\{\mu_A(x), \mu_B(x)\} \tag{1.19}$$

Different values of parameter β are corresponded by a weaker or stronger demand for "or."

1.6. FUZZY SET HEIGHT

Fuzzy set height comprises the greatest grade of membership of the elements belonging to a set. A fuzzy set is termed normalized if the grade of membership of at least one of its elements equals 1. *Figure 1.8* presents the membership functions of fuzzy set **A** whose height equals 0.7 and normalized fuzzy set **B** whose height equals 1.

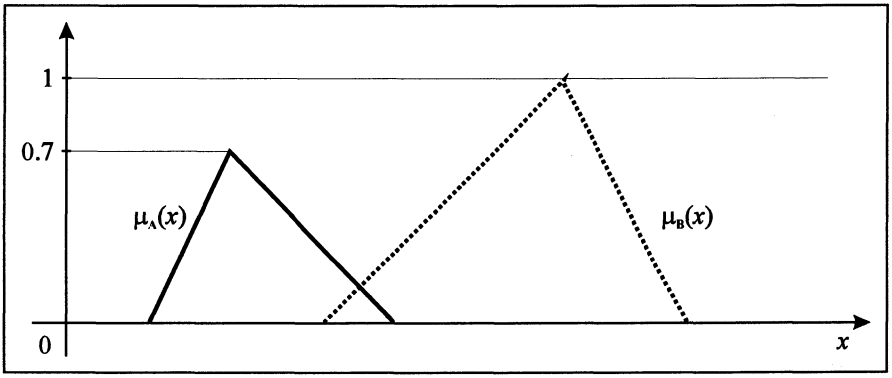

Figure 1.8. Membership functions of fuzzy set **A** whose height equals 0.7 and normalized fuzzy set **B**

1.7. SUPPORT OF FUZZY SET

In certain cases it is of particular interest to note those elements of a fuzzy set whose membership grade is greater than zero. Let us denote a fuzzy set by **A**. The support of fuzzy set **A** is understood to be a set of elements from X whose membership grade in fuzzy set **A** is greater than zero that is,

$$\text{Supp } \mathbf{A} = \left\{ x \in \mathrm{X} \,\middle|\, \mu_{\mathrm{A}}(x) > 0 \right\} \tag{1.20}$$

1.8. THE SCALAR CARDINALITY OF FUZZY SET

The scalar cardinality of fuzzy set **A** is defined as

$$|\mathbf{A}| = \sum_{x \in \mathrm{X}} \mu_{\mathrm{A}}(x) \tag{1.21}$$

whereas the relative cardinality of **A** is given by

$$\|\mathbf{A}\| = \frac{|\mathbf{A}|}{|x|} \tag{1.22}$$

It can therefore be said that cardinality and relative cardinality of fuzzy set **A** in a certain way indicate the proportion of elements from X having property **A**.

Example 1.6. Let us note fuzzy set **A** labeled as "set of old buses":

$$A = \frac{1}{12} + \frac{1}{10} + \frac{0.8}{7} + \frac{0.7}{6} + \frac{0.4}{4} + \frac{0.2}{2} + \frac{0.1}{1} \tag{1.23}$$

The age of the latest bus is one year while the age of the oldest bus is twelve years.

The cardinality of fuzzy set **A** is given by

$$|\mathbf{A}| = 1 + 1 + 0.8 + 0.7 + 0.4 + 0.2 + 0.1 = 4.2 \tag{1.24}$$

The relative cardinality of **A** is equal to

$$\|\mathbf{A}\| = \frac{|\mathbf{A}|}{|x|} = \frac{4.2}{7} = 0.6 \tag{1.25}$$

1.9. COMPLEMENT OF FUZZY SETS

A complement of fuzzy set A (whose elements have a grade of membership between 0 and 1) is understood to be fuzzy set $\bar{\mathrm{A}}$ whose membership function is calculated as

$$\mu_{\bar{\mathrm{A}}}(x) = 1 - \mu_{\mathrm{A}}(x) \tag{1.26}$$

Example 1.7. The precision in carrying out a planned timetable or airline schedule can be "measured" by delays in planned departure times. A delay is understood to be the difference between the actual and planned departure time. Let us denote by **A** and $\bar{\mathrm{A}}$, respectively, the fuzzy sets "punctual timetable" and "late timetable," whose membership functions are shown in *Figure 1.9.*

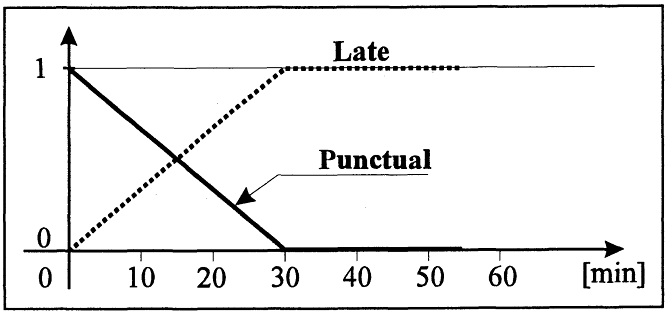

Figure 1.9. Membership functions of the fuzzy sets "punctual" and "late" timetables

It is clear that the fuzzy set "late" timetable is a complement of the fuzzy set "punctual" timetable.

1.10. CONVEX FUZZY SETS

Fuzzy set A is a convex fuzzy set if

$$\mu_{\mathrm{A}}(\lambda x_1 + (1-\lambda)x_2) \geq \mu_{\mathrm{A}}(x_1) \wedge \mu_{\mathrm{A}}(x_2) \tag{1.27}$$

where $x_1, x_2 \in \mathrm{X}$, $\lambda \in [0,1]$. *Figure 1.10* shows a convex fuzzy set and a nonconvex fuzzy set.

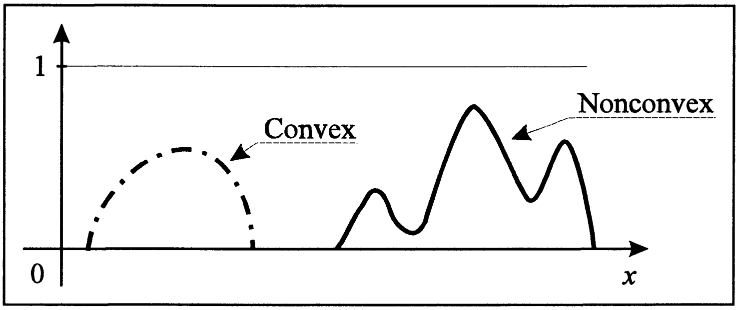

Figure 1.10. Convex and nonconvex fuzzy sets

1.11. LINGUISTIC HEDGES

Natural languages used in interpersonal communication are characterized by uncertainty, ambiguity, and imprecision. Nevertheless, we usually have no difficulty in understanding each other. Every driver will easily understand and accept the information that to a desired destination there is "about a ten-minute ride." The information to the effect that to a certain destination there is "9 min and 37s" is not normally used in everyday communication, nor is of any special interest. In some situations, insistence on the absolute precision that is, on the exclusive use of numerical variables may even cause confusion and ambiguity. When entering traffic from a parking lot or approaching an insufficiently visible intersection, the codriver is often requested information about the vehicles coming from the opposite direction. The information of the type "a vehicle is coming but it is quite far" will certainly be more easily accepted and understood than the following type of information "a vehicle is coming at the speed of 67 (km/h) and at a distance of 52.36 (m)." Certain situations are more conveniently described in terms of the so-called linguistic variables than by numerical variables. The concept of a linguistic variable was first introduced by Zadeh (1975a, 1975b). The values of numerical variables are numbers. Analogously, the values of linguistic variables are words or sentences. The stated examples show that linguistic variables are primarily used in either exceptionally complex or insufficiently defined or explained situations in which the application of numerical variables is not possible.

In order to explain a particular concept or situation, certain adjectives or adverbs are frequently used. These adjectives or adverbs are called *linguistic hedges*. The following are some of the typical linguistic hedges: *very*, *very very*, *more-or-less*, *almost*, *slightly*, and so on. By employing linguistic hedges, we can modify a membership function of a particular fuzzy set (Zadeh, 1973).

Let us denote by **A** a fuzzy set to which certain linguistic hedges will be applied. The linguistic hedges involved are those of concentration, dilatation, and intensification. Here are some examples of concentration:

$$\text{“very” } \mathbf{A} = \mathbf{A}^2 \tag{1.28}$$

$$\text{“very, very” } \mathbf{A} = \mathbf{A}^4 \tag{1.29}$$

The concentration defined by relation (1.30) will be denoted as CON_1 (**A**), while the concentration defined by relation (1.31) is denoted as CON_2 (**A**). The corresponding membership functions are

$$\mu_{CON_1(A)}(x) = (\mu_A(x))^2 \tag{1.30}$$

$$\mu_{CON_2(A)}(x) = (\mu_A(x))^4 \tag{1.31}$$

The membership functions of fuzzy set **A** and its concentration CON_1 (**A**) are shown in *Figure 1.11*.

As seen from *Figure 1.11*, the result of the concentration is a fuzzy set with a smaller membership grade. It can be observed that the degree of membership attained by the operation of concentration becomes lower as the original membership grade diminishes. The operation of dilatation yields the results contrary to those achieved by the operation of concentration. In other words, when the operation of dilatation is applied, the membership grade increases as the original membership grade diminishes. The following relation represents a typical dilatation:

$$\text{“slightly” } \mathbf{B} = \sqrt{\mathbf{B}} \tag{1.32}$$

that is,

$$\mu_{DIL(B)}(x) = (\mu_B(x))^{\frac{1}{2}} \tag{1.33}$$

Let us note a fuzzy set **A**. The membership function of intensification of fuzzy set **A** is defined as

$$\mu_{\text{INT}(A)}(x) = \begin{cases} 2\cdot(\mu_A(x))^2, & 0 \le \mu_A(x) \le 0.5 \\ 1 - 2\cdot[1-\mu_A(x)]^2, & 0.5 \le \mu_A(x) \le 1 \end{cases} \tag{1.34}$$

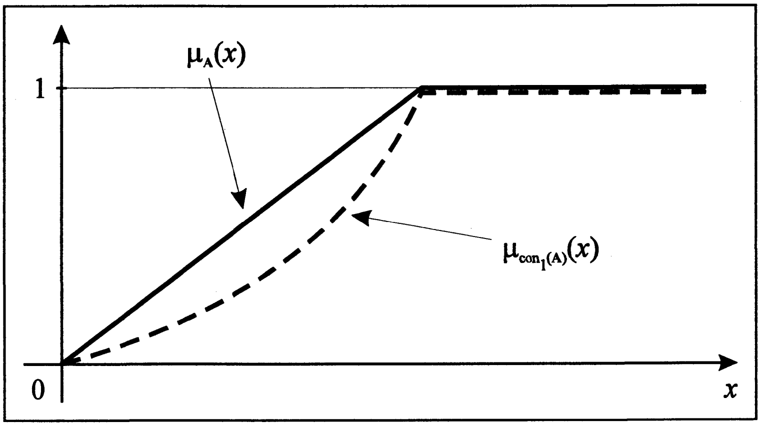

Figure 1.11. Membership functions of fuzzy set **A** and its concentration CON_1 (**A**)

The membership functions $\mu_{\mathbf{B}}(x)$ and $\mu_{\text{DIL}(\mathbf{B})}(x)$ are given in *Figure 1.12*.

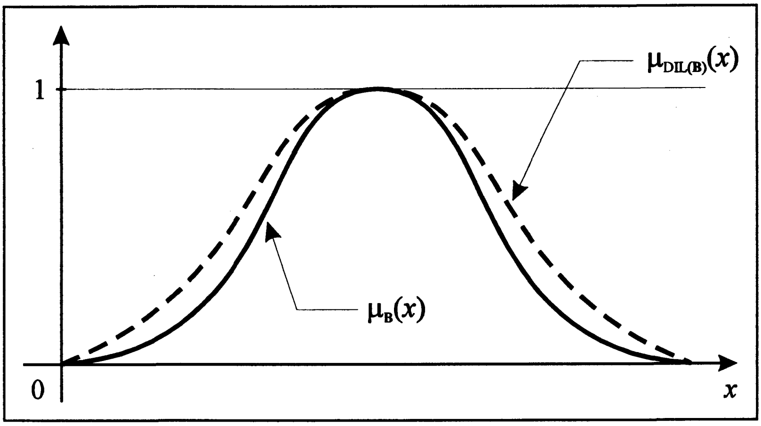

Figure 1.12. Membership functions $\mu_{\mathbf{B}}(x)$ and $\mu_{\text{DIL}(\mathbf{B})}(x)$

The membership functions $\mu_{\mathbf{A}}(x)$ and $\mu_{\text{INT}(\mathbf{A})}(x)$ are given in *Figure 1.13*.

As shown in *Figure 1.13*, the operation of intensification results in a growing difference in the grades of membership between the elements whose original membership grade is greater than 0.5 and those whose original membership grade is lower than 0.5.

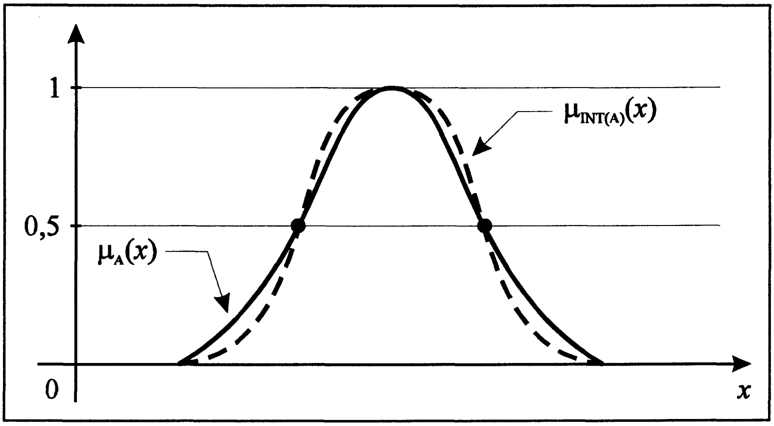

Figure 1.13. Membership functions $\mu_A(x)$ and $\mu_{INT(A)}(x)$

Let us note that concentrations, i.e., dilatations, can be defined by different relations. Thus, for example, one concentration and one dilatation can be, respectively, defined as:

$$\mu_{CON(A)}(x) = (\mu_A(x))^{1.5} \tag{1.35}$$

$$\mu_{DIL(A)}(x) = (\mu_A(x))^{0.75} \tag{1.36}$$

In the case of complex linguistic statements, i.e., in calculating their membership functions, Zadeh (1973) suggests first the application of linguistic hedges and the logical operator **Not** (if there is such in the statement), then the logical operator **And**, and finally the logical operator **Or**.

Example 1.8. We denote by **L** and **S** the fuzzy sets representing, respectively, "a large number of passengers waiting at a station" and "a small number of passengers waiting at a station." The membership functions of fuzzy sets **L** and **S**, $\mu_L(x)$ and $\mu_S(x)$ are given in *Figure 1.14*. These membership functions are calculated as

$$\mu_S(x) = \begin{cases} -\dfrac{x}{50} + 1, & 0 \le x \le 50 \\ 0, & x > 50 \end{cases} \tag{1.37}$$

$$\mu_L(x) = \begin{cases} \dfrac{x}{50}, & 0 \le x \le 50 \\ 1, & x > 50 \end{cases} \tag{1.38}$$

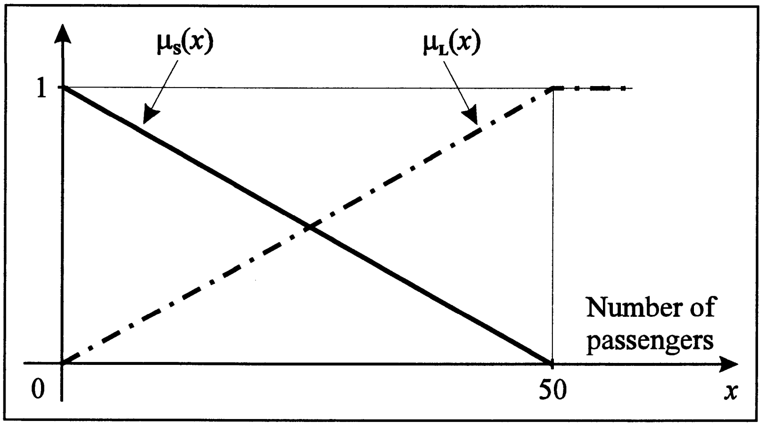

Figure 1.14. Membership functions $\mu_L(x)$ and $\mu_S(x)$

Let us denote by **VL** and **VVL**, respectively, the fuzzy sets representing a "very large" and a "very, very large" number of passengers waiting at a station. The corresponding membership functions are defined as follows:

$$\mu_{VL}(x) = (\mu_L(x))^2 = \begin{cases} \dfrac{x^2}{2500}, & 0 \le x \le 50 \\ 1, & x > 50 \end{cases} \tag{1.39}$$

$$\mu_{VVL}(x) = (\mu_L(x))^4 = \begin{cases} \dfrac{x^4}{6250000}, & 0 \le x \le 50 \\ 1, & x > 50 \end{cases} \tag{1.40}$$

The membership functions of fuzzy sets "very small" (**VS**) and "not very small" (**NVS**) number of passengers are, respectively, equal to

$$\mu_{VS}(x) = (\mu_S(x))^2 = \left(-\frac{x}{50} + 1\right)^2 \tag{1.41}$$

$$\mu_{NVS}(x) = 1 - \mu_{VS}(x) = 1 - \left(-\frac{x}{50} + 1\right)^2 \tag{1.42}$$

1.12. FUZZY SETS AS POINTS IN HYPERCUBES

In his books devoted to the fuzzy set theory Kosko (1992b, 1993) suggests the possibility of representing fuzzy sets as points in hypercubes. A specific graphical interpretation accompanied by appropriate explanations is intended to facilitate our reception of the concept of fuzzy sets. The presentation of fuzzy sets as points in hypercubes allows us to see more clearly the relation between a fuzzy set and its complement.

Let the universe comprise three elements, i.e., $X = \{x_1, x_2, x_3\}$. *Figure 1.15* represents fuzzy sets as points in hypercubes in the case of a three-element universe.

All the points at cube corners represent crisp sets. Thus, for example, point (0,1,0) represents a crisp subset to which x_1 and x_3 do not belong, while x_2 has a full membership. In other words, point (0,1,0) represents subset $\{x_2\}$. Subset $\{x_1\}$ is represented by point (1,0,0), subset $\{x_1, x_3\}$ by point (1,0,1), subset $\{x_3\}$ by point (0,0,1), and so on. Let us also note that point (0,0,0) represents a null set Ø, while point (1,1,1) refers to the whole set of *X*. The fact that bivalent opposites are linked by long diagonals ((1,0,1) and (0,1,0), (1,0,0) and (0,1,1), (0,0,1) and (1,1,0), and (0,0,0) and (1,1,1)) is particularly interesting.

Kosko (1992b, 1993) raised the following question: "If bivalent sets lie at the corners of a cube, what lies inside?" Kosko's perfectly clear reply was that "the fuzzy sets fill in the cube." Thus, for example, point (1/4, 2/3, 1/3) designated in *Figure 1.15* by *x*, represents a fuzzy set in which the membership grade of element x_1 is equal to 1/4, the membership grade of element x_2 is equal to 2/3, while the membership grade of element x_3 is equal to 1/3. It is obvious that each point in the cube shown in *Figure 1.15* represents a fuzzy set. The point at the cube center, at the intersection of the main diagonals, bears special significance. The coordinates of this point are (1/2, 1/2, 1/2).

These membership values indicate that the elements as much belong to the fuzzy set as they do not. In other words, these elements have equal membership in both the fuzzy set and its complement. The point at the cube center can be said to represent a set corresponded by a maximal fuzziness. Geometrically, the point at the cube center is viewed as equally remote from all corners of the cube.

Fuzzy sets as points in the case of a one-element universe and two-element universe are illustrated in *Figure 1.16*.

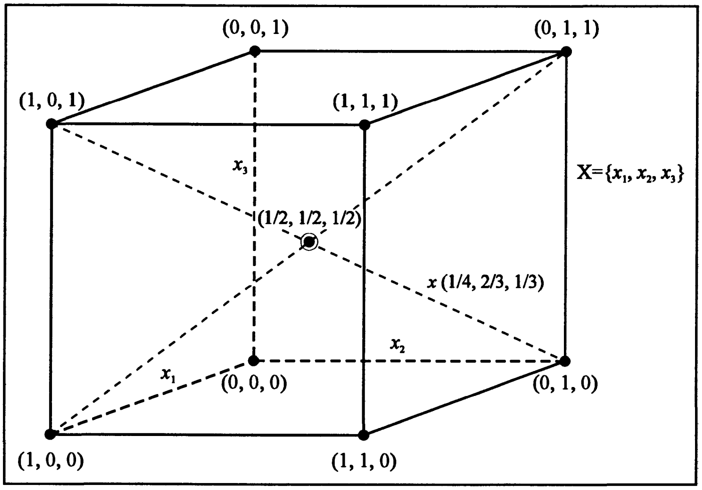

Figure 1.15. Fuzzy sets as points in the case of a three-element universe

It should be stressed once again that only points at cube corners (*Figure 1.16*), i.e., points at square corners, points at line ends represent crisp sets. All other points represent fuzzy sets. The fuzzier things are, the closer we get to the center of a section, square, cube, ..., hypercube in the n-space (in the case of a universe with n elements). As Kosko (1993) noted: "Bivalence holds at cube corners. Multivalence holds everywhere else."

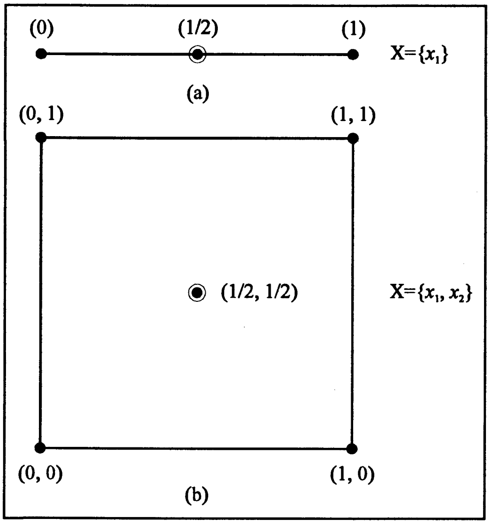

Figure 1.16. Fuzzy sets as points in the case of (a) one-element universe and (b) two-element universe

1.13. FUZZY RELATIONS

The concept of a fuzzy relation can be explained similarly to that of a fuzzy set. As we have already mentioned, unlike classical sets, where membership functions of set elements can take only the values 0 and 1, membership functions of fuzzy sets can take any value in the interval [0,1]. A higher membership function value indicates an element's "stronger" membership in a fuzzy set. There is a specific relation between two classical sets if their elements are interdependent, are connected, or condition each other. The transition from the concept of classical relations to the concept of fuzzy relations can be made by introducing the concept of "strength" of the relation between individual elements.

Before a formal definition of the concept of fuzzy relations is given, let us recall the definition of Cartesian product.

1.13.1. Cartesian product

Let us note two classical sets A and B. The Cartesian product of sets A and B is denoted by $A \times B$ and is defined as the set of ordered pairs in which the first element is from set A and the second element is from set B. The Cartesian product of $A \times B$ can be written as

$$A \times B = \{(a, b) \mid a \in A, b \in B\} \tag{1.43}$$

Relation R between sets A and B is a subset of the Cartesian product $A \times B$.

Example 1.9. Let $X = \{x_1, x_2, x_3\}$ denote the set of international airports in a country. Let $Y = \{y_1, y_2\}$ denote the set of international airports in one of the neighboring countries. Let there be air traffic established between cities x_1 and y_2 and cities x_3 and y_1. We denote by R the relation of establishing air traffic between these cities. The Cartesian product $X \times Y$ is

$$X \times Y = \{(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2), (x_3, y_1), (x_3, y_2)\} \tag{1.44}$$

Relation R(X,Y) between sets X and Y is

$$R(X, Y) = \{(x_1, y_2), (x_3, y_1)\} \tag{1.45}$$

Figure 1.17 presents the cities and the established air traffic.

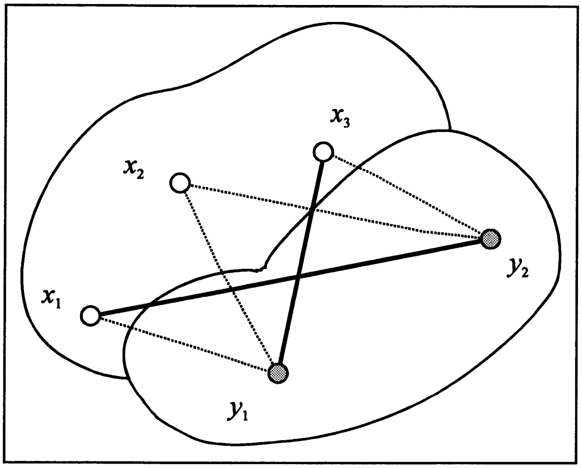

Figure 1.17. Airline network

The Cartesian product $X \times Y$ contains all possible air traffic routes (dotted lines), while relation R(X,Y) consists of only those routes where air traffic has been established. We can write that $R(X,Y) \subset X \times Y$.

1.13.2. Definition of a fuzzy relation

The membership function of relation $\mu_R(a, b)$ can be defined as follows:

$$\mu_R(a,b) = \begin{cases} 1, & \text{if and only if } (a,b) \text{ belongs to relation R} \\ 0, & \text{otherwise} \end{cases} \quad (1.46)$$

It is clear that for a classical relation, membership of a pair in the relation is expressed by only 0 or 1. Let us assume that membership of a pair in the relation can be expressed by any number between 0 and 1. We also designate this value to comprise the "strength" of membership of a pair in the relation. Let us note sets X_1 and X_2.

A fuzzy relation between sets X_1 and X_2 consists of a fuzzy set defined over the Cartesian product of $X_1 \times X_2$, where the ordered pairs (x_1, x_2) belong to a relation with a grade of membership ranging from 0 to 1.

Example 1.10. Let A and B denote sets of new housing areas on one side and the other of a river passing through a city (*Figure 1.18*). Set A consists of housing areas a_1, a_2, a_3 and set B consists of housing areas b_1, b_2: $A=\{a_1, a_2, a_3\}$ $B=\{b_1, b_2\}$.

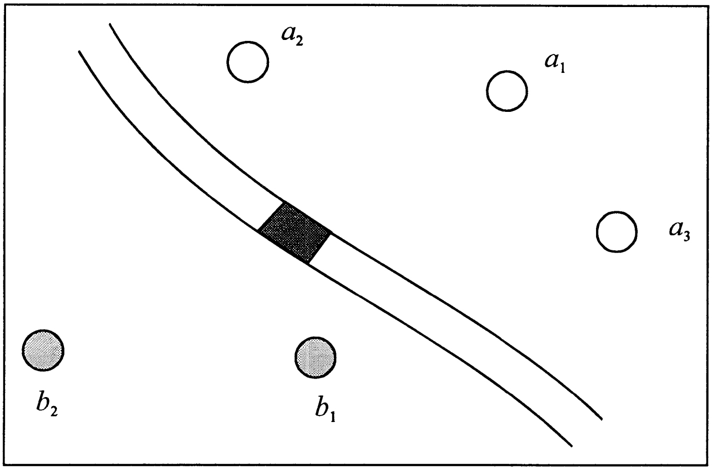

Figure 1.18. Distribution of new housing areas

Let us assume that fuzzy binary relation **R** between sets A and B denotes "short travel time." The convenient representations of a fuzzy binary relation **R** referring to "short travel time" are shown in *Figure 1.19*.

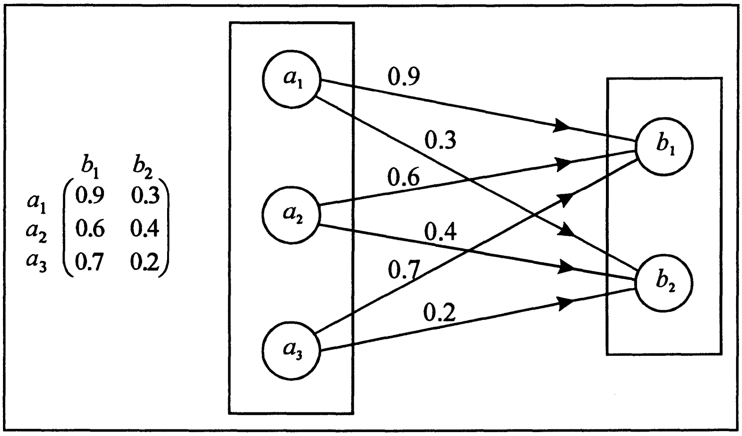

Figure 1.19. Representations of a fuzzy relation: membership matrix and sagittal diagram

Example 1.11. The dial-a-ride passenger transportation service is used in areas with smaller population density in some of the West European countries and in the United States. This system is also used in large cities to transport older and handicapped persons. Although there are various modifications to the dial-a-ride system, it most often has the following characteristics:

- The user calls the dispatcher service of the transportation organization and requests a ride.
- The ride request is described by an origin, a destination, and the time (or time interval) when the vehicle should pick up the client.
- Ride requests are usually made the day before they are needed.
- During the period from receiving the requests to starting the pickups, the ride organizer plans the vehicle routes.
- When designing the vehicle routes, the ride organizer endeavors to plan trips with the smallest number of vehicles and to design a set of routes with the shortest possible length.

When designing vehicle routes, some models first group the requested rides and then design the routes within each group. Kagaya et al. (1994) developed a model to group the requested rides. The model is based on the concept of fuzzy relations. Let us note *Figure 1.20.*

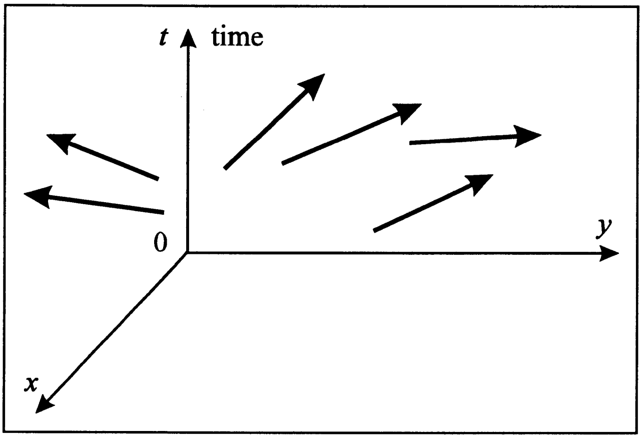

Figure 1.20. Requested rides described by an origin and destination direction and requested pickup time

Figure 1.20 presents a group of "similar" rides. Kagaya et al. (1994) determined the "similarity" of individual rides according to the following attributes:

- Distance between origin of ride t_i and origin of ride t_j,
- Distance between destination of ride t_i and destination of ride t_j,
- Time difference between pickup time of ride t_i and pickup time of ride t_j,
- Time difference between dropoff time of ride t_i and dropoff time of ride t_j,

- The difference between the angle that covers rides t_i and t_j, and the x (or y) axis,
- The difference between destination of ride t_i and origin of ride t_j.

Let there be a total of n rides to be made. We denote these rides respectively by $t_1, t_2, t_3, ..., t_n$. Each of these rides is described by origin and destination coordinates and by the pickup time of the ride. Let us denote by $\mathbf{R}_1$ the fuzzy relation referring to the "similarity of the origin of two rides." The grade of membership in fuzzy relation $\mathbf{R}_1$ can be presented in matrix form:

$$\begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_n \end{array} \begin{array}{c} \begin{array}{ccccc} \quad t_1 \quad & \quad t_2 \quad & \quad t_3 \quad & \cdots & \quad t_n \quad \end{array} \\ \begin{pmatrix} \mu_{\mathbf{R}_1}(t_1,t_1) & \mu_{\mathbf{R}_1}(t_1,t_2) & \mu_{\mathbf{R}_1}(t_1,t_3) & \cdots & \mu_{\mathbf{R}_1}(t_1,t_n) \\ \mu_{\mathbf{R}_1}(t_2,t_1) & \mu_{\mathbf{R}_1}(t_2,t_2) & \mu_{\mathbf{R}_1}(t_2,t_3) & \cdots & \mu_{\mathbf{R}_1}(t_2,t_n) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mu_{\mathbf{R}_1}(t_n,t_1) & \mu_{\mathbf{R}_1}(t_n,t_2) & \mu_{\mathbf{R}_1}(t_n,t_3) & \cdots & \mu_{\mathbf{R}_1}(t_n,t_n) \end{pmatrix} \end{array} \tag{1.47}$$

Kagaya et al. (1994) proposed that the grade of membership in fuzzy relation $\mathbf{R}_1$ representing the "similarity of two rides" can be calculated using the following relation:

$$\mu_{\mathbf{R}_1}(t_i,t_j) = \frac{1}{\left\{1 + a\left[\left(XO(t_i) - XO(t_j)\right)^2 + \left(YO(t_i) - YO(t_j)\right)^2\right]^b\right\}} \tag{1.48}$$

where $XO(t_i)$, $YO(t_i)$ are origin coordinates of ride t_i; $XO(t_j)$, $YO(t_j)$ are origin coordinates of ride t_j; a, b are parameters whose values are subjectively defined after many attempts to create a functional dependence that will generate a wide range of grades of membership in the fuzzy relation within the interval of 0 to 1.

We denote the relations $\mathbf{R}_2$, $\mathbf{R}_3$, $\mathbf{R}_4$, $\mathbf{R}_5$ and $\mathbf{R}_6$ as follows:

$\mathbf{R}_2$ is fuzzy relation of similarity (proximity) between the destination coordinates of two rides
$\mathbf{R}_3$ is fuzzy relation of similarity between the pickup times of two rides
$\mathbf{R}_4$ is fuzzy relation of similarity between the dropoff times of two rides
$\mathbf{R}_5$ is fuzzy relation of similarity between the angles that cover rides with the x (or) y axis
$\mathbf{R}_6$ is fuzzy relation of similarity (proximity) between the destination of the first and the origin of the second ride under consideration

Kagaya et al. (1994) proposed the following functional dependencies to calculate the grades of membership in these fuzzy relations:

$$\mu_{\mathbf{R}_2}(t_i, t_j) = \\ = \frac{1}{\left\{1 + 0.00015\left[\left(XD(t_i) - XD(t_j)\right)^2 + \left(YD(t_i) - YD(t_j)\right)^2\right]^{2.23}\right\}} \tag{1.49}$$

$$\mu_{\mathbf{R}_3}(t_i, t_j) = \frac{1}{\left\{1 + 0.007\left[TD(t_i) - TD(t_j)\right]\right\}^{1.02}} \tag{1.50}$$

$$\mu_{\mathbf{R}_4}(t_i, t_j) = \frac{1}{\left\{1 + 0.007\left[TA(t_i) - TA(t_j)\right]\right\}^{1.02}} \tag{1.51}$$

$$\mu_{\mathbf{R}_5}(t_i, t_j) = \frac{1}{\left\{1 + 0.5\left[AG(t_i) - AG(t_j)\right]^2\right\}} \tag{1.52}$$

$$\mu_{\mathbf{R}_6}(t_i, t_j) = \\ = \frac{1}{\left\{1 + 0.00014\left[\left(XD(t_i) - XD(t_j)\right)^2 + \left(YO(t_i) - YO(t_j)\right)^2\right]^{1.75}\right\}} \tag{1.53}$$

$XO(t_i)$, $YO(t_i)$, $XD(t_i)$, $YD(t_i)$, $TD(t_i)$, $TA(t_i)$ are, respectively, the origin and destination coordinates of the ride, the pickup time, and dropoff time of the ith ride.

1.13.3. The Cartesian product of fuzzy sets

Let us denote by $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n$, respectively, the fuzzy sets defined over sets $X_1, X_2, \ldots, X_n$. The Cartesian product of fuzzy sets $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n$ is defined as $\mathbf{A}_1 \times \mathbf{A}_2 \times \ldots \times \mathbf{A}_n$. The membership function of the Cartesian product of the fuzzy sets is calculated as

$$\mu_{\mathbf{A}_1 \times \mathbf{A}_2 \times \ldots \times \mathbf{A}_n}(x_1, x_2, \ldots, x_n) = \mu_{\mathbf{A}_1}(x_1) \wedge \mu_{\mathbf{A}_2}(x_2) \wedge \ldots \wedge \mu_{\mathbf{A}_n}(x_n) \tag{1.54}$$

Example 1.12. The following example illustrates how to calculate the membership function of the Cartesian product of fuzzy sets. A great variety of factors influence the trips that are made between two cities. Urban population is one of the most important factors generating a total number of trips.

Let $\mathbf{A}_1$ and $\mathbf{A}_2$ denote fuzzy sets that represent "large populations" in cities in two neighboring countries. Let the fuzzy sets $\mathbf{A}_1$ and $\mathbf{A}_2$ be

$$\mathbf{A}_1 = \frac{0.3}{200000} + \frac{0.6}{450000} + \frac{1}{1100000} \qquad \mathbf{A}_2 = \frac{0.25}{180000} + \frac{0.85}{820000} \tag{1.55}$$

The fuzzy set $\mathbf{A}_1 \times \mathbf{A}_2$, which represents the Cartesian product of sets $\mathbf{A}_1$ and $\mathbf{A}_2$, refers to "pairs of cities with large populations." We can write that

$$\begin{aligned}\mathbf{A}_1 \times \mathbf{A}_2 = {} & \frac{\min(0.3, 0.25)}{(200000, 180000)} + \frac{\min(0.3, 0.85)}{(200000, 820000)} + \\ & + \frac{\min(0.6, 0.25)}{(450000, 180000)} + \frac{\min(0.6, 0.85)}{(450000, 820000)} + \\ & + \frac{\min(1, 0.25)}{(1100000, 180000)} + \frac{\min(1, 0.85)}{(1100000, 820000)}\end{aligned} \tag{1.56}$$

which is

$$\begin{aligned}\mathbf{A}_1 \times \mathbf{A}_2 = {} & \frac{0.25}{(200000, 180000)} + \frac{0.3}{(200000, 820000)} + \\ & + \frac{0.25}{(450000, 180000)} + \frac{0.6}{(450000, 820000)} + \\ & + \frac{0.25}{(1100000, 180000)} + \frac{0.85}{(1100000, 820000)}\end{aligned} \tag{1.57}$$

1.14. MAX-MIN COMPOSITION

Let us note sets X, Y, and Z and relations $R_1(X,Y)$ and $R_2(Y,Z)$. The composition of these relations R(X,Z) is denoted as

$$R(X,Z) = R_1(X,Y) \circ R_2(X,Z) \tag{1.58}$$

The composition R(X,Z) is a subset of the Cartesian product X × Z such that $(x, z) \in R$ if and only if there is at least one $y \in Y$ such that $(x, y) \in R_1$ and $(y, z) \in R_2$.

For fuzzy relations $\mathbf{R}_1$ and $\mathbf{R}_2$, which are defined, respectively, over the Cartesian products X × Y and Y × Z, the max-min composition is most often used, which calculates the membership function as

$$\mu_{\mathbf{R}_1 \circ \mathbf{R}_2}(x,z) = \max_{y \in Y} \left\{ \min \left[\mu_{\mathbf{R}_1}(x,y), \mu_{\mathbf{R}_2}(y,z) \right] \right\} \tag{1.59}$$

Example 1.13. Let us note three sets of airports X, Y, and Z. The following airports belong to these sets:

$$X = \{x_1, x_2, x_3\} \quad Y = \{y_1, y_2\} \quad Z = \{z_1, z_2, z_3, z_4\} \tag{1.60}$$

We assume that due to the existing flight schedule, trips going from airports in set X to airports in set Z can be made only through airports in set Y. In other words, airports y_1 and y_2 are stopover points on trips from x_1, x_2, or x_3 to z_1, z_2, z_3, or z_4. We also assume that passengers subjectively estimate total travel time between individual airports, bearing in mind planned departure times, expected delays, and waiting times at the stopover airports. Let fuzzy relation $\mathbf{R}_1$ refer to "short travel time" between the airports of set X and the airports of set Y. $\mathbf{R}_2$ denotes the fuzzy relation "short travel time" between the airports of set Y and the airports of set Z. Let $\mathbf{R}_1$ and $\mathbf{R}_2$ be defined by the following matrices:

$$\mathbf{R}_1 = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{pmatrix} 0.4 & 0.8 \\ 0.65 & 0.75 \\ 0.9 & 0.2 \end{pmatrix} \end{array} \tag{1.61}$$

$$\mathbf{R}_2 = \begin{array}{c} \\ y_1 \\ y_2 \end{array} \begin{array}{c} \begin{array}{cccc} z_1 & z_2 & z_3 & z_4 \end{array} \\ \begin{pmatrix} 0.15 & 0.45 & 0.35 & 0.2 \\ 0.1 & 0.95 & 1 & 0.55 \end{pmatrix} \end{array} \tag{1.62}$$

Fuzzy relation $\mathbf{R}_1 \circ \mathbf{R}_2$ refers to "short travel time" between the airports of set X and the airports of set Z through the airports of set Y. The calculations for the grades of membership for airports x_2 and z_3 in fuzzy relation $\mathbf{R}_1 \circ \mathbf{R}_2$ are illustrated as follows:

$$\min\left\{\mu_{\mathbf{R}_1}(x_2, y_1), \mu_{\mathbf{R}_2}(y_1, z_3)\right\} = \min\left\{(0.65, 0.35)\right\} = 0.35 \tag{1.63}$$

$$\min\left\{\mu_{\mathbf{R}_1}(x_2, y_2), \mu_{\mathbf{R}_2}(y_2, z_3)\right\} = \min\left\{(0.75, 1)\right\} = 0.75 \tag{1.64}$$

$$\mu_{\mathbf{R}_1 \circ \mathbf{R}_2}(x_2, z_3) = \max(0.35, 0.75) = 0.75 \tag{1.65}$$

The grades of membership in fuzzy relation $\mathbf{R}_1 \circ \mathbf{R}_2$ for the other pairs of airports are determined in the same manner. These grades of membership are given in the following matrix:

$$\mathbf{R}_1 \circ \mathbf{R}_2 = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cccc} z_1 & z_2 & z_3 & z_4 \end{array} \\ \begin{pmatrix} 0.15 & 0.8 & 0.8 & 0.55 \\ 0.15 & 0.75 & 0.75 & 0.55 \\ 0.15 & 0.2 & 0.2 & 0.2 \end{pmatrix} \end{array} \tag{1.66}$$

1.15. EXTENSION PRINCIPLE

The extension principle is one of the most important principles in the theory of fuzzy sets. Let us note sets X and Y. Let function f map set X onto set Y. We also note fuzzy set **A** defined over set X:

$$\mathbf{A} = \frac{\mu_{\mathbf{A}}(x_1)}{x_1} + \frac{\mu_{\mathbf{A}}(x_2)}{x_2} + \ldots + \frac{\mu_{\mathbf{A}}(x_n)}{x_n} \tag{1.67}$$

The extension principle states

$$\begin{aligned} f(\mathbf{A}) &= f\left(\frac{\mu_{\mathbf{A}}(x_1)}{x_1} + \frac{\mu_{\mathbf{A}}(x_2)}{x_2} + \ldots + \frac{\mu_{\mathbf{A}}(x_n)}{x_n}\right) = \\ &= \frac{\mu_{\mathbf{A}}(x_1)}{f(x_1)} + \frac{\mu_{\mathbf{A}}(x_2)}{f(x_2)} + \ldots + \frac{\mu_{\mathbf{A}}(x_n)}{f(x_n)} \end{aligned} \tag{1.68}$$

Example 1.14. Time loss caused by an airline schedule, i.e., schedule delay, is a difference between a passenger's desired departure time and the time the passenger chooses based on the offered flight schedule. Thus, for example, if a passenger wants to leave at 9:00 a.m. and the airline schedule provides him with a departure at 11:00 a.m., there is a 2-hour schedule delay. There is a 3-hour schedule delay if the passenger wants to leave at 2:00 p.m. and the airline schedule offers him a departure at 11:00 a.m. Teodorovic (1983) established the following relation between average schedule delay per passenger s_d and flight frequency:

$$s_d(N) = \frac{T}{4N} \tag{1.69}$$

where s_d is schedule delay per passenger caused by the airline schedule, T is time period during which passengers have requested flights (passenger requests for flights are most often from 6:00 a.m. to 10:00 p.m.), and N is flight frequency during time period T.

Let there be 18 hours of passenger requests for flights during the day and let there be a planned flight frequency of "approximately 6 flights." We assume that fuzzy set **A** denoting "approximately 6 flights" contains the following elements:

$$\mathbf{A} = \frac{0.25}{4} + \frac{0.5}{5} + \frac{1}{6} + \frac{0.5}{7} + \frac{0.25}{8} \tag{1.70}$$

It is clear that

$$\begin{aligned} &s_d(4) = \frac{18}{16} = 1.125 \quad s_d(5) = \frac{18}{20} = 0.9 \quad s_d(6) = \frac{18}{24} = 0.75 \\ &s_d(7) = \frac{18}{28} = 0.642 \quad s_d(8) = \frac{18}{32} = 0.562 \end{aligned} \tag{1.71}$$

Fuzzy set **B** denotes "average schedule delay per passenger when there is a flight frequency of approximately 6 flights." The following elements belong to set **B**:

$$\mathbf{B} = \frac{0.25}{1.125} + \frac{0.5}{0.9} + \frac{1}{0.75} + \frac{0.5}{0.642} + \frac{0.25}{0.562} \tag{1.72}$$

Let us consider the extension principle when the dependent variable is a function of a large number of independent variables. Let us note sets $X_1, X_2, \ldots, X_n$. The Cartesian product of these sets is denoted by X:

$$X = X_1 \times X_2 \times \ldots \times X_n \tag{1.73}$$

We denote by $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_n$ the fuzzy sets determined respectively over sets $X_1, X_2, \ldots, X_n$. Let function f maps X onto set Y:

$$y = f(x_1, x_2, \ldots, x_n) \tag{1.74}$$

where $x_1 \in X_1, x_2 \in X_2, \ldots, x_n \in X_n$.

The extension principle states

$$\mu_Q(Y) = \max_{\substack{x_1, x_2, \ldots, x_n \\ y = f(x_1, x_2, \ldots, x_n)}} \left\{ \min\left(\mu_{P_1}(x_1), \ldots, \mu_{P_n}(x_n) \right) \right\} \qquad (1.75)$$

where **Q** is a fuzzy set defined over set Y.

Example 1.15. Figure 1.21 presents part of a transportation network. The solid line indicates link (A,B), which is common to paths p_1 and p_2. Origins and destinations along paths p_1 and p_2 are indicated, respectively, by r_1, r_2 and s_1, s_2.

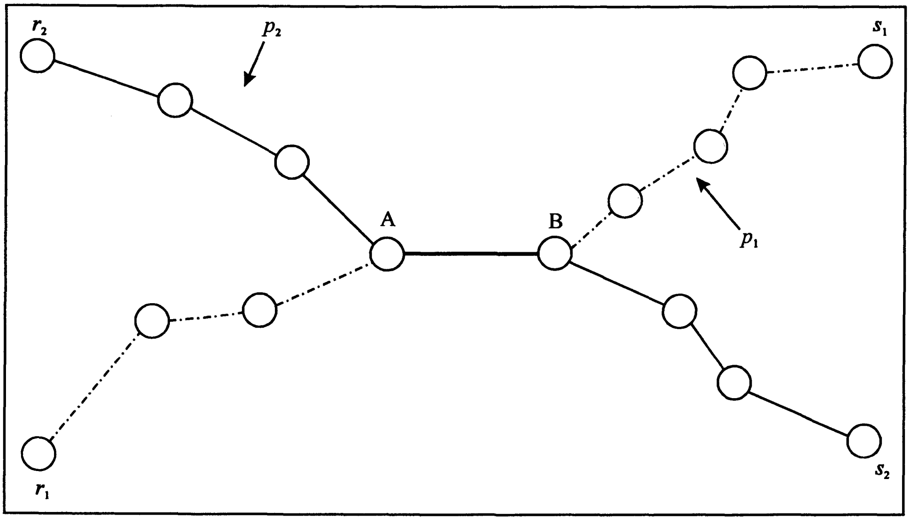

Figure 1.21. Paths p_1, p_2 and common link (A,B)

The traffic volume along common link (A,B), q, equals

$$q = q_1 + q_2 \qquad (1.76)$$

where q_i is traffic volume along path p_i, and $i = 1, 2$.

Let us assume that we know the approximate values of the volumes q_i, ($i = 1, 2$). We have described these approximate values by the following fuzzy sets: $\mathbf{P}_1$ is the traffic volume along path p_1 of approximately 500 (vehicles/h), and $\mathbf{P}_2$ is the traffic volume along path p_2 of approximately 700 (vehicles/h).

Let **Q** denote the fuzzy set that represents the traffic volume along link (A,B). The following elements belong to fuzzy sets $\mathbf{P}_1$ and $\mathbf{P}_2$:

$$\mathbf{P}_1 = \frac{0.2}{400} + \frac{0.6}{450} + \frac{1}{500} + \frac{0.6}{550} + \frac{0.2}{600} \qquad \mathbf{P}_2 = \frac{0.5}{600} + \frac{1}{700} + \frac{0.5}{800} \tag{1.77}$$

$$\mu_Q(400 + 600) = \mu_Q(1000) = \min(0.2, 0.5) = 0.2 \tag{1.78}$$

$$\mu_Q(400 + 700) = \mu_Q(1100) = \min(0.2, 1) = 0.2 \tag{1.79}$$

$$\mu_Q(400 + 800) = \mu_Q(1200) = \min(0.2, 0.5) = 0.2 \tag{1.80}$$

$$\mu_Q(450 + 600) = \mu_Q(1050) = \min(0.6, 0.5) = 0.5 \tag{1.81}$$

$$\mu_Q(450 + 700) = \mu_Q(1150) = \min(0.6, 1) = 0.6 \tag{1.82}$$

$$\mu_Q(450 + 800) = \mu_Q(1250) = \min(0.6, 0.5) = 0.5 \tag{1.83}$$

$$\mu_Q(500 + 600) = \mu_Q(1100) = \min(1, 0.5) = 0.5 \tag{1.84}$$

$$\mu_Q(500 + 700) = \mu_Q(1200) = \min(1, 1) = 1 \tag{1.85}$$

$$\mu_Q(500 + 800) = \mu_Q(1300) = \min(1, 0.5) = 0.5 \tag{1.86}$$

$$\mu_Q(550 + 600) = \mu_Q(1150) = \min(0.6, 0.5) = 0.5 \tag{1.87}$$

$$\mu_Q(550 + 700) = \mu_Q(1250) = \min(0.6, 1) = 0.6 \tag{1.88}$$

$$\mu_Q(550 + 800) = \mu_Q(1350) = \min(0.6, 0.5) = 0.5 \tag{1.89}$$

$$\mu_Q(600+600)=\mu_Q(1200)=\min(0.2,\,0.5)=0.2 \tag{1.90}$$

$$\mu_Q(600+700)=\mu_Q(1300)=\min(0.2,\,1)=0.2 \tag{1.91}$$

$$\mu_Q(600+800)=\mu_Q(1400)=\min(0.2,\,0.5)=0.2 \tag{1.92}$$

The final grades of membership of the elements in set **Q** equal

$$\mu_Q(1000)=0.2\,,\ \mu_Q(1050)=0.5 \tag{1.93}$$

$$\mu_Q(1100)=\max\,(0.2,\,0.5)=0.5 \tag{1.94}$$

$$\mu_Q(1150)=\max\,(0.6,\,0.5)=0.6 \tag{1.95}$$

$$\mu_Q(1200)=\max\,(0.2,\,1,\,0.2)=1 \tag{1.96}$$

$$\mu_Q(1250)=\max\,(0.5,\,0.6)=0.6 \tag{1.97}$$

$$\mu_Q(1300)=\max\,(0.5,\,0.2)=0.5 \tag{1.98}$$

$$\mu_Q(1350)=0.5\,,\ \mu_Q(1400)=0.2 \tag{1.99}$$

The membership functions of fuzzy sets $\mathbf{P}_1$, $\mathbf{P}_2$, and **Q** are shown in *Figure 1.22*.

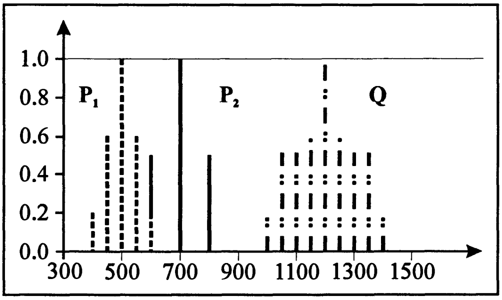

Figure 1.22. Membership functions of fuzzy sets $\mathbf{P}_1$, $\mathbf{P}_2$, and $\mathbf{Q}$

1.16. ALPHA CUT

The alpha cut (denoted as α - cut) of fuzzy set **A** is understood to be set A_α, which contains all elements whose grade of membership in fuzzy set **A** is greater than or equal to α, with $\alpha \in [0, 1]$. In other words, we can write

$$A_\alpha = \{x \mid \mu_A(x) \geq \alpha\} \qquad \forall x \in X \tag{1.100}$$

In addition to the α - cut, there is also a strong α - cut, which is defined as follows:

$$A_\alpha' = \{x \mid \mu_A(x) > \alpha\} \qquad \forall x \in X \tag{1.101}$$

Example 1.16. The claim that there are "relatively few aircraft" on the airport apron in late morning hours can be expressed by a corresponding fuzzy set **A**. Let us assume that the following elements belong to fuzzy set **A**:

$$\begin{aligned}\mathbf{A} = \; & \frac{0.95}{1} + \frac{0.9}{2} + \frac{0.85}{3} + \frac{0.8}{4} + \frac{0.75}{5} + \frac{0.7}{6} + \frac{0.65}{7} + \\ & + \frac{0.6}{8} + \frac{0.55}{9} + \frac{0.5}{10} + \frac{0.45}{11} + \frac{0.4}{12} + \frac{0.35}{13} + \frac{0.3}{14} + \\ & + \frac{0.25}{15} + \frac{0.2}{16} + \frac{0.15}{17} + \frac{0.1}{18} + \frac{0.05}{19} + \frac{0}{20}\end{aligned} \tag{1.102}$$

Let us define the α - set of fuzzy set **A** for $\alpha = 0.85$ and $\alpha = 0.55$. These α - cuts contain the following elements:

$$A_{0.85} = \{1, 2, 3\} \qquad A_{0.55} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\} \tag{1.103}$$

Figure 1.23 shows fuzzy set **A** and the α - cuts $A_{0.85}$ and $A_{0.55}$.

The previous sections defined the concept of a convex fuzzy set. A convex fuzzy set can be defined using the α - cut in the following way: a fuzzy set is convex if and only if each of its α - cuts is convex. *Figure 1.24* presents a convex and a nonconvex fuzzy set with corresponding α - cuts.

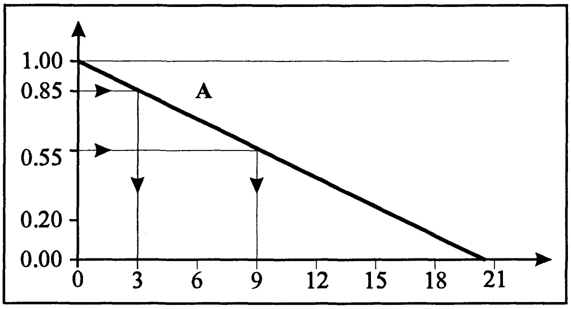

Figure 1.23. Fuzzy set **A** and α - cuts $A_{0.85}$ and $A_{0.55}$

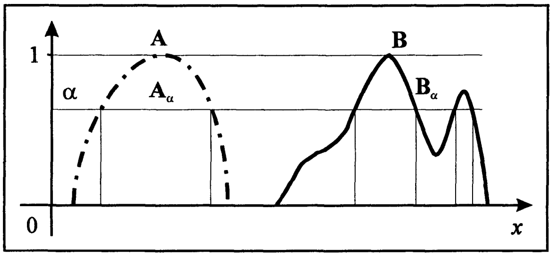

Figure 1.24. Convex and nonconvex fuzzy sets and corresponding α - cuts

1.17. THE CONCEPT OF A FUZZY NUMBER

Subjective estimations that deal with vehicle time loss at a signal or the cost of making a flight can be expressed by corresponding fuzzy sets. Based on experience or intuition, an expert is able to state that vehicle time loss at a traffic signal is "around 30 seconds" or that the cost of making a flight is "approximately $3,000." These subjective estimations are characterized by certain numerical values. It is intuitively clear that a flight cost that is "approximately $3,000" is less than a flight cost of "approximately $4,000." In other words, the fuzzy sets "around 30," "approximately 3,000" or "approximately 4,000" can be treated as numbers. Numbers expressed in this manner are called *fuzzy numbers*. The literature often uses the expression uncertain number in addition to fuzzy number. A fuzzy number is a fuzzy set that is convex and normalized. *Figure 1.25* shows the membership function of the fuzzy number "approximately 10."

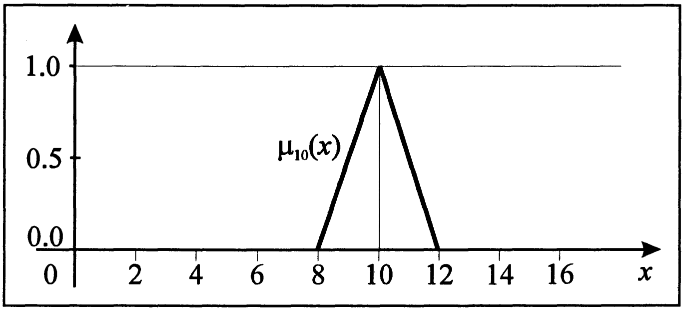

Figure 1.25. Membership function of the fuzzy number "approximately 10"

Fuzzy sets **B** and **C** are presented in *Figure 1.26.* These fuzzy sets cannot be fuzzy numbers since set **B** is not convex and set **C** is not a normalized fuzzy set.

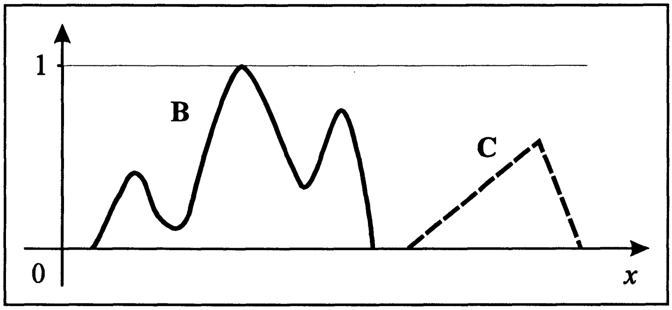

Figure 1.26. Fuzzy sets **B** and **C** which cannot be fuzzy numbers

In recent years, many authors have considered the problem of fuzzy numbers or fuzzy arithmetic. The extremely valuable monograph by Kaufmann and Gupta (1985) devoted to the problems of fuzzy arithmetic is of particular consequence. As they noted, fuzzy numbers can be treated as a "generalization of the concept of the confidence interval." We will briefly present the manner in which Kaufmann and Gupta generalized the concept of the confidence interval in an attempt to explain the concept of a fuzzy number. Our discussion involves traffic and transportation examples.

Let us assume that the travel time between two points is uncertain. Let us also assume that we are able to estimate travel time to be not less than t_1 and not greater than t_2. In other words, we are certain that travel time is within the closed interval $[t_1, t_2]$. Such an interval is called a *confidence interval* and is denoted as $T = [t_1, t_2]$.

A fuzzy number is also characterized by a level of presumption, which can be explained by the following example. Let us assume that we are certain that at 11:00 a.m. there will be ten aircraft in parking positions at an airport. This situation is assigned a level of presumption of 1. In the same manner, we can also estimate that due to disturbances in the airline schedule, there will be between eight and fourteen aircraft in parking positions at the airport at 11:00 a.m. This situation is assigned a level of presumption of 0. The estimation that there will be between nine and twelve aircraft in parking positions can be given a level of presumption of 0.6. As we have seen, the estimated number of aircraft in parking positions can be expressed by confidence intervals, with each interval characterized by a level of presumption. Let us denote this level of presumption by α. In the above example, estimates are expressed in the following confidence intervals and levels of presumption: $[8, 14]$, $\alpha = 0$; $[9, 12]$, $\alpha = 0.6$; $[10, 10]$, $\alpha = 1$.

Figure 1.27 shows fuzzy number **A**. The confidence interval corresponding to level of presumption α is denoted as $[a_1^\alpha, a_2^\alpha]$.

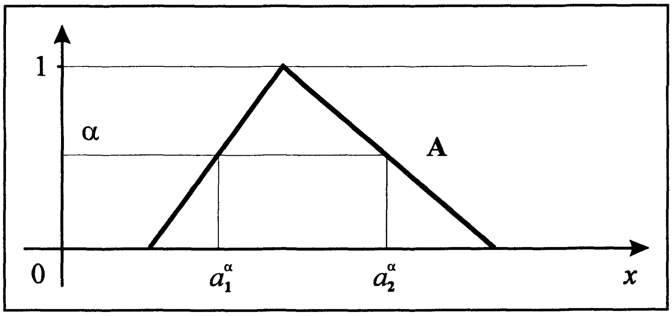

Figure 1.27. Fuzzy number **A** with its confidence interval and level of presumption

1.18. ADDING FUZZY NUMBERS

Before going into the addition of fuzzy numbers, we will briefly discuss how to add confidence intervals. Let us note the following two confidence intervals: $X = [x_1, x_2]$ and $Y = [y_1, y_2]$. If $m \in [x_1, x_2]$ and $n \in [y_1, y_2]$, then

$$m + n \in [x_1 + y_1,\ x_2 + y_2] \tag{1.104}$$

The left and right boundaries of the sum of the confidence intervals are, respectively, equal to the sum of the left boundaries and the sum of the right boundaries of the confidence intervals being added. The addition of confidence intervals can be symbolically expressed as follows:

$$X\,(+)Y = [x_1,\, x_2]\,(+)\,[y_1,\, y_2] = [x_1 + y_1,\, x_2 + y_2] \qquad (1.105)$$

Fuzzy numbers are added in the same manner as confidence intervals. Since every confidence interval of a fuzzy number has a specific level of presumption, when adding fuzzy numbers, the confidence intervals with the same levels of presumption are added. This addition is made for every level of presumption.

Let us denote the fuzzy numbers to be added by **X** and **Y**. The confidence intervals of these numbers corresponding to the level of presumption α are denoted by X_α and Y_α. Based in the above, we can write

$$X_\alpha\,(+)\,Y_\alpha = [x_1^\alpha, x_2^\alpha]\,(+)\,[y_1^\alpha, y_2^\alpha] = [x_1^\alpha + y_1^\alpha, x_2^\alpha + y_2^\alpha] \qquad (1.106)$$

Example 1.17. Direct flights are flown between cities A and C. While elaborating the airline schedule for the upcoming season, the possibility is considered of making stopovers in city B when flying from city A to city C (*Figure 1.28*).

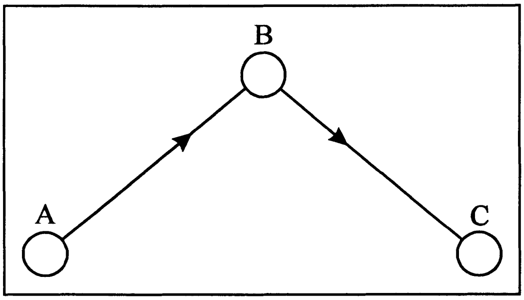

Figure 1.28. Flight from city A to city C with stopover in city B

Let us assume that we have subjectively estimated the daily number of passengers between the cities A and B as follows: $\mathbf{Q_1}$ is approximately 200; $\mathbf{Q_2}$ is approximately 400; $\mathbf{Q_3}$ is approximately 300. Fuzzy sets $\mathbf{Q_1}$, $\mathbf{Q_2}$ and $\mathbf{Q_3}$ depict the fuzzy numbers shown in *Figure 1.29.*

As can be seen from *Figure 1.29,* the level of presumption $\alpha = 0$ corresponds to the following confidence intervals of fuzzy numbers: $Q_{10} = [150, 250]$; $Q_{20} = [350, 450]$; $Q_{30} = [250, 350]$. Fuzzy numbers $\mathbf{Q_1}$, $\mathbf{Q_2}$, and $\mathbf{Q_3}$ represent triangular fuzzy numbers. In the general case, fuzzy numbers do not have a triangular shape, although triangular fuzzy numbers are most often encountered in examples.

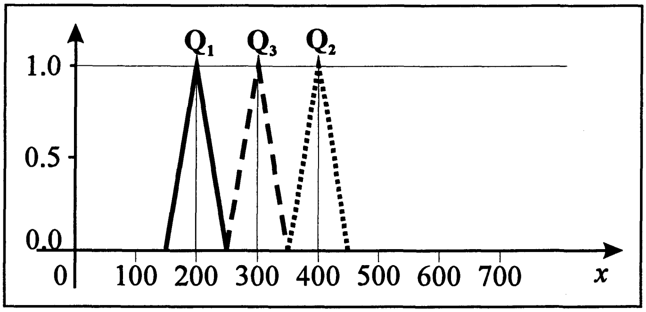

Figure 1.29. Fuzzy numbers $\mathbf{Q_1}$, $\mathbf{Q_2}$, and $\mathbf{Q_3}$

Let us return to our problem and estimate the number of passengers who will appear in flights between city A and city B and between city B and city C. The numbers of passengers traveling between these cities are also fuzzy numbers. Let us denote them by **X** and **Y**, respectively. It is clear that

$$\mathbf{X} = \mathbf{Q}_1 + \mathbf{Q}_2 \qquad \mathbf{Y} = \mathbf{Q}_2 + \mathbf{Q}_3 \tag{1.107}$$

Considering the coordinates of the corner points of the triangular fuzzy numbers shown in *Figure 1.29*, after determining the equations of the corresponding lines it is very easy to establish the membership functions of fuzzy numbers $\mathbf{Q_1}$, $\mathbf{Q_2}$, and $\mathbf{Q_3}$. These membership functions are

$$\mu_{\mathrm{Q}_1}(x) = \begin{cases} 0, & x \le 50 \\ \dfrac{x}{50} - 3, & 150 \le x \le 200 \\ -\dfrac{x}{50} + 5, & 200 \le x \le 250 \\ 0, & x \ge 250 \end{cases} \tag{1.108}$$

$$\mu_{\mathrm{Q}_3}(x) = \begin{cases} 0, & x \le 250 \\ \dfrac{x}{50} - 5, & 250 \le x \le 300 \\ -\dfrac{x}{50} + 7, & 300 \le x \le 350 \\ 0, & x \ge 350 \end{cases} \tag{1.109}$$

$$\mu_{Q_2}(x) = \begin{cases} 0, & x \le 350 \\ \dfrac{x}{50} - 7, & 350 \le x \le 400 \\ -\dfrac{x}{50} + 9, & 400 \le x \le 450 \\ 0, & x \ge 450 \end{cases} \tag{1.110}$$

We can see that the membership function of fuzzy number $\mathbf{Q_1}$ is described using the equations of the two lines. When we make an α - cut (for any level of presumption α), the left boundary of the corresponding confidence interval belongs to the first of these two lines. The right boundary of the confidence interval is found in the other line. In other words, the equations of the lines that define the membership function of fuzzy number $\mathbf{Q_1}$ can be written as follows:

$$\alpha = \left(\frac{q_{1_1}^{(\alpha)}}{50}\right) - 3 \qquad \alpha = -\left(\frac{q_{1_2}^{(\alpha)}}{50}\right) + 5 \tag{1.111}$$

The left and right boundaries of the confidence interval for the level of presumption α equal:

$$q_{1_1}^{(\alpha)} = 50\alpha + 150 \qquad q_{1_2}^{(\alpha)} = -50\alpha + 250 \tag{1.112}$$

The confidence interval $Q_{1\alpha}$ of fuzzy number $\mathbf{Q_1}$ with the level of presumption α equals $Q_{1\alpha} = [50\alpha+150, -50\alpha+250]$.

The confidence intervals of fuzzy numbers $\mathbf{Q_2}$ and $\mathbf{Q_3}$ are defined in the same manner $Q_{2\alpha} = [50\alpha+350, -50\alpha+450]$ and $Q_{3\alpha} = [50\alpha+250, -50\alpha+350]$.

Since

$$\mathbf{X} = \mathbf{Q}_1 + \mathbf{Q}_2 \qquad \mathbf{Y} = \mathbf{Q}_2 + \mathbf{Q}_3 \tag{1.113}$$

then

$$X_\alpha = Q_{1\alpha} \,(+)\, Q_{2\alpha} = [100\alpha + 500, -100\alpha + 700] \tag{1.114}$$

Thus, the left and right boundaries of fuzzy number $\mathbf{X}$ equal

$$x_1^{(\alpha)} = 100\alpha + 500 \qquad x_2^{(\alpha)} = -100\alpha + 700 \tag{1.115}$$

The equations of the lines representing the left and right boundaries of fuzzy number **X** are

$$\alpha = \left(\frac{x_1^{(\alpha)}}{100}\right) - 5 \qquad \alpha = -\left(\frac{x_2^{(\alpha)}}{100}\right) + 7 \tag{1.116}$$

We can finally write the membership function of fuzzy number **X**:

$$\mu_X(x) = \begin{cases} 0, & x \le 500 \\ \dfrac{x}{100} - 5, & 500 \le x \le 600 \\ -\dfrac{x}{100} + 7, & 600 \le x \le 700 \\ 0, & x \ge 700 \end{cases} \tag{1.117}$$

In the same fashion, for fuzzy number **Y** we have

$$Y_\alpha = Q_{2\alpha} (+) Q_{3\alpha} = [100\alpha + 600, -100\alpha + 800] \tag{1.118}$$

and

$$y_1^{(\alpha)} = 100\alpha + 600 \qquad y_2^{(\alpha)} = -100\alpha + 800 \tag{1.119}$$

The equations of the lines representing the left and right boundaries of fuzzy number **Y** are

$$\alpha = \left(\frac{y_1^{(\alpha)}}{100}\right) - 6 \qquad \alpha = -\left(\frac{y_2^{(\alpha)}}{100}\right) + 8 \tag{1.120}$$

The membership function of fuzzy number **Y** is

$$\mu_Y(x) = \begin{cases} 0, & x \le 600 \\ \dfrac{x}{100} - 6, & 600 \le x \le 700 \\ -\dfrac{x}{100} + 8, & 700 \le x \le 800 \\ 0, & x \ge 800 \end{cases} \tag{1.121}$$

Fuzzy numbers **X** and **Y** represent the expressions "approximately 600" and "approximately 700" passengers. The membership functions of fuzzy numbers $\mathbf{Q_1}$, $\mathbf{Q_2}$, $\mathbf{Q_3}$, **X**, and **Y** are shown in *Figure 1.30.*

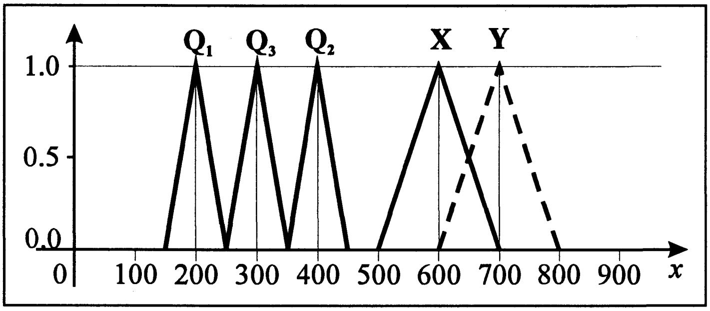

Figure 1.30. Membership functions of fuzzy numbers $\mathbf{Q_1}$, $\mathbf{Q_2}$, $\mathbf{Q_3}$, **X**, and **Y**

1.19. SUBTRACTING FUZZY NUMBERS

Let us first discuss the subtraction of confidence intervals. We note confidence intervals $X = [x_1, x_2]$ and $Y = [y_1, y_2]$. Let $m \in [x_1, x_2]$ and $n \in [y_1, y_2]$. Then

$$m - n \in [x_1 - y_2,\ x_2 - y_1] \tag{1.122}$$

We can write that the difference between confidence interval X and confidence interval Y is

$$X(-)Y = [x_1, x_2](-)[y_1, y_2] = [x_1 - y_2, x_2 - y_1] \tag{1.123}$$

When subtracting confidence intervals, the largest subtrahend value is subtracted from the smallest minuend value. The smallest subtrahend value is also subtracted from the largest minuend value.

Fuzzy numbers are subtracted just like confidence intervals. When subtracting fuzzy numbers, confidence intervals with the same level of presumption are subtracted. This is done for all the levels of presumption. Let us denote by **X** and **Y** the fuzzy numbers to be subtracted. The confidence intervals corresponding to the level of presumption α are denoted by X_α and Y_α. Based in the above, we can write

$$X_\alpha - Y_\alpha = [x_1^\alpha, x_2^\alpha] - [y_1^\alpha, y_2^\alpha] = [x_1^\alpha - y_2^\alpha, x_2^\alpha - y_1^\alpha] \tag{1.124}$$

Example 1.18. Many papers in literature are devoted to the vehicle routing problem. One of the most important algorithms in designing a vehicle route is the Clarke-Wright "savings" algorithm. There are many modification to this algorithm. Clarke and Wright (1964) based their algorithm in calculating the savings in cost or distance covered that was achieved when two nodes in a network are linked into one route instead of each node being served individually. Let us note *Figure 1.31*.

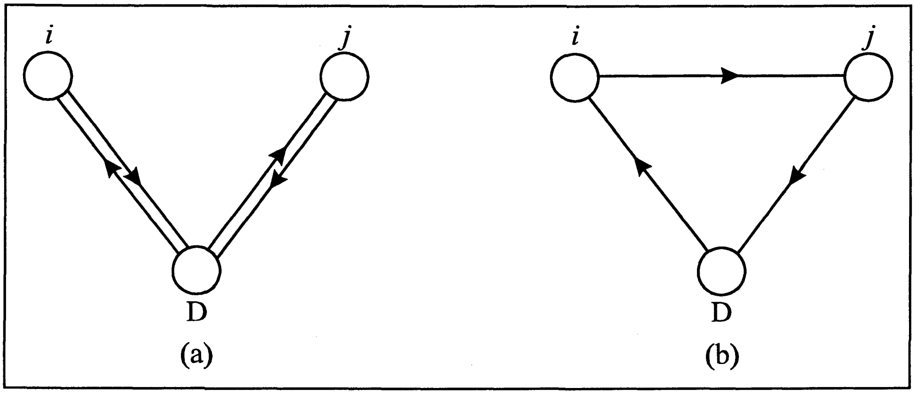

Figure 1.31. Depot D and nodes i and j

Figure 1.31(a) presents the case when each node is served by a vehicle individually. *Figure 1.31(b)* presents the case when the same vehicle serves both node i and node j. In this case, the vehicle departs from depot D, serves node i, goes to node j, and then returns to the depot. This clearly shortens the distance covered and decreases costs. Let us denote by $s(i,j)$ the savings achieved by joining nodes i and j into one route. We assume that all distances are symmetrical, i.e. the distance from i to j, $d(i,j)$, equals distance $d(j, i)$ from j to i. Since the operating costs are a function of the distance covered, these costs equal $c(i,j) = c(j, i)$. The savings in cost $s(i,j)$ realized when nodes i and j are joined into one route equal:

$$s(i,j) = \left[2c(\mathrm{D},i) + 2c(\mathrm{D},j)\right] - \left[c(\mathrm{D},i) + c(i,j) + c(\mathrm{D},j)\right] \qquad (1.125)$$

$$s(i,j) = c(\mathrm{D},i) + c(\mathrm{D},j) - c(i,j) \qquad (1.126)$$

We assume that we are not able to precisely determine costs, thus we can attribute fuzzy numbers to all costs. We can write that

$$\mathbf{S}(i,j) = \mathbf{C}(\mathrm{D},i) + \mathbf{C}(\mathrm{D},j) - \mathbf{C}(i,j) \qquad (1.127)$$

where **S**(i,j), **C**(D, i), **C**(D,j), and **C**(i,j) are fuzzy numbers. In order to simplify the expression, let us introduce the following substitutions:

$$\mathbf{S}(i,j) = \mathbf{S} \quad \mathbf{C}(\mathrm{D},i) = \mathbf{A} \quad \mathbf{C}(\mathrm{D},j) = \mathbf{B} \quad \mathbf{C}(i,j) = \mathbf{C} \tag{1.128}$$

We can write that

$$\mathbf{S} = \mathbf{A} + \mathbf{B} - \mathbf{C} \tag{1.129}$$

The sum of fuzzy numbers **A** and **B** is denoted by **E**. Savings **S** equal

$$\mathbf{S} = \mathbf{E} - \mathbf{C} \tag{1.130}$$

Fuzzy numbers **A**, **B**, and **C** are shown in *Figure 1.32*.

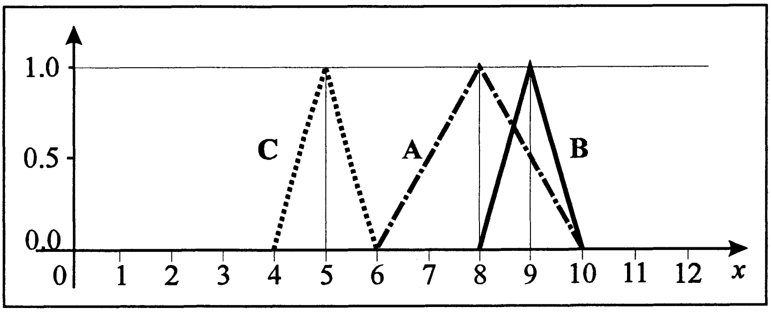

Figure 1.32. Fuzzy numbers **A**, **B**, and **C**

As we can see from *Figure 1.32*, the level of presumption $\alpha = 0$ corresponds to the following confidence intervals of fuzzy numbers **A**, **B**, and **C**: $A_0 = [6, 10]$, $B_0 = [8, 10]$, $C_0 = [4, 6]$.

Considering the coordinates of the corner points of the triangular fuzzy numbers shown in *Figure 1.32*, after determining the corresponding line equations, the following membership functions of fuzzy numbers **A**, **B**, and **C** are established:

$$\mu_A(x) = \begin{cases} 0, & x \le 6 \\ \dfrac{x}{2} - 3, & 6 \le x \le 8 \\ -\dfrac{x}{2} + 5, & 8 \le x \le 10 \\ 0, & x \ge 10 \end{cases} \tag{1.131}$$

$$\mu_B(x) = \begin{cases} 0, & x \le 8 \\ x-8, & 8 \le x \le 9 \\ -x+10, & 9 \le x \le 10 \\ 0, & x \ge 10 \end{cases} \tag{1.132}$$

$$\mu_C(x) = \begin{cases} 0, & x \le 4 \\ x-4, & 4 \le x \le 5 \\ -x+6, & 5 \le x \le 6 \\ 0, & x \ge 6 \end{cases} \tag{1.133}$$

The equations of the lines that determine the membership function of fuzzy set **A** can be written as follows:

$$\alpha = \left(\frac{a_1^{(\alpha)}}{2}\right) - 3 \qquad \alpha = -\left(\frac{a_2^{(\alpha)}}{2}\right) + 5 \tag{1.134}$$

where $a_1(\alpha)$ and $a_2(\alpha)$ are, respectively, the left and right boundaries of the confidence interval for the level of presumption α.

These boundaries equal

$$a_1^{(\alpha)} = 2\alpha + 6 \qquad a_2^{(\alpha)} = -2\alpha + 10 \tag{1.135}$$

The confidence interval A_α of fuzzy number **A** with the level of presumption α equals $A_\alpha = [2\alpha+6, -2\alpha+10]$.

The confidence intervals B_α and C_α of fuzzy numbers **B** and **C** are obtained in the same way $B_\alpha = [\alpha+8, -\alpha+10]$ and $C_\alpha = [\alpha+4, -\alpha+6]$.

Since

$$E_\alpha = A_\alpha \,(+)\, B_\alpha \tag{1.136}$$

then

$$E_\alpha = [3\alpha + 14, -3\alpha + 20] \tag{1.137}$$

The left and right boundaries of fuzzy number **E** for the level of presumption α are

$$e_1^{(\alpha)} = 3\alpha + 14 \qquad e_2^{(\alpha)} = -3\alpha + 20 \tag{1.138}$$

The equations of the lines representing the left and right boundaries of fuzzy number **E** are

$$\alpha = \left(\frac{e_1^{(\alpha)}}{3}\right) - \frac{14}{3} \qquad \alpha = -\left(\frac{e_2^{(\alpha)}}{3}\right) + \frac{20}{3} \tag{1.139}$$

The membership function of fuzzy number **E** equals

$$\mu_{\mathbf{E}}(x) = \begin{cases} 0, & x \le 14 \\ \frac{x}{3} - \frac{14}{3}, & 14 \le x \le 17 \\ -\frac{x}{3} + \frac{20}{3}, & 17 \le x \le 20 \\ 0, & x \ge 20 \end{cases} \tag{1.140}$$

Since **S** = **E** - **C**, then

$$\mathrm{S}_\alpha = \mathrm{E}_\alpha \,(-)\, \mathrm{C}_\alpha \tag{1.141}$$

which is

$$\begin{aligned} \mathrm{S}_\alpha &= [3\alpha + 14, -3\alpha + 20]\,(-)\,[\alpha + 4, -\alpha + 6] \\ \mathrm{S}_\alpha &= [(3\alpha + 14) - (\alpha + 6), (-3\alpha + 20) - (\alpha + 4)] \\ \mathrm{S}_\alpha &= [4\alpha + 8, -4\alpha + 16] \end{aligned} \tag{1.142}$$

The left and right boundaries of fuzzy number **S** with the level of presumption α equal

$$s_1^{(\alpha)} = 4\alpha + 8 \qquad s_2^{(\alpha)} = -4\alpha + 16 \tag{1.143}$$

The equations of the lines representing the left and right boundaries of fuzzy number **S** are as follows:

$$\alpha = \left(\frac{s_1^{(\alpha)}}{4}\right) - 2 \qquad \alpha = -\left(\frac{s_2^{(\alpha)}}{4}\right) + 4 \tag{1.144}$$

The membership function of fuzzy number **S** is

$$\mu_S(x) = \begin{cases} 0, & x \le 8 \\ \dfrac{x}{4} - 2, & 8 \le x \le 12 \\ -\dfrac{x}{4} + 4, & 12 \le x \le 16 \\ 0, & x \ge 16 \end{cases} \tag{1.145}$$

The membership functions of fuzzy numbers **A**, **B**, **C**, **E**, and **S** are shown in *Figure 1.33*.

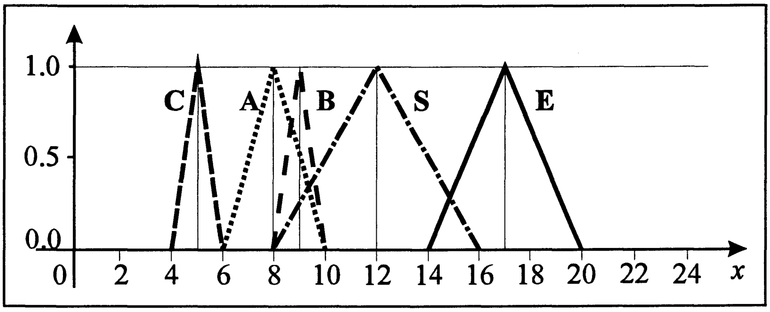

Figure 1.33. Membership functions of fuzzy numbers **A**, **B**, **C**, **E**, and **S**

1.20. MULTIPLYING AND DIVIDING FUZZY NUMBERS

Let us note fuzzy numbers **X** and **Y** in Set R^+. For a level of presumption of α, the multiplication and division of the confidence intervals X_α and Y_α are defined as

$$X_\alpha\,(\cdot)\,Y_\alpha = \left[x_1^{(\alpha)}, x_2^{(\alpha)}\right](\cdot)\left[y_1^{(\alpha)}, y_2^{(\alpha)}\right] = \left[x_1^{(\alpha)}\,y_1^{(\alpha)}, x_2^{(\alpha)}\,y_2^{(\alpha)}\right] \tag{1.146}$$

$$X_\alpha\,(\div)\,Y_\alpha = \left[x_1^{(\alpha)}, x_2^{(\alpha)}\right](\div)\left[y_1^{(\alpha)}, y_2^{(\alpha)}\right] = \left[\frac{x_1^{(\alpha)}}{y_2^{(\alpha)}}, \frac{x_2^{(\alpha)}}{y_1^{(\alpha)}}\right] \tag{1.147}$$

Example 1.19. Let fuzzy number **C** be the product of fuzzy numbers **A** and **B**: **C** = **A B**.

Let fuzzy numbers **A** and **B** correspond to the following membership functions:

$$\mu_A(x) = \begin{cases} 0, & x \le 1 \\ x-1, & 1 \le x \le 2 \\ -x+3, & 2 \le x \le 3 \\ 0, & x \ge 3 \end{cases} \tag{1.148}$$

$$\mu_B(x) = \begin{cases} 0, & x \le 4 \\ x-4, & 4 \le x \le 5 \\ -x+6, & 5 \le x \le 6 \\ 0, & x \ge 6 \end{cases} \tag{1.149}$$

Based in the membership function, we can ascertain that for fuzzy number **A** with the level of presumption α, the following relations are satisfied:

$$\alpha = a_1^{(\alpha)} - 1 \qquad \alpha = -a_2^{(\alpha)} + 3 \tag{1.150}$$

$$a_1^{(\alpha)} = \alpha + 1 \qquad a_2^{(\alpha)} = -\alpha + 3 \tag{1.151}$$

In the same manner, confidence interval B_α with the level of presumption α is given as $B_\alpha = [\alpha+4, -\alpha+6]$. The product of confidence intervals A_α and B_α is

$$\begin{aligned} A_\alpha (\cdot) B_\alpha &= [\alpha+1, -\alpha+3] (\cdot) [\alpha+4, -\alpha+6] = \\ &= [(\alpha+1)(\alpha+4), (-\alpha+3)(-\alpha+6)] = \\ &= [\alpha^2+5\alpha+4, \alpha^2-9\alpha+18] \end{aligned} \tag{1.152}$$

Values x, which define the left boundary of fuzzy number **AB**, equal $\alpha^2 + 5\alpha + 4$, while corresponding values x, which define the right boundary, equal $\alpha^2 - \alpha + 18$. After solving the equations

$$\alpha^2 + 5\alpha + 4 = x \qquad \alpha^2 - 9\alpha + 18 = x \tag{1.153}$$

we get

$$\alpha = \frac{\left(-5 + \sqrt{9+4x}\right)}{2} \qquad \alpha = \frac{\left(9 - \sqrt{9+4x}\right)}{2} \tag{1.154}$$

The membership function of fuzzy number **AB** equals

$$\mu_C(x) = \mu_{AB}(x) = \begin{cases} 0, & x \le 4 \\ \dfrac{-5+\sqrt{9+4x}}{2}, & 4 \le x \le 10 \\ \dfrac{9-\sqrt{9+4x}}{2}, & 10 \le x \le 18 \\ 0, & x \ge 18 \end{cases} \tag{1.155}$$

Fuzzy numbers **A**, **B**, and **C** are shown in *Figure 1.34*.

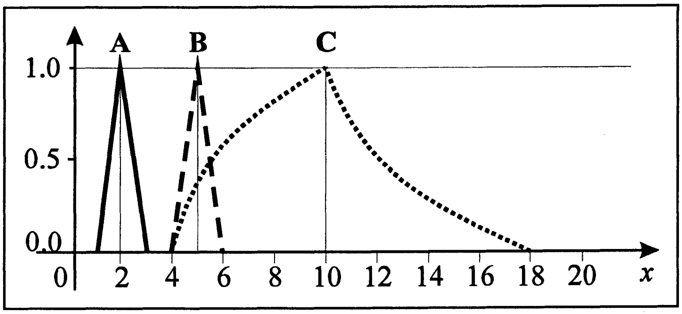

Figure 1.34. Multiplying fuzzy numbers **A** and **B**

Example 1.20. An airline company has estimated that in the coming season it can expect costs of "approximately \$4,000" per flight between city A and city B. It has also estimated that in average each flight can be expected to have "approximately 100 passengers." Let us determine expected costs per passenger. Let us denote by **X** and **Y**, respectively, fuzzy sets for the estimated costs per flight and the estimated number of passengers per flight. Fuzzy numbers **X** and **Y** are shown in *Figure 1.35* and *Figure 1.36.*

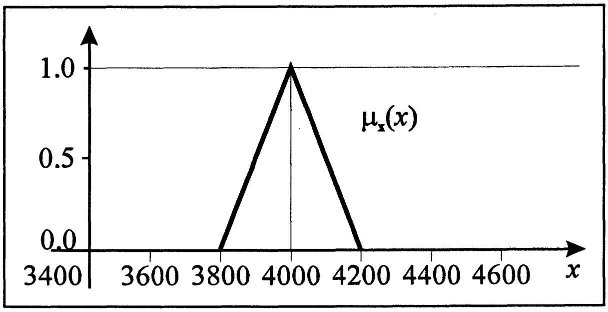

Figure 1.35. Fuzzy number **X** (estimated costs per flight)

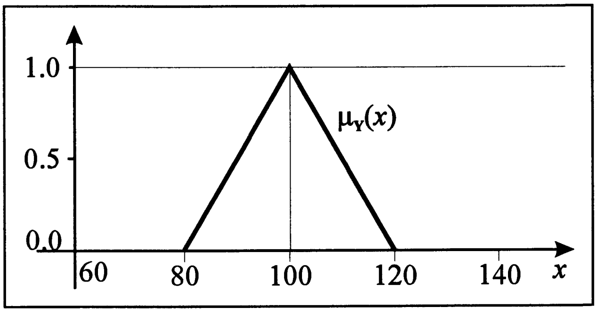

Figure 1.36. Fuzzy number **Y** (estimated number of passengers per flight)

The membership functions of fuzzy numbers **X** and **Y** are

$$\mu_X(x) = \begin{cases} 0, & x \le 3800 \\ \dfrac{x}{200} - 19, & 3800 \le x \le 4000 \\ -\dfrac{x}{200} + 21, & 4000 \le x \le 4200 \\ 0, & x \ge 4200 \end{cases} \tag{1.156}$$

$$\mu_Y(x) = \begin{cases} 0 & x \le 80 \\ \dfrac{x}{20} - 4, & 80 \le x \le 120 \\ -\dfrac{x}{20} + 6, & 100 \le x \le 120 \\ 0, & x \ge 120 \end{cases} \tag{1.157}$$

Since

$$\alpha = \left(\frac{x_1^{(\alpha)}}{200}\right) - 19 \qquad \alpha = -\left(\frac{x_2^{(\alpha)}}{200}\right) + 21 \tag{1.158}$$

then

$$X_\alpha = [200\alpha + 3800, -200\alpha + 4200] \tag{1.159}$$

It can be shown in the same manner that

$$Y_\alpha = [20\alpha + 80, -20\alpha + 120] \tag{1.160}$$

The division of the confidence intervals X_α and Y_α is defined as

$$X_\alpha\ (\div)\ Y_\alpha = = \left(\frac{200\alpha + 3800}{-20\alpha + 120}, \frac{-200\alpha + 4200}{-20\alpha + 80}\right) \tag{1.161}$$

It is easily shown that the obtained membership function by dividing the fuzzy numbers **X** and **Y** equals

$$\mu_{X(\div)Y}(x) = \begin{cases} 0, & x \le \dfrac{95}{3} \\ \dfrac{6x - 190}{x + 10}, & \dfrac{95}{3} \le x \le 40 \\ \dfrac{-4x + 120}{x + 10} + 6, & 40 \le x \le \dfrac{105}{2} \\ 0, & x \ge \dfrac{105}{2} \end{cases} \tag{1.162}$$

1.21. MULTIPLYING A FUZZY NUMBER BY A CONSTANT

Let us note fuzzy number **X** in set R and constant $c \in R_0^+$. For a level of presumption of α, the product of constant c and confidence interval X_α equals

$$cX_\alpha = [c, c]\,(\cdot)\,[x_1^{(\alpha)}, x_2^{(\alpha)}] = [cx_1^{(\alpha)}, cx_2^{(\alpha)}] \tag{1.163}$$

The operation of multiplying fuzzy number **X** by a constant is often shown using membership functions:

$$\mu_{cX}(x) = \mu_X\left(\frac{x}{c}\right) \qquad \forall x \in R \tag{1.164}$$

Example 1.21. Let travel costs be a linear function of travel time. When travel time is subjectively estimated and expressed by a fuzzy number, travel costs also represent a fuzzy number. Let there be a dependence between travel time and travel costs in the form

$$\mathbf{C} = a\mathbf{T} \tag{1.165}$$

where **C** is fuzzy number representing travel costs, **T** is fuzzy number representing subjectively estimated travel time, and a is a constant. Let us assume that a = 2 and that fuzzy number **T** corresponds to the following membership functions:

$$\mu_{\mathbf{T}}(x) = \begin{cases} 0, & x \le 10 \\ \dfrac{x}{5} - 2, & 10 \le x \le 15 \\ -\dfrac{x}{5} + 4, & 15 \le x \le 20 \\ 0, & x \ge 20 \end{cases} \tag{1.166}$$

Since

$$\mu_{a\mathbf{X}}(x) = \mu_{\mathbf{T}}\left(\frac{x}{a}\right) \tag{1.167}$$

$$\mu_{\mathbf{C}}(x) = \mu_{2\mathbf{T}}(x) = \mu_{\mathbf{T}}\left(\frac{x}{2}\right) \tag{1.168}$$

then

$$\mu_{\mathbf{C}}(x) = \begin{cases} 0, & x \le 20 \\ \dfrac{x}{10} - 2, & 20 \le x \le 30 \\ -\dfrac{x}{10} + 4, & 30 \le x \le 40 \\ 0, & x \ge 40 \end{cases} \tag{1.169}$$

Fuzzy numbers **T** and **C** are shown in *Figure 1.37*.

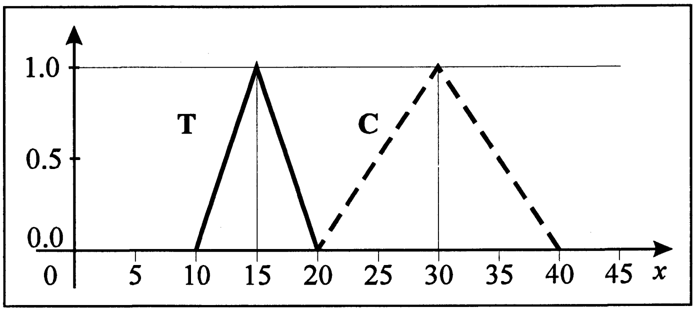

Figure 1.37. Fuzzy numbers **T** and **C**

1.22. TRIANGULAR AND TRAPEZOIDAL FUZZY NUMBERS

Triangular fuzzy numbers are a special class of fuzzy numbers; their name is derived from the shape of their membership functions. *Figure 1.38* shows a typical triangular number, most often presented in the form:

$$\mathbf{A} = (a_1, a_2, a_3) \tag{1.170}$$

where a_1 is lower (left) boundary of the triangular fuzzy number, a_2 is number corresponding to the highest level of presumption, and a_3 is upper (right) boundary of the fuzzy number.

The membership function of fuzzy number **A** is

$$\mu_{\mathbf{A}}(x) = \begin{cases} 0, & x \le a_1 \\ \left(\dfrac{x - a_1}{a_2 - a_1}\right), & a_1 \le x \le a_2 \\ \left(\dfrac{a_3 - x}{a_3 - a_2}\right), & a_2 \le x \le a_3 \\ 0, & x \ge a_3 \end{cases} \tag{1.171}$$

Triangular fuzzy numbers are added, subtracted, multiplied and divided in the same way as other fuzzy numbers. Using the previously explained methodological procedure for adding fuzzy numbers, we can easily show that the sum of fuzzy number $\mathbf{A} = (a_1, a_2, a_3)$ and fuzzy number $\mathbf{B} = (b_1, b_2, b_3)$ is:

$$\mathbf{A}(+)\mathbf{B} = (a_1, a_2, a_3)(+)(b_1, b_2, b_3) = (a_1 + b_1, a_2 + b_2, a_3 + b_3) \tag{1.172}$$

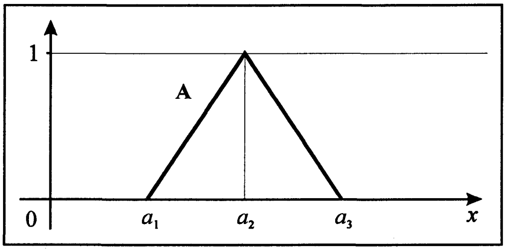

Figure 1.38. Triangular fuzzy number **A**

Fuzzy number $\mathbf{A}^{-} = (-a_3, -a_2, -a_1)$ is the opposite of fuzzy number $\mathbf{A} = (a_1, a_2, a_3)$. *Figure 1.39* presents fuzzy number $\mathbf{A} = (2, 3, 5)$ and its opposite number $\mathbf{A}^{-} = (-5, -3, -2)$.

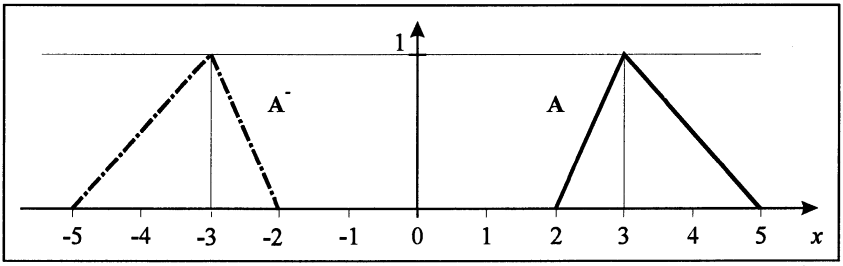

Figure 1.39. Fuzzy number **A** and its opposite number $\mathbf{A}^{-}$

Triangular fuzzy number **B** is subtracted from triangular fuzzy number **A** by adding fuzzy number **A** to fuzzy number $\mathbf{B}^{-}$. In other words:

$$\begin{aligned}\mathbf{A}(-)\mathbf{B} = \mathbf{A}(+)\mathbf{B}^{-} &= (a_1, a_2, a_3)(+)(-b_3, -b_2, -b_1) = \\ &= (a_1 - b_3, a_2 - b_2, a_3 - b_1)\end{aligned} \tag{1.173}$$

It can easily be shown that the multiplication or division of two triangular fuzzy numbers does not produce a triangular fuzzy number.

A triangular fuzzy number is multiplied by a constant in the same way that other fuzzy numbers are multiplied by a constant. Using the previously presented methodology for multiplying a fuzzy number by a constant, we get

$$c\mathbf{A} = c(a_1, a_2, a_3) = (ca_1, ca_2, ca_3) \tag{1.174}$$

where c is a positive constant, and **A** is a triangular fuzzy number.

The result of multiplying a triangular fuzzy number by a negative constant produces

$$b\mathbf{A} = b(a_1, a_2, a_3) = (ba_3, ba_2, ba_1) \tag{1.175}$$

where b is a negative constant, and **A** is a triangular fuzzy number.

Trapezoidal fuzzy numbers are characterized by membership functions in the form of a trapezoid. *Figure 1.40* presents trapezoidal number $\mathbf{A} = (a_1, a_2, a_3, a_4)$. It can easily be shown that the sum and difference of trapezoidal fuzzy numbers **A** and **B** are

$$\begin{aligned}\mathbf{A}(+)\mathbf{B} &= (a_1, a_2, a_3, a_4)(+)(b_1, b_2, b_3, b_4) = \\ &= (a_1 + b_1, a_2 + b_2, a_3 + b_3, a_4 + b_4)\end{aligned} \tag{1.176}$$

$$\begin{aligned}\mathbf{A}(-)\mathbf{B} &= (a_1, a_2, a_3, a_4)(-)(b_1, b_2, b_3, b_4) = \\ &= (a_1 - b_4, a_2 - b_3, a_3 - b_2, a_4 - b_1)\end{aligned} \tag{1.177}$$

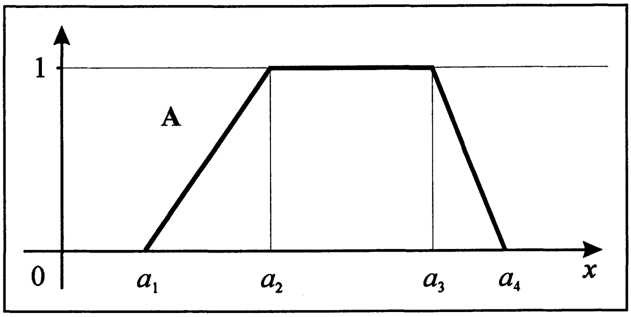

Figure 1.40. Trapezoidal fuzzy number $\mathbf{A} = (a_1, a_2, a_3, a_4)$

1.23. KAUFMANN AND GUPTA'S METHOD FOR COMPARING FUZZY NUMBERS

The problem of comparing fuzzy numbers has been the subject of many papers. This problem was discussed by Yager (1981), Adamo (1981), Baas and Kwakernaak (1977), Dubois and Prade (1983), and Kaufmann and Gupta (1988). Important reviews by Zimmermann (1986) and Bortolan and Degani (1985) analyzed various methods of comparing fuzzy numbers. We will briefly discuss Kaufmann and Gupta's method, which was primarily chosen for its simplicity. This method is based in the concept of "removal" of a fuzzy number, which will be explained through the algorithmic steps of Kaufmann and Gupta's algorithm.

Kaufmann and Gupta's (1988) method to compare fuzzy numbers is comprised of the following steps:

Step 1: Compare the "removal" of the numbers. If a conclusion can be made based in this comparison, the algorithm is ended. If not, go to Step 2.

Step 2: Compare the values that correspond to the highest grade of membership. If a number order can be determined after this comparison, the algorithm is ended. If a conclusion cannot be made after comparing the highest grades of membership, go to Step 3.

Step 3: Compare the length of the fuzzy numbers' bases.

Let us note *Figure 1.41*. "Left removal" $R_l(\mathbf{A},k)$ of fuzzy number **A** compared to real number k consists of the surface between real number k and the left side of fuzzy number **A**. The "right removal" $R_r(\mathbf{A},k)$ of fuzzy number **A** is defined as the surface between real number k and the right side of fuzzy number **A**. The "removal" of fuzzy number **A** compared to real number k is defined as

$$\mathrm{R}(\mathbf{A},k) = \frac{\left(\mathrm{R}_l(\mathbf{A},k) + \mathrm{R}_r(\mathbf{A},k)\right)}{2} \tag{1.178}$$

Figure 1.41 shows the "left removal," "right removal," and "removal" of fuzzy number **A** compared to real number k.

When fuzzy number **A** has a triangular shape and when $k = 0$, it can be easily shown that the "removal" of fuzzy number **A** equals

$$\mathrm{R}(\mathbf{A},k) = \frac{\left(a_1 + 2a_2 + a_3\right)}{4} \tag{1.179}$$

Fuzzy number **A** is smaller than fuzzy number **B** if

$$\mathrm{R}(\mathbf{A},k) < \mathrm{R}(\mathbf{B},k) \tag{1.180}$$

When $\mathrm{R}(\mathbf{A},k) = \mathrm{R}(\mathbf{B},k)$, the second algorithmic step must be used and the values of the highest grades of membership must be compared. Let x^*_{A} and x^*_{B} denote the highest grades of membership in fuzzy sets **A** and **B**. Fuzzy number **A** is smaller than fuzzy number **B** if

$$x^*_{\mathrm{A}} < x^*_{\mathrm{B}} \tag{1.181}$$

When $\mathrm{R}(\mathbf{A},k) = \mathrm{R}(\mathbf{B},k)$ and $x^*_{\mathrm{A}} = x^*_{\mathrm{B}}$, the third step must be used, which compares the bases of the fuzzy numbers. Fuzzy number **A** is less than fuzzy number **B** if the base of fuzzy number **A** is less than the base of fuzzy number **B**.

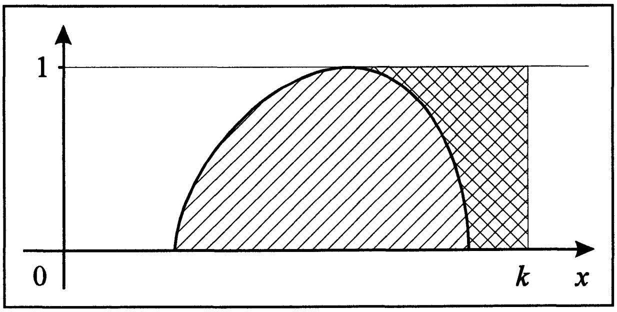

Figure 1.41. Fuzzy number **A**'s "removal" compared to real number k

Example 1.22. Let us compare triangular fuzzy number **A** = (6,9,11) with triangular fuzzy number **B** = (7,8,12) shown in *Figure 1.42.*

Since these are triangular fuzzy numbers, their "removals" equal

$$\mathrm{R}(\mathbf{A},k)=\frac{(a_1+2a_2+a_3)}{4}=\frac{(6+18+11)}{4}=8.75 \tag{1.182}$$

$$\mathrm{R}(\mathbf{A},k)=\frac{(a_1+2a_2+a_3)}{4}=\frac{(7+16+12)}{4}=8.75 \tag{1.183}$$

Since the shift of fuzzy numbers **A** and **B** are equal, their highest grades of membership must be compared. Since

$$x_{\mathrm{A}}^{*}=9>8=x_{\mathrm{B}}^{*} \tag{1.184}$$

we conclude that fuzzy number **A** is greater than fuzzy number **B**.

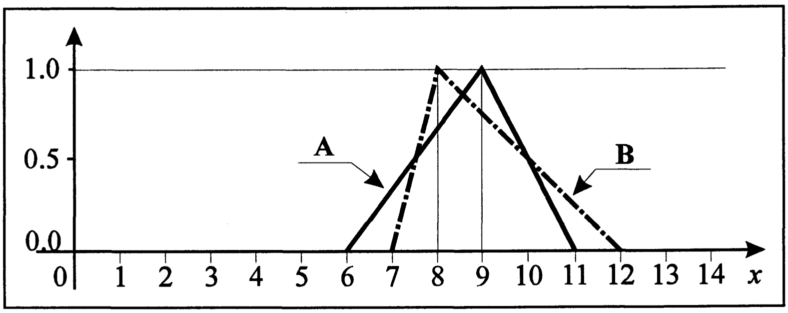

Figure 1.42. Membership function of fuzzy numbers **A** and **B**

1.24. THE BASIC ELEMENTS OF FUZZY SYSTEMS

When managing complex industrial and traffic systems, an experienced operator sometimes achieves better results then classical automatic control systems. The control strategies of the operator can often be formulated in terms of numerous descriptive rules, which are simple for a manual processing and execution but complicated when it comes to the use of classical algorithms. These difficulties arise from the fact that, when describing different decisions made at various stages of a process, human beings prefer to use qualitative expressions instead of quantitative ones. The qualitative or fuzzy nature of the human way of deciding has encouraged engineers to make an attempt at developing fuzzy systems that would control processes.

Fuzzy logic systems (fuzzy systems, fuzzy control systems, fuzzy logic controllers, fuzzy controllers, fuzzy expert systems, fuzzy associative memories) arise from the desire to model human experience, intuition, and behavior in decision making (Zimmermann, 1991). The original idea of the possibility of a decision making based in imprecise, qualitative data by combining descriptive linguistic rules through fuzzy logic was first introduced by Zadeh. The combination of imprecise logic rules in a single control strategy is called by Zadeh (1973) *approximate or fuzzy reasoning*. The fuzzy logic was first presented by Zadeh (1965, 1973) as a means of processing vague, linguistic information. Zadeh concluded that the conventional quantitative techniques of the system analysis are essentially inappropriate for humanistic systems and expressed it in the form of the incompatibility principle: "As the complexity of a system increases, our ability to make precise and yet significant statements about it behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics."

The following quotation (Zadeh, 1973) is also illustrative of the extensive application of fuzzy logic: "The closer one looks at a real-world problem, the fuzzier becomes its solution."

Fuzzy logic was developed to provide soft algorithms for data processing that can both make inferences about imprecise data and use the data. It enables the variables to partially (up to a certain degree) belong to a particular set and at the same time makes use of the generalizations of conventional Bull logic operators in data processing. Fuzzy logic is widely used in intelligent control systems (Brown and Harris, 1994) to make inferences about vague rules describing the relation between imprecise, qualitative linguistic estimations of the inputs and outputs of a system. These control rules usually represent the knowledge of an expert and provide an easily comprehended pattern of knowledge representation. Many conventional expert systems can be described by means of a set of binary rules (Quinlan, 1993), but the discontinuities occurring at the exit of a system, when various rules are activated, do not resemble human behavior, where usually a smooth relation exists between cause and consequence. Fuzzy rules include descriptive expressions such as small, medium, or large used to categorize the linguistic (fuzzy) input and output variables. A set of fuzzy rules, describing the control strategy of the operator, forms a fuzzy control algorithm, that is, approximate reasoning algorithm, whereas the linguistic expressions are represented and quantified by fuzzy sets. The main advantage of this approach is the possibility of introducing and using rules from experience, intuition, heuristics, and the fact that a model of the process is not required.

The development of fuzzy logic and the application of fuzzy systems to managing processes have a long history. By introducing the concept of approximate reasoning Zadeh managed to show that vague logical statements allow for the formation of algorithms by which vague inferences can be derived from vague data. Zadeh noted that his method was particularly useful in studying complex humanistic systems. The basic fuzzy set theories were developed in the 1960s, and the first applications were investigated by Mamdani (1974). A number of papers in which the approximate reasoning was applied to industrial systems were written by a group of researchers at Queen Mary College, London, headed by Mamdani. For example, Mamdani and Assilian (1975) apply *fuzzy logical controller* to managing a steam engine, demonstrating that the controller of a physical system can be purely logical. The linguistic rules that qualitatively express a control strategy are commonly obtained from a skilled operator of a plant. Among pioneering works in the field of fuzzy controllers the papers by Mamdani (1974), Kickert and Van nauta Lemke (1976), ∅stergaard (1976), Ruthersford and Carter (1976), and Tong (1976) must be mentioned. Mamdani and Assilian investigated the problem of controlling a small laboratory steam engine. Kickert and Van Nauta Lemke developed a fuzzy control algorithm for an experimental warm water plant. Ruthersford and Carter, with the help of their algorithm, controlled the raw mix permeability of a full-scale sinter strand, while ∅stergaard developed a fuzzy algorithm to control a heat exchanger. Tong (1976) controlled a pressurized tank containing liquid. Tong (1977) also performed the control of engineering review of fuzzy systems. In the 1970s, Mamdani and his researchers proposed the first self-controlling fuzzy controller (Procyk and Mamdani, 1979).

In the 1980s, only a small number of researchers continued work in Great Britain and the United States, while at the same time numerous applications of control by fuzzy logic were developed in Japan. The application of fuzzy control to industrial and transportation processes is still developing in Europe and Japan. To quote Ken Self (1990), "Japan has caught fuzzy logic fever." Realizing the potentials offered by fuzzy logic, large Japanese companies such as Toshiba, Sanyo, Canon, Matsushita, Mitsubishi, Sony, and Subaru have produced various products that have a fuzzy controller as a component part. The characteristics of these products exceed those of similar products which are made by other manufacturers and are not based in fuzzy logic. Fuzzy logic has been successfully used in managing elevator operations and rail traffic. Video camcorders, washing machines, televisions, water heaters, and vacuum cleaners are examples of successful products whose work is based in the principles of fuzzy logic. For the application of fuzzy logic in engineering a recently published monograph by Ross (1995) is of outstanding interest.

Fuzzy reasoning is also used in solving certain traffic and transportation problems. Pappis and Mamdani (1977) were the first to use the principles of fuzzy logic to control the isolated signalized intersection of two one-way streets. Sasaki and Akiyama (1986, 1987, 1988) developed a model to control freeway traffic. Chen et al. (1990) developed a model to control traffic in a freeway on-ramp. The Teodorovic and Kikuchi model (1990) based in fuzzy logic deals with the problem of route choice. Teodorovic and Babic (1993) developed an approximate reasoning model to control air traffic flows. The problem of dispatching vessels in river traffic was modeled by Vukadinovic and Teodorovic (1994) using fuzzy logic.

Throughout the literature, various aspects of fuzzy logic have been discussed in numerous papers. In the field of engineering, published papers by Mendel (1995) and Jang and Sun (1995) are of particular consequence.

The question of defining a fuzzy logic system naturally arises. Mendel (1995) explains the concept of a fuzzy logic system (FLS) in the following way: "In general, a FLS is a non-linear mapping of an input data (feature) vector into a scalar output (the vector output case decomposes into a collection of independent multi-input/single-output systems)." Let us note that a fuzzy logic system most often maps crisp inputs into crisp outputs.

The basic elements of each fuzzy logic system are rules, fuzzifier, inference engine, and defuzzifier (*Figure 1.43*).

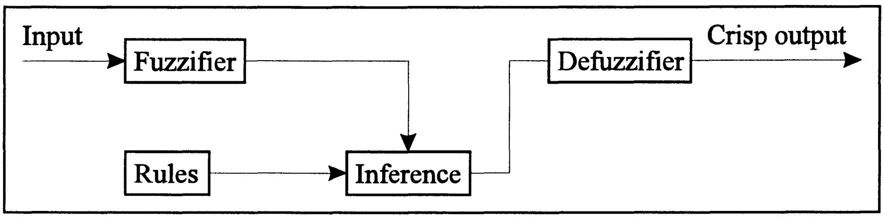

Figure 1.43. Basic elements of a fuzzy logic system

The input data are most commonly crisp values. The task of a fuzzifier is to map crisp numbers into fuzzy sets. Let us note that in certain cases the input consists of fuzzy, i.e., linguistic variables. Fuzzy or linguistic variables are those variables whose numerical values are fuzzy numbers or whose domain is composed of linguistic variables i.e., expressions rather than numerical variables. If x represents a fuzzy variable, $T(x)$ denotes a set of linguistic values (categories) of variable x. For example, for the fuzzy variable x = *Time of service*, the set of different linguistic values can be $T(x)$ = {very small, small, medium, large, very large}. The linguistic values are quantified by fuzzy sets. The numerical values of these variables can be hours (minutes) belonging to all fuzzy sets to a various degree. The numerical values can be continuous or discreet. Fuzzy variables are found by measuring, observing, and, what is very important and quite frequent, by subjective estimation based in experience and intuition. In the following discussions, the cases related to crisp input as well as those related to fuzzy input will be considered in greater detail.

Fuzzy rules can conveniently represent the knowledge of experienced operators used in control. These rules are arrived at either by verbalizing the operator's expertise or by conducting a carefully composed survey. This involves a long and close cooperation with a number of experienced operators, which can be very difficult to achieve. When it can be positively claimed that the operator is consciously or unconsciously using the rules in managing, the rules can be recorded by observing the operator's behavior. In other words, the rules can be formulated by using the observed decisions (input/output numerical data) of the operator.

Fuzzy rule (fuzzy implication) takes the following form:

If x is **A**, then y is **B**

where **A** and **B** represent linguistic values quantified by fuzzy sets defined over universes of discourse X and Y. The first part of the rule "x is **A**" is the premise or the condition preceding the second part of the rule "y is **B**" which constitutes the consequence or conclusion. Basically, this expression describes the relation between the variables x and y, i.e., a fuzzy rule is defined as a binary fuzzy relation R in the Cartesian space $X \times Y = \{(x,y)\}$. Each element (x,y) of the fuzzy relation is associated with the corresponding membership grade $\mu_{\mathbf{R}}(x,y)$. The binary fuzzy relation **R** can be treated as the fuzzy set defined in space $X \times Y$ characterized by a two-dimensional membership function $\mu_{\mathbf{R}}(x,y)$.

The fact after "If" is called the *premise* or *hypothesis* or *antecedent*. From this fact, another fact called *conclusion* or *consequent* (the fact after "then") can be inferred.

Note the following simple rule:

If variable x is **Big**, then variable y is **Small**

The membership functions of the fuzzy sets **Big** and **Small** are shown in *Figure 1.44*.

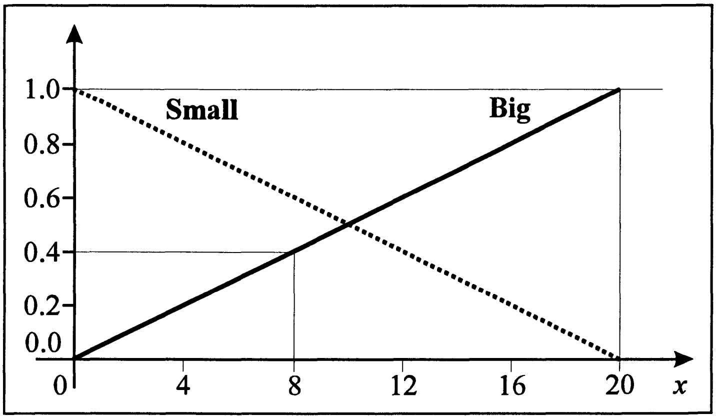

Figure 1.44. Membership functions of fuzzy sets **Big** and **Small**

As can be seen, the output of variable y is conditioned by the input of fuzzy variable x. Let us assume that we have obtained data or that we have estimated that the value of input variable $x = 8$. From *Figure 1.44*, we can see that $x = 8$ corresponds to the grade of membership in fuzzy set **Big** of 0.4. This grade of membership is actually the "value of the truth" contained in the claim that the value of input variable $x = 8$ can be treated as **Big**. Since output variable y is conditioned by input variable x, we conclude that the claim that variable y is **Small** is only as true as the truth in the claim that input variable x is **Big**.

Fuzzy reasoning (approximate reasoning) involves the transformation of a group of fuzzy rules into fuzzy relations in order to achieve a result. Fuzzy reasoning is an inference procedure, i.e., the way of generating the conclusion from the premises when the linguistic expressions are quantified by fuzzy sets. The inference engine of the fuzzy logic system maps fuzzy sets into fuzzy sets. The inference engine "handles the way in which rules are combined" (Mendel, 1995). There are a number of various inferential procedures in literature.

The basic inference rule in classical logic is *modus ponens*:

Premise 1 (fact):	x is A,
Premise 2 (rule):	If x is A, then y is B,
Consequence (conclusion):	y is B.

The expressions A and B are precisely defined. However, when the expressions A, B, A', and B' are described by fuzzy sets, the inference procedure becomes a *general modus ponens* or approximate reasoning:

Premise 1 (fact):	x is **A'**,
Premise 2 (rule):	If x is **A**, then y is **B**,
Consequence (conclusion):	y is **B'**.

If fuzzy implication $\mathbf{A} \Rightarrow \mathbf{B}$ is expressed by fuzzy relation **R** in the Cartesian space $X \times Y = \{(x,y)\}$, then fuzzy set **B'** can be expressed by the following formula:

$$\mu_{\mathbf{B}'}(y) = \max_{x} \min\left\{\mu_{\mathbf{A}'}(x), \mu_{\mathbf{R}}(x,y)\right\} \tag{1.185}$$

$$\mu_{\mathbf{B}'}(y) = w \wedge \mu_{\mathbf{B}}(y) \tag{1.186}$$

In order to determine fuzzy set **B'** the degree of match of sets **A** and **A'**, w, is first determined as the maximal value of the intersection of the sets (*Figure 1.45*). Fuzzy set **B'** is determined by the intersection of fuzzy sets **B** and w, whereas w represents the firing strength of the rule or the degree to which the rule is fulfilled.

Within one rule the following two claims must often be satisfied:

If x is **A** and y is **B**, then z is **C**.

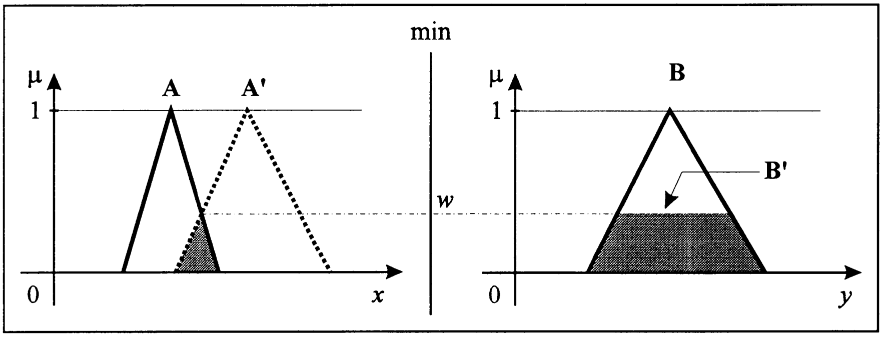

Figure 1.45. Approximate reasoning for a single rule

The inference procedure thus takes the following form:

Premise 1 (fact):	x is **A'** and y is **B'**,
Premise 2 (rule):	If x is **A** and y is **B**, then z is **C**,
Consequence (conclusion):	z is **C'**.

As in the previous case, the degree of match of sets **A** and **A'**, w_1, is first determined, then the degree of match of sets **B** and **B'**, w_2, and the resulting fuzzy set **C'** is obtained by the intersection of fuzzy set **C** and $w = \min(w_1, w_2)$, whereas w represents the firing strength of the rule or the degree to which the rule is fulfilled (*Figure 1.46*). In this inference procedure the operator min represents operation "and."

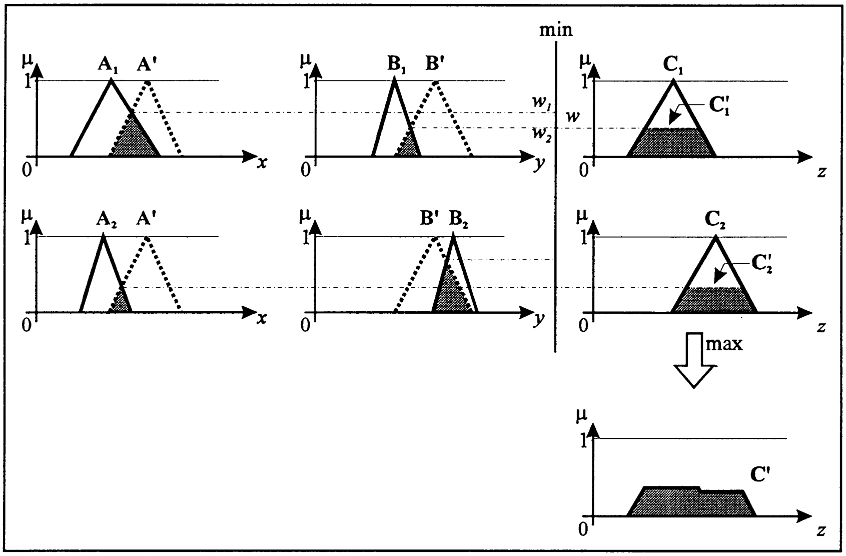

Figure 1.46. Approximate reasoning by max-min composition for two rules

A set of rules is transformed by the union of fuzzy relations. When operator max represents operation "or," the composition of inference is called the "max-min composition." The first model of the fuzzy system where the approximate reasoning is performed by max-min composition was proposed by Mamdani (Mamdani and Assilian, 1975). *Figure 1.46* shows the approximate reasoning performed by max-min composition when the algorithm is composed of two rules, i.e., the calculation of the output numerical value z when the input numerical values are x and y.

A number of papers in the field of fuzziness begin with the following assertion: "When the membership function of the fuzzy set is known let us assume that the min operator is the corresponding model of the intersection of fuzzy sets." However, to use the fuzzy sets theory in modelling real phenomena, i.e., if the developed models are to be applicable in real situations, it is necessary to perform a kind of empirical verification or confirmation of such assumptions (Zimmermann, 1987).

The following set of rules which is called the fuzzy rule base is a typical example of an approximate reasoning algorithm:

If x is **Big**, then y is **Small**
or
If x is **Medium**, then y is **Medium**
or
If x is **Small**, then y is **Big**

Our known input variable value ($x = 8$) must go through all the above-defined rules: we must determine how much truth is contained in the claim that $x = 8$ is **Big**, how much truth in $x = 8$ is **Medium**, and, finally, how much truth there is in the claim that $x = 8$ is **Small.** After going through all the rules in the approximate reasoning algorithm, all the possible values of output variable y are associated with a grade of membership.

The last step in the approximate reasoning algorithm is defuzzification, i.e., choosing one value for the output variable. Using the algorithm of fuzzy reasoning a fuzzy set is obtained as the output result in the Mamdani model with particular membership grades of possible numerical values of the output variable. By defuzzification the fuzzy information is compressed and given by a representative numerical information. In the original paper by Mamdani (Mamdani and Assilian, 1975) the center of gravity of the resulting fuzzy set represents the output numerical value.

In most applications an analyst or decision maker looks at the grades of membership of individual output variable values, and chooses one of them according to the following criteria: "the smallest maximal value," "the largest maximal value," "center of gravity," "mean of the range of maximal values," and so on. (*Figure 1.47*).

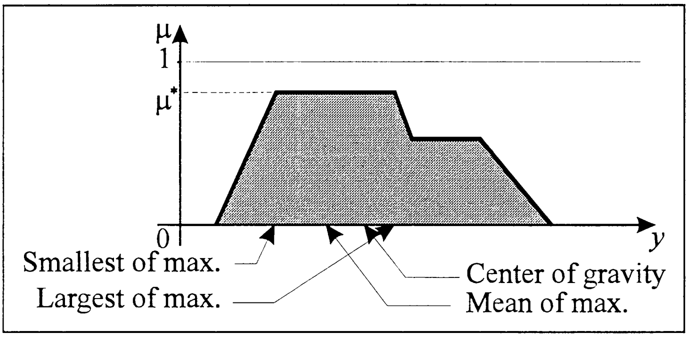

Figure 1.47. Approximate reasoning using max-min composition for two rules

A great deal of fuzzy information may seem to have been lost when the resulting fuzzy set is represented by a numerical value. The only way of analyzing these methods thus far has been experimentation and estimation of the obtained results.

Let us note that in classical expert systems rules are exclusively obtained from human experts. In the case of fuzzy rule-based systems, the formation of the rule base can be made by human experts, according to numerical data or by combining numerical data and human experts. The combination of numerical information obtained from measurement and linguistic information elicited from human experts is particularly interesting and important. This case will be considered more thoroughly in the discussions to follow.

The development of models of fuzzy logic most often requires several iterations. The first step defines the set of rules and the corresponding membership functions of the input/output variables. After looking at the results, corrections are made to the rules and/or membership functions, if necessary. Then the model is tested once again with the modified rules and/or membership functions, and so on.

A more detailed way of applying fuzzy logic will be given in the consideration of various traffic problems solved by fuzzy reasoning. The following chapter presents models based in fuzzy logic used to solve traffic and transportation problems, along with numerical examples.

1.24.1. Graphical interpretation of fuzzy logic inference

A graphical interpretation of a fuzzy logic inference can considerably help us to reach a deeper understanding of the nature of fuzzy logic inference. Graphical techniques associated with fuzzy logic inference may be particularly useful in the case of a small number of rules and a "manual" application of fuzzy logic (without the aid of a computer).

Let us consider a set of fuzzy rules containing three input variables x_1, x_2 and x_3 and one output variable y.

Rule 1: If x_1 is $\mathbf{P}_{11}$ and x_2 is $\mathbf{P}_{12}$ and x_3 is $\mathbf{P}_{13}$, then y is $\mathbf{Q}_1$,
or
Rule 2: If x_1 is $\mathbf{P}_{21}$ and x_2 is $\mathbf{P}_{22}$ and x_3 is $\mathbf{P}_{23}$, then y is $\mathbf{Q}_2$,
or

Rule k: If x_1 is $\mathbf{P}_{k1}$ and x_2 is $\mathbf{P}_{k2}$ and x_3 is $\mathbf{P}_{k3}$, then y is $\mathbf{Q}_k$.

The given rules are interrelated by the conjunction *or*. Such a set of rules is called a *disjunctive system of rules* and assumes the satisfaction of at least one rule. It is assumed that membership functions of fuzzy sets $\mathbf{P}_{k1}$ and $\mathbf{P}_{k3}$ (k = 1, 2, ..., K) are of a triangular shape, whereas membership functions of fuzzy sets $\mathbf{P}_{k2}$ and $\mathbf{Q}_k$ (k = 1, 2, ..., K) are of a trapezoidal shape. Let us note *Figure 1.48* in which our disjunctive system of rules is presented.

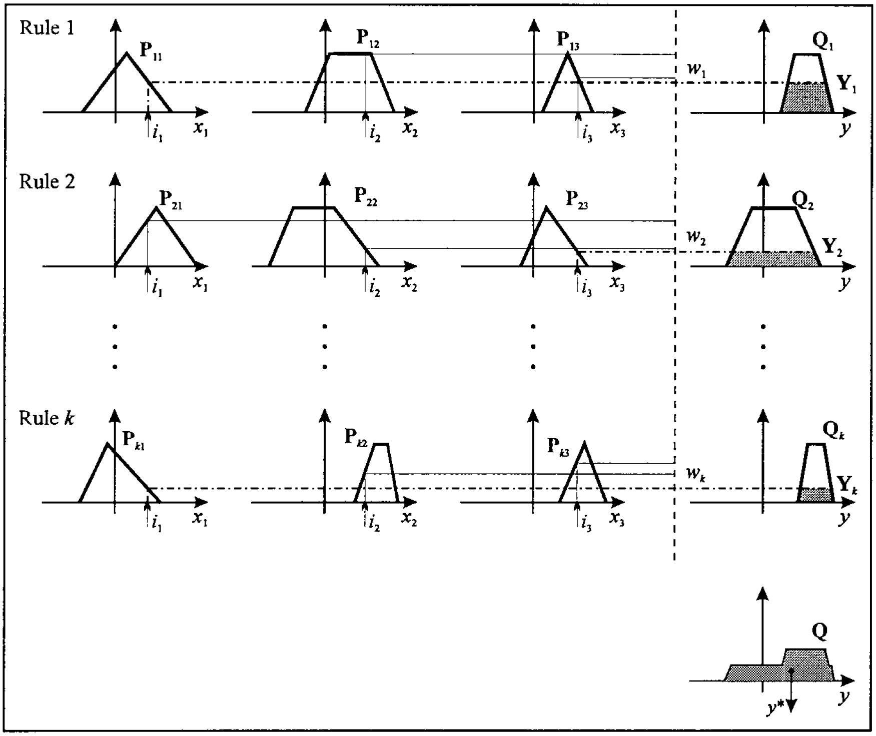

Figure 1.48. Graphical interpretation of a disjunctive system of rules

Let the values i_1, i_2, and i_3, respectively, taken by input variables x_1, x_2, and x_3, be known. (Depending in the context of the problem considered, these values can be obtained in the basis of the collected data, through measurements, or by an expert evaluation). In the considered case, the values i_1, i_2, and i_3 are crisp.

Figure 1.48 also represents the membership function of output **Q**. This membership function takes the following form:

$$\mu_{\mathbf{Q}}(y) = \max_k \left\{\min\left[\mu_{\mathbf{P}_{k1}}(i_1), \mu_{\mathbf{P}_{k2}}(i_2), \mu_{\mathbf{P}_{k3}}(i_3)\right]\right\}, k = 1,2,\ldots,\mathrm{K} \qquad (1.187)$$

whereas fuzzy set **Q** representing the output is actually a fuzzy union of all the rule contributions $\mathbf{Y}_1$, $\mathbf{Y}_2$, ..., $\mathbf{Y}_k$, i.e.:

$$\mathbf{Q} = \mathbf{Y}_1 \cup \mathbf{Y}_2 \cup \ldots \cup \mathbf{Y}_k \tag{1.188}$$

It is clear that

$$\mu_{\mathrm{Q}}(y) = \max\left\{\mu_{\mathrm{Y}_1}(y), \mu_{\mathrm{Y}_2}(y), \ldots, \mu_{\mathrm{Y}_k}(y)\right\} \tag{1.189}$$

Let us try to explain more thoroughly the manner in which, for the known values i_1, i_2, and i_3 of input variables x_1, x_2, and x_3, the corresponding value y^* of output variable y is calculated.

Consider rule 1, which reads as follows:

If x_1 is $\mathbf{P}_{11}$ and x_2 is $\mathbf{P}_{12}$ and x_3 is $\mathbf{P}_{13}$, then y is $\mathbf{Q}_1$

The value $\mu_{\mathbf{P}11}(i_1)$ indicates how much truth is contained in the claim that i_1 equals $\mathbf{P}_{11}$. Similarly, values $\mu_{\mathbf{P}12}(i_2)$ and $\mu_{\mathbf{P}13}(i_3)$, respectively, indicate the truth value of the claim that i_2 equals $\mathbf{P}_{12}$ and i_3 equals $\mathbf{P}_{13}$. Value w_1, which is equal to

$$w_1 = \min\left\{\mu_{\mathrm{P}_{11}}(i_1),\ \mu_{\mathrm{P}_{12}}(i_2),\ \mu_{\mathrm{P}_{13}}(i_3)\right\} \tag{1.190}$$

indicates the truth value of the claims that, simultaneously, i_1 equals $\mathbf{P}_{11}$, i_2 equals $\mathbf{P}_{12}$ and i_3 equals $\mathbf{P}_{13}$.

Since the conclusion contains as much truth as the premise, after calculating value w_1, the membership function of fuzzy set $\mathbf{Q}_1$ should be transformed. In this way, fuzzy set $\mathbf{Q}_1$ is transformed into fuzzy set $\mathbf{Y}_1$ (*Figure 1.49*).

Values w_2, w_3, ..., w_k are calculated in the same manner leading to the transformation of fuzzy sets $\mathbf{Q}_2$, $\mathbf{Q}_3$, ..., $\mathbf{Q}_k$ into fuzzy sets $\mathbf{Y}_2$,$\mathbf{Y}_3$, ..., $\mathbf{Y}_k$.

As this is a disjunctive system of rules, assuming the satisfaction of at least one rule, the membership function $\mu_{\mathrm{Q}}(y)$ of the output represents the outer envelope of the membership functions of fuzzy sets $\mathbf{Y}_1$, $\mathbf{Y}_2$, ..., $\mathbf{Y}_k$. The final value y^* of the output variable is arrived at upon defuzzification.

In some cases, the values taken by input variables are not crisp. In other words, inputs sometimes appear as fuzzy sets described by appropriate membership functions. The appearance of fuzzy sets may result from insufficiently precise measurements, approximately known data and subjective evaluation of particular parameters (*Figure 1.49*).

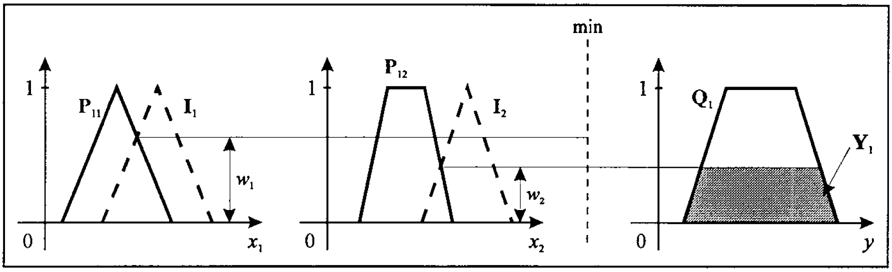

Figure 1.49. Graphical interpretation of fuzzy inputs

Figure 1.49 shows Rule 1 of a disjunctive system of rules. This rule reads as follows:

Rule 1: If x_1 is $\mathbf{P}_{11}$ and x_2 is $\mathbf{P}_{12}$, then y is $\mathbf{Q}_1$

Let the inputs be fuzzy, as shown in *Figure 1.49* by the appropriate membership functions of fuzzy sets $\mathbf{I}_1$ and $\mathbf{I}_2$. The intersections of these inputs and the appropriate membership functions for premises $\mathbf{P}_{11}$ and $\mathbf{P}_{12}$ are marked by hatched triangles shown in *Figure 1.49*. The maximum values w_1 and w_2 of these intersections represent, respectively, truth value that x_1 equals $\mathbf{P}_{11}$ and that x_2 equals $\mathbf{P}_{12}$. The value min $\{w_1, w_2\}$ indicates truth value contained in the claim that x_1 equals $\mathbf{P}_{11}$ and that, simultaneously, x_2 equals $\mathbf{P}_{12}$. The further procedure for calculating the values of the output variable is the same as in the case of calculating the crisp values of the input variable.

Chapter 2.

Presentation of Fuzzy Models Developed in Transportation Applications

2.1. VEHICLE ROUTING MODEL BASED ON THE RULES OF FUZZY ARITHMETIC

2.1.1. Basic assumptions of the vehicle routing problem

Vehicle routing problems appear in various transportation activities. Hundreds of papers have been published in world literature during the past three decades treating different aspects of the vehicle routing problem. Significant reviews were given by Larson and Odoni (1981), Bodin et al. (1983), Solomon and Desrosiers (1988), and Golden and Assad (1988).

Let us note depot D and nodes *n* shown in *Figure 2.1*. Vehicles leave the depot, serve nodes in the network, and on completion of their routes, return to the depot.

Every node is described by a certain *demand* (the amount to be delivered to the node or the amount to be picked up from the node). Other known values include the *coordinates* of the depot and nodes, the *distance* between all pairs of nodes, and the *capacity* of the vehicles providing service. The classical vehicle routing problem consists of finding the set of routes that minimizes transport costs.

In most vehicle routing models it is assumed that travel time, transport costs, and distance between pairs of nodes in the network are constant values known in advance. On the other hand, the travel time between two nodes in a network involves an uncertainty due to traffic conditions, type of driving, weather conditions, choice of streets, and so on. Using fuzzy arithmetic, Teodorovic and Kikuchi (1991a) developed a vehicle routing model containing uncertainty in terms of the travel time between individual nodes. We will briefly discuss the results achieved by Teodorovic and Kikuchi (1991a).

D. Teodorovič et al., *Traffic Control and Transport Planning:*

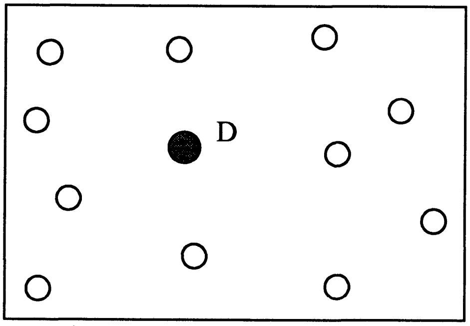

Figure 2.1. Depot D and nodes requiring service

Our subjective feeling regarding travel time is often not very precise. For example, the estimate is made that it takes "approximately 20 minutes" to go from one node to another. No one will claim that it takes 17 minutes when subjectively estimating travel time. Estimating travel time in this way is not the result of objective measurements but is a subjective estimation that differs among dispatchers and drivers. Travel time has often been treated as a random variable, and this required travel time measurements and the establishment of a certain probability density function. However, dispatchers-decision makers most often make a subjective estimate of travel time based on their experience and intuition, expressing the estimated travel time as "short," "long," "about 20 minutes," and so on.

Teodorovic and Kikuchi (1991a) treated travel time and transportation costs between two nodes in a network as fuzzy numbers. The vehicle routing algorithm developed by Teodorovic and Kikuchi assumed the following operational constraints:

1. The dispatcher knows the service requests in advance. Each request is characterized by a location and amount to be served.
2. All vehicles providing service have the same capacity, which is sufficient to serve at least one request.
3. All vehicles start their activity from the depot and return to it after completing their service. Vehicles may leave the depot several times during the day.
4. Travel times between pairs of nodes in the network are only approximately known.
5. Drivers must follow the route (sequence of nodes to be served) designed by the dispatcher or analyst. When stopping at successive nodes on the route, drivers are free to choose the streets they will take.

When travel time and transport costs are constant values, vehicle routing problems are most often solved using one of the large number of heuristic algorithms developed in the past three decades. Example 1.18 pointed out the classical Clarke-Wright "savings" algorithm (1964). Let us assume that initially we are using n vehicles and that we are dispatching one vehicle to each one of the nodes to be served. The transportation cost savings $s(i,j)$ achieved when one vehicle serves nodes i and j instead of each node being served by a separate vehicles equals

$$s(i,j) = c_{i,\mathrm{D}} + c_{\mathrm{D},j} - c_{i,j} \tag{2.1}$$

where D is depot, and $c_{i,j}$ are transportation costs between nodes i and j.

When the savings between all pairs of nodes have been made, the Clarke-Wright algorithm ranks the savings in descending order and then starts with the greatest savings and routes vehicles while bearing operating constraints in mind.

2.1.2. Calculating and ranking "savings" using fuzzy arithmetic

Let us denote by **T** the fuzzy set of subjectively estimated travel time between nodes i and j. In order to simplify the arithmetic operations, Teodorovic and Kikuchi (1991a) assumed that travel time **T** was a triangular fuzzy number (*Figure 2.2*). Triangular fuzzy number **T** is expressed as

$$\mathbf{T} = (t_1, t_2, t_3) \tag{2.2}$$

where t_1, t_2, and t_3 are the lower boundary, the value that corresponds to the highest grade of membership, and the upper boundary of fuzzy number **T**.

We assume that travel costs between any two nodes in the network are a linear function of time:

$$\mathbf{C} = a\mathbf{T} \tag{2.3}$$

where **C** is fuzzy number representing travel costs between two nodes whose travel time is fuzzy number **T**, and a is travel costs per unit of time.

Fuzzy travel costs **C** can also be represented by a triangular fuzzy number:

$$\mathbf{C} = (at_1, at_2, at_3) \tag{2.4}$$

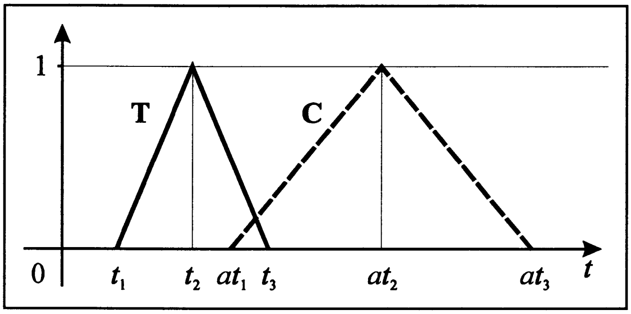

Figure 2.2 Fuzzy time and fuzzy travel costs

Since travel costs are fuzzy numbers, the "savings" are also triangular fuzzy numbers:

$$\mathbf{S}_{i,j} = \mathbf{C}_{i,\mathrm{D}} + \mathbf{C}_{\mathrm{D},j} - \mathbf{C}_{i,j} \tag{2.5}$$

The triangular fuzzy numbers representing savings can be calculated using fuzzy arithmetic (adding and subtracting fuzzy numbers). When the savings have been calculated (in the form of triangular fuzzy numbers), they must be ranked in descending order. Teodorovic and Kikuchi ranked the savings with Kaufmann and Gupta's method (1988).

Before ranking the calculated savings, they must be shifted along the x axis so that all calculated savings are positive. All savings $\mathbf{S}_1$, $\mathbf{S}_2$, ..., are shifted along the positive line of the x axis by value $\mathbf{r}$ (*Figure 2.3*):

$$\mathbf{S}_i(+)\mathbf{r} = \left(a_{1i} + \mathbf{r}, a_{2i} + \mathbf{r}, a_{31} + \mathbf{r}\right) \tag{2.6}$$

where $\mathbf{r}$=min (a_{1i}), $a_{1\mathrm{i}}$ is lower boundary of the ith saving, $\mathbf{S}_i = (\mathrm{a}_{1i}, \mathrm{a}_{2i}, \mathrm{a}_{3i})$.

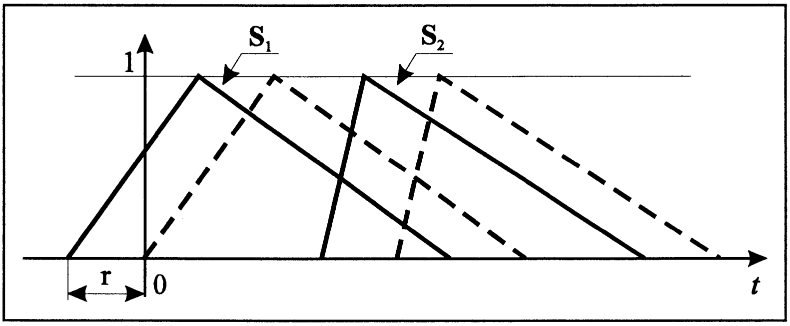

Figure 2.3. Shifting "savings"

2.1.3. Vehicle routing algorithm when travel times are fuzzy numbers

The algorithm to route vehicles when travel time and transport costs are treated as fuzzy numbers consists of the following algorithmic steps:

Step 1: Estimate travel time between all pairs in the network, and present them in the form of triangular fuzzy numbers.
Step 2: Using the functional dependence between travel time and transport costs, calculate transport costs between all pairs of nodes, and present them in the form of fuzzy numbers.
Step 3: Using fuzzy arithmetic, calculate the fuzzy savings.
Step 4: Shift the bases of all calculated fuzzy numbers along the positive line of the x axis.
Step 5: Rank the fuzzy savings using Kaufmann and Gupta's method.
Step 6: Route the vehicles using the classical Clarke-Wright algorithm.

Teodorovic and Kikuchi (1991a) tested their developed algorithm on a large number of numerical examples. During testing, in estimating travel times in the network, the effect of the level of uncertainty on the form of the vehicle routing was examined. The travel times between all nodes in the network were assumed to be triangular fuzzy numbers. The level of uncertainty used to estimate travel time was represented by the length of the base of the triangular fuzzy number. In other words, the greater the uncertainty in estimating travel time between two nodes in the network, the longer the base of the triangular fuzzy number representing travel time. Teodorovic and Kikuchi (1991a) expressed the level of uncertainty through parameter β, where $\beta = 0$ corresponds to the highest possible uncertainty, while $\beta = 1$ corresponds to the situation in which the exact travel time through the network is known with complete certainty. For a given parameter β, the lower and upper boundaries of the triangular fuzzy number are randomly chosen so as to be between βa_2 and $(2 - \beta)a_2$, with being a_2 the travel time with the greatest grade of membership. After generating all travel times using the algorithm, a set of vehicle routes is obtained that corresponds to a certain parameter β. Different sets of vehicle routes can be generated for different values of parameter β. *Figure 2.4* presents two generated sets of vehicle routes for different values of parameter β ($\beta = 0.1$, $\beta = 0.7$). Depending on the level of uncertainty used to estimate travel time in the network, a specific set of routes can be chosen to be used by the vehicles when serving the network.

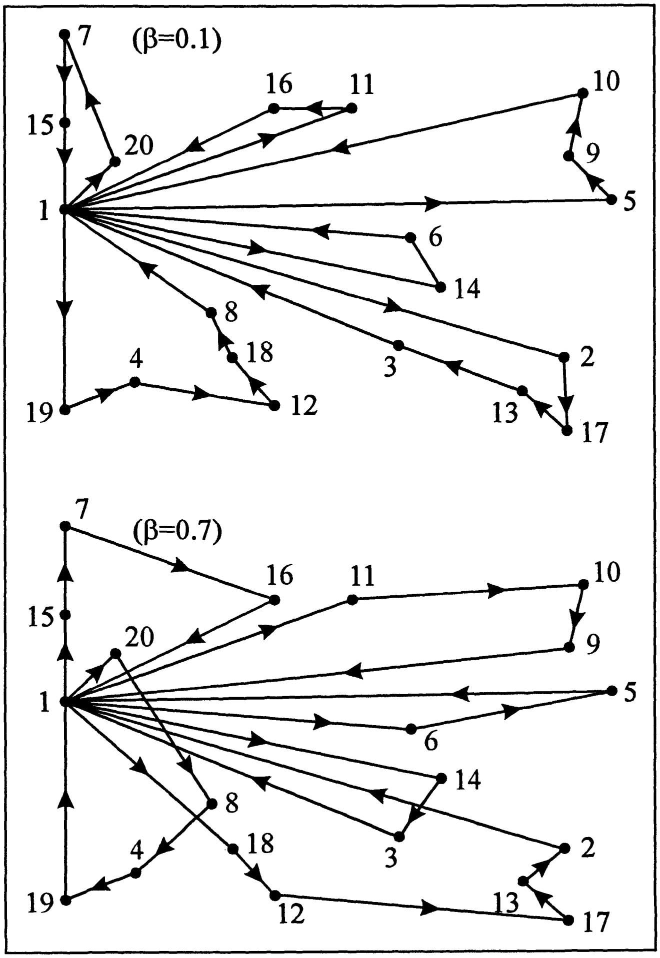

Figure 2.4. Generated sets of vehicle routes for different values of parameter β

2.2. SCHEDULING ALGORITHM FOR A STATIC DIAL-A-RIDE PROBLEM BASED ON FUZZY ARITHMETIC RULES

The dial-a-ride system of transport organization is used in some countries in the areas with low population density, as well as in big towns to transport the elderly and people with disabilities. There are several modifications to the dial-a-ride transport system. They share the following common features:

1. The user dials the dispatcher service of a transport organizer requesting a trip.
2. A trip request is characterized by the trip origin and destination as well as the time moment (or time interval) when the client is to be picked up.
3. Serving is usually carried out by vehicles of the same capacity.
4. When designing vehicle routes, the transport organizer tries to execute the scheduled rides by the smallest possible number of vehicles, to design a set of the shortest routes, to achieve the shortest possible waiting time, and so on.

There are two versions of the dial-a-ride problem: static and dynamic. The static version involves the collection of trip requests on a "today for tomorrow" basis. In this case, the transport organizer designs vehicle routes in the time interval ranging between collection of requests and execution of the requested rides. The routes designed for the next day remain unaltered throughout the serving.

In the dynamic version of the dial-a-ride problem the routes are designed in real time. In other words, once a new client has requested to be served, the dispatcher promptly redesigns the routes that is, tries to "insert" the new request into one of the existing routes. The "insertion" of a new client effects certain changes in the route. If it is impossible for a new client to be inserted into one of the existing routes, his request is rejected.

The dial-a-ride problem has been studied by a number of authors, most notably by Psaraftis (1983), Alfa (1986), Jaw et al. (1986), Rhee (1987), Solomon and Desrosiers (1988), and Kikuchi and Rhee (1989).

Kikuchi (1992) was the first to develop an algorithm based on the fuzzy sets theory to resolve the dial-a-ride problem. Let us briefly present the results achieved by Kikuchi (1992), who developed "a vehicle routing and scheduling method for time-window based, many-to-many, advanced reservation, multivehicle dial-a-ride problem." Kikuchi (1992) considered the dial-a-ride problem within the following operating environment:

1. In the transport network there is a depot from which vehicles depart and to which they return.
2. Each trip request is characterized by the trip origin, desired departure time, and time windows for the desired departure. (The desired time of pickup (and dropoff) of a passenger is assumed to be a fuzzy number (For example: "I would like to be picked up about 9 a.m.")).
3. Passengers with different origin-destination pairs may be transported together by the same vehicle.
4. Travel time between two nodes is represented by a fuzzy number. (The dispatcher subjectively estimates travel times according to traffic

conditions and the driver's characteristics (the manner of driving, the driver's choice of streets and paths). The dispatcher's subjective estimation of travel times is based on his experience and intuition).

5. Time windows within which the serving of passengers is to be started are of a "soft" type. This means that the vehicle is allowed to arrive before the time window but must depart within the time window.

As can be seen, Kikuchi (1992) was the first to introduce fuzziness in the dial-a-ride problem treating the desired time of vehicle arrival and the travel time between two nodes as fuzzy numbers (*Figure 2.5*).

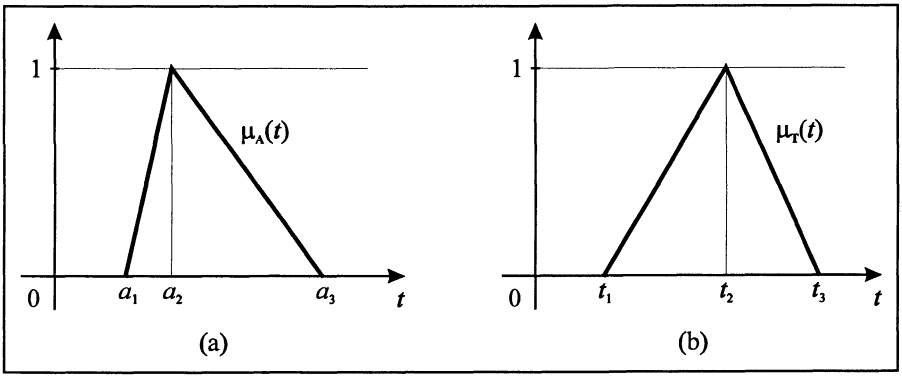

Figure 2.5. Membership functions of desired time of vehicle arrival (a) and travel time between two nodes (b)

Figure 2.5 represents the desired time of vehicle arrival at a certain node and the travel times between certain pair of nodes as triangular fuzzy numbers. Thus, for example, a_2 is the most desired time, whereas a_1 and a_3 represent the beginning and ending of the desired time range. The time window corresponds to the range between a_1 and a_3. The values of t_1 and t_3 correspond to the shortest and longest estimated travel times between the two observed nodes.

The algorithm we proceed to present is related to the problem of minimizing the number of vehicles required to meet a given set of travel requests.

As in the case of other routing problems, there again exist two possible approaches to the process of route designing: simultaneous and sequential. The simultaneous approach refers to the case of certain operations on routes (development of the initial routes, expansion of the routes, insertion of new requests into the route, and so on) being executed simultaneously for all the routes. The sequential approach involves the consideration of routes (development of the initial route, insertion of new requests, and so on) in sequences that is, one at a time.

In the case of certain dial-a-ride algorithms (Jaw et al., 1986; Alfa, 1986; Kikuchi and Rhee, 1989), the initial route is developed first and then the insertion of more trips is performed. Jaw et al. developed the initial routes for several vehicles simultaneously and then made the insertion of the remaining trips into any of the initial routes. Unlike them, Kikuchi (1992) first developed the initial route for one vehicle and then tried to insert into the initial route as many remaining trips as possible. Kikuchi (1992) began to develop the second initial route only when it was no longer possible for any new trip to be inserted into the first initial route. Kikuchi's work (1992) is characterized by sequential procedure (routes are designed one by one), a two-stage approach (the initial route is developed first and then an insertion is made), and fuzziness (desired times of departures and travel times are treated as fuzzy numbers).

In order to explain the first step of the algorithm, consisting of the development of the initial route, let us assume that all the given quantities are deterministic.

The initial route is developed by the greedy method. This method "develops the initial route by connecting trips such that the vehicle travels to the closest pickup point after dropping off the passenger of the previous trip." Let us assume that the serving of trip j has just been completed. By keeping to the original notation, we can write that the choice of the next trip, having completed trip i, is made according to the following relation:

$$\min_{x} \mathrm{W}(j,x) = \mathrm{ET}(x) - \mathrm{T}(j,x) - \mathrm{LT}(j) \qquad \mathrm{W}(j,x) \geq 0 \tag{2.7}$$

where $\mathrm{ET}(x)$ is the earliest time of pickup for trip x, $\mathrm{T}(j, x)$ is travel time between dropoff node of trip j and pickup node of trip x, and $\mathrm{LT}(j)$ is the latest allowable dropoff time of trip j.

Let us note that $\mathrm{W}(j, x)$ represents the idle time of the vehicle when trips j and x are connected (*Figure 2.6*).

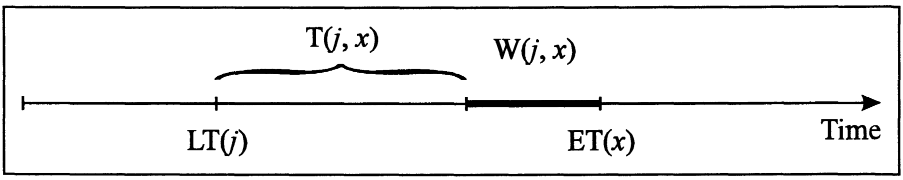

Figure 2.6. Idle time $\mathrm{W}(j, x)$ of vehicle when trips j and x are connected

Clearly, after making trip i, trip x is made for which $W(j, x)$ is the least. In this manner, an initial route is developed. For the purpose of reducing the computation time when $T(j, x)$ is a triangular fuzzy number, it is possible (for a start) to treat $T(j, x)$ as deterministic. Alternatively, it is possible to use the longest travel time $t_3(j, x)$ as estimation for travel time $T(j, x)$, or the shortest travel time $t_1(j, x)$, and so on. In the case of using quantities $t_3(j, x)$, for example, the initial route will contain a smaller number of trips, and the passengers included in the initial route will certainly arrive at the desired times.

Once we have designed the first initial route for one vehicle, we should try "to insert as many additional trips as possible in it before developing an initial route for the next vehicle."

In order to insert a previously uninserted trip into the initial route, three tests are to be conducted. The first is intended to test the insertability of its two stops (for pickup and dropoff) between the already existing stops (*Figure 2.7*).

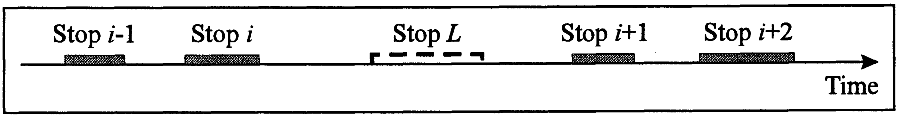

Figure 2.7. Testing of insertability of stops of a new trip between two existing stops in the initial route

In the following discussion we shall keep to the original notation suggested by Kikuchi (1992). It is assumed that trip L, which is not included in the initial route, is to be inserted into the initial route between stops i and i+1 (*Figure 2.7*). It should be determined whether the vehicle, after executing trip i, can execute trip L followed by trip i+1, beginning trips i, L and i+1 within the defined time windows.

Let us denote as ATE(L) the earliest possible arrival time at L after leaving i, and as DTL(L) the latest possible time that the vehicle can depart L to reach stop i+1 within time window. It is clear that (*Figure 2.8*)

$$\mathrm{ATE}(L) = \mathrm{ET}(i) + \mathrm{T}(i, L) \tag{2.8}$$

$$\mathrm{DTL}(L) = \mathrm{LT}(i+1) - \mathrm{T}(L, i+1) \tag{2.9}$$

The vehicle can travel from i to L to i+1 if $\mathrm{ATE}(L) \leq \mathrm{DTL}(L)$. In other words, the vehicle can travel from i to L to i+1 if the earliest possible arrival time at L after leaving i is less than the latest possible time that the vehicle can depart L to reach stop i+1 within the time interval (*Figure 2.8*). As can be seen from *Figure 2.8*, when inserting trip L between two existing stops, "a decrease" in the original time window of trip L can frequently occur (shaded section of the original time window).

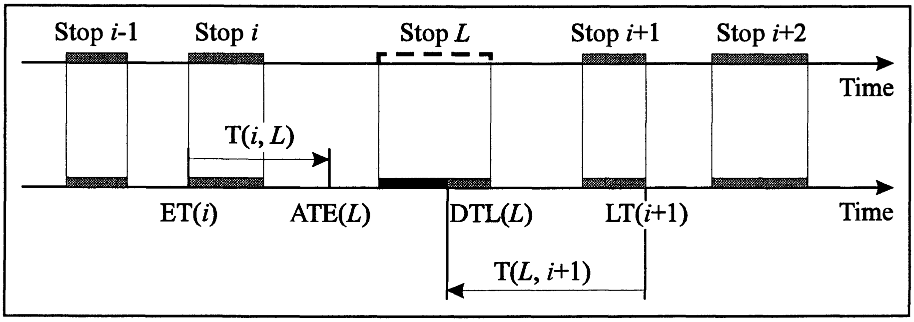

Figure 2.8. Insertability of stop L between two existing stops

Let us introduce in discussion the following binary variables:

$$\mathrm{AT}(L)_t = \begin{cases} 1, & t \geq \mathrm{ATE}(L) \\ 0, & t < \mathrm{ATE}(L) \end{cases} \tag{2.10}$$

$$\mathrm{DT}(L)_t = \begin{cases} 1, & t \leq \mathrm{DTL}(L) \\ 0, & t > \mathrm{DTL}(L) \end{cases} \tag{2.11}$$

$$\mathrm{STP}(L)_t = \min\,[\mathrm{AT}(L)_t, \mathrm{DT}(L)_t] \tag{2.12}$$

As can be seen, binary variables $\mathrm{AT}(L)_t$, $\mathrm{DT}(L)_t$, and $\mathrm{STP}(L)_t$ are dependent on time. Fuzzy set **AT**(L) is called "after ATE(L)" or "larger than ATE(L)." The name of the fuzzy set **DT**(L) is "before DTL(L)" or "smaller than DTL(L)." It is clear that the membership functions of fuzzy sets **AT**(L) and **DT**(L) can also be defined in a somewhat different manner.

Let us denote by **STP**(L) a fuzzy set representing the intersection of fuzzy sets **AT**(L) and **DT**(L) (*Figure 2.9*). The membership function $\mu_{\mathbf{STP}(L)}(t)$ is equal to:

$$\mu_{\mathbf{STP}(L)}(t) = \min\,[\mu_{\mathbf{AT}(L)}(t), \mu_{\mathbf{DT}(L)}(t)] \tag{2.13}$$

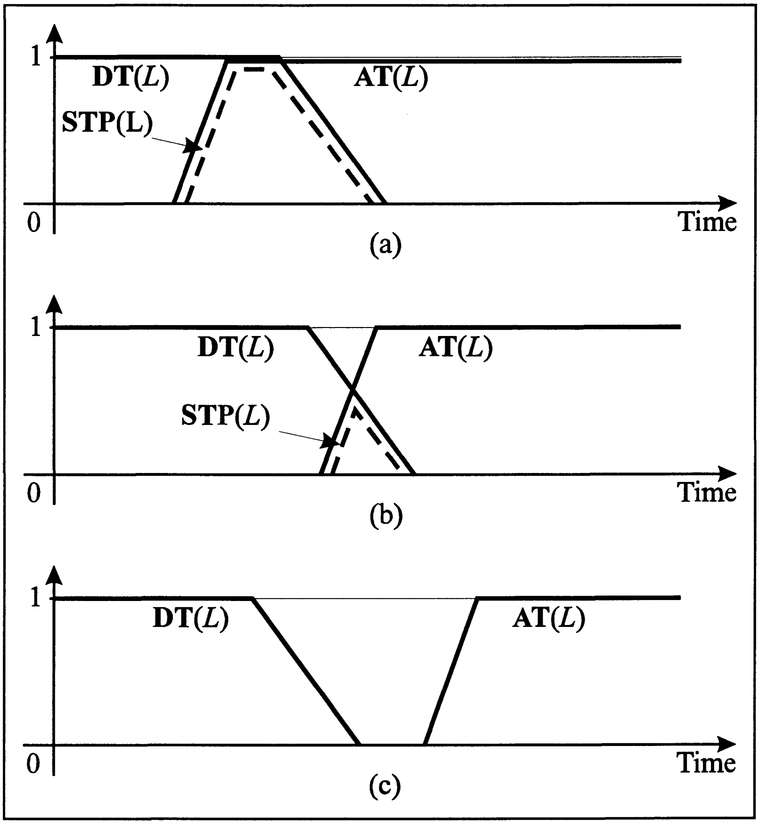

Figure 2.9. Fuzzy set **STP**(L) representing the intersection of fuzzy sets **AT**(L) and **DT**(L)

Clearly, the maximum value of the membership function represents the maximum possibility of stopping at L at time t, while still satisfying the requirement for stopping at i and i+1 within the appropriate time windows.

The variable AT$(L)_t$ represents "the measure of whether the vehicle is able to arrive at L by time t," while DT$(L)_t$ represents "the measure of whether the vehicle can arrive at the next stop on time if it leaves L at time t." In the case when

$$\mathrm{STP}(L)_t = \min\,[\mathrm{AT}(L)_t, \mathrm{DT}(L)_t] = 1 \qquad (2.14)$$

it can be concluded that the vehicle is able to stop at L at time t.

As already noted, the estimated travel times between any two nodes in the network are treated as fuzzy numbers. Travel times **T**(i, L) and **T**(L, i+1) represent triangular fuzzy numbers: **T**(i, L) = ($T_1(i, L)$, $T_2(i, L)$, $T_3(i, L)$) and **T**(L, i+1)=($T_1(L, i+1)$, $T_2(L, i+1)$, $T_3(L, i+1)$).

According to the rules of fuzzy arithmetic, it can be concluded that both **ATE**(L) and **DTL**(L) are triangular fuzzy numbers (*Figure 2.10*).

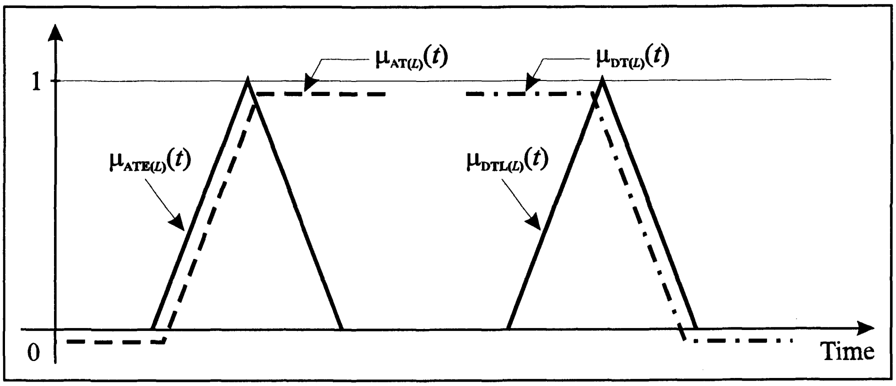

Figure 2.10. Membership functions of fuzzy sets **ATE**(L), **AT**(L), **DTL**(L), and **DT**(L)

As shown in *Figure 2.9*, the possibility of stopping at L, Pos[Stop(L)] is in certain cases (case (a)) equal to 1, which means that the vehicle can certainly make a stop at L. The case shown in *Figure 2.9(b)* refers to the situation when 0<Pos[Stop(L)]<1, which means that it is possible to make a stop at L, but it is not completely certain. The closer the value of Pos[Stop(L)] approaches 1, the more certain we are of the possibility to make a stop at L. *Figure 2.9(c)* illustrates the case when the vehicle cannot make a stop at L, since Pos[Stop(L)] = 0.

The stated procedure for computing the possibility of stopping at L, Pos[Stop(L)], refers to the case of examining the insertion of trip L between trips i and i+1, assuming that trips i and i+1 belong to the initial route from the beginning. Since trips i and i+1 are included in the initial route from the beginning, the corresponding possibilities of stopping are calculated as follows:

$$\text{Pos [Stop}(i)] = \text{Pos [Stop}(i+1)] = 1 \tag{2.15}$$

Kikuchi (1992) also presented the case in which he tried to insert stop L between stop i and stop i+1, of which at least one stop has previously been inserted into the initial route. In that case, we obtain

$$\text{Pos [Stop}(i)] \leq 1 \tag{2.16}$$

$$\text{Pos [Stop}(i+1)] \leq 1 \tag{2.17}$$

Given that possibilities of stopping at i and i+1 are 1, let us denote the possibility of stopping at L as

$$\text{Pos}\ \{\text{Stop}(L) \mid \text{Pos}\ [\text{Stop}(i)] = \text{Pos}[\text{Stop}(i+1)] = 1\} \tag{2.18}$$

In the case when stop i and/or stop i+1 is a previously inserted stop, the possibility of stopping at L is equal to

$$\text{Pos}[\text{Stop}(L)]=\text{Pos}\{\text{Stop}(L) \mid \text{Pos}[\text{Stop}(i)]=\text{Pos}[\text{Stop}(i+1)]=1\} \wedge \\ \text{Pos}[\text{Stop(i)}] \wedge \text{Pos}[\text{Stop}(i+1)] \tag{2.19}$$

Figure 2.11 illustrates the case of trying to insert stop L between previously inserted stops i and i+1.

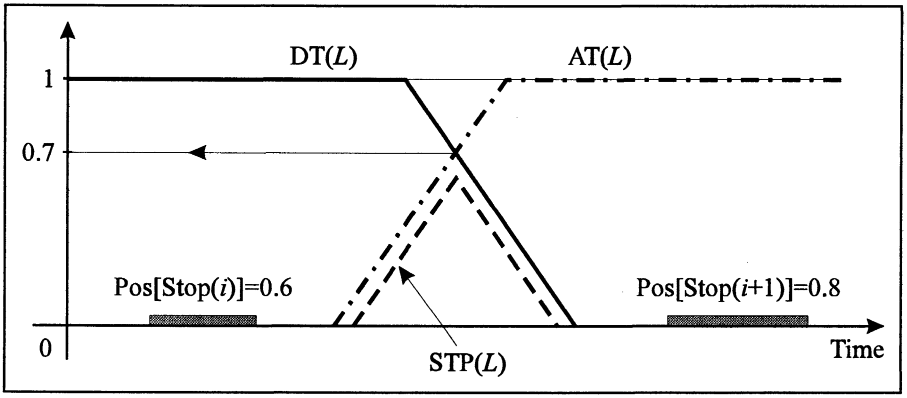

Figure 2.11. Insertion of stop L between previously inserted stops i and i+1

Suppose that in the case when stop L is inserted into the initial route between previously inserted stops i and i+1 we obtain

$$\text{Pos}\ [\text{Stop}(L)] = 0.7 \tag{2.20}$$

In the case of inserting stop L between i and i+1 that have previously been inserted and have the corresponding possibilities of stopping, the final possibility of stopping at L is equal to

$$\text{Pos}\ [\text{Stop}(L)] = \min\ [0.6, 0.7, 0.8] = 0.6 \tag{2.21}$$

The decision whether to insert stop L between stop i and stop i+1 should be made based on a previously defined value $0 \leq c_1 \leq 1$.

If Pos[Stop(L)] < c_1, stop L between stop i and stop i+1 should not be made, as it is considered to be "risky." Obviously, as the values of quantity c_1 increase, so does the analyst's optimism concerning the possibility of inserting stop L between stop i and stop i+1.

After testing the insertability of stop L between the two existing stops (i and i+1), it is necessary to test the satisfaction of the desired time of stopping. In other words, it is necessary to determine if a possible range of the arrival time at stop L overlaps with the passenger's time window.

Let us denote by **UT**(L) the fuzzy set of the desired time of a vehicle's visit at L (*Figure 2.12*).

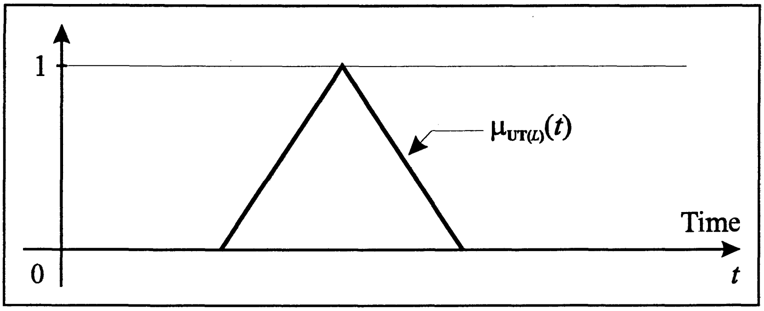

Figure 2.12. Membership function of fuzzy set **UT**(L) representing desired time of vehicle's visit at L

Let us, also, denote by **US**(L) the fuzzy set of user's satisfaction of stopping at L. This fuzzy set represents the intersection of fuzzy set **STP**(L) and fuzzy set **UT**(L) (*Figure 2.13*).

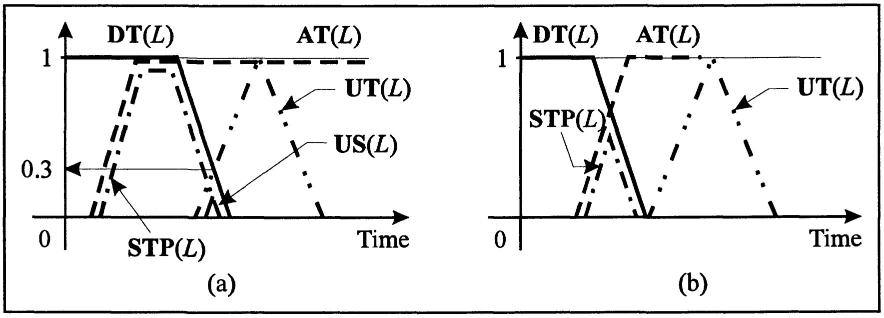

Figure 2.13. Membership function of fuzzy set **US**(L) representing user's satisfaction

In *Figure 2.13* are given two of several possible cases of the intersection of **UT**(L) and **STP**(L). As Kikuchi (1992) noted: "the maximum degree of satisfaction of the passenger can be measured at the point where the membership function of **US**(L) takes the highest value." In the case shown in *Figure 2.13(a)* this value is equal to 0.3, whereas in *Figure 2.13(b)* it is equal to zero. In other words, the possibility of satisfaction of the passenger is

$$\text{Pos}\left[\text{Sat}\,(L)\right] = \max \mu_{\text{US}\,(L)}(t) \tag{2.22}$$

As in the case of the previous test, it is possible to introduce a minimum acceptable level of satisfaction c_2. An insertion should be made only when

$$\text{Pos}\,[\text{Sat}\,(L)] \geq c_2 \tag{2.23}$$

The value of c_2 is also defined in advance. By specifying greater values of c_2, we take more account of the passenger's level of satisfaction.

The completion of the second test results (in a large number of cases) in a decrease in the time window of stop L (*Figure 2.14*).

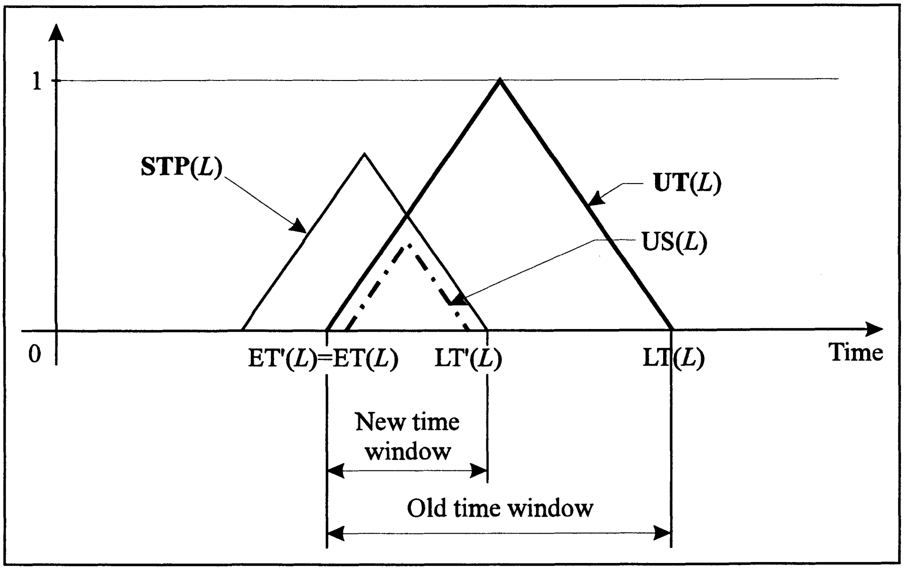

Figure 2.14. Decrease in time window

ET'(L) and LT'(L) denote, respectively, the new beginning of the time window at stop L and the new ending of the time window at stop L.

The new time window is as follows:

$$\left[\mathrm{ET}'(L), \mathrm{LT}'(L)\right] = \left[t \mid \mu_{\mathrm{US}(L)}(t) \geq 0\right] \tag{2.24}$$

After completing the second test, the time windows of all other stops belonging to the initial route must be adjusted. Since all travel times between all pairs of nodes are fuzzy numbers, new bounds ET'(k) and LT'(k) (k = 1, 2, ...) of the time windows of all other stops are obtained according to the rules of addition and subtraction of fuzzy numbers (*Figure 2.15*). It should be noted that the new bounds of time windows are also fuzzy sets.

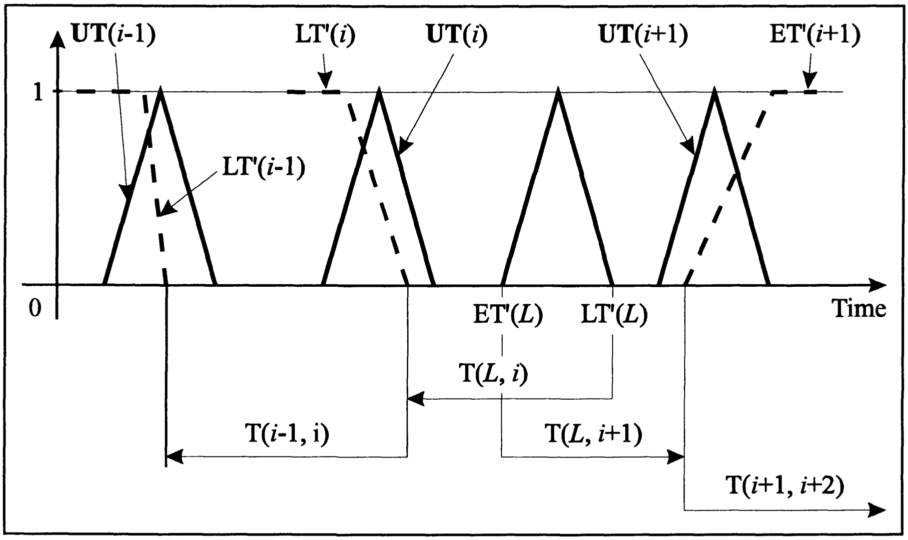

Figure 2.15. Adjustment of time windows of other stops

Parameter c_2 in the second test which reflects a minimum acceptable level of the passenger's satisfaction can also be used to determine the passenger's satisfaction in the case of other stops belonging to the initial route. In other words, to insert stop L into the initial route, for any stop k already belonging to the initial route the following requirement must be satisfied:

$$\max_{\mathrm{ET}'(k) \leq t \leq \mathrm{LT}'(k)} \mu_{\mathrm{US}(k)}(t) \geq c_2 \tag{2.25}$$

Kikuchi (1992) performed a large number of computing experiments, varying the values of parameters c_1 and c_2 and the range of time windows. In that way, relationships were established between time window and number of vehicles required, possibilities of vehicle stopping, passengers' satisfaction, and different versions of the vehicle schedule.

2.3. FUZZY MULTIPLE-ATTRIBUTE DECISION MAKING: THE CHOICE OF THE BEST ALTERNATIVE FOR ORGANIZING AIR SHUTTLE SERVICE

In performing our professional duties as well as in our private life we are faced with making various decisions. Generally, we need to pick up one alternative from a particular set of possible alternatives or, in certain cases, all the alternatives considered have to be ranked. The ranking of the alternatives is usually done according to a number of criteria that, as a rule, are mutually conflicting. For this purpose, a number of different methods have been developed, belonging to the field called *multiple-attribute decision making*. Within this field, a remarkable number of papers have been published. The monograph by Hwang and Yoon (1981) is certainly among the most prominent. Choosing a signaling schedule for an isolated intersection, choosing a convenient urban public transportation network, choosing a flight schedule, or choosing the best location for a new railroad station represent some of the decisions that traffic experts are sometimes expected to make.

In matrix D, values x_{ij} are given that certain alternatives A_i $(i = 1, 2, ..., m)$ take by particular criteria X_j $(j = 1, 2, ..., n)$:

$$D = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_m \end{array} \begin{array}{c} \begin{array}{cccc} X_1 & X_2 & \cdots & X_n \end{array} \\ \left| \begin{array}{cccc} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{array} \right| \end{array} \tag{2.26}$$

By m we denote the total number of alternatives, and by n the total number of criteria according to which the considered alternatives are compared.

As already noted, within the field of multiple-attribute decision making various methods for ranking the alternatives have been developed. A detailed classification and analysis of these methods have been made by Hwang and Yoon (1981). The most famous classical methods include the *dominance method*, the *maximin method*, the *maximax method*, the *conjuctive method*, the *disjunctive method*, the *lexicographic method*, the *simple additive weighting method*, *ELECTRE*, *TOPSIS*, *AHP*, the *weighted* *product method*, and the *distance from target method*. In each of these classical methods it is assumed that values x_{ij} that certain alternatives A_i (i = 1, 2, ..., m) take by particular criteria are crisp numbers.

However, as Chen and Hwang (1992) point out: "most of the real world multiple-attribute decision-making problems contain mixture of fuzzy and crisp data." It is clear that, in some cases, the values x_{ij} that certain alternatives take by particular criteria are not given quantitatively but rather in terms of appropriate linguistic expressions. Thus, for example, a car is often described as "very comfortable." In other words, when ranking different types of car, certain criteria may be numerical variables (price of car, fuel consumption in litres per 100 km, monthly installment, and so on), while others are found to be linguistic variables (comfort, safety, aesthetic appeal). Since the values of linguistic variables are words or sentences, certain values of x_{ij} are words, such as, a "very comfortable," "quite safe," or "extraordinarily beautiful" car.

The fact that both crisp and fuzzy data occur in matrix D brought about the need to modify the classical multiple-attribute decision-making methods. In their monograph called "Fuzzy Multiple Attribute Decision Making," Chen and Hwang (1992) made a classification and analysis of the existing fuzzy multiple-attribute decision-making methods. In the following discussion, only a comparatively recent fuzzy multiple-attribute decision-making method proposed by Chen and Hwang is presented. The reader interested in this area can find more information on this issue in Chen and Hwang's book (1992).

The reasons determining whether the criteria by which a ranking is made will be linguistic variables or fuzzy numbers are various. In some cases, certain criteria simply cannot be quantified ("very comfortable," "extraordinarily beautiful"). Some of x_{ij} values represent fuzzy numbers. The appearance of fuzzy numbers in matrix D is conditioned by the impossibility of the appropriate values to be precisely established. Thus, for example, when choosing the type of plane for an airline company, it is very difficult to determine with absolute precision the annual number of passengers between certain pairs of cities or the operational costs of the carrier on one line for the five incoming years. Also, in certain cases it is not possible to provide precise numerical data, and we are forced to make appropriate approximations.

The proposed Chen and Hwang's method (1992) for solving fuzzy multiple-attribute decision-making problems consists of two steps. In the first step, the linguistic terms or fuzzy numbers are transformed into crisp scores. When a particular linguistic term is transformed into a crisp score, it is first converted into a fuzzy number, and then the obtained fuzzy number is converted into a crisp score. On the transformation, all the data in matrix

D are crisp numbers. In the second step, any of the classical multiple-attribute decision-making methods for obtaining the final rank of the considered alternatives can be applied.

Considering the results achieved by other authors (Baas and Kwakernaak, 1977; Efstathiou and Rajkovic, 1979; Efstathiou and Tong, 1982), Chen and Hwang proposed eight conversion scales by which the converting of linguistic terms into fuzzy numbers can be made. Which of these scales will be used depends on the total number of linguistic terms that are to be converted into fuzzy numbers. *Figure 2.16* represents two possible conversion scales.

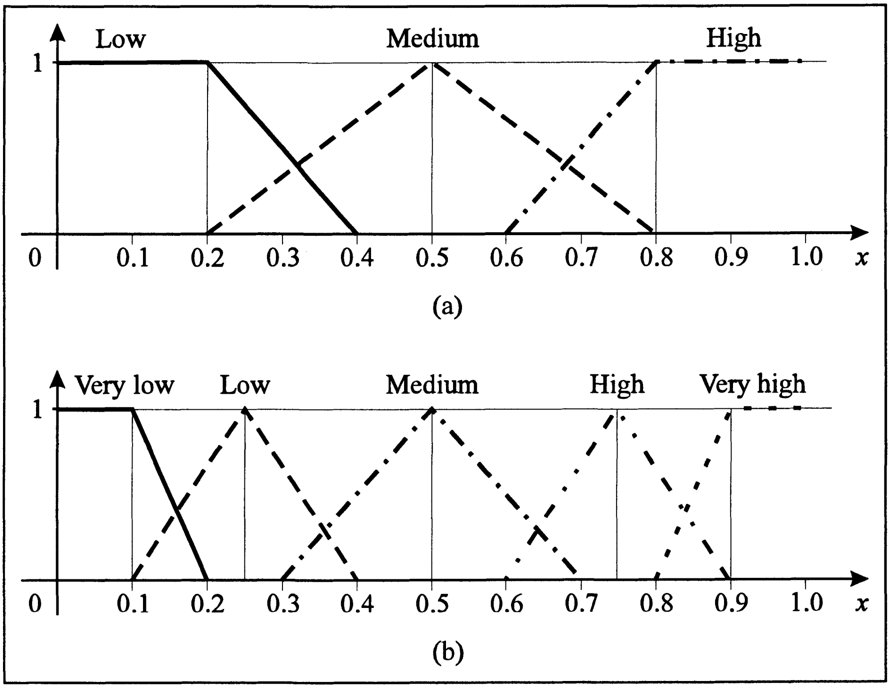

Figure 2.16. Conversion scales in the case of three (a) and five (b) linguistic terms

By means of conversion scales certain linguistic terms are easily converted into fuzzy numbers. Thus, for example, the linguistic term "medium" (when five linguistic terms are used) is corresponded by a triangular fuzzy number (0.3, 0.5, 0.7), as shown in *Figure 2.16(b)*.

In the next step, fuzzy numbers are converted into crisp scores. Jain (1976), Chen (1985), and Chen and Hwang (1992) proposed that a crisp score of a fuzzy number A be obtained by means of the so-called maximizing and minimizing sets. The membership functions of these sets are defined as

$$\mu_{\max}(x) = \begin{cases} x, & 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases} \tag{2.27}$$

$$\mu_{\min}(x) = \begin{cases} 1-x, & 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases} \tag{2.28}$$

The right score, $\mu_R(\mathbf{A})$, the left score, $\mu_L(\mathbf{A})$, and the total score, $\mu_T(\mathbf{A})$, of a fuzzy number **A** are defined as

$$\mu_R(\mathbf{A}) = \max_x \left[\min\,(\mu_A(x), \mu_{\max}(x))\right] \tag{2.29}$$

$$\mu_L(\mathbf{A}) = \max_x \left[\min\,(\mu_A(x), \mu_{\min}(x))\right] \tag{2.30}$$

$$\mu_T(\mathbf{A}) = \frac{\mu_R(\mathbf{A}) + 1 - \mu_L(\mathbf{A})}{2} \tag{2.31}$$

For each fuzzy number in matrix D the total score is calculated. On the calculation of all total scores, matrix D contains only crisp data, so to the ranked alternatives any of the classical multiple-attribute decision-making methods can be applied.

2.3.1. The Choice of the Best Alternative for Organizing Air Shuttle Service

For the purpose of providing the best possible transportation service and achieving the best economic effects for the air transport carrier a number of different organizations of air transportation have been developed. The so-called air shuttle service includes equal time intervals between aircraft departures and the impossibility of reserving a seat in advance. Teodorovic (1985, 1988) studied the basic characteristics of the air shuttle service between two cities. Since the passenger in the air shuttle service has lost the possibility of reserving a seat in advance, some sort of guarantee must be provided in terms of the waiting time. In order to determine the flight frequency for the air shuttle, calculations must be made for several different alternatives (several different time intervals between consecutive departures), concerning the required transportation capacities, total operating costs, average passenger waiting time, probability of finding a vacant seat on the first departing plane, and so on.

Let us present a somewhat modified example of choice of the best alternatives examined by Teodorovic (1985). A total of six alternatives A_i ($i = 1, 2, ..., 6$) for establishing the air shuttle between two cities have been considered. The alternatives differ in terms of the interval between the departures of certain planes and the probability of guaranteeing the passenger a vacant seat on the first departing plane. The alternatives shown in *Table 2.1* were considered.

Table 2.1. Characteristics of air shuttle alternatives

Alternative	Interval between the departure of two consecutive planes	Probability of guarantying the passengers a vacant seat on the first departing plane (q)
A_1	1h	$q<1$
A_2	1.5h	$q<1$
A_3	2h	$q<1$
A_4	1h	$q=1$
A_5	1.5h	$q=1$
A_6	2h	$q=1$

For each given alternative A_i, $i = 1, 2, ..., 6$, the following calculations were made: X_1 is number of engaged planes in time intervals 5 a.m. to 11 a.m. and 7 p.m. to 9 p.m., X_2 is number of engaged planes in time interval 11 a.m. to 7 p.m., X_3 are transportation costs in [$], X_4 is average waiting time of one passenger in [min], X_5 is probability of finding a vacant seat on the first departing plane.

The values that alternatives A_i, $i = 1, 2,..., 6$ take according to criteria X_j, $j = 1, 2, ..., 5$ are given in matrix D:

$$D = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \end{array} \begin{array}{c} \begin{array}{ccccc} X_1 & X_2 & X_3 & X_4 & X_5 \end{array} \\ \begin{bmatrix} 4 & 4 & \text{about } 80{,}000 & 21 & 0.98 \\ 4 & 2 & \text{about } 60{,}000 & 35 & 0.92 \\ 4 & 2 & \text{about } 60{,}000 & 51 & 0.97 \\ 8 & 4 & \text{about } 90{,}000 & 17 & 1 \\ 4 & 2 & \text{about } 70{,}000 & 31 & 1 \\ 6 & 2 & \text{about } 60{,}000 & 48 & 1 \end{bmatrix} \end{array} \tag{2.32}$$

In matrix D all the data are crisp except the data in column X_3, which represent linguistic terms. Let us first convert these linguistic terms into the corresponding fuzzy numbers. The membership functions of fuzzy numbers "about 60,000," "about 70,000," "about 80,000," and "about 90,000" are given in *Figure 2.17* (fuzzy numbers **A**, **B**, **C**, and **D**).

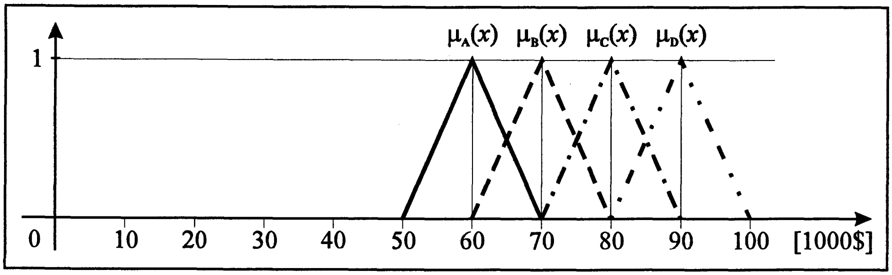

Figure 2.17. Membership functions of fuzzy numbers **A**, **B**, **C**, and **D**

In order for all the data in matrix D to be crisp, fuzzy numbers must be converted into crisp scores. The membership functions of the maximizing set, the minimizing set, and the considered fuzzy numbers are given in *Figure 2.18.*

The right score, the left score and the total score of fuzzy numbers **A**, **B**, **C**, and **D** are given in *Table 2.2.*

Table 2.2. The right score, the left score, and the total score of fuzzy numbers **A**, **B**, **C**, and **D**

Fuzzy number **I**	The left score $\mu_L(\mathbf{I})$	The right score $\mu_R(\mathbf{I})$	The total score $\mu_T(\mathbf{I})=(\mu_R(\mathbf{I})+1-\mu_L(\mathbf{I}))/2$
A	5/11	7/11	13/22
B	4/11	8/11	15/22
C	3/11	9/11	17/22
D	2/11	10/11	19/22

The calculated values of total scores allow us to form matrix D in which all the data are crisp. A new, transformed matrix is as follows:

$$D = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \end{array} \begin{array}{c} \begin{array}{ccccc} X_1 & X_2 & X_3 & X_4 & X_5 \end{array} \\ \begin{bmatrix} 4 & 4 & 17/22 & 21 & 0.98 \\ 4 & 2 & 13/22 & 35 & 0.92 \\ 4 & 2 & 13/22 & 51 & 0.97 \\ 8 & 4 & 19/22 & 17 & 1 \\ 4 & 2 & 15/22 & 31 & 1 \\ 6 & 2 & 13/22 & 48 & 1 \end{bmatrix} \end{array} \tag{2.33}$$

It is now possible to rank the alternatives using one of the classical multiple-attribute decision-making methods. In this example, the method used is TOPSIS (Hwang and Yoon, 1981). This method is based on simultaneous measurements of the distance of a particular alternative from the so-called ideal and negative ideal solution that is, the measurement of the relative distance of an alternative from the ideal solution.

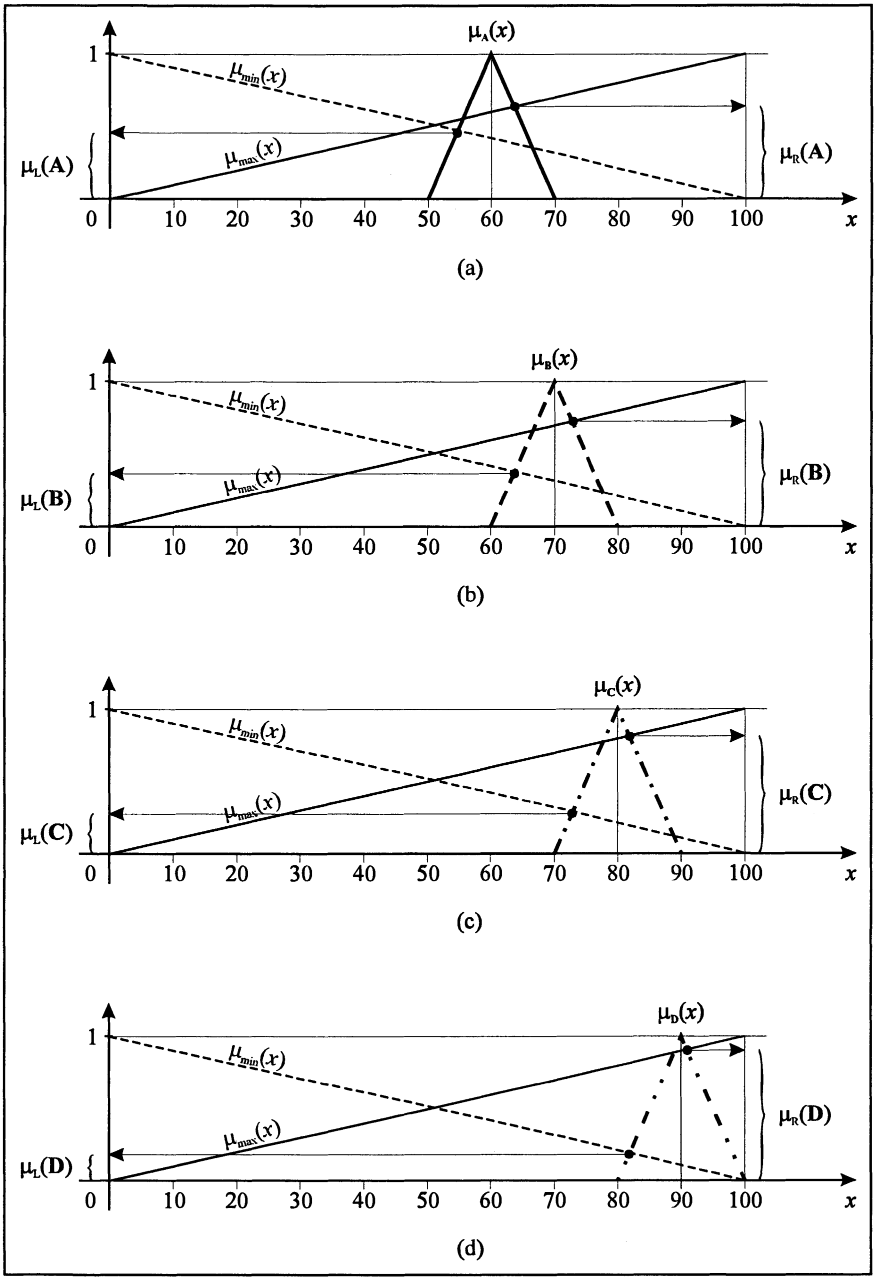

Figure 2.18. Membership functions of maximizing set, minimizing set, and fuzzy numbers **A**, **B**, **C**, and **D**

As in the case of many other methods, a normalized decision matrix is calculated first.

Normalized values r_{ij} are calculated as

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}} \quad i = 1, 2, \ldots, m, \; j = 1, 2, \ldots, n \tag{2.34}$$

In the next step, each column's elements in matrix R are multiplied by weight w_j (significance of a criterion) corresponding to a particular column. In this manner, matrix V is obtained such that the values of its elements express the weights (significance) of individual criteria as well. Matrix V is found to be

$$\mathrm{V} = \begin{bmatrix} v_{11} & \cdots & v_{1j} & \cdots & v_{1n} \\ \vdots & & & & \vdots \\ v_{i1} & \cdots & v_{ij} & \cdots & v_{in} \\ \vdots & & & & \vdots \\ v_{m1} & \cdots & v_{mj} & \cdots & v_{mn} \end{bmatrix} = \begin{bmatrix} w_1 r_{11} & \cdots & w_j r_{1j} & \cdots & w_n r_{1n} \\ \vdots & & & & \vdots \\ w_1 r_{i1} & \cdots & w_j r_{ij} & \cdots & w_n r_{in} \\ \vdots & & & & \vdots \\ w_1 r_{m1} & \cdots & w_j r_{mj} & \cdots & w_n r_{mn} \end{bmatrix} \tag{2.35}$$

On calculating the elements of matrix V, the ideal solution A* and the negative ideal solution A⁻ are determined. These solutions are defined as

$$\begin{aligned} \mathrm{A}^* &= \left\{ (\max_i v_{ij} \,|\, j \in \mathrm{J}), (\min_i v_{ij} \,|\, j \in \mathrm{J}') \,|\, i = 1, 2, \ldots, m \right\} = \\ &= \left\{ v_1^*, v_2^*, \ldots, v_j^*, \ldots, v_n^* \right\} \end{aligned} \tag{2.36}$$

$$\begin{aligned} \mathrm{A}^- &= \left\{ (\min_i v_{ij} \,|\, j \in \mathrm{J}), (\max_i v_{ij} \,|\, j \in \mathrm{J}') \,|\, i = 1, 2, \ldots, m \right\} = \\ &= \left\{ v_1^-, v_2^-, \ldots, v_j^-, \ldots, v_n^- \right\} \end{aligned} \tag{2.37}$$

where

$$\mathrm{J} = \left\{ j = 1, 2, \ldots, n \,|\, j \text{ belongs to the benefit criteria} \right\} \tag{2.38}$$

$$\mathrm{J}' = \left\{ j = 1, 2, \ldots, n \,|\, j \text{ belongs to the cost criteria} \right\} \tag{2.39}$$

Let us note that the benefit criteria are understood to be those by which an alternative is better if it takes greater values. As far as the cost criteria are concerned, an alternative is better if by these criteria it takes lower values. Distance S_i^* of each alternative from the ideal alternative is

$$S_i^* = \sqrt{\sum_{j=1}^{n}(v_{ij} - v_j^*)^2} \qquad i = 1, 2, \ldots, m \tag{2.40}$$

Distance S_i^- of each alternative from the negative ideal solution is

$$S_i^- = \sqrt{\sum_{j=1}^{n}(v_{ij} - v_j^-)^2} \qquad i = 1, 2, \ldots, m \tag{2.41}$$

Relative closeness C_i^* of the alternative A_i to the ideal solution A^* is

$$C_i^* = \frac{S_i^-}{S_i^* + S_i^-}, \qquad 0 \leq C_i^* \leq 1 \quad i = 1, 2, \ldots, m \tag{2.42}$$

Since $C_i^* = 1$ if $A_i = A^*$ and $C_i^* = 0$ if $A_i = A^-$, the alternative A_i is better if C_i^* is closer to 1. It is clear that from the set of alternatives $A_1, A_2, \ldots, A_m$ the best alternative is A_i with the largest value of C_i^*.

In our case, X_1, X_2, X_3, and X_4 are cost criteria, while X_5 is a benefit criterion. To determine the weights of the criteria, Teodorovic (1985) used the method of entropy. The following criteria weights were obtained:

$w_1 = 0.1683$; $w_2 = 0.333$; $w_3 = 0.2257$; $w_4 = 0.2724$; $w_5 = 0.0006$.

The ideal and negative ideal solutions are

$$\left.\begin{aligned} A^* &= \{0.052568, 0.0961288, 0.1153388, 0.0523963, 0.000250136\} \\ A^- &= \{0.1051361, 0.1922576, 0.078916, 0.1581084, 0.000230105\} \end{aligned}\right\} \tag{2.43}$$

The values of C_i^* are given in *Table 2.3*.

The values of C_i^*, $i = 1, 2, \ldots, 6$ indicate that the alternatives are ranked in the following manner: A_2, A_5, A_6, A_3, A_1, A_4.

Table 2.3. Relative closeness C_i^* of the alternative A_i to the ideal solution A^*

Alternative A_i	C_i^*
A_1	0.5119817
A_2	0.8810983
A_3	0.5235411
A_4	0.4763717
A_5	0.5579323
A_6	0.5240586

2.4. CONTROLLING AN ISOLATED SIGNALIZED INTERSECTION USING FUZZY LOGIC

The problem of controlling an isolated signalized intersection has been dealt with by many authors worldwide. The first paper that tried to solve this problem using fuzzy logic was written by Pappis and Mamdani (1977). They considered the isolated signalized intersection of two one-way streets and developed a model based on linguistic control instructions. A comparison of the results from the model based on fuzzy logic with results from the classical approach to controlling an isolated signalized intersection indicated that better results (from the viewpoint of average time loss per vehicle) were achieved by the model based on fuzzy logic.

Pappis and Mamdani developed the idea of a model that would imitate the work of the decision maker. Let us assume for a moment that there is a policeman directing traffic at the intersection of two one-way streets. The policeman lets traffic pass in one direction for a while and then at one moment he decides to stop traffic from the direction it has been coming and to let traffic pass from the other direction. The following questions are logically raised: What criteria does the policeman use to make his decision? At what moment in time does the policeman conclude that he should change the one-way street that has priority in the intersection? The policeman is certainly affected by both the number of vehicles waiting to be allowed through the intersection and the time they have been waiting. At one point, the policeman might note that a "large" number of vehicles has accumulated, and he decides to let them through the intersection. The policeman might also be influenced by the fact that a "smaller" number of vehicles have been waiting to cross the intersection for a relatively "long" time. We can see that "smaller," "large," or "very large" number of vehicles waiting in line to pass through the intersection can be represented by fuzzy sets, as can "small," "medium," "long," or "very long" waiting time.

We will briefly present the Pappis and Mamdani model to control an isolated signalized intersection. *Figure 2.19* presents the isolated signalized intersection of two one-way streets.

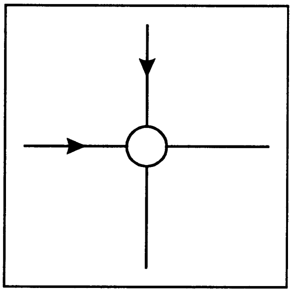

Figure 2.19. Isolated signalized intersection of two one-way streets

Pappis and Mamdani (1977) assumed a uniform distribution of vehicles arriving at the intersection. They also assumed that the cycle was divided into two periods ("actually red" and "actually green") and that vehicles leave the line at the same intensity at which they join it. Keeping to the same symbols as Pappis and Mamdani, we will introduce the following variables: T is a fuzzy variable that denotes the time that has lapsed since the last light change at the intersection, A is a fuzzy variable denoting the number of vehicles from the priority direction that have passed through the green light during the considered time period, Q is a fuzzy variable representing the number of vehicles waiting in line on the one-way street that does not have priority (the number of vehicles waiting for the light to turn green), and E is a fuzzy variable "extension" that has values identical to fuzzy variable T representing the extension given to the present state of the system.

Fuzzy variables T, A, and Q are input variables whose values should determine the value of output variable E. Fuzzy variable A could be assigned the value "many" vehicles, "more than several" vehicles, "few" vehicles, and so on. Fuzzy variable Q could be assigned similar values. Variables T and E are assigned values "very short," "short," "medium" time, and so on. Pappis and Mamdani also used fuzzy sets "any" number of vehicles "more than" and "less than." We would note that the grade of membership of every element belonging to the fuzzy set "any" equals 1. Let us note fuzzy set **M**. The membership functions of fuzzy sets "greater than M" and "less than M" can be defined as follows:

$$\mu_{\textbf{greater than M}}(x) = \begin{cases} 0, & x \le x^* \\ 1 - \mu_{\mathrm{M}}(x), & x \ge x^* \end{cases} \tag{2.44}$$

$$\mu_{\text{less than M}}(x) = \begin{cases} 1-\mu_{\text{M}}(x), & x \le x^* \\ 0, & x \ge x^* \end{cases} \tag{2.45}$$

where element x^* of set **M** has the largest grade of membership in set **M**. *Figure 2.20* presents the membership functions of fuzzy sets **M**, "greater than M," and "less than M."

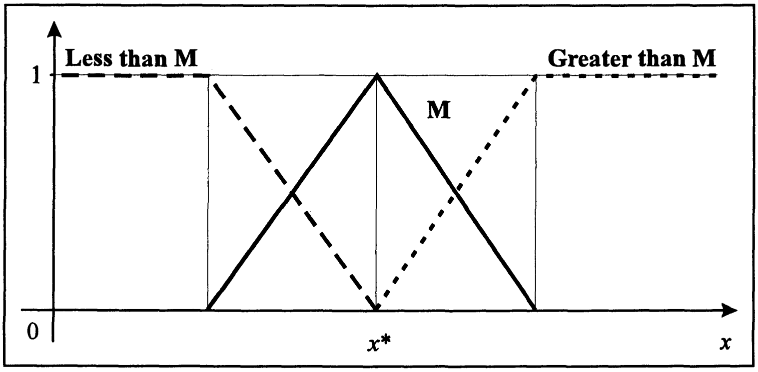

Figure 2.20. Membership functions of fuzzy sets **M**, **greater than M**, and **less than M**

The algorithm to control traffic at an isolated intersection proposed by Pappis and Mamdani (1977) consists of the rules of the following type:

Rule 1: If T is very short and A is greater than none and Q is any, then E is very short,
or
Rule 2: If T is short and A is greater than few and Q is less than very small, then E is short,
or
Rule 3: If T is medium and A is greater than few and Q is less than very small, then E is medium,
or
Rule 4: If T is long and A is greater than medium and Q is less than very small, then E is long,
or
Rule 5: If T is very long and A is greater than many and Q is less than very small, then E is very long.

The values of fuzzy variable E representing the extension given to the present state of the system were between 1 and 10 seconds. Every 10 seconds a different set of five rules was used to make the decision on the length of time to the next light change at the intersection.

Note Rule 5: If T is very long and A is greater than many and Q is less than very small, then E is very long. This rule represents the fuzzy relation between the time that has lapsed since the last light change at the intersection (T), the number of vehicles that have gone through the green light during time period (A), the number of vehicles waiting in line for the light to turn green (Q), and the extension given to the present state of the system (E). The expression

If T is very long and A is greater than many and Q is less than very small represents the fuzzy phrase defined over the T × A × Q.

Let us denote this fuzzy phrase by **N**. The membership function of this fuzzy phrase equals

$$\mu_{\mathrm{N}}(t,a,q) = \min\left\{\mu_{\text{very long}}(t), \mu_{\text{greater than many}}(a), \mu_{\text{less than very small}}(q)\right\} \quad (2.46)$$

Rule 5 is a fuzzy implication that can be written as

If **N** Then E is very long

This fuzzy implication also represents the fuzzy phrase defined over the set T × A × Q × E. Let us denote this fuzzy phrase by **M**. The membership function of fuzzy phrase **M** equals

$$\mu_{\mathbf{M}}(t,a,q,e) = \min\left\{\mu_{\mathrm{N}}(t,a,q), \mu_{\text{very long}}(t)\right\} \quad (2.47)$$

Let us denote by $\mathbf{M}_1$, $\mathbf{M}_2$, ..., $\mathbf{M}_5$, respectively, the fuzzy phrases referring to the five earlier-defined rules. Two or more fuzzy phrases linked by "or" ("else") are called a *fuzzy clause* (Pappis and Mamdani, 1977). In our case fuzzy phrases $\mathbf{M}_1$, $\mathbf{M}_2$, ..., $\mathbf{M}_5$ comprise a fuzzy clause. Let us denote this fuzzy clause by **R**. The membership function of fuzzy clause **R** equals

$$\mu_{\mathbf{R}} = \max\left\{\mu_{\mathbf{M}_1}(t,a,q,e), \mu_{\mathbf{M}_2}(t,a,q,e), \ldots, \mu_{\mathbf{M}_5}(t,a,q,e)\right\} \quad (2.48)$$

For known values of input variables T, A and Q, using the approximate reasoning by max-min composition, each of the possible output values of variable E is associated with a grade of membership.

During defuzzification, when one numerical value is chosen for the output variable, one of the following criteria, "the smallest maximal value," "the largest maximal value," "center of gravity," "mean of the range of maximal values," is usually applied. In the model developed by Pappis and Mamdani (1977), the center of gravity of the resulting fuzzy set represents the output numerical value.

2.5. A FUZZY TRAFFIC CONTROL SYSTEM ON AN URBAN EXPRESSWAY

In the majority of cases traffic control on urban expressways is managed by experienced operators. In order for a smooth traffic flow to be achieved, the operators should certainly be highly skilled. The operators are often exposed to exceptionally hard-working conditions that is, stress-causing situations, as they are frequently expected to make managing decisions in a very short time. It has also been observed that the manner in which the operators make their decisions is not easy to describe.

Realizing the importance of the development of "an automatic decision-making system in place of operators," Sasaki and Akiyama (1986, 1987, 1988) successfully applied fuzzy logic to a traffic control process on urban expressways. Along with the pioneering work of Pappis and Mamdani (1977), the work of Sasaki and Akiyama (1986, 1987, 1988) indicated great possibilities offered by fuzzy logic in solving complex traffic engineering problems. Let us present the basic results achieved by Sasaki and Akiyama (1986, 1987, 1988).

The reasons for the application of fuzzy logic in the traffic control on urban expressways come from the fact that both the particular data used to make a decision and the human decision are characterized by fuzziness. In most cases, the available data on the values of certain traffic parameters are not absolutely precise. Thus, for example, due to the fact that a detector is located at 500 meter intervals, the information that traffic congestion length ranges from 2 km to 2.5 km is obtained, although the real traffic congestion length is 2.1 km. According to Sasaki and Akiyama (1988), fuzziness is inherent in other traffic parameters as well, such as traffic volume, traffic demand, weather, and so on. Furthermore, in making decisions experienced operators primarily rely on their intuition and experience. Consequently, the existing "fuzziness in operators" suggests the application of fuzzy logic in order to make the appropriate control procedure.

Sasaki and Akiyama (1988) examined a route having three on-ramps. If traffic congestion becomes large, it is necessary to reduce the number of booths or close the toll gate. Reducing the number of booths or closing the toll gate is interpreted as a traffic control procedure. The control decision involves the determination of the number of open booths of each on-ramp. In order to avoid all possible combinations of the open booths that is, all control patterns, after studying the diary of traffic control, the authors observed four kinds of the most common patterns. The observed control patterns are given in *Table 2.4.*

Table 2.4. The observed control patterns

	Number of open booths			
Pattern	On-ramp 1	On-ramp 2	On-ramp 3	Expected control volume
1	5	2	2	0
2	5	1	1	70-80
3	4	1	1	125-140
4	3	1	1	180-200

The operator chooses an appropriate pattern when he concludes that it is necessary to make a traffic restriction. As Sasaki and Akiyama (1988) noted, "the necessity of control that the operators feel is shown as the operators' expected control volume."

For the sake of simplification, the authors assumed that the operator makes the decisions about the control level (*LEVEL*) that is, his choice of patterns by considering the length of traffic congestion (*CON*) and the expected traffic demand (*DEM*). The data on the length of traffic congestion are obtained by traffic detectors every 5 minutes. It was observed that in estimating the expected traffic demand the operators made use both of the obtained data and of their experience and intuition.

The approximate reasoning algorithm obtained after consulting the operator consisted of the following rules:

If *CON* = low, then *LEVEL* = low,
or
If *CON* = medium and *DEM* = low, then *LEVEL* = low,
or
If *CON* = medium and *DEM* = medium, then *LEVEL* = medium,
or
If *CON* = medium and *DEM* = high, then *LEVEL* = high,
or
If *CON* = high, then *LEVEL* = high.

Fuzzy variables *CON* and *DEM* are input variables whose values should determine the value of the output variable *LEVEL*. Variables *CON*, *DEM* and *LEVEL* could take the values "low," "medium," and "high." The shapes of the chosen membership functions of fuzzy sets "low," "medium," and "high" are shown in *Figure 2.21.*

The membership functions $\mu(x)$ take the form

$$\mu(x) = e^{-\frac{\left(x-\frac{m}{\sigma}\right)^2}{2}} \tag{2.49}$$

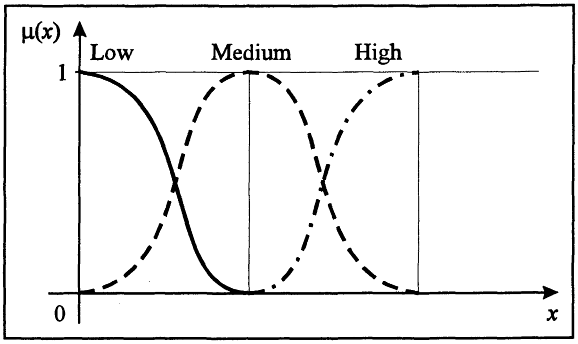

Figure 2.21. Membership functions of fuzzy sets "low," "medium," and "high"

The values of parameters m and σ were obtained after repeating a number of field tests. The authors included into consideration the so-called adaptation index P defined as

$$P = \frac{N_1}{N} \tag{2.50}$$

where N_1 is number of cases in which the pattern from the model is the same as the actual pattern, and N is total number of cases.

The values of parameters in the membership function were chosen leading to the largest value of the adaptation index P.

The developed approximate reasoning algorithm was employed every 5 minutes on the collection of new input data. The results obtained based on the model were found to be almost equal to actual operators' decisions. In other words, it was demonstrated that this model can satisfactorily substitute a human operator.

2.6. SOLVING THE ROUTE-CHOICE PROBLEMS IN URBAN NETWORKS USING FUZZY LOGIC

The problem of assigning traffic in a network is one of the most important problems encountered by traffic specialists. When traffic flows between origins and destinations of movement are known, and when the patterns between the characteristics of certain links and the traffic volumes on them are also known, the problem is to estimate the expected traffic volumes on the links. Different variations of this problem appear in city, noncity, air, and rail traffic. Let us note *Figure 2.22* representing a network with five nodes that are the origins and destinations of movement. The volumes between individual origins and destinations are given in matrix F.

$$F = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} - & 10 & 30 & 10 & 10 \\ 10 & - & 20 & 10 & 70 \\ 10 & 20 & - & 10 & 50 \\ 10 & 20 & 50 & - & 50 \\ 30 & 10 & 50 & 20 & - \end{pmatrix} \end{array} \tag{2.51}$$

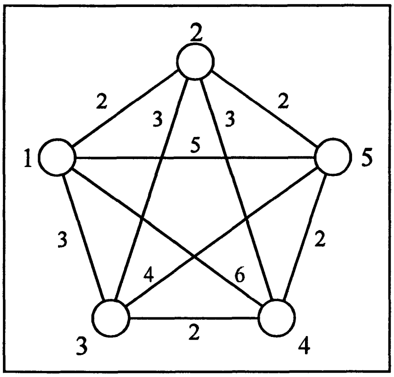

Figure 2.22. Transportation network whose expected volumes on its links must be determined

Travel times along the links are noted in *Figure 2.22*. It is assumed that travel times are constant and independent of the volume assigned to a link. Assuming that the travelers, drivers (in general, the network users) will choose to travel along a path exclusively according to the criterion of travel time, all network users will choose the shortest paths between the origins and destinations of travel. Assigning traffic in a network according to this type of path choice is shown in *Figure 2.23.*

Traffic assignment based on the assumption that all network users are oriented exclusively to the shortest path between pairs of nodes is known as "all or nothing." When travel time along a link is a function of the volume assigned to that link (this is most often a parabolic dependence in city traffic), the assignment shown in *Figure 2.23* would have a completely different look. When assigning traffic in city traffic, the operating assumption is that travel time is the dominant factor influencing users in their choice of path.

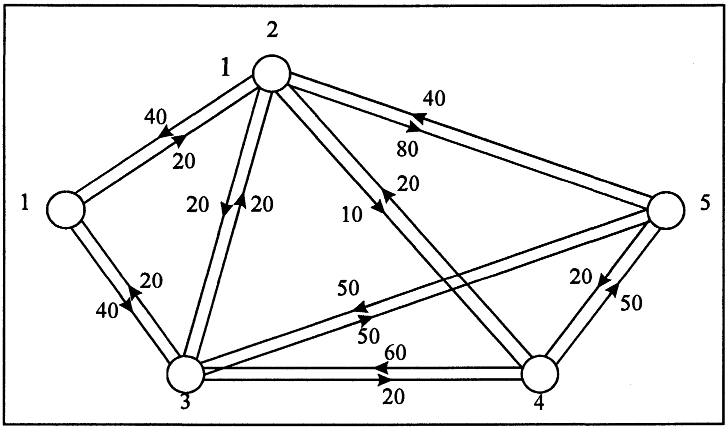

Figure 2.23. Traffic assignment in a network based on the "all or nothing" principle

The deciding factors behind the choice of path on airline networks are travel time, price of transportation on the paths, flight frequency, and number of stopovers.

In the past three decades, the problem of traffic assignment in a network has been considered by a large number of authors worldwide. In addition to the large number of prominent works devoted to the problem of traffic assignment, mention should also be made of the books by Florian (1976) and Sheffi (1985). The problem of traffic assignment in an air traffic network has been discussed by Kanafani and Chang (1979), Kanafani (1981), and Kanafani and Ghobrial (1985).

When travel time along a link (or some other factor that influences the user) is a function of the volume assigned to the link, assuming that every network user tries to minimize his travel time, the network will be in stable conditions when no user is able to decrease his travel time through the network by changing his path. These conditions are known in the literature as *user equilibrium conditions*, and corresponding assignment models are called *user equilibrium models*. The main components of user equilibrium models are traffic assignment based on the principle "all or nothing" (users always choose the shortest path between the origin and destination) and a specific functional dependence between travel time along a link and the traffic volume assigned to the link.

In more recent user equilibrium models a very strict distinction is made between real and perceived travel time (or real and perceived travel costs, real and perceived flight frequency, and so on). Improved user equilibrium models are based on the assumption that the choice of path depends on perceived, and not on observed, travel time. Perceived travel time is treated in such models as a random variable, whereby every network user corresponds to a specific value of perceived travel time. Stable conditions in this case arise when no network user believes that any change in his path could decrease his travel time. Such conditions are known in the literature as *stochastic user equilibrium*, and corresponding models are called *stochastic user equilibrium* models. A stochastic model to "load" the network and a functional dependence between travel time on a link and the volume assigned to the link are the main components of stochastic user equilibrium models.

The system approach to assigning traffic in a network endeavors to assign user flows so that the total travel time of all users through the network is minimized.

2.6.1. A fuzzy logic route-choice model

When choosing which path to take between the origin and destination of movement, the user usually does not have exact information concerning travel times along different paths. In the same vein, an air traveler usually does not have all information (particularly for large networks) regarding the number of departures, all departure times, all tariffs, the number of stopovers on certain paths, and so on. Using these assumptions as a point of departure, Teodorovic and Kikuchi (1990) developed a model to assign traffic in a network based on fuzzy logic. The following discussion will present the results of their work.

Let us assume that every network user chooses the path he will take through the network based on perceived travel time along certain paths. Perceived travel time is very often "fuzzy." In other words, when subjectively estimating travel time between two points, expressions are used such as "it takes about 20 minutes from point A to point B." It is rarely if ever heard that travel time between points A and B is 19 minutes. The claim that travel time between two points is "about 20 minutes" is the result of a subjective feeling, an individual's subjective estimate. This is not the result of any measuring or the realization of a random variable representing travel time. If we were to record travel time between points A and B over a longer period of time, we would receive a series of different values, each representing one realization of the random variable travel time. When we subjectively estimate travel time, we do not have information regarding the probability density function of travel time; rather, we base our estimate on experience and intuition. The estimated travel time (or travel costs, flight frequency, and so on) differs from passenger to passenger, from user to user.

User perceived travel times can be treated as fuzzy sets. Thus, the fuzzy set "travel time is about 20 minutes" denotes the travel time estimated by the first user, while the fuzzy set "travel time is about 25 minutes" can denote the travel time estimated by the second user.

Let us assume that a user has a specific preference regarding the choice of each of the possible paths through the network. This preference can be "stronger" or "weaker." Let us introduce into the discussion a preference index that can take values from the interval of 0 to 1. When the user has an absolute preference for a specific path, we consider the preference index to be equal to 1. This preference index decreases along with a decrease in the strength of the preference. Let us assume that the strength of the user's preference can be expressed by fuzzy sets such as "very strong," "strong," "medium," "weak," and "very weak." Let us consider the simplest case when the user is to choose one of two possible paths between the origin and destination of movement (*Figure 2.24*).

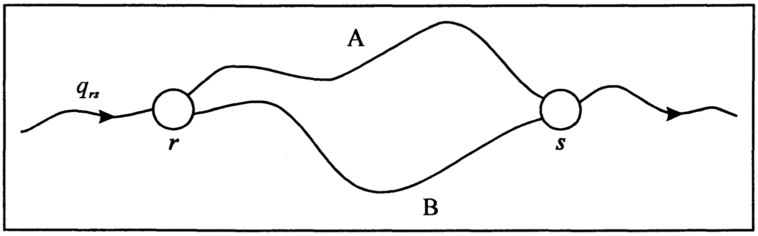

Figure 2.24. Choosing one of two possible paths between the origin and destination

The origin is denoted by r, the destination by s. We assume that we know the traffic volume between r and s. This volume is denoted by q_{rs}. Possible paths are marked by A and B. The number of possible paths between two nodes in a network is most often very large, but in practice network users usually consider only two or three alternative paths. Teodorovic and Kikuchi (1990) assumed that users choose their paths based on a comparison of the characteristics of alternative paths. Let us denote by **TA** and **TB**, respectively, fuzzy sets representing estimated travel time (by one user) along path A and path B. When comparing travel times along paths A and B, the user might estimate, for example, that travel time along path A, **TA** is "much shorter than **TB**," "shorter than **TB**," "equal to **TB**," "longer than **TB**," and "much longer than **TB**." These fuzzy sets are denoted in the following manner: **MLTB** is much less than **TB**, **LTB** is less than **TB**, **GTB** is greater than **TB**, and **MGTB** is much greater than **TB**.

Figure 2.25 presents perceived travel time along path B and fuzzy sets **MLTB**, **LTB**, **GTB**, and **MGTB**.

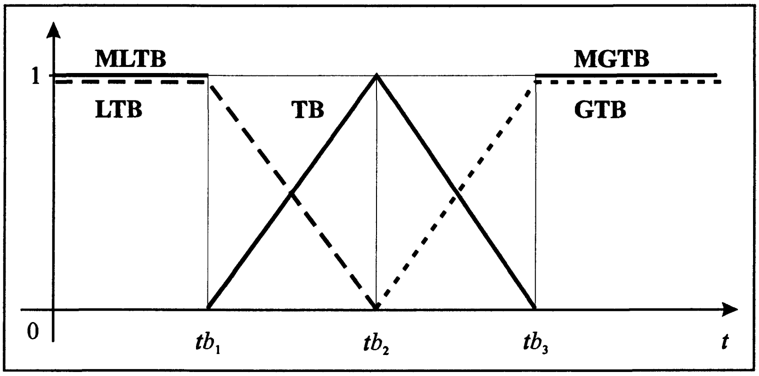

Figure 2.25. Perceived travel time along path B and fuzzy sets **MLTB**, **LTB**, **GTB**, and **MGTB**

In the example in *Figure 2.25*, perceived travel time is presented as a triangular fuzzy number. The main reason for assuming that perceived travel time is a fuzzy triangular number is that arithmetic operations are the easiest when triangular fuzzy numbers are concerned. Future research should examine the form of the membership function of perceived travel time in greater detail. The membership functions corresponding to the fuzzy sets shown in *Figure 2.25* are

$$\mu_{\mathbf{MLTB}}(t) = \begin{cases} 1, & t \le tb_1 \\ 0, & t \ge tb_1 \end{cases} \tag{2.52}$$

$$\mu_{\mathbf{LTB}}(t) = \begin{cases} 1 - \mu_{\mathbf{TB}}(t), & t \le tb_2 \\ 0, & t \ge tb_2 \end{cases} \tag{2.53}$$

$$\mu_{\mathbf{GTB}}(t) = \begin{cases} 0, & t \le tb_2 \\ 1 - \mu_{\mathbf{TB}}(t), & t \ge tb_2 \end{cases} \tag{2.54}$$

$$\mu_{\text{MGTB}}(t) = \begin{cases} 0, & t \le tb_3 \\ 1, & t \ge tb_3 \end{cases} \tag{2.55}$$

where tb_1, tb_2, and tb_3 are the smallest expected, the largest expected, and the largest possible values of perceived travel time TB.

Let us denote by p_A and p_B the preference indexes associated with paths A and B. The fuzzy sets "very strong," "strong," "average," "weak," and "very weak" preferences are shown in *Figure 2.26.*

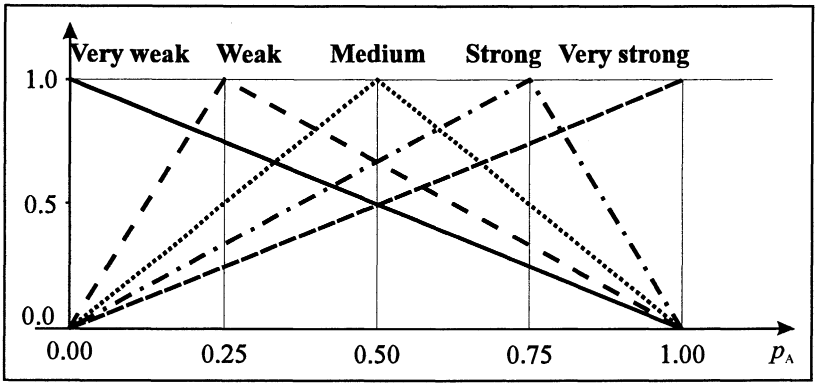

Figure 2.26. Fuzzy sets describing preference strength

As already mentioned, Teodorovic and Kikuchi (1990) assumed that the user chooses his path based on a comparison with the characteristics of alternative paths. For example, it is logical to assume that the user's preference for path A will be "very strong" should the user estimate that travel time along path A is "much shorter" than travel time along path B. Teodorovic and Kikuchi proposed the following fuzzy rules to establish network user's preference strength:

If **TA** = **MLTB**, then **PA** = **Very strong,**
or
If **TA** = **LTB**, then **PA** = **Strong,**
or
If **TA** = **TB**, then **PA** = **Medium,**
or
If **TA** = **GTB**, then **PA** = **Weak,**
or
If **TA** = **MGTB**, then **PA** = **Very weak.**

For network user's known perceived travel times **TA** and **TB**, approximate reasoning by max-min composition can be used to establish the preference strength for every network user.

This approximate reasoning procedure is graphically shown in *Figure 2.27*. For perceived travel times **TA** and **TB**, we must first establish the extent to which the perceived travel times satisfy the premises of the rules. In other words, we must first establish the extent to which perceived travel time **TA** is "much shorter than **TB**," "shorter than **TB**," "longer than **TB**," or "much longer than **TB**" (*Figure 2.27(a) - (e)*). *Figure 2.27(f)* presents the membership function of the preference index that is obtained after applying the approximate reasoning procedure. One numerical preference index value that is equal to the center of gravity of the obtained fuzzy set is chosen.

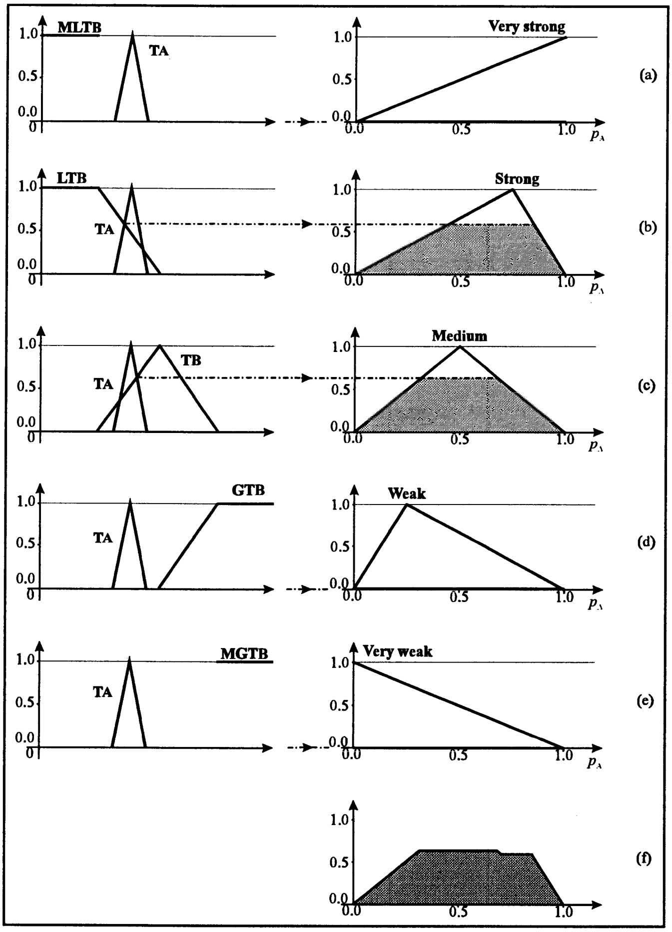

Figure 2.27. Graphical interpretation of the approximate reasoning algorithm to determine preference strength

2.6.1.1. Network "loading" algorithm based on hybrid numbers

Once the preference index has been determined for each network user, an algorithm must be developed that will determine the number of users along individual links. Such algorithms are known in the literature as network *loading algorithms*. Teodorovic and Kikuchi (1990) developed a network loading algorithm in which the basic input data are the preference indexes of individual network users. We will briefly present the Teodorovic and Kikuchi algorithm.

Let us assume that q_{rs} users want to travel from node r to node s. We also assume that each of the q_{rs} users perceives travel time as a triangular fuzzy number, and that a user's perceived travel times are distributed around the "most probable" travel time arrived at by previous measurements. Let us denote by **TA** and **TB**, respectively, fuzzy numbers representing the "most probable" travel times along paths A and B.

Kaufmann and Gupta (1985) introduced the concept of a hybrid number representing the "combination" of a fuzzy number and a random variable. Hybrid number (**TA**, **L**) "composed" of fuzzy number **TA** and random variable **L** is shown in *Figure 2.28.*

Hybrid number (**TA**, **L**) is "composed" of a series of fuzzy numbers, each obtained by "shifting" fuzzy number **TA** in a random way along the abscissa. Depending on the normal distribution of random variable **L**, diverse "distributions" of fuzzy numbers are obtained along the abscissa.

Teodorovic and Kikuchi assumed that user's perceived travel time could be a hybrid number. In other words, it was assumed that perceived travel time along path A could be represented as a hybrid number (**TA**, **L**) with **L** being a random variable having a normal distribution. Using fuzzy numbers **TA** and **TB** and random variable **L**, it is possible to generate q_{rs} perceived times along path A, with each of the generated perceived travel times corresponding to one user. Applying the earlier-presented algorithm of approximate reasoning to determine the preference strength for each q_{rs} user, it is possible to determine corresponding preference indexes.

Let us denote by $P_{\mathrm{A}i}$ the preference index of the ith network user in terms of using path A. The total number of network users who will use path A can be expressed as

$$q_{\mathrm{A}} = \sum_{i=1}^{q_{rs}} P_{\mathrm{A}_i} \tag{2.56}$$

The number of users on path B is calculated as

$$q_B = q_{rs} - q_A \tag{2.57}$$

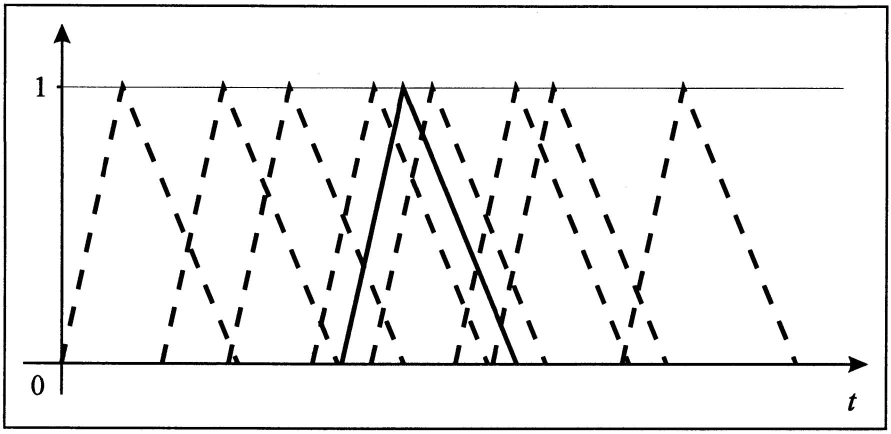

Figure 2.28. Hybrid number (**TA, L**)

As already mentioned, individual perceived travel time is calculated using the assumption of a normal distribution of random variable **L**. In other words, in order to calculate perceived travel times along paths A for q_{rs} users, q_{rs} random numbers with a normal distribution must be generated. It is clear that calculated values q_A and q_B considerably depend on the series of generated q_{rs} random numbers. In the same vein, for different series of generated random numbers, various values of estimated volumes q_A and q_B are obtained. Using Daganzo and Sheffi's idea (1977), presented in a model where perceived travel time was treated as a random variable with a normal distribution, Teodorovic and Kikuchi (1990) proposed reaching the estimation of values q_A and q_B after repeating the entire calculation greater number of times for different generated sets of random numbers. We denote by q_A^j the estimated value of the volume along path A in the *j*th iteration. Volume q_{AK} along path A is estimated after K iterations that is, after generating K sets of random numbers and making an estimation for each of these sets. It is calculated as

$$q_{AK} = \frac{1}{K} \sum_{j=1}^{K} q_A^j \tag{2.58}$$

The corresponding standard deviation σ_{AK} is

$$\sigma_{AK} = \frac{1}{K} \sum_{j=1}^{K} \left(q_A^j - q_{AK} \right)^2 \tag{2.59}$$

The iterations end when, for example,

$$\frac{\sigma_{AK}}{q_{AK}} < c_0 \tag{2.60}$$

where c_0 is a previously given constant whose value determines the accuracy of the entire procedure.

The network "loading" algorithm based on hybrid numbers consists of the following algorithmic steps:

Step 1: Define the "highest expected" perceived travel times along paths A and B. Present these travel times in the form of triangular fuzzy numbers. Give a value to constant c_0, which determines the accuracy of the output results.

Step 2: Generate (using a normal distribution) q_{rs} random numbers, which are used to determine perceived travel times for each q_{rs} network user along path A. Using new q_{rs} random numbers determine perceived travel times along path B.

Step 3: Using the approximate reasoning model to choose the path, calculate the volumes along paths A and B.

Step 4: Repeat Steps 2 and 3 K times, and calculate corresponding volumes.

Step 5: If a satisfactory accuracy has been achieved (defined by the constant c_0), end the algorithm. Otherwise, increase K by 1 (K = K+1), and continue with Steps 2, 3, and 4.

The above approximate reasoning algorithm and the network "loading" algorithm are based on the assumption that perceived travel times are a dominant factor that influences the user to choose certain paths in the network. It is certain that numerous other factors influence the choice of path through the network. Involving other factors into analysis would require the development of more complex approximate reasoning algorithms.

2.6.2. Route-choice behavior by multistage fuzzy reasoning

Akiyama and Tsuboi (1996) used multistage fuzzy reasoning to describe the driver decision making process on road networks. Their paper considers the multiroute choice problem. Akiyama and Tsuboi (1996) first made a survey and collected data. Experiment participants were asked about the number of alternative routes they use going from their origin to their destination (the authors asked the respondents to cite at most three alternatives they may use). For each route considered they defined its characterizing factors (travel time, degree of congestion, and risk of accidents). All the participants estimated the values of all the factors using their own perception and experience. Appropriate fuzzy numbers could then represent these values. The third survey question dealt with the values of the utilities of the nominated routes. The authors assumed the existence of two stages in the driver decision-making process. The first approximate reasoning algorithm they proposed determined the utilities of the alternative routes. A typical rule in this approximate reasoning algorithm reads:

If T is large and CR is small and RA is medium, then V is small

where T is perceived travel time, CR is congestion of a route, RA is risk of accident, and V is utility for the route.

The first input variable DF in the second approximate reasoning algorithm is the difference between the utilities associated with the shortest path and the second-shortest path. The second input variable DS is the difference in the utilities between the second-shortest path and the third-shortest path. The third input variable N refers to the number of alternative routes. When $N = 2$, variable DS does not appear in the premises. A typical rule of the second approximate reasoning algorithm reads:

If $N = 3$ and DF is medium and DS is medium, then $FR(1)$ is large $FR(2)$ is medium $FR(3)$ is medium

where DF is difference of utilities between the shortest path and the second-shortest path, DS is difference of utilities between the second-shortest path and the third-shortest path, and $FR(i)$ is degree of frequency for route i.

Akiyama and Tsuboi (1996) also developed a neural network model for the second stage of estimation. The neurons in the input layer represent the number of alternative routes and the values of the utilities of individual alternative routes. The network output is route frequency. Connecting weights are determined using the back propagation method. The two-stage model based on a combination of fuzzy logic (first stage) and neural network (second stage) produced somewhat better results than the two-stage model based on fuzzy logic in both stages.

2.7. MODELING ROUTE CHOICE WITH ADVANCED TRAVELER INFORMATION BY FUZZY LOGIC

Lotan and Koutsopoulos (1993a, 1993b) studied the route-choice behavior problem. They indicated the great importance of studying this problem, especially in the light of intensive research connected with the development of *intelligent vehicle highway systems* (IVHS). When developing their approximate reasoning algorithm Lotan and Koutsopoulos used rules dealing with perceived travel times (or other measures of attractiveness) and rules dealing with real-time traffic information. Real-time traffic information on current traffic conditions was primarily used to indicate how default behavior changes. The attractiveness of different alternatives was measured on a scale ranging from -1 to 1. A value of -1 indicates a complete aversion toward taking the alternative, while a value of +1 corresponds to an assured choice of the alternative. Once the initial set of rules was established, in order to improve the model it was necessary to test, update, and expand the rules.

Continuing the research of Lotan and Koutsopoulos (1993a, 1993b), Vythoulkas and Koutsopoulos (1994) studied the modeling of discrete choice behavior using techniques and concepts from fuzzy set theory and neural networks. The authors used the neuro-fuzzy approach in order to calibrate the proposed model. They assumed that the membership functions of the fuzzy sets that appear in different rules are bell-shaped (gaussian functions). The authors presented the problem of rule generation as an integer programming optimization problem. Vythoulkas and Koutsopoulos (1994) used the fuzzy neural network procedure proposed by Lin and Lee (1991) to optimize the parameters of the membership functions, as well as to adjust the rule weights. The developed neural network was trained using an adaptation of the generalized delta rule. Two sets of data were used in order to test the proposed approach. The first group represented stated preference data collected by Lotan and Koutsopoulos (1993b) referring to the choice of one out of three alternative routes in Boston in the presence of information. The second group consisted of revealed preference data of the choice between rail and car for intercity travel. Data were obtained by a population survey in the city of Nijmegen in the Netherlands. The authors made a number of experiments with different combinations of input variables and finally concluded that the most important variables for mode choice decision are travel time, travel cost, and rail access/egress time. Rule premises used differences between travel time by rail and travel time by car and differences between travel cost by rail and travel cost by car. The model was calibrated using an error function that was the sum of the squares of the deviations between actual choice and calculated preference.

In the past several years there has been an intense study of different aspects of *advanced traveler information systems* (ATIS) in the world. ATIS is certainly one of the very important elements of intelligent vehicle highway systems that are being intensively developed in a large number of research centers. The basic task of ATIS is to provide certain information to travelers so that they will be better informed when they make their travel decisions. ATIS studies to date have used field experiments, route choice surveys, and computer simulation games. The route-choice problem is one of the most important transportation planning problems, and further study of route choice with advanced traveler information is certainly of great interest. How do the characteristics of competitive routes influence route choice when there is ATIS? How do travelers' characteristics influence route choice? How much confidence does the driver have in information received from ATIS? How do travelers perceive the information they receive? What is the extent of previously gained traveler experience, and how does it affect route choice?

Research to date has provided answers to some of these questions. Kanafani and Al-Deek (1991) studied the benefits of vehicle route guidance in urban networks. They concluded that "saving in total system travel time on the order of 3 to 4% can be achieved from route guidance." Caplice and Mahmassani (1992) considered commuters' preferred arrival times at the workplace, the use of traffic information and switching propensity. Koutsopoulos and Xu (1993) developed a routing strategy based on information discounting for travel time projection. They also considered different aspects of ATIS (frequency of information update, location of information nodes, approaches to estimate travel times). Al-Deek and Kanafani (1993) dealt with the problem of determining the benefits from ATIS in corridors under incident conditions. Their paper also develops a deterministic queuing model that is applied to an idealized corridor composed of two routes. The interactions between traffic information reliability and travelers' response (route-choice decisions) were studied by Mahmassani and Shen-Te Chen (1993). Hall's research (1993) concluded that ATIS cannot be a solution to peak-period congestion. Yang et al. (1993) investigated driver route choice with ATIS using neural network concepts, first conducting a large number of experiments using interactive computer simulation in order to collect necessary data. They determined that the greatest number of users choose a route primarily based on recent experience and developed a model based on neural network concepts. Further consideration of different aspects of ATIS and finding partial answers to some of the most important questions is most certainly an extremely important area for future research in the field of traffic planning.

Depending on the type of system, ATIS can provide drivers with information that can be qualitative, quantitative, predictive, and prescriptive. Teodorovic et al. (1998) that ATIS provides drivers with quantitative and predictive information. In other words, they assumed that ATIS provides drivers with certain quantitative information regarding estimated travel time along competitive routes. Travelers use this information to compare the competitive routes. When comparing travel time along alternative routes, the user may estimate, for example, that travel time along a certain route is "equal to," or "much shorter," "shorter," "longer," or "much longer" than travel time along a competitive route. It is also logical to assume that previously gained experience in using ATIS has a great influence on the route choice. In this regard, drivers are able to make a daily comparison between actual travel time and travel time estimated by ATIS. Differences between real and estimated travel time (which can be, for example, "very small," "small," "large," and so on) along alternative routes have an extremely important impact on the degree of user preference for a certain route. The degree of preference can be linguistically stated as "very weak," "weak," "medium," "strong," or "very strong."

Teodorovic et al. (1998) assumed that the differences in estimated travel time along alternative routes, differences between estimated and real travel times along different routes, and the degree of preference to take certain routes can be represented by corresponding fuzzy sets. Therefore, the basic goal of their paper was to attempt to develop a travelers' route-choice behavior model based on fuzzy set theory techniques. The travelers' route-choice behavior model developed is based on fuzzy logic. The rules contained in the approximate reasoning algorithm they developed were reached by learning from examples. Model development and testing used data obtained from computer simulation; twenty six travelers participated in the simulation experiment.

2.7.1. Interactive computer simulation for data collecting

The experiment conducted by Teodorovic et al. (1998) included twenty six drivers (students and professors at the Faculty of Transport and Traffic Engineering, University of Belgrade). We would note that the interactive computer simulation conducted by Teodorovic et al. (1998) was inspired by Yang et al. (1993). The drivers chose one of two alternative routes during every day for thirty days of the month. The experiment participants were told the "estimated travel time" along each alternative route, and it was up to them to choose one of the competitive routes. After they had made their choice, the drivers were told the "real travel time" along the chosen route, and the difference between "estimated" and "real" travel time. This allowed the drivers to gain experience connected with the accuracy of ATIS predictions. Every driver made a route choice every day of the month. This enabled the user to accumulate experience in using ATIS. "Estimated" and "real" travel times for every day in the month were generated randomly using the Gaussian probability density function. In other words, the following probability density function was used to generate travel times:

$$f(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.61}$$

For "estimated" travel times, the values of corresponding parameters were

Route 1: $\mu = 20\,\text{min}$ $\sigma = 5\,\text{min}$
Route 2: $\mu = 22\,\text{min}$ $\sigma = 5\,\text{min}$

For "real" travel times, the parameter values were

Route 1: $\mu = 20\,\text{min}$ $\sigma = 7\,\text{min}$
Route 2: $\mu = 22\,\text{min}$ $\sigma = 7\,\text{min}$

2.7.2. Analysis of the factors that influence route choice using ATIS

Drivers make their route choice after comparing the characteristics of the alternative routes. When ATIS exists, travelers certainly bear in mind information received from ATIS when making their route choice. Teodorovic et al. (1998) assumed that users make their route-choice decisions based on immediately perceived travel time along different routes (based on travel time information received from ATIS) and based on experience gained from "using" ATIS. Generally speaking, regardless of whether ATIS exists or not, perceived travel time on different routes is characterized by fuzziness. Users' perceived travel times can be represented by corresponding fuzzy sets. We also assume that the user has a certain preference for the choice of a certain route. This preference can be "stronger" or "weaker." We will introduce into the discussion a preference index that can take a value between 0 and 1. When a user has an absolute preference to use a certain route we consider the preference index to be 1. Should all travelers have a preference index of 1 regarding a certain route, this would mean that 100% of the travelers would use that route. It is clear that a decrease in the degree of user preference brings a decline in the percentage of travelers using that route. The percentage of users on a certain route the *i*th day P(i) can be expressed by fuzzy sets called "very great percentage," "great percentage," "small percentage," and so on. Let us consider the simplest situation when users must choose one of two alternative routes.

Let us denote by $\Delta T(i)$ the difference in travel times estimated by ATIS between route I and route II on the *i*th day under observation. This difference is shown to all drivers. Each driver experiences this value in a different way. For some drivers it might be "big," while another group of drivers might consider it "very big." In this research it is assumed that when there is ATIS, the value of $\Delta T(i)$ is one of several fuzzy variables having an important influence on route choice. Once information is received about the value of difference $\Delta T(i)$, and taking other factors into consideration, drivers make their route choice. When real travel time along the chosen route is shorter than that estimated by ATIS, drivers feel that they have made a good route choice, and in a repeated situation the next day they will certainly make the same choice. On the other hand, when real travel time is longer than expected, the degree of preference for that route decreases, and the degree of preference for the alternative route increases the next day. We can therefore ascertain that each of the alternative routes is characterized by a certain convenience and a certain shortcoming. Let us try to find specific "measures" that will be used to measure the conveniences and shortcomings of alternative routes. Let us denote by $C_I(i)$ the convenience of using route I the *i*th day in question. Convenience $C_I(i)$ is greater the greater the number of users who were satisfied with the choice of route I the (i-1)th day, and the greater the number of users who were dissatisfied with the choice of route II the (i-1)th day. Let us denote by $C_I^{II}(i)$ the "share" of convenience $C_I(i)$ originating from users who were satisfied with their choice of route I the (i-1)th day. Let us also denote by $C_I^{II}(i)$ the "share" of convenience $C_I(i)$ originating from users who were dissatisfied with their choice of route II the (i-1)th day. Values $C_I^I(i)$ and $C_I^{II}(i)$ are defined as follows:

$$C_I^I(i) = \begin{cases} \dfrac{\left|t_r^I(i-1) - t_e^I(i-1)\right|}{n} \cdot n_I(i-1), & t_r^I(i-1) \le t_e^I(i-1) \\ 0, & \text{otherwise} \end{cases} \tag{2.62}$$

where $t_r^{\mathrm{I}}(i-1)$ is real travel time along route I the (i-1)th day, $t_e^{\mathrm{I}}(i-1)$ is estimated (by ATIS) travel time along route I the (i-1)th day, $n_{\mathrm{I}}(i\text{-}1)$ is number of travelers who chose route I the (i-1)th day, and n is total number of travelers

$$\mathrm{C}_{\mathrm{I}}^{\mathrm{II}}(i)=\begin{cases}\left|t_r^{\mathrm{II}}(i-1)-t_e^{\mathrm{II}}(i-1)\right|\cdot\dfrac{n_{\mathrm{II}}(i-1)}{n}, & t_r^{\mathrm{II}}(i-1)>t_e^{\mathrm{II}}(i-1)\\ 0, & \text{otherwise}\end{cases}\tag{2.63}$$

where $t_r^{\mathrm{II}}(i-1)$ is real travel time along route II the (i-1)th day, $t_e^{\mathrm{II}}(i-1)$ is estimated (by ATIS) travel time along route II the (i-1)th, $n_{\mathrm{II}}(i-1)$ is number of travelers who choose route II the (i-1)th day, and n is total number of travelers.

Let us denote by $\mathrm{S}_{\mathrm{I}}(i)$ the shortcoming of using route I the ith day in question. Shortcoming $\mathrm{S}_{\mathrm{I}}(i)$ is greater the greater the number of users who were dissatisfied with their choice of route I the (i-1)th day, and the greater the number of users who were satisfied with their choice of route II the (i-1)th day. Let us also denote by $\mathrm{S}_{\mathrm{I}}^{\mathrm{I}}(\mathrm{i})$ the "share" of shortcoming $\mathrm{S}_{\mathrm{I}}(\mathrm{i})$ originating from users who were dissatisfied with their choice of route I the (i-1)th day. Let us also denote by $\mathrm{S}_{\mathrm{I}}^{\mathrm{II}}(i)$ the "share" of shortcoming $\mathrm{S}_{\mathrm{I}}(i)$ originating from users who were satisfied with their choice of route II the (i-1)th day. Values $\mathrm{S}_{\mathrm{I}}^{\mathrm{I}}(i)$ and $\mathrm{S}_{\mathrm{I}}^{\mathrm{II}}(i)$ are defined as follows:

$$\mathrm{S}_{\mathrm{I}}^{\mathrm{I}}(i)=\begin{cases}\dfrac{\left|t_r^{\mathrm{I}}(i-1)-t_e^{\mathrm{I}}(i-1)\right|}{n}\cdot n_{\mathrm{I}}(i-1), & t_r^{\mathrm{I}}(i-1)>t_e^{\mathrm{I}}(i-1)\\ 0, & \text{otherwise}\end{cases}\tag{2.64}$$

$$\mathrm{S}_{\mathrm{I}}^{\mathrm{II}}(i)=\begin{cases}\left|t_r^{\mathrm{II}}(i-1)-t_e^{\mathrm{II}}(i-1)\right|\cdot\dfrac{n_{\mathrm{II}}(i-1)}{n}, & t_r^{\mathrm{II}}(i-1)\le t_e^{\mathrm{II}}(i-1)\\ 0, & \text{otherwise}\end{cases}\tag{2.65}$$

Values $\mathrm{C}_{\mathrm{II}}^{\mathrm{I}}(i)$, $\mathrm{C}_{\mathrm{II}}^{\mathrm{II}}(i)$, $\mathrm{S}_{\mathrm{II}}^{\mathrm{I}}(i)$ and $\mathrm{S}_{\mathrm{II}}^{\mathrm{II}}(i)$ are defined in the same manner. The total convenience and the total shortcoming of route I and route II, respectively, are

$$\mathrm{C}_{\mathrm{I}}(i)=\mathrm{C}_{\mathrm{I}}^{\mathrm{I}}(i)+\mathrm{C}_{\mathrm{I}}^{\mathrm{II}}(i)\tag{2.66}$$

$$C_{II}(i) = C_{II}^{I}(i) + C_{II}^{II}(i) \tag{2.67}$$

$$S_{I}(i) = S_{I}^{I}(i) + S_{I}^{II}(i) \tag{2.68}$$

$$S_{II}(i) = S_{II}^{I}(i) + S_{II}^{II}(i) \tag{2.69}$$

As already noted, participants in the simulation experiment chose one of two alternative routes every day for thirty days of the month. Since the first simulated day (i=1) the drivers had no information relating to the previous day, the corresponding values of the "convenience" and "shortcoming" were zero ($C_I(1) = C_{II}(1) = S_I(1) = S_{II}(1) = 0$). Values $C_I(i)$, $C_{II}(i)$, $S_I(i)$, $S_{II}(i)$, as already noted, are the "measures" of the convenience or shortcoming of each route. They indicate the influence of the previous day's route choice on the route choice of the day under observation. In this research it is considered that the feeling of having made a "good" or "bad" route choice the previous day is very present when making the route choice the following day. In this way, these values play the role of a special "short-term memory." Teodorovic et al. (1998) assumed that in addition to this "short-term memory" there is also a "long-term memory" that is also present in the route choice thought mechanism. This "long-term memory" should in a way indicate the "cumulative belief" that the travelers might have reached (based on ATIS suggestions) regarding which route is "better" that is, "shorter" or "faster." Let us denote by $B(i)$ the "measure" of the cumulative belief regarding the ith day of the month. This "measure" is the mean value of the differences between the estimated travel times along competitive routes, until ith day that is,

$$B(i) = \frac{\sum_{j=1}^{i-1} (t_e^{I}(j) - t_e^{II}(j))}{i-1} \tag{2.70}$$

Depending on value $B(i)$ and the individual's subjective feeling, a route might be treated as "bad," "average," "good," and so on. We can therefore state the input fuzzy variables: ΔT is difference in travel times estimated by ATIS, C is convenience of the route, S is shortcoming of the route, and B is cumulative belief regarding route quality.

The output fuzzy variable P is percentage of travelers who choose to use the route in question on the ith day in question.

2.7.3. Fuzzy logic and the route-choice problem

As we already mentioned, Teodorovic and Kikuchi (1990) were the first to use fuzzy logic in considering the route-choice problem. It should be underscored that Lotan and Koutsopoulos (1993b) were the first to link fuzzy logic and the route-choice problem within the context of ATIS, whereby a certain theoretical consideration is supported by limited empirical data.

In the considered route choice using ATIS, one of the possible rules might be, for example,

If ΔT is T_1 and C is C_3 and S is S_3 and B is B_3, then P is P_{11}

This rule represents the fuzzy relation between the difference in travel times estimated by ATIS (ΔT), route convenience (C), route shortcoming (S), cumulative belief regarding route "quality" (B), and the percentage of travelers who choose the route in question (P).

As mentioned earlier, the simulation experiment whose goal was to collect appropriate data included twenty six drivers. The drivers were able to gain experience regarding the accuracy of ATIS predictions. Each driver made a route choice for every day of the month, resulting in the driver's cumulated experience regarding ATIS. It was noted during conversations with the simulation experiment participants that they had difficulty in expressing how they made their route choice using linguistic rules. For this reason, Teodorovic et al. (1998) decided to generate fuzzy rules from available numerical data. The method they used to generate fuzzy rules from numerical data was that proposed by Wang and Mendel (1992c).

During each of the thirty days in the simulation experiment, a certain number of drivers chose route I. Since on the first day the experiment participants did not have any information regarding the previous day, we received twenty nine input-output data pairs for the first route and twenty nine input-output data pairs for the second route. The authors therefore had at their disposal the following fifty eight input-output data pairs: (ΔT(1), C(1), S(1), B(1), P(1)), (ΔT(2), C(2), S(2), B(2), P(2)), ..., (ΔT(58), C(58), S(58), B(58), P(58)).

In the Teodorovic et al.'s (1998) paper the fuzzy rules base was obtained exclusively based on available numerical information. The results of the simulation experiment conducted in order to collect necessary data are shown in *Table 2.5*, *Table 2.6* and *Table 2.7*. The data shown in *Table 2.6* refer to route I, while those in *Table 2.7* refer to route II. As can be seen, the tables show not only data obtained from the examinees' answers, but also the corresponding values of variables ΔT, C, S, B, and P.

Based on the data given in *Table 2.6* and *Table 2.7*, using the method proposed by Wang and Mendel (1992c), the following thirty eight fuzzy rules were generated:

Rule 1: If $\Delta T = T_1$ and $C = C_1$ and $S = S_1$ and $B = B_1$, then $P=P_1$,
or
Rule 2: If $\Delta T = T_1$ and $C = C_1$ and $S = S_2$ and $B = B_1$, then $P=P_1$,
or
...
Rule 38: If $\Delta T = T_{11}$ and $C = C_1$ and $S = S_3$ and $B = B_3$, then $P=P_{10}$.

Table 2.5. Route characteristics

	Estimated travel time (min)		"Real" travel time (min)		Difference between estimated and "real" travel time	
	Route I	Route II	Route I	Route II	Route I	Route II
Day	t_{EI}	t_{EII}	t_{RI}	t_{RII}	$t_{RI}-t_{EI}$	$t_{RII}-t_{EII}$
1.	25.61	18.65	22.45	21.78	-3.16	3.13
2.	16.98	24.83	19.91	25.41	2.93	0.58
3.	17.30	29.22	15.14	31.47	-2.16	2.25
4.	18.14	31.51	20.32	13.32	2.18	-18.19
5.	25.91	22.61	20.26	22.78	-5.65	0.17
6.	24.25	19.60	22.21	19.76	-2.04	0.16
7.	14.33	32.05	22.93	24.87	8.60	-7.18
8.	19.38	15.89	24.81	19.82	5.43	3.93
9.	28.72	30.87	20.69	21.04	-8.03	-9.83
10.	13.51	17.68	19.01	43.51	5.50	25.83
11.	28.95	32.36	24.81	20.22	-4.14	-12.14
12.	13.45	23.30	24.12	14.38	10.67	-8.92
13.	18.40	38.98	23.87	20.70	5.47	-18.28
14.	20.98	21.57	23.60	22.16	2.62	0.59
15.	17.51	28.67	16.31	26.90	-1.20	-1.77
16.	14.24	27.16	28.32	19.62	14.08	-7.54
17.	18.20	28.99	20.87	12.10	2.67	-16.89
18.	18.64	21.60	20.92	34.79	2.28	13.19
19.	20.29	14.55	22.27	26.86	1.98	12.31
20.	19.65	31.17	15.16	24.44	-4.49	-6.73
21.	18.54	24.57	14.41	31.90	-4.13	7.33
22.	17.65	18.55	15.64	24.27	-2.01	5.72
23.	15.19	19.92	23.40	29.47	8.21	9.55
24.	23.81	21.62	19.01	28.65	-4.80	7.03
25.	23.10	40.16	18.08	33.67	-5.02	-6.49
26.	12.26	25.48	21.12	31.60	8.86	6.12
27.	15.71	25.21	28.45	23.61	12.74	-1.60
28.	23.93	20.46	21.65	27.87	-2.28	7.41
29.	20.74	23.79	20.08	30.27	-0.66	6.48
30.	15.94	22.16	27.90	35.05	11.96	12.89

The comparison was made between the results obtained from the approximate reasoning algorithm and real results obtained from the simulation experiment. The real results and the results obtained from the approximate reasoning algorithm are shown in *Table 2.8* and *Figure 2.29.*

Table 2.6. Percentage of drivers choosing route I

Number of drivers along route						
Route I	Route II					
n_{I}	n_{II}	ΔT	C	S	B	P
9	17	6.96				35
24	2	-7.85	3.1	0.0	0.44	92
26	0	-11.91	0.0	2.7	4.27	100
26	0	-13.37	2.2	0.5	6.54	100
14	12	3.30	0.0	2.2	4.58	54
10	16	4.65	3.1	0.0	3.04	38
25	1	-17.71	1.0	0.0	5.13	96
14	12	3.49	0.0	8.5	4.05	54
16	10	-2.16	1.8	2.9	3.84	62
22	4	-4.17	4.9	3.8	3.88	85
21	5	-3.40	4.0	4.7	3.83	81
25	1	-9.85	3.4	2.3	4.33	96
23	3	-20.58	0.0	11.0	5.58	88
13	13	-0.59	0.0	6.9	5.23	50
25	1	-11.16	0.0	1.3	5.62	96
26	0	-12.92	1.2	0.1	6.08	100
20	6	-10.78	0.0	14.0	6.36	77
18	8	-2.96	0.0	5.9	6.17	69
7	19	5.74	4.1	1.6	5.54	27
24	2	-11.53	9.0	0.5	5.84	92
22	4	-6.03	4.1	0.5	5.85	85
17	9	-0.90	4.6	0.0	5.62	65
21	5	-4.73	3.3	0.0	5.58	81
16	10	2.19	1.8	6.6	5.26	62
24	2	-17.05	5.7	0.0	5.73	92
26	0	-13.22	4.6	0.5	6.02	100
23	3	-9.50	0.0	8.9	6.15	88
10	16	3.47	0.0	12.0	5.80	38
21	5	-3.05	5.4	0.0	5.71	81
26	0	-6.22	1.8	0.0	5.73	100

The results obtained based on the model showed extremely good agreement with results obtained from the computer simulation. Certainly, some other fuzzy input variables could have been used within the proposed approximate reasoning algorithm. Modifying the proposed variables or introducing new fuzzy input variables is one of the interesting and important directions for future research. It would also be interesting to test the proposed approach when there are a greater number of alternative routes. It can be ascertained that the proposed approach is "model-free," which means it is not necessary to have a mathematical model for the problem considered. The proposed system also has the possibility to learn from examples, which means that it is adaptable. With new examples, from time to time, changes appear in the rules and/or new fuzzy rules are added. The input space is high-dimensional. On the other hand, available data pairs are limited. Therefore, it is often not possible to build a fuzzy model over the whole input space. In such situations the problem arises of estimating output using the existing fuzzy rule base. It should be noted that different approaches have been developed in the literature to resolve this problem (Wang and Mendel, 1992c; Sugeno and Yasukawa, 1993).

Table 2.7. Percentage of drivers choosing route II

Number of drivers along route						
Route I	Route II					
n_I	n_{II}	ΔT	C	S	B	P
9	17	-6.96				65
24	2	7.85	0.0	3.1	-0.44	8
26	0	11.91	2.7	0.0	-4.27	0
26	0	13.37	0.5	2.2	-6.54	0
14	12	-3.30	2.2	0.0	-4.58	46
10	16	-4.65	0.0	3.1	-3.04	62
25	1	17.71	0.0	1.0	-5.13	4
14	12	-3.49	8.5	0.0	-4.05	46
16	10	2.16	2.9	1.8	-3.84	38
22	4	4.17	3.8	4.9	-3.88	15
21	5	3.40	4.7	4.0	-3.83	19
25	1	9.85	2.3	3.4	-4.33	4
23	3	20.58	11.0	0.0	-5.58	12
13	13	0.59	6.9	0.0	-5.23	50
25	1	11.16	1.3	0.0	-5.62	4
26	0	12.92	0.1	1.2	-6.08	0
20	6	10.78	14.0	0.0	-6.36	23
18	8	2.96	5.9	0.0	-6.17	31
7	19	-5.74	1.6	4.1	-5.54	73
24	2	11.53	0.5	9.0	-5.84	8
22	4	6.03	0.5	4.1	-5.85	15
17	9	0.90	0.0	4.6	-5.62	35
21	5	4.73	0.0	3.3	-5.58	19
16	10	-2.19	6.6	1.8	-5.26	38
24	2	17.05	0.0	5.7	-5.73	8
26	0	13.22	0.5	4.6	-6.02	0
23	3	9.50	8.9	0.0	-6.15	12
10	16	-3.47	12.0	0.0	-5.80	62
21	5	3.05	0.0	5.4	-5.71	19
26	0	6.22	0.0	1.8	-5.73	0

Table 2.8. Comparison between real results and the results obtained using the approximate reasoning algorithm

	% of Drivers			% of Drivers	
Ordinal number	Real	Estimated	Ordinal number	Real	Estimated
1.	92	97	31.	0	3
2.	100	92	32.	0	3
3.	100	92	33.	46	57
4.	54	47	34.	61	66
5.	39	42	35.	4	3
6.	96	92	36.	46	51
7.	54	49	37.	38	38
8.	60	60	38.	15	22
9.	85	78	39.	19	27
10.	81	73	40.	4	5
11.	96	96	41.	11	10
12.	89	90	42.	50	43
13.	50	53	43.	4	4
14.	96	91	44.	0	3
15.	100	92	45.	23	16
16.	77	83	46.	31	29
17.	69	70	47.	70	70
18.	27	37	48.	8	4
19.	92	91	49.	19	10
20.	85	90	50.	35	30
21.	65	62	51.	19	15
22.	81	85	52.	39	45
23.	62	52	53.	8	4
24.	92	91	54.	0	4
25.	100	91	55.	11	12
26.	89	88	56.	61	57
27.	39	43	57.	19	28
28.	81	72	58.	0	10
29.	100	90			
30.	8	10			

As already underscored, the results obtained from the model showed extremely good agreement with the results obtained from the computer simulation. This indicates that approaches based on fuzzy logic can be developed in future research on different aspects of ATIS.

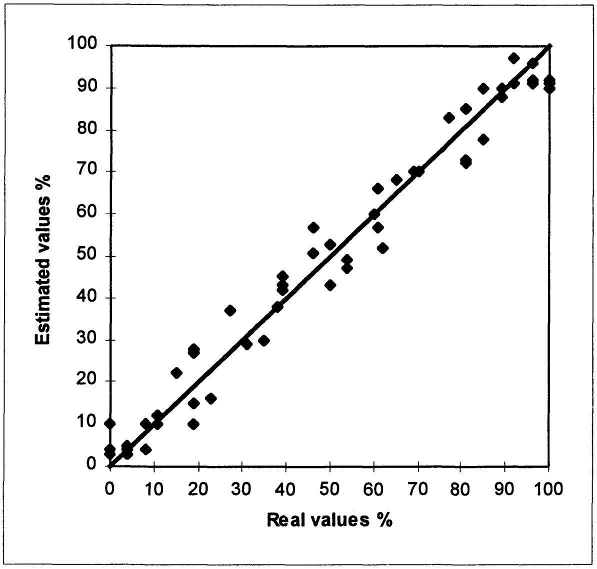

Figure 2.29. Comparison between real results and results obtained using the approximate reasoning algorithm

2.8. APPROXIMATE REASONING ROUTE-CHOICE MODEL ON AN AIRLINE NETWORK

Countless trips are made on a daily basis in intercity transportation, and each trip is linked to a certain number of decisions that must be made. Passengers decide whether or not to make a certain trip, the choice of destination and the modal split, the choice of carrier and route, the choice of tariff, and the class of transportation (first and tourist class in airplanes). During the past four decades, a large number of models have been developed that describe trip generation, trip distribution, modal split, and route choice. Some models are based on the socioeconomic characteristics of a city, region, or zone and/or the characteristics of the transport system. A second group of models starts with the passenger as the trip's decision maker.

How do passengers choose? What are the most important factors that influence their decision making? Do passengers stick to strictly rational reasons when deciding about their trips? Do they change their minds (and how often) in the middle of a trip? Do they always act and choose in the same manner? Research into transportation demand in the past four decades has certainly answered some of these questions (Kanafani, 1983). On the other hand, due to the extremely great impact that transportation demand has on transportation supply and the carrier's economic effects, further research is needed to try to explain the manner in which passengers decide.

Depending on the choice problem being considered, basic input data needed to make decisions include travel time, travel costs, waiting time to be served, frequency, comfort, safety, and so on. In other words, some of the parameters that are essential when choosing among alternatives are characterized by uncertainty. When making certain decisions, passengers do not have absolutely precise data at their disposal regarding some parameters.

On the other hand, some other figures are known with absolute precision. For example, passengers are given the information that there are five flights a day between two cities. Passengers appraise this information in different ways. For some passengers, this is a "relatively small number of flights"; for others it is a "satisfactory" or "medium" number, and so on. In other words, passengers use different subjective appraisals to characterize the values of certain parameters that are known with absolute precision.

The problems of route choice and traffic assignment are certainly among the most important problems in transportation. When transportation flows are known between trip origins and destinations, and when the characteristics of the paths linking them are also known, then the expected passenger flows along these links need to be estimated. Different variations of this problem appear in city, intercity, air, and train transportation. In air transportation, these problems were considered by Kanafani and Chang (1979), Kanafani (1981), Ghobrial (1983), and Kanafani and Ghobrial (1985). The last three papers developed an iterative procedure to assign traffic on an airline network. The route-choice model is a component part of this procedure. The model developed herein belongs to the class of Logit models.

Deciding factors when choosing among routes in an airline network include travel time, the cost of transport on different routes, flight frequency, and the number of stopovers.

Passengers who are flying have very precise information available regarding travel time on specific routes. They also know the price of the trip and the number of stopovers. Passengers are most often aware of flight frequency, too. We can state that the factors influencing route choice in air transportation are not characterized by imprecision. In other words, passengers (most often) have very precise information regarding these factors. Teodorovic and Kalic assumed (1996) that perceived (appraised) travel time, perceived flight frequency, or perceived number of stopovers are most often "fuzzy" amounts. A passenger might appraise a certain travel time as being "very small," "small," "medium," and so on. Maintaining that travel time is "very small" is the result of a subjective feeling or the individual's subjective evaluation. This is neither the result of any measurement nor the realization of a random variable representing travel time. Evaluations of travel time, as well as evaluations of the offered flight frequency or the price of transportation, differ from passenger to passenger, from user to user. As already pointed out, these evaluations are most often fuzzy.

The basic goal of the paper by Teodorovic and Kalic (1996) was to try to develop a route-choice model on an airline network that suitably describes the "fuzziness" in the perceived values of factors that influence route choice. The model was based on the rules of approximate reasoning that is, on the principles of fuzzy logic. It evaluates passenger flows on individual links when information is available regarding passenger flows between origins and destinations, travel time, costs, flight frequency, and number of stopovers on the routes.

The route-choice problem in air transportation is similar to the route-choice problem in city or intercity road transportation. The differences in these problems are primarily found in the parameters that influence route choice. Let us assume that an airplane trip between city r and city s can be made either by taking route A or path B (*Figure 2.30*).

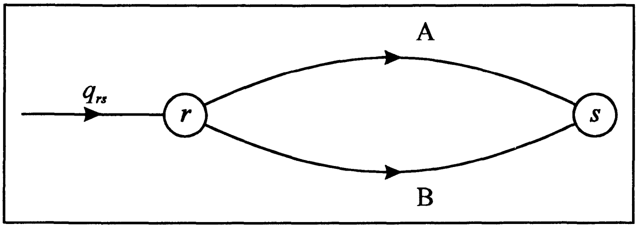

Figure 2.30. Choosing one of two possible routes between origin and destination

The origin is denoted by r and the destination by s. Let us assume that the passenger flow between r and s is known. This flow is denoted by q_{rs}. Possible routes are designated by A and B. Although the number of possible routes between any two airports is considerably larger, passengers most often take only two or three alternate routes into consideration. Teodorovic and Kikuchi (1990) assumed that the passenger makes his route choice based on a comparison of the characteristics of alternate routes; after making this comparison, the passenger chooses his route through the network with a certain probability. The probability that he will choose a specific route can be "greater" or "lesser." In addition, the total number of passengers on any route can be "larger" or "smaller." When a passengers has an absolute preference for a specific route, we will consider the choice probability as being equal to 1. When the strength of a passenger's preference diminishes, the probability of that route being chosen drops along with it. Let us assume that the probability of a passenger choosing any specific route can be expressed by fuzzy sets such as "very great," "great," "medium," "small," "very small," and so on. In other words, the percentage of passengers taking a certain route is "very large," "large," "medium," and so on.

Teodorovic and Kalic (1996) also assumed that when transport costs are relatively uniform, then travel time and flight frequency on alternate routes are the dominant factors that influence a passenger's route choice. The number of stopovers is also an extremely important factor, however, to a certain extent it is already included in travel time. Other factors such as the type of aircraft, changing planes during the trip, and so on. Teodorovic and Kalic (1996) did not include in this analysis, primarily in an attempt to make the proposed model as simple as possible.

We will consider the simplest case when a passenger must choose between two possible routes going from his origin to his destination. We assume that passengers base their route choice on a comparison of travel time and flight frequency on the alternate routes. We will denote, respectively, by δT and δF the differences in travel time and flight frequency between one route and the other. Passengers can perceive these differences as "lesser frequency," "approximately the same frequency," "greater frequency," "negligible difference in travel time," and so on. We will also denote by P the percentage of passengers who use route A. Let us introduce into the discussion the following fuzzy sets:

VBN very big negative difference in travel time
BN big negative difference in travel time
MN medium negative difference in travel time
SN small negative difference in travel time
N negligible difference in travel time
SP small positive difference in travel time
MP medium positive difference in travel time
BP big positive difference in travel time
VBP very big positive difference in travel time
SF smaller flight frequency than on the alternate route
ASF approximately the same flight frequency as on the alternate route
BF bigger frequency than on the alternate route
VVS very very small percentage of passengers using route A
VS very small percentage of passengers using route A
S small percentage of passengers using route A
MS medium-small percentage of passengers using route A
M medium percentage of passengers using route A
MB medium-big percentage of passengers using route A
B big percentage of passengers using route A
VB very big percentage of passengers using route A
VVB very very big percentage of passengers using route A

The membership functions of these fuzzy sets (subjectively defined by Teodorovic and Kalic, 1996) are shown in *Figure 2.31*.

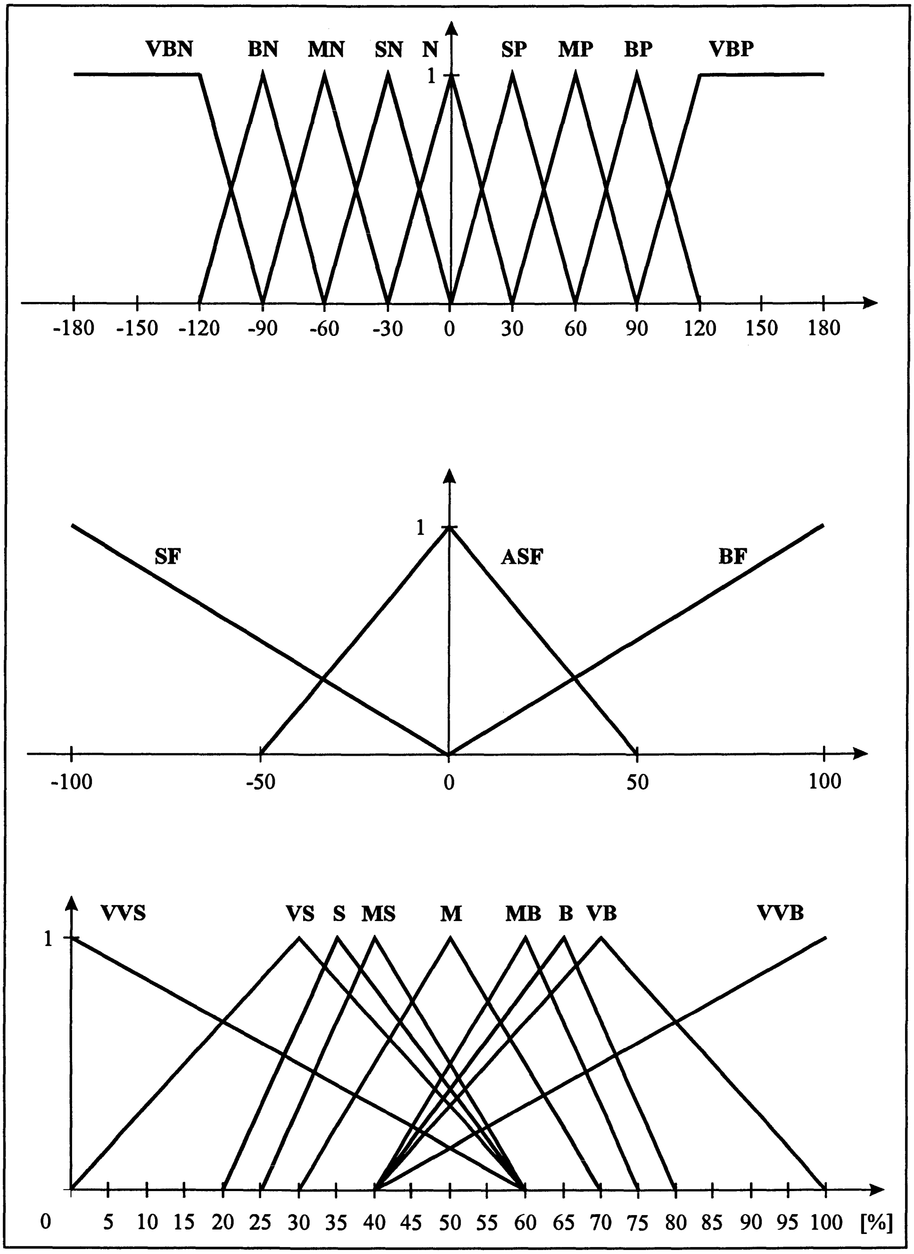

Figure 2.31. Membership functions of the fuzzy sets

In order to determine route-choice probability, Teodorovic and Kalic (1996) developed an approximate reasoning algorithm composed of the following rules:

Rule 1: If δT = **VBN** and δF = **ANY**, then P = **VVB**,
or
Rule 2: If δT = **BN** and δF = **ANY**, then P = **VB**,
or
...
Rule 15: If δT = **VBP** and δF = **ANY**, then P = **VVS**.

For known travel time and flight frequency along alternate routes, the approximate reasoning rules can be used to determine the route-choice probability for each of the routes under consideration.

The approximate reasoning algorithm developed by Teodorovic and Kalic (1996) was tested using real data taken from the doctoral dissertation of Ghobrial (1983).

Table 2.9 presents data on traffic in both directions between thirteen pairs of cities in the Atlanta, Georgia, region. These thirteen cities were chosen due to the fact that they were served by the same type of aircraft. The *Table 2.9* presents flight frequency, travel time, differences in flight frequency, differences in travel time, and percentage of the total number of passengers that chose the first route under consideration. Based on these figures, membership functions were subjectively defined of the fuzzy sets used in the approximate reasoning algorithm.

Table 2.9. Traffic characteristics of the network

Pair of cities	Travel time (min) Route 1	Travel time (min) Route 2	δT	Weekly frequency Route 1	Weekly frequency Route 2	δF	Percentage of passengers using Route 1
1	180	40	140	52	28	24	19
2	40	180	-140	28	52	-24	81
3	170	40	130	48	7	41	19
4	40	170	-130	7	48	-41	81
5	180	60	120	56	13	43	25
6	60	180	-120	13	56	-43	75
7	150	67	83	14	14	0	31
8	67	150	-83	14	14	0	69
9	200	110	90	27	28	-1	40
10	110	200	-90	28	27	1	60
11	210	60	150	90	7	83	22
12	60	210	-150	7	90	-83	78
13	170	110	60	48	14	34	39

Table 2.9. Traffic characteristics of the network (continued)

Pair of cities	Travel time (min)			Weekly frequency			Percentage of passengers using Route 1
	Route 1	Route 2	δT	Route 1	Route 2	δF	
14	110	170	-60	14	48	-34	61
15	135	85	50	14	14	0	39
16	85	135	-50	14	14	0	61
17	90	45	45	14	36	-22	33
18	45	90	-45	36	14	22	67
19	120	90	30	31	21	10	43
20	90	120	-30	21	31	-10	57
21	200	105	95	84	14	70	34
22	105	200	-95	14	84	-70	66
23	180	100	80	84	7	77	36
24	100	180	-80	7	84	-77	64
25	250	160	90	28	14	14	39
26	160	250	-90	14	28	-14	61

The percentages of passengers on individual routes were calculated based on the approximate reasoning algorithm. *Table 2.10* and *Figure 2.32* present a comparison of the real values and the values obtained using the approximate reasoning algorithm.

Table 2.10. Comparison of real values and values obtained using the approximate reasoning algorithm

Pair of cities	Percentage of passengers using route 1	
	Real value	Value obtained using the approximate reasoning algorithm
1	19	19.7
2	81	80.3
3	19	19.7
4	81	80.3
5	25	19.7
6	75	80.3
7	31	30.1
8	69	69.9
9	40	30
10	60	70
11	22	19.7
12	78	80.3
13	39	38.3
14	61	61.7
15	39	38.9
16	61	61.1
17	33	39.4
18	67	60.6
19	43	43

Table 2.10. Comparison of real values and values obtained using the approximate reasoning algorithm (continued)

Pair of cities	Percentage of passengers using route 1	
	Real value	Value obtained using the approximate reasoning algorithm
20	57	57
21	34	29.5
22	66	70.5
23	36	30.2
24	64	69.8
25	39	30
26	61	70

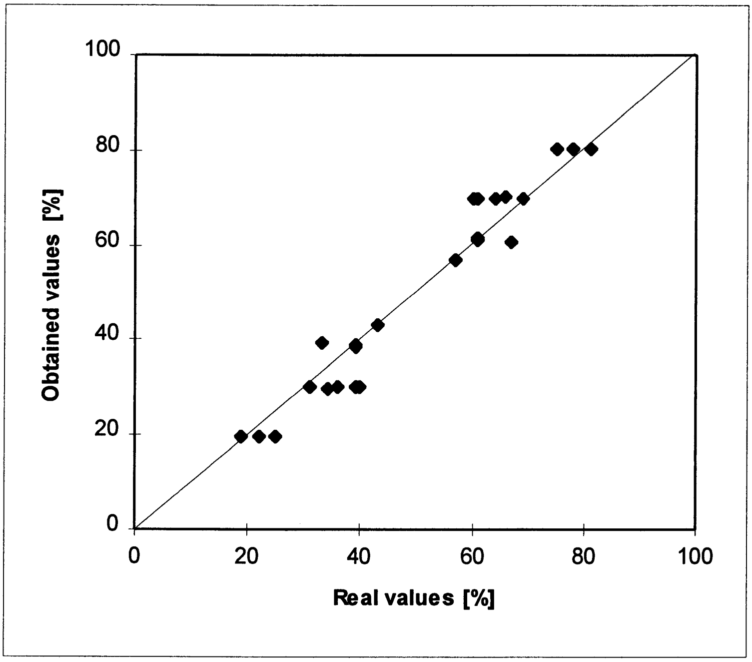

Figure 2.32. Comparison of real values and values obtained using the approximate reasoning algorithm

As can be seen from *Table 2.10* and *Figure 2.32*, extremely good agreement was obtained between the values calculated using the approximate reasoning algorithm and real values. Such good agreement was achieved based on an iterative procedure composed of several steps. First, the membership functions of the fuzzy sets were subjectively defined after an analysis of the data on traffic characteristics. In the next step, using the approximate reasoning algorithm, the percentages of passengers on individual routes were calculated and compared with the real values. After the comparison, some membership functions and/or rules were modified, and the values were calculated once again. Then another comparison was made followed by another modification of some membership functions and/or rules, and so on. Results were obtained practically momentarily (computer time is less than one second). This fact considerably accelerated the iterative procedure and enabled greater experimentation with different membership functions and/or rules in a relatively short time.

2.9. A FUZZY LOGIC APPROACH TO THE VEHICLE ROUTING PROBLEM WHEN DEMAND AT NODES IS UNCERTAIN

Hundreds of papers in world literature have been devoted to the vehicle routing problem. Most of them assume that the quantities to be picked up (or dropped off) at the nodes are deterministic. In recent years, papers have started to appear (Dror and Trudeau, 1986; Dror et al., 1989; Teodorovic and Pavkovic, 1992) in which demand at the nodes is treated as a random variable. The problem of routing vehicles when stochastic demand at the nodes is known is referred to in the literature as the *stochastic vehicle routing problem.* The basic input data to solve such a problem are the probability density functions of the random variables representing demand at the nodes.

In order to verify the probability density functions of demand at the nodes, demand must be "recorded" over a longer period of time, and a detailed statistical analysis must be made of the collected data. On the other hand, we often do not have precise data available regarding demand at some nodes (start-up operations of a distributing system are just starting up, the records of demand at some nodes that are not up-to-date, and so on). In other words, the information about vehicle demand at some nodes is often not precise enough. For example, based on experience, it can be concluded that demand at a node is "around 100 units," "between 80 and 120 units," "approximately 200 units," and so on. Thus, there is often uncertainty regarding the amount of demand at some nodes. Teodorovic and Pavkovic (1996) developed a model to design vehicle routing when there is uncertain demand at the nodes. The model is based on the heuristic "sweeping" algorithm, rules of fuzzy arithmetic, and fuzzy logic.

2.9.1. Problem assumptions

Let us assume that there are n nodes in the network to be served. Vehicles set out from depot D, serve a number of nodes, and on completion of their service, return to the depot. We also assume that the demand at each node is only approximately known. Such demand can be represented by a triangular fuzzy number. Triangular fuzzy number $\mathbf{D} = (d_1, d_2, d_3)$ is described by its left boundary d_1 and its right boundary d_3. Thus, the dispatcher or analyst studying the problem can subjectively estimate, based on his experience and intuition and/or available data, that demand at the node will not be less than d_1 or greater than d_3. The value of d_2 corresponding to a grade of membership of 1 can also be determined by a subjective estimate.

The problem that Teodorovic and Pavkovic discussed in their paper can be defined as follows: For known locations of the depot and nodes to be served, known vehicle capacity, and demand at the nodes that is only approximate (represented by triangular fuzzy numbers), design a set of vehicle routes that minimizes costs.

2.9.2. Proposed solution to the problem

In the past three decades, a large number of different heuristic algorithms have been developed to route vehicles. One of the simplest is the sweeping algorithm proposed by Gillett and Miller (1974). This algorithm is applied to polar coordinates, and the depot is considered to be the origin of the coordinate system. Then the depot is joined with an arbitrarily chosen point that is called the *seed point.* All other points are joined to the depot and then aligned by increasing angles that are formed by the segments that connect the points to the depot and the segment that connects the depot to the seed point. The route starts with the seed point, and then the points aligned by increasing angles are included, respecting given constraints all the while. When a point cannot be included in the route since this would violate a certain constraint, this point becomes the seed point of a new route, and so on. The process is completed when all points are included in a route (*Figure 2.33*).

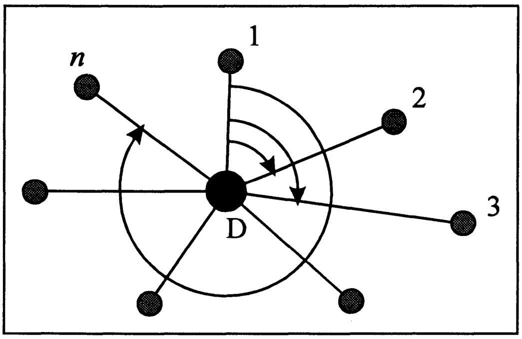

Figure 2.33. Sweeping algorithm

Increasing the number of nodes served along a route decreases the available (remaining) capacity of the vehicle. When demand at the nodes is deterministic, after completing service at one node it is easy to calculate whether the vehicle is able to serve the next node. On the other hand, when demand at the nodes is characterized by triangular fuzzy numbers, it is not a simple task to decide whether the vehicle should serve the next node or return to the depot. It is clear that the greater the vehicle's remaining capacity and the lesser the demand at the next node, the greater the vehicle's "chances" of being able to serve the next node. In other words, the greater the difference between the remaining capacity and demand at the next node, the greater our preference to send the vehicle to serve that next node.

We assume that service is provided by vehicles of the same size. We will denote vehicle capacity by C and the fuzzy number representing demand at the ith node by $\mathbf{D}_i$. After serving the first k nodes, the available capacity of vehicle $\mathbf{B}_k$ will equal

$$\mathbf{B}_k = C - \sum_{i=1}^{k} \mathbf{D}_i \tag{2.71}$$

Demand at the nodes $\mathbf{D}_i$ ($i = 1, 2, ..., k$) is represented by triangular fuzzy numbers. The capacity of vehicle C can also be presented as a triangular fuzzy number $\mathbf{C} = (C, C, C)$. Using the rules of fuzzy arithmetic, we can easily show that $\mathbf{B}_k$ is also a triangular fuzzy number that is,

$$\mathbf{B}_k = \left(C - \sum_{i=1}^{k} d_{3i},\ C - \sum_{i=1}^{k} d_{2i},\ C - \sum_{i=1}^{k} d_{1i} \right) \tag{2.72}$$

where d_{1i} is left boundary of fuzzy number $\mathbf{D}_i$, d_{2i} is value of fuzzy number $\mathbf{D}_i$ corresponding to a grade of membership of 1, and d_{3i} is right boundary of fuzzy number $\mathbf{D}_i$.

It is clear that the "strength" of our preference for the vehicle to serve the next node after it has served k nodes depends on available capacity $\mathbf{B}_k$. Our preference can be, for example, "low," "medium," or "high." We will denote by p_k the preference index that expresses the strength of our preference to send the vehicle to the next node after it has served k nodes. Let the preference index be between 0 and 1 that is,

$$0 \le p_k \le 1 \qquad k = 1, 2, \ldots, n \tag{2.73}$$

When $p_k = 1$, we are absolutely certain that we want the vehicle to serve the next node. When $p_k = 0$, we are completely sure that the vehicle should return to the depot.

The linguistic expressions "low preference," "medium preference," and "high preference" can be represented by corresponding fuzzy sets. *Figure 2.34* presents the membership functions of fuzzy sets "low," "medium," and "high" preference.

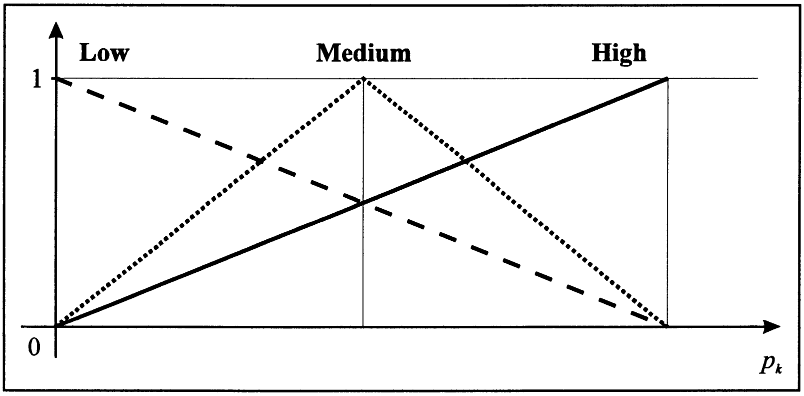

Figure 2.34. Fuzzy sets describing preference strength

Available capacity can also be subjectively estimated as "small," "medium," and "large." We denote respectively by **S**, **M**, and **L** the fuzzy sets denoting small, medium, and large available vehicle capacity. The membership functions of these fuzzy sets are shown in *Figure 2.35*.

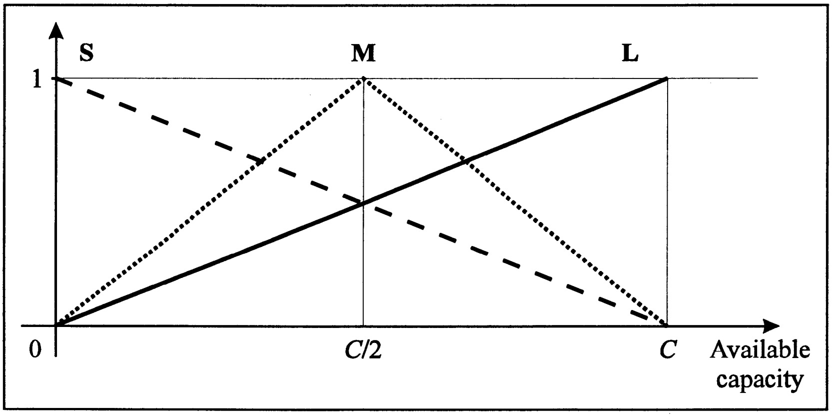

Figure 2.35. Fuzzy sets describing available capacity

Two approximate reasoning algorithms were developed by Teodorovic and Pavkovic (1996) to determine preference strength regarding sending a vehicle to the next node. In the first, simpler approximate reasoning algorithm, it is assumed that the strength of our preference depends exclusively on available capacity. This approximate reasoning algorithm reads

If $\mathbf{B}_k$ = **S**, then $\mathbf{P}_k$ = **Low**,
or
If $\mathbf{B}_k$ = **M**, then $\mathbf{P}_k$ = **Medium**,
or
If $\mathbf{B}_k$ = **L**, then $\mathbf{P}_k$ = **High**.

For known available capacity $\mathbf{B}_k$ that remains after serving k nodes, it is possible to use the approximate reasoning rules to determine the strength of our preference to send the vehicle to the next node. We are now able to approximately answer the following question: Should we send the vehicle to the next node or return it to the depot after completing service to k nodes? In other words, we can now use known available capacity $\mathbf{B}_k$ to calculate the strength of our preference p_k^* to send the vehicle to the next node. This approximate reasoning procedure is graphically shown in *Figure 2.36*.

For known capacity $\mathbf{B}_k$, the first step is to establish how much this available data satisfies the premises of the rules. In other words, first we must determine whether available capacity is “small,” “medium,” or “large” (*Figure 2.36*). *Figure 2.36* presents the membership function of the index preference obtained by applying the approximate reasoning rules. In the Teodorovic and Pavkovic's paper the preference index chosen was the center of gravity.

Let the chosen preference index equal p_k^*. Based on this value, a decision must be made whether to send the vehicle to the next node or return it to the depot. Teodorovic and Pavkovic (1996) made the decision as follows. The vehicle should be sent to the next node if the following relation is fulfilled:

$$p_k^* \geq p^{**} \tag{2.74}$$

where p^{**} is a parameter subjectively given from the interval [0,1]. When

$$p_k^* < p^{**} \tag{2.75}$$

the vehicle should be returned to the depot. Lower values of parameter p^{**} express our endeavor to use the vehicle capacity as much as possible. On the other hand, when parameter p^{**} is low, there is a rise in the number of cases in which the vehicle arrives at the next node and is not able to carry out planned service due to small available capacity.

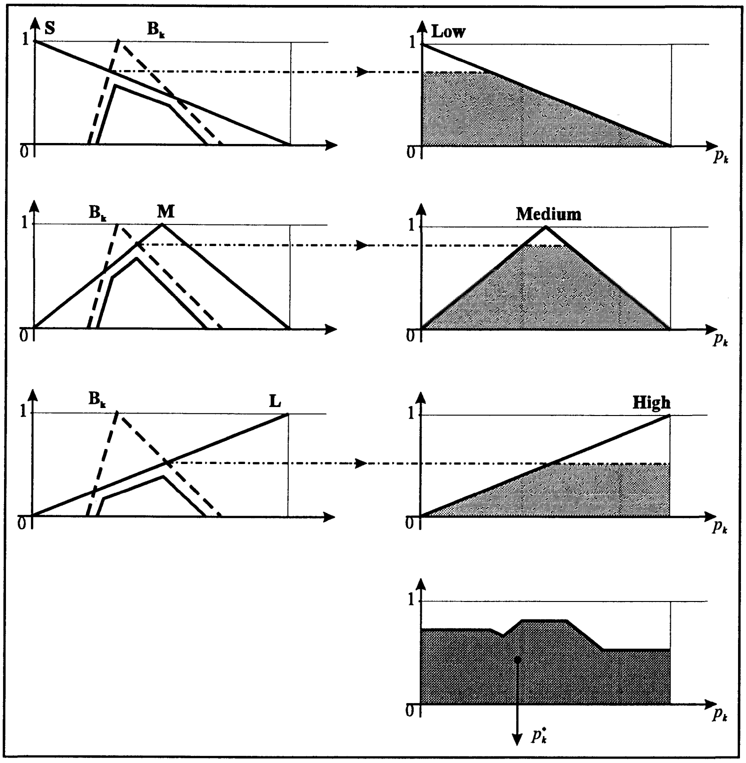

Figure 2.36. Determining preference strength p_k* for known available capacity $\mathbf{B}_k$

The algorithm developed by Teodorovic and Pavkovic (1996) to route vehicles when demand at the nodes is uncertain consists of the following algorithmic steps:

Step 1: Express demand at the nodes in the form of triangular fuzzy numbers. Assign a value to parameter p^{**}.

Step 2: If all nodes have been assigned to routes, end the algorithm. Otherwise, go to Step 3.

Step 3: Choose a seed point. Link this seed point to the depot. Link all nodes to the depot that are not included in a route.

Step 4: First include the seed point in a route. Then include nodes in the route by increasing order of the angles that are made by the lines that link the nodes with the depot and the line that links the seed point to the depot. Before deciding to include a node into the route, first use the approximate reasoning algorithm to calculate the preference index. If the calculated preference index is greater than or equal to the previously assigned value of p^{**}, include the node in the route. Otherwise, this node becomes a new seed point. Return to Step 2.

2.9.2.1. Determining route length

The developed algorithm was tested on a large number of different numerical examples. The location of the nodes requiring service and of the depot were generated in a random manner. Demand at the nodes was also arbitrarily determined. After concluding the computer experiments, the first question to be answered was the efficiency of the proposed algorithm. CPU times are shown in *Table 2.11.*

Table 2.11. CPU times

Number of nodes	CPU time (sec)	Number of nodes	CPU time (sec)
50	1	550	15
100	2	600	16
150	3	650	18
200	5	700	19
250	6	750	21
300	7	800	23
350	9	850	24
400	10	900	26
450	12	950	28
500	13	1000	30

All computer experiments were done on a PC computer (386 processor). Very small CPU times were achieved. Bearing in mind the fact that the problem did not have to be solved in "real time," the achieved CPU times were absolutely acceptable.

The real value of demand at a node is only known when the vehicle reaches the node. On the other hand, the vehicle routes are designed in advance by applying the proposed algorithm. Due to the uncertainty of demand at the nodes, a vehicle might not be able to service a node once it arrives there due to insufficient capacity. Teodorovic and Pavkovic (1996) assumed in such situations that the vehicle returns to the depot, empties what it has picked up thus far, returns to the node where it had a "failure," and continues service along the rest of the planned route (*Figure 2.37*).

When evaluating the planned route, the additional distance that the vehicle makes due to "failure" arising in some nodes along the route must be taken into consideration. Parameter p^{**}, which is subjectively determined, has an extremely great impact on both the total length of the planned routes and on the additional distance covered by vehicles due to "failures" at some nodes. As already mentioned, lower values of parameter p^{**} express our desire to use vehicle capacity the best we can. These values correspond to routes with shorter total distances. On the other hand, lower values of parameter p^{**} increase the number of cases in which vehicles arrive at a node and are unable to service it, thereby increasing the total distance they cover due to the "failure." Higher values of parameter p^{**} are characterized by less utilization of vehicle capacity along the planned routes and less additional distance to cover due to failures. The problem logically arises of determining the value of parameter p^{**}, which will result in the least total sum of planned route lengths and additional distance covered by vehicles due to failure.

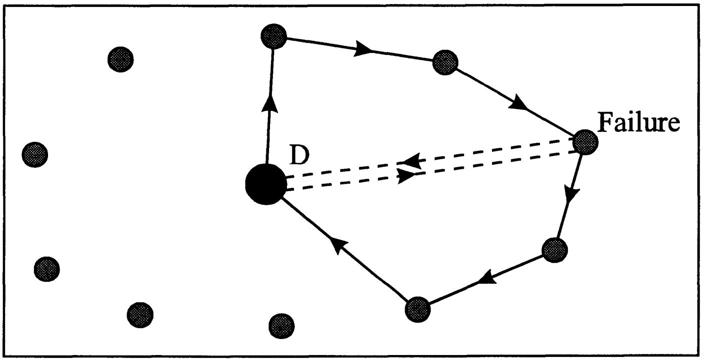

Figure 2.37. "Failure" at a node of the planned route

Teodorovic and Pavkovic (1996) decided to use simulation to reach the value of parameter p^{**} corresponding to the least total distance. As noted by Kaufmann and Gupta (1985), "In the theory of fuzzy subsets the law of possibility plays a role similar to that played by the law of probability in measurement theory for ordinary sets."

If

$$\forall x \in \mathrm{R}: \quad h(x) \in [0,1] \quad \text{and} \quad \bigvee_{x \in \mathrm{R}} h(x) = 1 \tag{2.76}$$

then $h(x)$ is called the possibility law on R. If **A** is a fuzzy subset of R, then the possibility of **A** for the law $h(x)$ is defined as

$$\mathrm{poss}_{\mathrm{H}}\mathbf{A} = \bigvee_{x \in \mathrm{R}} \left(\mu_{\mathrm{A}}(x) \wedge h(x)\right) \tag{2.77}$$

where $\mu_A(x)$ is membership function of fuzzy set **A**, $\vee$ is maximum symbol, and $\wedge$ is minimum symbol. *Figure 2.38* presents the possibility of **A** (left) and the possibility of **B** (right) for the law $h(x)$.

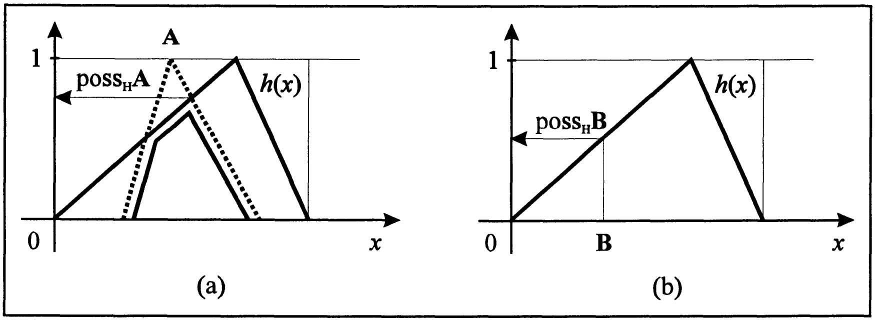

Figure 2.38. The possibility of **A** and the possibility of **B** for the law $h(x)$

The case shown in *Figure 2.38(b)* is particularly interesting for the problem considered. Let possibility $h(x)$ refer to demand at a specific node. The heavy line in *Figure 2.38(b)* denotes the possibility that demand in the node is equal to B. It was noted earlier that node demand is considered to be a triangular number. In other words, all possibility laws that refer to node demand have a triangular form. The node demand was simulated as follows. First, random number B (*Figure 2.38(b)*) was generated in the interval between the left and right boundaries of the triangular fuzzy number representing demand at the node. After generating random number B, $\text{poss}_H\mathbf{B}$ was calculated which is the possibility that demand at the node equals B. Then random number r was generated from the interval [0,1]. When

$$r \leq \text{poss}_H \mathbf{B} \tag{2.78}$$

demand at the node is adopted as being equal to B.

In the opposite case, when $r > \text{poss}_H\mathbf{B}$, it is not accepted that demand at the node equals B. In this case, random numbers B and r are generated again and again until random numbers B and r are found that satisfy relation (2.78).

Demand for every node in the network is generated in the above manner. In other words, demand at each node is a deterministic amount that is obtained by simulation. Moving along the route designed by the approximate reasoning algorithm and accumulating the amounts picked up at each node, it was easy to determine the nodes where failures occurred and to calculate the additional distance that the vehicles had to make. Five hundred simulation experiments were made for the considered set of routes, resulting in 500 additional distances that the vehicles made due to failures. In other words, the total additional distance covered by the vehicles due to failures was treated as a random variable with a corresponding empirical frequency distribution.

The values of parameter p^{**} varied within the interval of 0 to 1 with a step of 0.05. As parameter p^{**} rose, a general rising tendency was noted in the total length of planned routes with a decrease in the unplanned distances that vehicles had to make due to failures at the nodes. *Table 2.12* and *Figure 2.39* show the lengths of the planned routes, additional distances covered due to failures at the nodes, and the total expected distance that the vehicles were to cover (the total distance is the sum of the total lengths of the planned routes and the additional distances that are covered due to failures at the nodes). *Table 2.12* and *Figure 2.39* refer to a hypothetical example with 100 nodes.

Table 2.12. Total length of planned routes, additional distance covered due to failures, and total expected distance to be covered by vehicles

p^{**}	Total planned length of all routes	Additional distance covered due to failures	Total expected length of all routes
0.00	1762.63	546.41	2309.04
0.05	1785.10	589.08	2374.18
0.10	1785.10	571.23	2356.33
0.15	1785.10	571.23	2356.33
0.20	1785.10	589.08	2374.18
0.25	1785.10	571.23	2356.33
0.30	1785.10	571.23	2356.33
0.35	1785.10	571.23	2356.33
0.40	1785.10	571.23	2356.33
0.45	2084.30	0.00	2084.30
0.50	2381.88	0.00	2381.88
0.55	2828.11	0.00	2828.11
0.60	3793.04	0.00	3793.04
0.65	4823.94	0.00	4823.94
0.70	7878.01	0.00	7878.01
0.75	7878.01	0.00	7878.01
0.80	7878.01	0.00	7878.01
0.85	7878.01	0.00	7878.01
0.90	7878.01	0.00	7878.01
0.95	7878.01	0.00	7878.01
1.00	7878.01	0.00	7878.01
0.42	1948.54	487.14	2435.68
0.44	2292.62	137.56	2430.17
0.45	2084.30	0.00	2084.30
0.46	2389.46	0.00	2389.46
0.48	2363.08	0.00	2363.46

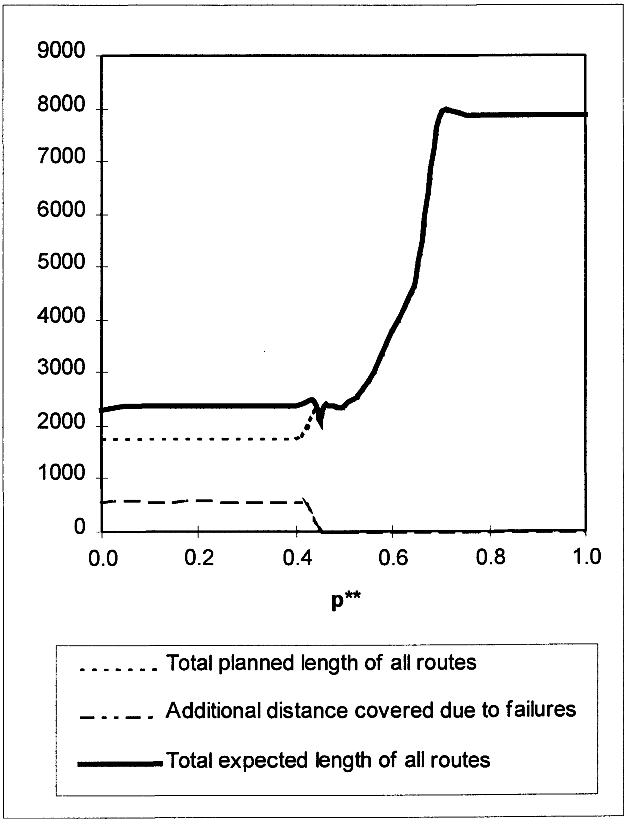

Figure 2.39. Lengths of planned routes, additional distances covered due to failures, and total expected distances to be covered by vehicles

When $p^{**} = 0$, only one big route is planned. When $p^{**} = 1$, the number of routes equals the number of nodes that is, each route consists of only one node. We note from *Table 2.12* and *Figure 2.39* that the least total expected distance to be covered by the vehicles is realized when $0.40 < p^{**} < 0.50$. A more "refined" search in this interval (lower part of *Table 2.12*) indicates that the value $p^{**} = 0.45$ corresponds to the least total distance covered by the vehicles.

The algorithms enabled a quick determination of the set of vehicle routes, even for larger problems. The value of parameter p^{**} had to be determined before using the algorithms. The choice of this parameter greatly influences the length of the planned routes and the additional distances covered by vehicles due to failures at the nodes. The "best" value of parameter p^{**} was found by simulation. The "best" value of parameter p^{**} was conditioned by the spatial distribution of the nodes, the shape of the triangular fuzzy numbers representing demand at the nodes, and vehicle capacity.

2.10. AIR TRAFFIC FLOW MANAGEMENT MODEL BASED ON FUZZY LOGIC

In recent years there has been considerable congestion in some elements of the air traffic control system. Congestion has appeared in some airports, air routes, and air traffic control sectors. Congestion is due to the fact that in certain time periods "demand" at an airport or on an air route is greater than capacity. This situation results in delayed flights, greater airline and airport operating costs, and a decrease in the quality of air traffic service. The problem of air traffic flow management in congested air traffic control conditions (the *flow management problem*) was considered by Odoni (1987), Andreatta and Romanin-Jacur (1987), Terrab (1990) and Teodorovic and Babic (1993).

The most frequent congestion occurs at airports. Meteorological conditions (such as visibility and wind velocity) have a direct influence on airport capacity. Since meteorological conditions cannot be precisely forecast over a longer period of time, the capacity of an airport can be only approximately determined over the long run. In the same vein, some airports frequently have time periods in which "demand" is greater than the airport's capacity. In such conditions, the air traffic control service might not allow a certain number of aircraft to take off at their planned departure times. In this way, aircraft delay (caused by congestion) takes place on the ground instead of in the air. Airport congestion can appear during one time period (one hour) or over a greater number of time intervals. Air traffic flow management consists of undertaking actions to minimize the negative consequences of congestion. In other words, there is a need to determine which aircraft will not be served during a considered time period and to set their new departure times.

Odoni (1987) considered different aspects of the problem of air traffic flow management; he gave a mathematical formulation of the problem and indicated further research orientation. Studying this problem over one time period, Andreatta and Romanin-Jacur (1987) developed a model based on dynamic programming. Terrab (1990) considered the problem of managing air traffic flows in his doctoral dissertation. He treated airport capacity as a deterministic value where the problem of air traffic flow management was reduced to the problem of determining the assignment of flows through a network with a restricted capacity corresponding to the least costs. Terrab also treated airport capacity as a random variable and developed a model based on dynamic programming for this case. The problem of managing air traffic flows was considered for several time periods in which there was airport congestion. Hormann (1987) developed a simulation model that was able to assess different solutions to the problem of managing air traffic flows. Teodorovic and Babic (1993) also developed a model to manage air traffic flows based on fuzzy logic.

2.10.1. Approximate reasoning model to manage air traffic flows when there is airport congestion

In the Teodorovic and Babic (1993) air traffic flow management model based on fuzzy logic the authors considered a starlike network with several departure airports and one landing airport. This starlike network is shown in *Figure 2.40.*

Let us assume that the capacity of the landing airport and landing "demand" vary during the considered time period (0,T). A time period is usually several hours long or one day. Landing "demand" and landing airport capacity during the considered time period are shown in *Figure 2.41.*

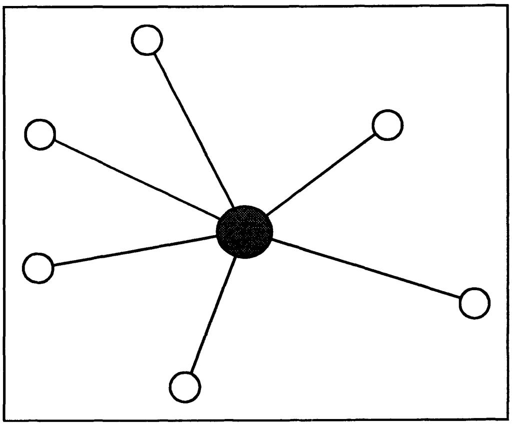

Figure 2.40. Starlike network with several departure airports and one landing airport

Let us divide the considered time period into P smaller time intervals. We will denote these smaller time periods by T_1, T_2, ..., T_p, respectively. Let us assume that landing airport capacity is a deterministic variable whose values in individual time intervals are K_1, K_2, ..., K_p. We also assume that these values are known in advance. Let there be a total of N aircraft requesting to land at the landing airport during the considered time period (0,T). Each of the aircraft A_i, $i = 1, 2, ..., N$ requests to land at a certain time interval. We denote by P_i ($i = 1, 2, ..., P$) the requested landing interval of aircraft A_i. We note from *Figure 2.41* that during some time intervals landing "demand" is greater than the capacity. We therefore conclude that during these time intervals it is not possible to serve all aircraft that is, some of these aircraft will have to be served in one of the subsequent time intervals. Let us assume that aircraft A_i will most certainly be served in time interval T_{p+1} (whose capacity $K_{p+1} = \infty$) if it could not be served in any of the time intervals $T_1, T_2, ..., T_p$. When deciding which aircraft will be served outside of the requested time interval, care must be taken that the total negative effects arising from aircraft delay are as small as possible. Let us assume that the decision maker considers aircraft "size" and aircraft "delay" when deciding which aircraft to serve within a requested time interval. Aircraft size is denoted by the number of passengers in the aircraft, while delay consists of the time difference between the allotted and the requested aircraft landing.

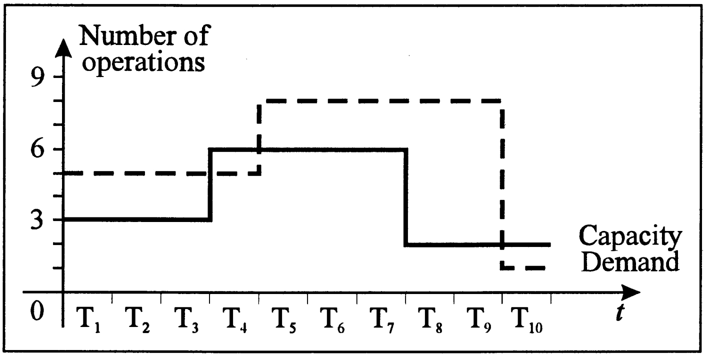

Figure 2.41. Landing "demand" and landing airport capacity during time period (0,T)

The air traffic flow management problem for insufficient capacity at the landing airport is similar to the problem of disturbances in carrying out a planned airline schedule which is periodically encountered by airline carriers. When one or more aircraft are canceled for technical reasons, air carriers must carry out the planned airline schedule with a smaller number of aircraft. Such situations bring about cancellations and/or delays of some flights. The dispatcher in charge of traffic decides which flights will be canceled and which flights will have landing delays. When determining new landing times, the dispatcher might try to "distribute" the total costs caused by the delay approximately evenly over certain flights. This tactic in removing the disturbance treats all aircraft equally but causes a large number of delays. Another possibility is the tactic of the dispatcher "sacrificing" a smaller number of aircraft and giving them the greatest share of the cost of delay. This results in the cancellation and/or great delay of a small number of flights with the simultaneous departure of a large number of aircraft on time. Decisions can be similarly made regarding aircraft to be served outside of their requested time intervals when airport capacity is smaller than demand.

Let us note aircraft A_i which requests landing in time interval P_i. This aircraft will heretofore be called "original" and will be denoted by A_{i0}. This aircraft ("original") A_{i0} can be served during one of the time intervals P_i, P_{i+1}, ..., P_{i+m}. Let us introduce k_i = P - P_i, "copies" of aircraft A_{i0}. These copies are denoted respectively as A_{i1}, A_{i2}, ..., A_{im}, ..., A_{ik_i}. Copy A_{im} denotes the situation in which aircraft A_{im} is served with *m* time intervals of delay (the aircraft is served at time interval P_{i+m}). Let us make a "copy" of all "originals." Thus, in each of the time intervals, in addition to the "originals" requesting service in the considered time interval, there appear "copies" of all aircraft that requested service in previous time intervals. These copies "compete" among themselves and with the "originals" for "survival" in the considered time interval. Let us assume that the decision maker has certain preferences toward servicing certain "originals" or "copies". These preferences must be dependent on aircraft size and delay (either of the originals or of the copies). The "copies" have the same aircraft size as the "original" aircraft. The copies differ from their originals by a certain amount of delay. For example, while the delay of original A_{i0} is zero, the delay of copy A_{im} is *m* time intervals. The decision maker has a subjecting feeling regarding aircraft size and delay times. The decision maker feels that some aircraft are "big," "medium," or "little." The decision maker might also estimate delay time to be "big," "medium," or "little." In other words, aircraft size and delay can be treated as fuzzy variables. The categories "big aircraft," "medium aircraft," or "little aircraft," and the categories "big delay," "little delay," or "medium delay" can be represented as fuzzy sets. The decision maker assigns aircraft to certain time intervals, bearing in mind the landing airport capacity at certain time intervals, and he subjectively estimates the aircraft size and time of delay. When distributing aircraft by intervals, the decision maker operates with certain rules. For example, the decision maker can have a "very strong" preference to serve an aircraft in the requested time interval that is "big" and has "little" delay. Decision maker preference can be "very weak" in the case of a "little" aircraft with "big" delay (the "sacrifice" tactic is used).

Let us denote by **S** and **T**, respectively, aircraft size and delay. The categories of "big," "medium," and "little" aircraft can be represented by corresponding fuzzy sets. The membership functions of these sets are shown in *Figure 2.42*.

The categories of "big," "medium," and "little" aircraft are shown by fuzzy sets $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{A}_3$. "Big," "medium," and "little" delays are shown in the same manner by fuzzy sets $\mathbf{D}_1$, $\mathbf{D}_2$, and $\mathbf{D}_3$. The membership functions of these sets are shown in *Figure 2.43*.

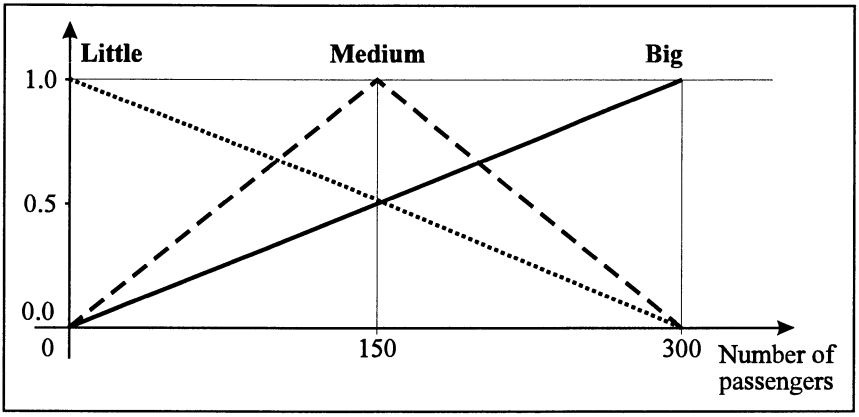

Figure 2.42. Membership functions of fuzzy sets: $\mathbf{A}_1$ is "big" aircraft, $\mathbf{A}_2$ is "medium" aircraft, and $\mathbf{A}_3$ is "little" aircraft

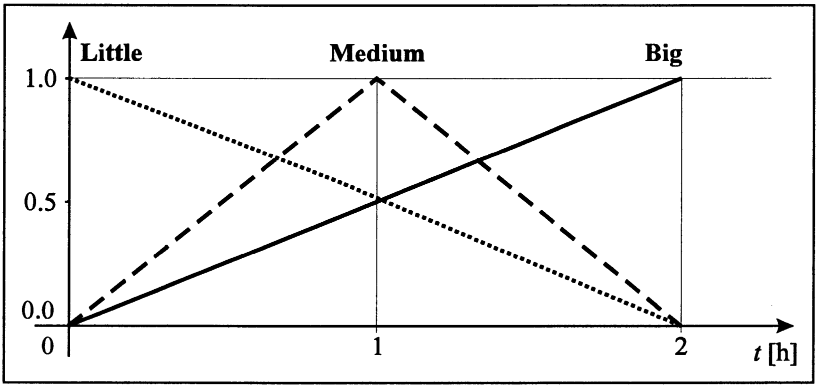

Figure 2.43. Membership functions of fuzzy sets: $\mathbf{D}_1$ is "big" delay, $\mathbf{D}_2$ is "medium" delay, and $\mathbf{D}_3$ is "little" delay

The strength of the decision maker's preference can be expressed by preference index **PI**, which is between 0 and 1. When preference equals 1, the decision maker has no dilemma in wanting to serve the considered aircraft at the requested time interval. As the preference index decreases, so does the "strength" of the decision maker's desire to serve the considered aircraft. We assume that the decision maker's preference can be "very strong," "strong," "medium," "weak," and "very weak." These categories are represented by fuzzy sets $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, $\mathbf{P}_4$, and $\mathbf{P}_5$, whose membership functions are shown in *Figure 2.44*.

The approximate reasoning algorithm to establish preference strength regarding aircraft to be served, developed by Teodorovic and Babic (1993), consists of nine rules, some of which read as follows:

Rule 1: If the aircraft is **Little** and the delay is **Little**, then preference is **Medium**,

or

Rule 2: If the aircraft is **Small** and the delay is **Medium**, then preference is **Weak**,

or

...

Rule 9: If the aircraft is **Big** and the delay is **Small**, then preference is **Very strong**.

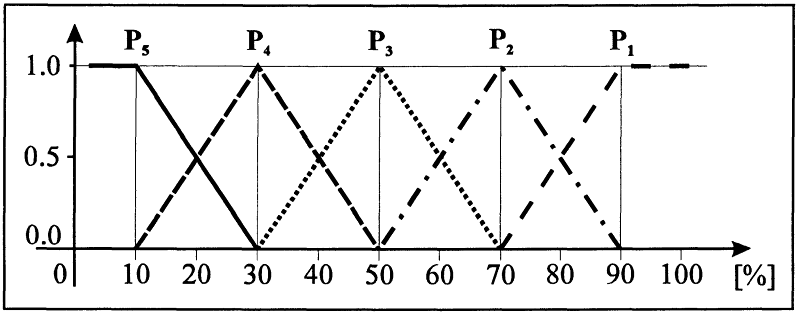

Figure 2.44. Membership functions of fuzzy sets $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, $\mathbf{P}_4$, and $\mathbf{P}_5$

The presented approximate reasoning by max-min composition enables the establishment of the preference index value, or the strength of the decision maker's preference regarding service to a specific aircraft in the considered time interval.

Let us denote by p_{i0p} the decision maker's preference index regarding service to original A_{i0} in time interval $p = P_i$. We denote by p_{imp} the decision maker's preference index regarding service to aircraft A_{i0} m time intervals later. In other words, p_{imp} represents the preference index of copy A_{im} that is served in time interval $p = P_i + m$.

Teodorovic and Babic (1993) developed an algorithm that determines the aircraft that should be served in time intervals T_p, p = 1, 2, ..., P. The algorithm also determines the new time intervals in which to serve the aircraft that could not be served in their requested time intervals. The algorithm is composed of the following steps:

Step 1: Make k_i "copies" of every "original A_{i0}.

Step 2: Using approximate reasoning algorithm calculate the preference index for each "original" A_{i0} and "copy" A_{im} (i = 1, 2, ..., N; m =

1, 2, ..., k_i). Sort the preference indexes by descending order and put them in list form. Denote by L_p the number of aircraft allotted interval T_p (p = 1, 2, ..., P). Put L_p equal to zero ($L_p = 0$).

Step 3: Note the highest preference index on the list. If this value corresponds to "original" A_{i0}, put p equal to P_i ($p = P_i$). Should the highest value correspond to a "copy," put p equal to $P_i + m$ ($p = P_i + m$). Increase value L_p by 1. Check whether $L_p < K_p$. If this inequality is not satisfied, put L_p equal to K_p ($L_p = K_p$). If the inequality is satisfied, allot interval T_p to "original" A_{i0} or "copy" A_{im}. The delay of aircraft A_i equals zero if the highest preference index corresponds to the "original." Aircraft A_i will be delayed m time intervals if the highest preference index corresponds to "copy" A_{im}. Remove the preference index of aircraft A_i (the "original" and all its "copies") from the list. Go to Step 4.

Step 4: Check whether this inequality is satisfied:

$$\sum_{p=1}^{P} L_p = \sum_{p=1}^{P} K_p \tag{2.79}$$

If the inequality is not satisfied, go to Step 3. Otherwise, go to Step 5.

Step 5: Allot the last interval P + 1 to the remaining "originals" on the list. These aircraft are delayed P + 1 time intervals. Since all aircraft are distributed by time intervals, the algorithm is ended.

2.11. CONTROLLING A FLEET IN RIVER TRAFFIC USING FUZZY LOGIC

The problem of transporting bulk freight in river traffic will be presented using the example of loading, transporting, and unloading gravel. Let the suction dredger be located downstream from the ports where the gravel is to be transported. A pusher tug pushes tows that is, compositions of two, three or at most four barges. After loading gravel at the loading point, the tug pushes the barges upstream toward the ports where the gravel is to be delivered.

When the barges and gravel reach the unloading ports, the dispatcher in charge of traffic control decides on the number of barges that should be left at each port. The barges that the dispatcher has decided will stay at a particular port are pushed by the tug to be anchored in the reloading machinery zone. After leaving a certain number of barges in the port, the tug continues to navigate upstream until the entire tow (all the barges it has pushed) has been left in ports. Then the tug changes direction, heading downstream to pick up the empty barges along the way. The dispatcher in charge of traffic control decides on the number of empty barges that the tug should pick up at each port. After forming a tow of empty barges, the tug continues downstream to the suction dredging machine. The process of loading, transporting, and unloading gravel is shown in *Figure 2.45*.

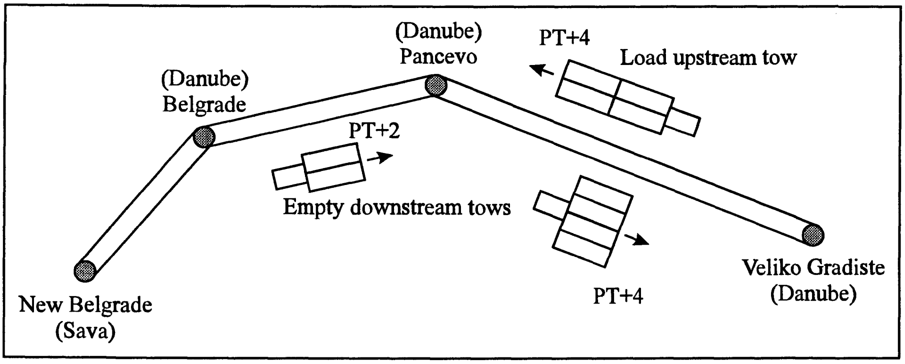

Figure 2.45. Loading, transporting, and unloading gravel on a river network

All processes of loading, transporting, and unloading take place simultaneously. In other words, as we observe the river network, some tugs are navigating upstream pushing barges filled with gravel, unloading and picking up of empty barges is taking place in the ports, while the remaining tugs are navigating downstream towards the dredging machine. A dispatcher manages the entire process of loading, transporting, and unloading in conditions of uncertainty, since the goals, constraints, and consequences of actions undertaken cannot be perceived with precision. Subjectively evaluating the situation, the dispatcher makes certain decisions with very little time at his disposition. The entire process of loading, transporting, and unloading becomes even more complex with an increase in the number of tugs, barges, and ports that need gravel.

2.11.1. Approximate reasoning model to control the process of loading, transporting, and unloading gravel

Studying the process of transporting bulk freight in river traffic, Vukadinovic and Teodorovic (1994) developed an approximate reasoning model to control the process of loading, transporting, and unloading gravel. We will briefly present their results.

When a tug enters a port, the dispatcher makes a decision about the number of full barges to be left or the number of empty barges to be picked up. When deciding on the number of full barges to leave, the dispatcher endeavors to minimize the tug's waiting time, bearing in mind the fuzzy variable "Waiting time to unload." Let us assume that the dispatcher subjectively notes three categories as "small," "average," and "large" waiting time to unload. These categories can be represented by fuzzy sets. The fuzzy variable "Number of barges left" can also be in the categories "large," "average," and "small" number of barges to be left. Bearing in mind the fuzzy variables "Waiting time to unload" and "Number of barges left," the dispatcher applies certain rules to make a final decision on the number of barges left in a specific port. For example, one of the rules might be: If waiting time to unload is "large," then the number of barges left is "small."

Endeavoring to form an empty tow for the tug navigating downstream, the dispatcher takes into consideration the fuzzy variable "Number of empty barges in the port," differentiating the categories "small," "average," and "large" number of empty barges. The fuzzy variable "Number of empty barges taken by a tug" also has the categories "large," "average," and "small" number. As in the case of the upstream tug, the dispatcher applies certain rules to arrive at a final decision regarding the number of empty barges to be taken by the tug.

The approximate reasoning algorithm developed by Vukadinovic and Teodorovic (1994) was applied in the case of two unloading ports - Pancevo and Belgrade (Yugoslavia). The dredging machine is located at Veliko Gradiste (*Figure 2.45*). When navigating upstream from Veliko Gradiste, the tow first comes to Pancevo where 0, 1, 2, 3, or 4 barges may be left. The dispatcher is the one to decide how many barges will be left in Pancevo. All freight barges not left in Pancevo are pushed to Belgrade. Thus, the following question must be answered: How many barges from the tow should be left in Pancevo? When making this decision, the dispatcher bears in mind the fuzzy variables "Waiting time to unload in Pancevo" and "Waiting time to unload in Belgrade." For example, if "Waiting time to unload in Pancevo" is "large" and "Waiting time to unload in Belgrade" is "small," then the "Number of barges left in Pancevo" is "small."

The membership functions of fuzzy sets "small," "average," and "large" waiting time to unload are shown in *Figure 2.46*.

Vukadinovic and Teodorovic (1994) defined the fuzzy sets "small," "average," and "large" number of barges left as positive integers of real variable y within the interval [0,4]. The fuzzy sets "small," "average," and "large" number of barges left are denoted respectively by **A**, **B**, and **C**. These sets are expressed as

$$
\begin{aligned}
A &= \frac{1}{0} + \frac{0.75}{1} + \frac{0.5}{2} + \frac{0.25}{3} + \frac{0}{4} \\
B &= \frac{0}{0} + \frac{0.5}{1} + \frac{1}{2} + \frac{0.5}{3} + \frac{0}{4} \\
C &= \frac{0}{0} + \frac{0.25}{1} + \frac{0.5}{2} + \frac{0.75}{3} + \frac{1}{4}
\end{aligned}
\tag{2.80}
$$

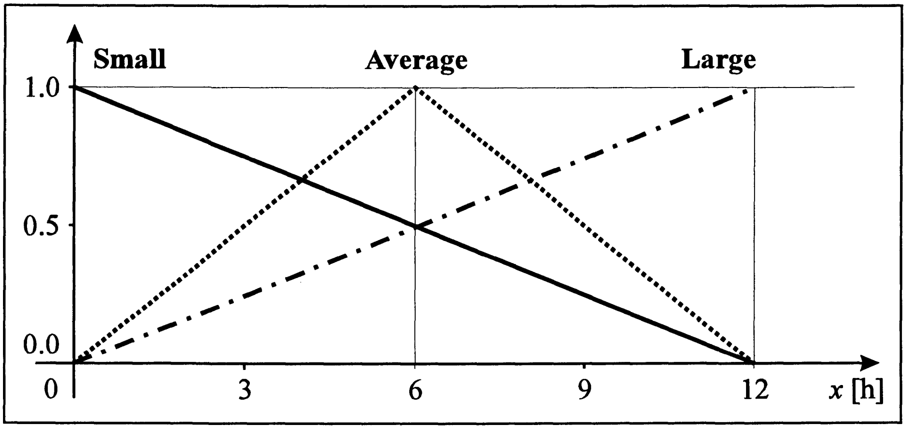

Figure 2.46. Membership functions of the fuzzy sets "small," "average," and "large" amount of waiting time to unload

When navigating downstream, the tug can take 0, 1, 2, 3, or 4 empty barges in Belgrade. After taking a certain number of empty barges in Belgrade, the tug completes its tow in Pancevo and then continues navigating downstream toward the dredging machine. The dispatcher endeavors to make decisions that will enable the tug going downstream to form a complete barge tow as quickly as possible. The following question must therefore be answered: How many empty barges should the tug take in Belgrade? When answering this question, the dispatcher must bear in mind the number of empty barges waiting in Pancevo. The fuzzy variables that the dispatcher uses to make his decisions are "Number of empty barges in Belgrade" and "Number of empty barges in Pancevo." The number of empty barges in Belgrade or Pancevo can be "small," "average," or "large." Let us denote by $\mathbf{A}_1$, $\mathbf{B}_1$, and $\mathbf{C}_1$, respectively, the fuzzy sets representing "small," "average," and "large" number of empty barges. These sets are expressed as

$$
\begin{aligned}
A_1 &= \frac{1}{0} + \frac{1}{1} + \frac{0.67}{2} + \frac{0.33}{3} + \frac{0}{4} \\
B_1 &= \frac{0}{0} + \frac{0.5}{1} + \frac{1}{2} + \frac{0.5}{3} + \frac{0}{4} \\
C_1 &= \frac{0}{0} + \frac{0.33}{1} + \frac{0.67}{2} + \frac{1}{3} + \frac{1}{4}
\end{aligned}
\tag{2.81}
$$

Let us introduce into the discussion the fuzzy set called "any." For example, we can say that there is "any" number of barges left or "any" amount of waiting time to unload. The membership function of every element belonging to the fuzzy set "any" equals 1. *Figure 2.47* presents the membership functions of fuzzy sets "any" number of barges left and "any" waiting time to unload.

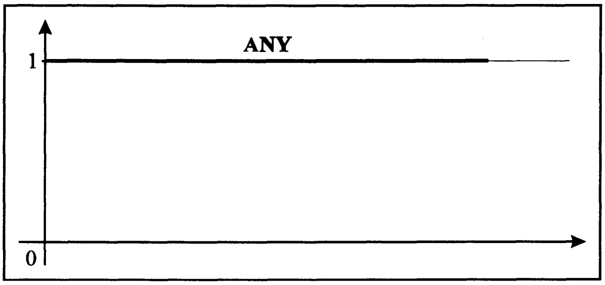

Figure 2.47. Membership function of fuzzy sets "any" number of barges left and "any" waiting time to unload

Vukadinovic and Teodorovic (1994) developed two approximate reasoning algorithms. The first calculates the number of barges that the tug should leave in Pancevo, and the second the number of empty barges it should take in Belgrade.

The algorithm that calculates the number of barges to be left in Pancevo consists of the following rules:

Rule 1: If waiting time to unload in Pancevo is **Large**, and waiting time to unload in Belgrade is **Large** or **Average**, then the number of barges left in Pancevo is **Average**,
or

...

Rule 5: If waiting time to unload in Pancevo is **Small**, and waiting time to unload in Belgrade is **Average** or **Small**, then the number of barges left in Pancevo is **Average**.

Figure 2.48 graphically presents the approximate reasoning by max-min composition to calculate the number of barges left in Pancevo.

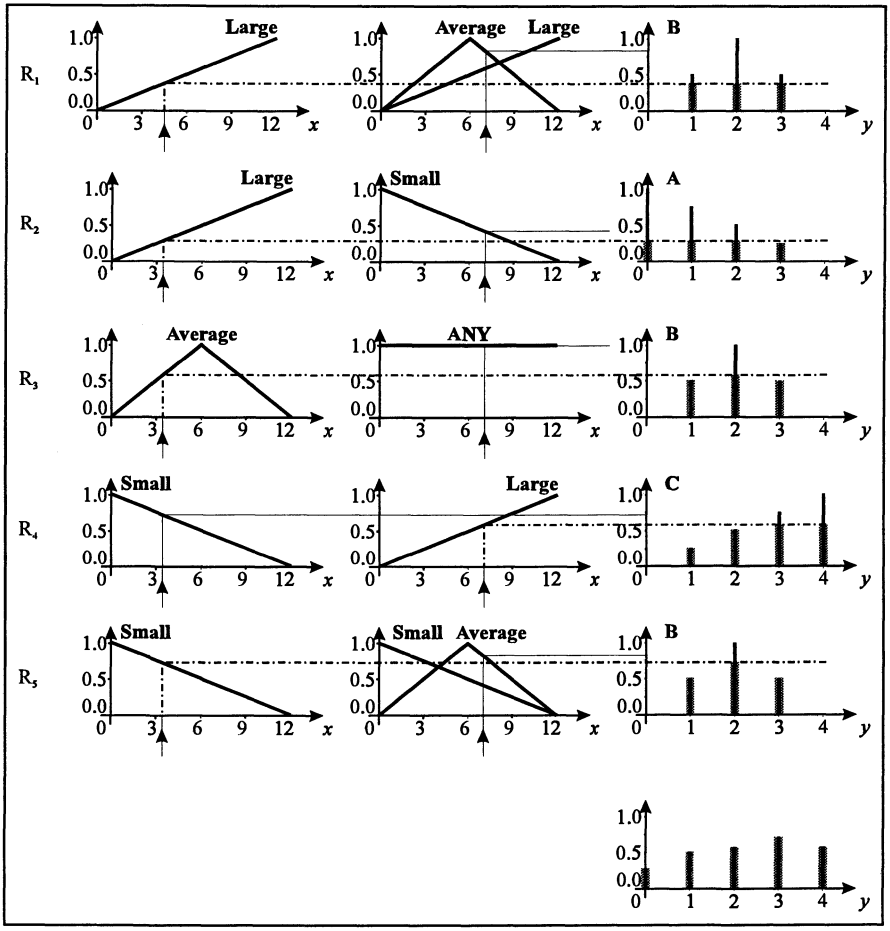

Figure 2.48. Graphical presentation of the approximate reasoning algorithm to calculate the number of barges left in Pancevo

The approximate reasoning algorithm to calculate the number of empty barges that are picked up in Belgrade consists of the following rules:

Rule 1: If the number of empty barges in Belgrade is **Large**, and the number of empty barges in Pancevo is **ANY**, then the number picked up in Belgrade is **Large**,

or

...

Rule 4: If the number of empty barges in Belgrade is **Small**, and the number of empty barges in Pancevo is **Small**, then the number of barges picked up in Belgrade is **Average**.

Rule 4 means that the tug should wait in Belgrade until the number of empty barges becomes an "average" number of barges.

Vukadinovic and Teodorovic tested these approximate reasoning algorithms on a large number of numerical examples. Waiting times to unload varied from zero to 12 hours. The number of empty barges was from zero to 4. In addition to calculating the number of barges that were left or picked up for different input values, dispatchers were surveyed as well. Dispatchers were asked the number of barges they would leave or the number of empty barges they would pick up within the framework of an offered scenario. In other words, the dispatchers were asked such questions as: "If you estimate that the waiting time to unload is 3 hours in Pancevo and 8 hours in Belgrade, how many barges would you leave in Pancevo?" *Table 2.13* and *Table 2.14*, respectively, present the number of barges left based on the approximate reasoning algorithm and the number left based on the dispatcher survey.

Table 2.13. Number of barges left based on the model, for different waiting times in Pancevo and Belgrade

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	2	2	2	2	2	2	2	2	1	1	1	1	1
1	2	2	2	2	2	2	2	2	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	1	1	1	1
3	2	2	2	2	2	2	2	2	2	2	1	1	1
4	2	2	2	2	2	2	2	2	2	2	2	1	1
5	2	2	2	2	2	2	2	2	2	2	2	2	2
6	2	2	2	2	2	2	2	2	2	2	2	2	2
7	2	2	2	2	2	2	2	2	2	2	2	2	2
8	3	3	2	2	2	2	2	2	2	2	2	2	2
9	3	3	3	2	2	2	2	2	2	2	2	2	2
10	3	3	3	3	2	2	2	2	2	2	2	2	2
11	3	3	3	3	3	2	2	2	2	2	2	2	2
12	3	3	3	3	3	2	2	2	2	2	2	2	2

A comparison of *Table 2.13* with *Table 2.14* indicates the very good agreement between the dispatcher decisions and the results of the approximate reasoning algorithms. Computer testing has shown that the developed models are applicable in real time.

Table 2.14. Number of barges left determined by dispatchers, for different waiting times in Pancevo and Belgrade

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	2	2	2	2	2	2	2	1	1	1	1	1	1
1	2	2	2	2	2	2	2	2	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	1	1	1	1
3	2	2	2	2	2	2	2	2	2	2	1	1	1
4	2	2	2	2	2	2	2	2	2	2	2	1	1
5	2	2	2	2	2	2	2	2	2	2	2	2	1
6	3	2	2	2	2	2	2	2	2	2	2	2	2
7	3	3	2	2	2	2	2	2	2	2	2	2	2
8	3	3	3	2	2	2	2	2	2	2	2	2	2
9	3	3	3	3	2	2	2	2	2	2	2	2	2
10	3	3	3	3	3	2	2	2	2	2	2	2	2
11	4	3	3	3	3	3	2	2	2	2	2	2	2
12	4	4	3	3	3	3	3	2	2	2	2	2	2

2.12. A FUZZY LOGIC APPROACH TO THE AIRCREW ROSTERING PROBLEM

Once the airline schedule has been designed, scheduling crews and assigning them to planned flights are among the most important problems facing air carriers. The aircrew scheduling problem can be formulated as follows: for a given airline schedule, schedule the crew "in the best way possible" while simultaneously respecting all provisions that regulate crew operations so as not to endanger the security of the flight. The expression "the best way possible" means that a suitable objective function must be defined that will be used to schedule the crew. Solving the crew scheduling problem results in a set of rotations from a given set of flight segments. A rotation is a sequence of flight segments on consecutive days that are made by a crew that begin and end their trip in the same air carrier base.

Once the crew scheduling problem is solved that is, a set of rotations has been defined to be carried out by the crew members, air carriers are faced with the problem of crew rostering. The crew rostering problem entails the assignment of different crew members to planned rotations. In other words, the crew rostering problem includes the construction of personalized monthly schedules (rosters). When the crew rostering problem has been solved, each crew member will be assigned rotations to be made during the following month. When assigning crew members, consideration must be taken of planned vacations, sick leave, days planned for schooling and training, requested free days, days when medical examinations are held, and so on. The problem of assigning crew members to planned work duties is combinatorial by nature. Because the problem is complex and has large dimensions, various heuristic algorithms are most often used to solve it.

Numerous models have been developed in the past three decades. Some of the most important works dealing with the aircrew rostering problem are Agard (1970), Nicolleti (1975), Antosik (1978), Buhr (1978), Moore et al. (1978), Tingley (1979), Marchettini (1980), Giafferri et al. (1982), Glaneert (1984), Sarra (1988), and Ryan (1992). Aircrew members are usually assigned to work duties to be carried out the following month.

Large crew rostering problems are usually decomposed into numerous smaller problems. This decomposition usually takes the form of the "day-by-day" method (Nicoletti, 1975; Buhr, 1978; Antosik, 1978; Marchettini, 1980; Glaneert, 1984) or the "pilot-by-pilot" method (Moore et al., 1978). Ryan (1992) and Gamache and Soumis (1993) modeled the crew rostering problem as a generalized set partitioning problem.

When solving the crew rostering problem, consideration must be taken of both the air carrier's interests and the crew members' interests. Experience from different airline companies indicates that crew members are particularly interested in approximately equal workloads. Bearing this in mind and the days in the month when each crew member cannot be assigned to a work duty (schooling and training, medical examinations, free days, and so on), when solving the crew rostering problem effort must be made so that all crew members have approximately the same number of monthly flying time, approximately the same number of free days and work days, approximately the same number of preferred and undesirable rotations, approximately the same number of weekend work days, and so on.

Instead of finding the optimal solution, Teodorovic and Lucic (1998) tried to find a "satisfactory solution" to the aircrew rostering problem. A "satisfactory solution" is defined as a solution that enables all crew members to have "approximately equal workloads." Let us briefly show Teodorovic and Lucic's (1998) model.

2.12.1. Mathematical formulation of the problem of assigning crew members to planned rotations

The maximum amount of crew workload (maximum amount of flight time, maximum number of takeoffs, and so on) is defined by law in every country. Every air carrier can define its own internal constraints that are stricter than those defined by law. Some crew workload constraints refer to a single day, while other constraints refer to a longer time period (usually one month). Constraints that refer to a single day (maximum allowed daily flight time, maximum allowed daily number of takeoffs, and so on) are taken into consideration when constructing crew rotations that is, when solving the crew scheduling problem. Therefore, these constraints will not be considered any further. The one-month constraints in the airline company (cumulative pilot flight time during the month, the number of takeoffs during the month, and so on) are called *hard constraints* since they must not be breached when constructing the crew rosters. Let us bring to attention that those hard constraints cannot be violated as they are prescribed by law and thus bring about legal responsibility of the violator on account of his compromising the safety of air transport. In further considerations we will use the same notation as Teodorovic and Lucic (1998).

Let us denote by m the total number of pilots to be assigned to work duties during the following month. The total number of rotations to be made the following month is denoted by k.

Let us introduce the following binary variables:

$$p_{il} = \begin{cases} 1, & \text{if the } i\text{th pilot can spend the } l\text{th day at work} \\ 0, & \text{otherwise} \end{cases} \tag{2.82}$$

$$q_{jl} = \begin{cases} 1, & \text{if the } j\text{th rotation starts on the } l\text{th day of the month} \\ 0, & \text{otherwise} \end{cases} \tag{2.83}$$

where $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, k$, and $l = 1, 2, \ldots, 30$.

When $p_{il} = 0$, then it has been planned in advance for the ith pilot to be absent from work on the lth day (vacation, medical examination, schooling and training, and so on).

The total number of possible working days P of all m pilots during the following month equals

$$P = \sum_{i=1}^{m} \sum_{l=1}^{30} p_{il} \tag{2.84}$$

We denote by d_j the "length" of the jth rotation expressed as flight time. Total "length" D of all k rotations to be made the following month is

$$D = \sum_{j=1}^{k} d_j \tag{2.85}$$

Let us denote by a the ideal average daily flight time of each of the m pilots. This number equals

$$a = \frac{D}{P} = \frac{\sum_{j=1}^{k} d_j}{\sum_{i=1}^{m}\sum_{l=1}^{30} p_{il}} \tag{2.86}$$

It is clear that the best situation upon solving the crew rostering problem would be for each of the m pilots to have an average daily flight time equal to a. Let us denote by a_i^* the ideal monthly flight time of the ith pilot. This is

$$a_i^* = a\sum_{l=1}^{30} p_{il}, \qquad i = 1, 2, \ldots, m \tag{2.87}$$

As we can see, when calculating the ideal monthly flight time for each pilot, the total number of days in the month that the pilot can be assigned to a work duty must be taken into consideration. Value a_i^* clearly differs from pilot to pilot.

Let us introduce into the discussion binary variable x_{ij} that is defined as follows:

$$x_{ij} = \begin{cases} 1, & \text{if the } i\text{th pilot is assigned to the } j\text{th rotation} \\ 0, & \text{otherwise} \end{cases} \tag{2.88}$$

Let us denote by a_i the real monthly flight time to be flown by the ith pilot. Value a_i equals

$$a_i = \sum_{j=1}^{k} d_j x_{ij}, \qquad i = 1, 2, \ldots, m \tag{2.89}$$

We denote by F_1 the average relative deviation (per pilot) of real monthly flight time from the ideal. Then

$$F_1 = \frac{1}{m}\sum_{i=1}^{m} \left| \frac{\sum_{j=1}^{k} d_j x_{ij} - a\sum_{l=1}^{30} p_{il}}{a\sum_{l=1}^{30} p_{il}} \right| \tag{2.90}$$

It is clear that the smaller the value of F_1 the closer the real monthly flight time of most pilots is to the ideal possible monthly flight time. Value F_1 can serve as one of the objective functions when solving the crew rostering problem. Of course, instead of minimizing the average monthly deviation (per pilot), it would be possible to minimize the maximum deviation per pilot, and so on.

As noted previously, the relatively equal workload per pilot can also be understood as an equal number of weekend days spent outside the home, an equal number of departures before 8:00 a.m., or an equal number of foreign per diem allowances during the month (in some companies pilots are much better paid when they fly foreign routes). It was shown that in the company used as an illustrative example, equaling the number of foreign per diem allowances per pilot was extremely important. The foreign per diem included a special monetary compensation received by the pilot for every workday spent in the cockpit on an international route. The second objective function was the average absolute deviation (per pilot) between the real and ideally possible number of foreign per diem allowances during the month.

Let us denote by b the "ideal" share of foreign per diem allowances that each pilot should receive during the month. Then b equals

$$b = \frac{\sum_{j=1}^{k} c_j}{\sum_{i=1}^{m} \sum_{l=1}^{30} p_{il}} \tag{2.91}$$

where c_j is total number of foreign per diem allowances included in the jth rotation.

Let us denote by b_i^* the ideal monthly number of foreign per diem allowances of the ith pilot. Then b_i^* equals:

$$b_i^* = \left\lceil b \sum_{l=1}^{30} p_{il} \right\rceil, \qquad i = 1, 2, \ldots, m \tag{2.92}$$

where $\lceil y \rceil$ is real number y rounded to the nearest integer.

Let us denote by F_2 the average absolute deviation (per pilot) between the real and ideal number of foreign per diem allowances during the month. Then we have

$$F_2 = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{\sum_{j=1}^{k} c_j x_{ij} - \left\lceil b\sum_{l=1}^{30} p_{il}\right\rceil}{\left\lceil b\sum_{l=1}^{30} p_{il}\right\rceil}\right| \tag{2.93}$$

When solving the crew rostering problem, the emphasis is certainly placed on minimizing F_2 that is, equaling the number of foreign per diem allowances each pilot earns during the month.

It should be noted that the methodology proposed by Teodorovic and Lucic (1998) to solve the crew rostering problem loses none of its generality if some other criteria are chosen for the objective functions instead of F_1 and F_2.

Various constraints that must be satisfied are a component part of the crew rostering problem. The Teodorovic and Lucic's paper (1998) has taken into consideration the following constraints:

Flight time limitations: When solving the crew rostering problem, consideration must be taken of the amount of flight time accumulated since the beginning of the month. In the case under consideration, the following relation must be satisfied:

$$\sum_{j=1}^{k} d_j x_{ij} \le 85, \qquad i = 1, 2, \ldots, m \tag{2.94}$$

In other words, the amount of flight time accumulated during the entire month must not exceed 85 hours.

Total number of takeoffs during the month: The total number of takeoffs per pilot per month must not exceed 90. In other words, the following relation must be satisfied:

$$\sum_{j=1}^{k} f_j x_{ij} \le 90, \qquad i = 1, 2, \ldots, m \tag{2.95}$$

where f_j is the number of takeoffs contained in the jth rotation.

In addition to these constraints, the following constraints were also taken into consideration when solving the crew rostering problem:

- The total monthly number of working hours must not exceed 160 hours per pilot.

- Every pilot must have a free day no later than the fifth consecutive working day.
- While carrying out his rotation, a pilot may not be given a free day.
- One rotation may be given to only one pilot.
- The rotations assigned to a pilot must not overlap in time.

When solving the crew rostering problem, emphasis was put on achieving the smallest possible values for F_1 and F_2, while at the same time satisfying all existing constraints.

2.12.1.1. Modified "day-by-day" method for solving the aircrew rostering problem

The algorithm to solve the aircrew rostering problem developed by Teodorovic and Lucic (1998) is a modification of the "day-by-day" heuristic method. The algorithm consists of the following algorithmic steps:

Step 1: Note and set aside all rotations that start on the first day of the month.

Step 2: Assign crew members to the above rotations. In doing so, constantly endeavor to achieve the smallest possible values for F_1 and F_2 and make sure that no constraints are breached concerning both the day in question and all previous days for which assignments have already been make.

Step 3: Note and set aside all rotations that start on the following day.

Step 4: Check whether assignments have been made for all the days in the month. If they have, the algorithm is finished. Otherwise, return to Step 2.

The only difference between the proposed method and the standard "day-by-day" method is Step 2. When assigning pilots to rotations within Step 2, attention must be given to achieving the "smallest possible values" for F_1 and F_2. This reflects our desire to achieve the "most equal" workload possible among the pilots. Statements along the line "approximately equal monthly flight time" or "approximately equal monthly number of foreign per diem allowances" contain a certain fuzziness within them. Each individual crew rostering plan contains a greater or lesser amount of equality regarding monthly flight time and/or monthly number of foreign per diem allowance. Our satisfaction with the assignment of a specific rotation to a specific pilot is larger or smaller depending on whether it results in larger or smaller equality in the pilots' workload.

2.12.1.2. Approximate reasoning algorithm for assigning a specific pilot to one of the rotations starting on the day in question

Let us note the sth day of the month ($1 \le s \le 30$). We assume that the ith pilot should be assigned one of the rotations that starts on the sth day of the month. Since we are keeping to the "day-by-day" procedure, the ith pilot has already been assigned rotations that start on the first, second, or (s-1)th day. The ith pilot's total accumulative flight time including day s-1 equals

$$a_i(s-1) = \sum_{j=1}^{k} \sum_{l=1}^{s-1} q_{jl} x_{ij} d_j \tag{2.96}$$

Let the zth rotation be a candidate for assignment to the ith pilot. Should the zth rotation be assigned to the ith pilot, the amount of accumulated flight time by the ith pilot including the sth day would be:

$$a_{iz}(s) = a_i(s-1) + d_z \tag{2.97}$$

The ideal amount of flight time of the ith pilot including the sth day is:

$$a_i^*(s) = a \sum_{l=1}^{s} p_{il} \tag{2.98}$$

Let us denote by $q_{iz}(s)$ the relative deviation of the assigned flight time from the ith pilot's ideal flight time ending with the sth day, should the zth rotation be assigned to the ith pilot. It is clear that

$$q_{iz}(s) = \frac{a_{iz}(s) - a_i^*(s)}{a_i^*(s)} \tag{2.99}$$

The amount of $q_{iz}(s)$ is extremely important for the analyst-crew planner's decision as to whether or not to assign the zth rotation to the ith pilot. The smaller the relative deviation $q_{iz}(s)$, the greater the analyst's preference to assign the ith pilot to the zth rotation starting on the sth day. This relative deviation can be positive or negative. It is assumed that every analyst-crew planner has a subjective idea about what is highly, moderately, or slightly positive or negative regarding relative deviation.

In a similar manner, we can define $w_{iz}(s)$ representing the relative deviation of the assigned number of foreign per diem allowances from the ith pilot's ideal number of foreign per diem allowances ending with the sth day should the ith pilot be assigned the zth rotation. Deviation $w_{iz}(s)$ equals

$$w_{iz}(s) = \frac{b_{iz}(s) - b_i^*(s)}{b_i^*(s)} \tag{2.100}$$

where $b_{iz}(s)$ is number of the ith pilot's foreign per diem allowances ending with the sth day should he be assigned the zth rotation, and $b_i^*(s)$ is ideal number of foreign per diem allowances for the ith pilot ending with the sth day.

Amounts $b_{iz}(s)$ and $b_i^*(s)$ are calculated in a similar way as amounts $a_{iz}(s)$ and $a_i^*(s)$.

Let us introduce into the discussion the following fuzzy sets with which we will try to describe the relative deviation of the assigned flight time from the ideal flight time, or the assigned number of foreign per diem allowances from the ideal number: **HN** is highly negative relative deviation of the assigned from the ideal, **MN** is moderately negative relative deviation of the assigned from the ideal, **S** is small relative deviation of the assigned from the ideal, **MP** is moderately positive relative deviation of the assigned from the ideal, **HP** is highly positive relative deviation of the assigned from the ideal, and **ANY** is any deviation whatsoever of the assigned from the ideal.

The membership functions of the sets **HN**, **MN**, **S**, **MP**, and **HP** are shown in *Figure 2.49*. As we can see, particular membership functions are asymmetric. Their shapes that is, the limit values of the triangular membership functions have been reached after the examination of many different shapes of the membership functions. It was shown that those shapes led to the best results.

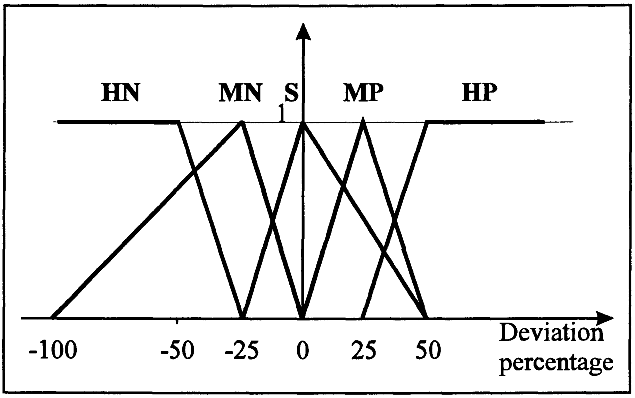

Figure 2.49. The membership functions of the fuzzy sets **HN**, **MN**, **S**, **MP**, and **HP**

Let us introduce into the discussion the preference index $p_{iz}(s)$ that expresses the analyst's preference to assign the ith pilot to the zth rotation that starts on the sth day. Let

$$0 \leq p_{iz}(s) \leq 1, \quad i = 1,2 \ldots, m \quad z = 1,2 \ldots, k \quad s = 1,2 \ldots, 30 \qquad (2.101)$$

It is clear that the analyst's readiness to assign the zth rotation starting on the sth day to the ith pilot is greater, the greater the value of preference index $p_{iz}(s)$.

Let us also introduce into the discussion the following fuzzy sets that refer to the strength of the decision maker's preference: **VW** is very weak preference, **W** is weak preference, **M** is medium preference, **S** is strong preference, and **VS** is very strong preference.

The membership functions of fuzzy sets **VW**, **W**, **M**, **S**, and **VS** are shown in *Figure 2.50.*

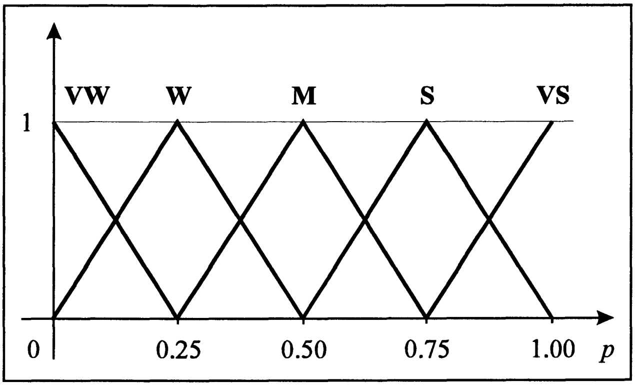

Figure 2.50. The membership functions of fuzzy sets **VW**, **W**, **M**, **S**, and **VS**

The approximate reasoning algorithm that determines the strength of the decision maker's preference regarding the assignment of the zth rotation starting on the sth day to the ith pilot consists of the following algorithmic steps:

Rule 1: If $q_{iz}(s)$ is **BN** and $w_{iz}(s)$ is **HN**, then $p_{iz}(s)$ is **M**,
or

Rule 2: If $q_{iz}(s)$ is **HN** and $w_{iz}(s)$ is **MN** or **S**, then $p_{iz}(s)$ is **W**,
or

...

Rule 13: If $q_{iz}(s)$ is **HP** and $w_{iz}(s)$ is **ANY**, then $p_{iz}(s)$ is **VW.**

In this manner, the preferences of assigning different rotations starting on the day in question are calculated for each pilot. Let us denote by P_s the matrix whose elements $p_{ij}(s)$ are the preference indexes of assigning the *i*th pilot ($i = 1, ..., m$) to the *j*th rotation ($j = 1, ..., k$) starting on the *s*th day of the month (preference indexes $p_{ij}(s)$ are only calculated for the rotations for which $q_{js} = 1$ that is, only for rotations starting on the *s*th day of the month). In the next step, it is necessary to solve the assignment problem that is, it is essential to make the final choice of rotation assigned to a particular pilot on the day in question.

The input data for the approximate reasoning algorithm are the values of $q_{iz}(s)$ and $w_{iz}(s)$. We recall that $q_{iz}(s)$ is the relative deviation of the *i*th pilot's assigned flight time from the ideal flight time ending with the *s*th day, should the *z*th rotation be assigned to the *i*th pilot. We would also repeat that $w_{iz}(s)$ is the relative deviation of the *i*th pilot's assigned number of foreign per diem allowances from the ideal number, ending with the *s*th day, should the *i*th pilot be assigned the *z*th rotation. A question naturally arises regarding how to use the approximate reasoning algorithm at the very beginning. In other words, how to use the approximate reasoning algorithm on the first day when none of the pilots has been assigned any rotation. Assigning rotations that begin on the first day of the month to the pilots was done in the manner described below. First, the pilots are sorted in increasing order by the number of days they can work during the month in question. Also, the rotations are sorted in decreasing order by their length of time. Then, the longest rotation from the list is assigned to the pilot with the smallest number of available working days, the second longest to the second pilot on the list and so on.

In a described manner, pilots have been assigned rotations that start the first day. Thus, $q_{iz}(s)$ and $w_{iz}(s)$ have received their initial values and the approximate reasoning algorithm can be used to assign pilots to rotations that begin on the second, third, ..., thirtieth day of the month.

The algorithms developed were tested by Teodorovic and Lucic (1998) on the example of assigning fifty three flight captains to 422 rotations to be flown during the subsequent thirty days. The data are real and have been obtained from one airline company. The rotation characteristics are shown in *Table 2.15*.

Table 2.15. Characteristics of the rotations to be assigned

Rotation number	Rotation starting day	Rotation starting time (min)	The earliest day of the following rotation start	The earliest time of the following rotation start (min)
1	1	355	2	270
2	1	365	2	1175
3	1	380	3	340
4	1	390	2	1105
5	1	400	1	1210
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
422	30	1325	31	930

Rotation number	Number of legs in rotation	Rotation flying time (min)	Rotation working time (min)	Rotation foreign per diem allowance	Rotation working days
1	5	240	485	0	1
2	4	355	585	1	2
3	6	590	1090	2	2
4	5	225	445	0	2
5	4	210	300	0	1
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
422	2	365	535	2	2

The values of binary variables p_{il} are given in *Table 2.16* (when $p_{il} = 1$, the *i*th pilot can work on the *l*th day of the month).

Table 2.16. Pilot availability by days during the month

Pilot	Days				
Number	1	2	3	...	30
1	1	1	1	...	1
2	1	1	0	...	1
3	1	1	1	...	1
...	...	...	...	...	...
53	1	1	1	...	1

The average deviation of real flight time from ideal flight time per captain is 7.17%. Average deviation of foreign per diem allowances per captain is 9.31%. The relative deviations of the flight time and number of foreign per diem allowances are shown in *Figure 2.51* and *Figure 2.52*.

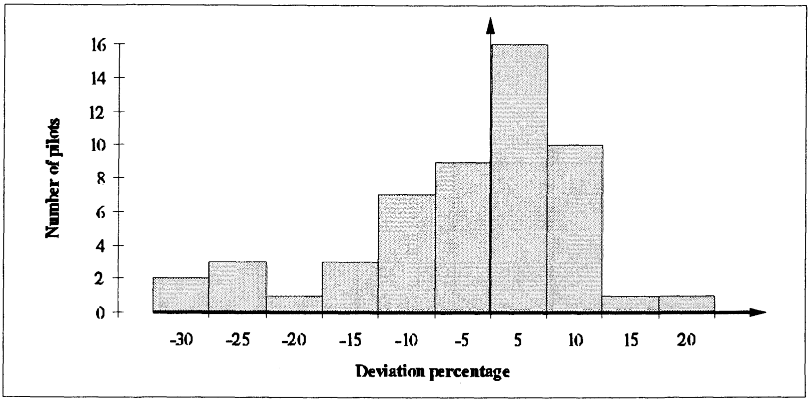

Figure 2.51. Distribution of the deviation of real flight time from ideal flight time

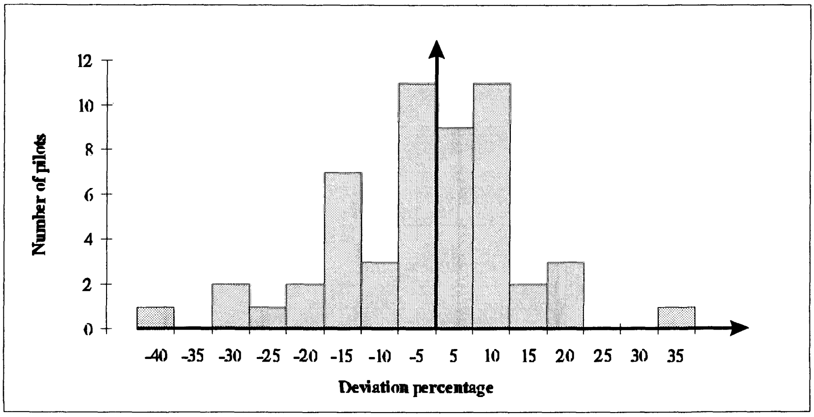

Figure 2.52. Distribution of the deviation of real number of foreign per diem allowances from the ideal number

Based on the average deviation values regarding flight time (7.17%) and number of foreign per diem allowances (9.31%), and based on the shape of the distributions shown in *Figure 2.51* and *Figure 2.52*, we can conclude that we have achieved satisfactory results in resolving the numerical example.

The considered multicriteria decision-making problem has been solved using fuzzy control methods. The reasons for such approach can be found in the fact that, using fuzzy control methods, it is possible to accommodate possible criteria of the qualitative character such are defining of the "less desirable airport for landing," equalizing the monthly number of landings at the "less desirable airports," and so on. Using fuzzy control methods in order to determine the strength of the decision maker's preference to assign rotation to a particular pilot in a satisfactory way mimics the job of an analyst doing the crew rostering. In the conversation with crew planners it is noted that they think and base their decision in a manner similar to the laid out algorithm of the approximate reasoning.

2.13. A FUZZY LOGIC APPROACH TO THE VEHICLE ASSIGNMENT PROBLEM

Assigning vehicles to planned transportation tasks is a daily problem in every transportation company. In most cases, dispatchers responsible for assigning the vehicles rely primarily on their experience and intuition in the course of decision making. Experienced dispatchers usually have built-in criteria ("rules") which they use to assign a given amount of freight to be sent a given distance to a given vehicle with given structural and technical-operational characteristics (capacity, ability to carry freight certain distances, and so on).

Transportation companies receive a great number of requests every day from clients wanting to send goods to different destinations. Each transportation request is characterized by a large number of attributes, including the most important: type of freight, amount of freight (weight and volume), loading and unloading sites, preferred time of loading and/or unloading, and distance the freight is to be transported. Transportation companies usually have fleets of vehicles consisting of several different types of vehicle. In addition to the characteristics of the transportation request, when assigning a specific type of vehicle to a specific transportation request, the dispatcher must also bear in mind the total number of available vehicles, the available number of vehicles by vehicle type, the number of vehicles temporarily out of working order, and vehicles undergoing technical examinations or preventive maintenance work. When meeting transportation requests, one or more of the same type of vehicle might be used. In other cases, several different types of vehicle might be used. Depending on the characteristics of the transportation request and how the transportation company operates, vehicle assignments to transportation requests can be made several times a day, once a day, once a week, and so on. Without loss of generality, we considered the case when dispatching is carried out every day based on the principle "today for tomorrow." In other words, dispatchers have a set amount of time (one day) to match available vehicles to transportation tasks that are to begin the following day.

In some transportation companies, drivers are "married" to their vehicles (a driver always drives the same vehicle). On the other hand, in some companies a driver might drive a different vehicle every day. Depending on their driver's license, some drivers can drive one or several different types of vehicle. Legal and/or company constraints regarding the length of the driver's working day, rest time required between two trips, the length of regulated monthly or annual vacations, and so on, also have an important influence on the way that drivers will be assigned to different tasks. We will not consider the problem of assigning drivers to specific tasks. The basic problem studied here is the assignment of vehicles to specific trips. The results reached after matching vehicles to transportation requests comprise the input data to solve crew scheduling and crew rostering problems.

Recently Milosavljevic et al. (1996) considered the vehicle assignment problem. The basic goal of their research is to develop a model that will be able to imitate the work of an experienced dispatcher. A good dispatcher must have suitable abilities and skills, and his training usually requires a long period. The problem considered in the mentioned paper was not one requiring "real-time" dispatching (which is needed to dispatch ambulances, fire department vehicles, police patrol units, taxis, dial-a-ride systems, and so on). However, in spite of this, the large number of different input data and the relatively limited time in which to solve the problem of assigning vehicles to requests can certainly create stressful situations for the dispatcher. All of this supports the need to develop a system that will help the dispatcher make decisions.

Milosavljevic et al. (1996) attempted to formalize the dispatcher's knowledge that is, to approximately determine the rules used by the dispatcher to make decisions. Let us present the results achieved by Milosavljevic et al. (1996).

2.13.1. Statement of the problem

Let us consider the vehicle assignment problem within the scope of the following scenario. We assume that a transportation company has several different types of vehicle at its disposal. The number of different vehicle types is denoted by n. Individual types of vehicle differ from each other in terms of structural and technical-operational characteristics. We also assume that the transportation company has a depot from which the vehicles depart and to which they return after completing their trip (*Figure 2.53*):

Let us consider a delivery system in which different types of freight are delivered to different nodes. We also assume that after serving a node, the vehicle returns to the depot. The reasons for such a delivery tactic are often due to the fact that different types of freight cannot be legally delivered in the same vehicle, and that different types of freight belong to different clients of the transportation company. Since the vehicle returns to the depot after serving a node, we note that the routes the vehicle is to take are known (*Figure 2.53*). As shown in *Figure 2.53*, we are dealing with a set of routes in the form of a star, with each route containing a node to be served. Let us denote by m the total number of transportation requests to be undertaken the following day. Let us also denote by T_i the *i*th transportation request, ($i = 1, 2, ..., m$). Every transportation request T_i is characterized by four parameters (v_i, Q_i, D_i, n_i), where v_i is the node where freight is to be delivered when executing transportation request T_i, Q_i is the amount of freight to be transported by request T_i, D_i is the distance freight is to be transported in request T_i (the distance between depot D and node v_i), and n_i is the number of trips along route {D, v_i, D} that can be made by one vehicle during the time period under consideration (one day)

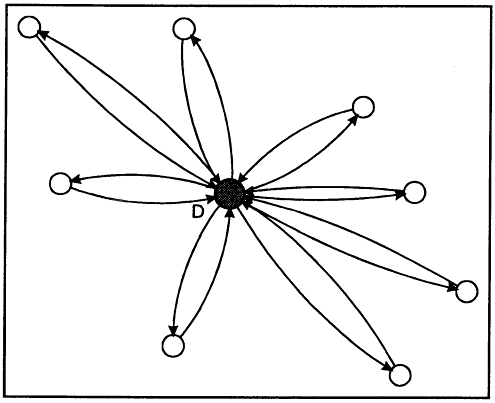

Figure 2.53. Depot D and nodes to be served

In order to simplify the problem, we will assume that the number of possible trips n_i that can be made along route {D, v_i, D} is independent of the vehicle type.

We shall denote by C_j the capacity of vehicle type j taking part in the service ($j = 1, 2, ..., n$). The number of available type j vehicles is denoted by N_j. We also assume that

$$n_i \geq 1 \qquad i = 1, 2, \ldots, m \tag{2.102}$$

Based on relation (2.102), we conclude that the vehicle can serve any node within the geographical region under consideration at least once a day and return to the depot.

Depending on the values of D_i and Q_i and the capacity C_j of the vehicle serving node v_i, one or more trips will be made along route {D, v_i, D} during the day being considered. One type or a variety of vehicle types can take part in the delivery to node v_i. Let us first consider the case when only one type of vehicle takes part in serving any node. The more complicated case when several different types of vehicle serve a node is considered later. We would also note that in some cases there is the possibility of the transportation company not being able to serve all nodes with its available transportation capacities.

The standard routing problem (Bodin and Golden, 1981) consists of designing a route to be taken by the vehicles when serving the nodes. In most papers devoted to the classical routing problem, it is assumed that the capacity of the serving vehicle is greater than or equal to demand in any node. In our case, the routes to be taken by the vehicles are known (*Figure 2.53*). We shall denote by f_{ij} the number of trips (frequency) to be made by a type j vehicle when executing transport request T_i. It is clear that $f_{ij} \geq 0$ ($i = 1, 2, ..., m, j = 1, 2, ..., n$).

The problem considered by Milosavljevic et al. (1996) is to determine the value of f_{ij} ($i = 1, 2, ..., m, j = 1, 2, ..., n$) so that the available vehicles are assigned to planned transportation tasks "in the best possible way." The expression "the best possible way" naturally requires more detailed discussion. The next section will consider the criteria to be used when assigning vehicles to planned transportation tasks.

2.13.2. Proposed solution to the problem

The total number of vehicles N available to the dispatcher at the moment he assigns vehicles is

$$N = \sum_{j=1}^{n} N_j \tag{2.103}$$

As already mentioned, the problem considered is the assignment of N available vehicles to m transportation requests. This belongs to the category of operations research problems known as assignment problems. The "classical" approach to solving the vehicle assignment problem is to define a corresponding objective function and set of constraints. In other words, the vehicle assignment problem can be treated as a mathematical programming problem. Let us assume for the moment that in meeting every transport request only one type of vehicle is involved. Let us denote by c_{ij} the cost of assigning vehicle type j to the ith transportation request, and introduce into the discussion binary variable x_{ij} defined as follows:

$$x_{ij} = \begin{cases} 1, & \text{if vehicle type } j \text{ is assigned to meet} \\ & \text{transportation request } T_i \\ 0, & \text{if vehicle type } j \text{ is not assigned to meet} \\ & \text{transportation request } T_i \end{cases} \tag{2.104}$$

Let us denote by F the total cost of assigning the available vehicles to the transportation requests.

Minimizing F, along with defining the set of constraints that refer to the utilization of individual vehicles and to "coverage" of the planned transportation tasks, leads to the formulation of the vehicle assignment problem as an integer programming problem.

On the other hand, there are numerous factors that cannot be built into the formulation of the vehicle assignment problem as an integer programming problem. It is often impossible to precisely determine assignment cost c_{ij}. Some transportation requests are "more important" than others. In other words, some clients have signed long-term transportation contracts, and others randomly request transportation that will engage transportation capacities for longer or shorter periods of time. In some cases there is no absolutely precise information about the number of individual types of vehicle that will be ready for operation the following day. Bearing in mind the number of operating vehicles and the number of vehicles expected to be operational the following day, the dispatcher subjectively estimates the total number of available vehicles by type. Some vehicle types are more "suitable" for certain types of transportation tasks than other. Naturally, vehicles with a 5t capacity are more suitable to deliver good within a city area than those with a 25t capacity. On the other hand, 25t vehicles are considerably more suitable than 5t or 7t vehicles for long-distance freighting.

As we can see, the vehicle assignment problem is often characterized by uncertainty regarding input data necessary to make certain decisions (assignment cost c_{ij} cannot always be estimated precisely, while the future number of available vehicles is liable to the dispatcher's subjective estimation). It should be emphasized that the subjective estimation of individual parameters differs from dispatcher to dispatcher, or from decision maker to decision maker. The number of available vehicles of a specific type might be "sufficient" for one dispatcher, while another dispatcher might think this number "insufficient" or "approximately sufficient." Also, one dispatcher might consider a certain type of vehicle "highly suitable" regarding a certain distance, while other dispatchers consider this type of vehicle "suitable" or "relatively suitable." Clearly, a number of parameters that appear in the vehicle assignment problem are characterized by uncertainty, subjectivity, imprecision and ambiguity. This raises the need in the mathematically modeling phase of the problem to use methods that can satisfactorily treat uncertainty, ambiguity, imprecision and subjectivity. The approximate reasoning model presented in the following section is an attempt to formalize the dispatcher's knowledge that is, to determine the rules used by dispatchers in assigning vehicles to transportation requests.

2.13.3. Approximate reasoning model for calculating the dispatcher's preference when only one type of vehicle is used to meet every transportation request

It can be stated that every dispatcher has a pronounced subjective feeling about which type of vehicle corresponds to which transportation request. This subjective feeling concerns both the suitability of the vehicle in terms of the distance to be traveled and vehicle capacity in terms of the amount of freight to be transported.

Dispatchers consider the suitability of different types of vehicles as being “low” (**LS**), “medium” (**MS**), and “high” (**HS**) in terms of the given distance the freight is to be transported. Also, capacity utilization (the relationship between the amount of freight and the vehicle's declared capacity, expressed as a percentage) is often estimated by the decision maker as “low” (**LCU**), “medium” (**MCU**), or “high” (**HCU**).

The suitability of a certain type of vehicle to transport freight different distances, and its capacity utilization can be treated or represented as fuzzy sets (*Figure 2.54* and *Figure 2.55*).

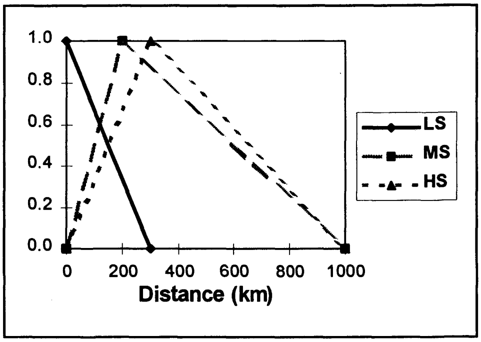

Figure 2.54. Membership functions of fuzzy sets: **LS** is low, **MS** is medium, and **HS** is high suitability in terms of distance

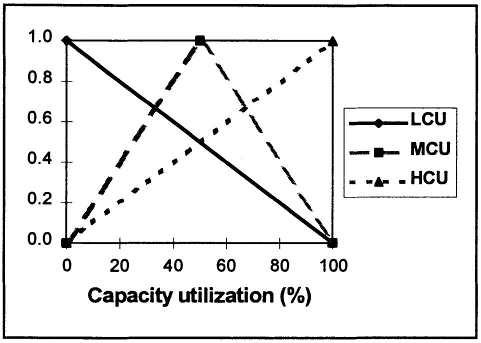

Figure 2.55. Membership functions of fuzzy sets: **LCU** is low, **MCU** is medium, **HCU** is high vehicle capacity utilization

Vehicle capacity utilization is the ratio of the amount of freight transported by a vehicle to the vehicle's capacity.

The membership functions of the fuzzy sets shown in *Figure 2.54* and *Figure 2.55* must be defined individually for every type of vehicle.

The decision maker assigns transportation requests to individual types of vehicle bearing in mind above all the distance to be traveled and the capacity utilization of the specific type of vehicle. When dispatching, the decision maker - dispatcher operates with certain rules. Based on conversations with dispatchers who deal with the vehicle assignment problem every day, Milosavljevic et al. (1996) noted that the decision maker has certain preferences:

- "Very strong" preference is given to a decision that will meet the request with a vehicle type having "high" suitability in terms of distance and "high" capacity utilization,

  or
- "Very weak" preference is given to a decision that will meet the request with a vehicle that has "low" suitability regarding distance and "low" capacity utilization.

The strength of the dispatcher's preference can be "very strong," "strong," "medium," "weak," and "very weak." The problem of converting linguist terms into corresponding fuzzy sets has been treated by many authors (Baas and Kwaakernak, 1977; Bonissone, 1982; Efstathiou and Rajkovic, 1979; Efstathiou and Tong, 1982; Kerre, 1982; Wenstop, 1976; and particularly Chen and Hwang, 1992). Milosavljevic et al. (1996) noted that dispatchers most often use five terms to express the strength of their preference regarding the meeting of a specific transportation request with a specific type of vehicle. These five preference categories can be presented as corresponding fuzzy sets **P1**, **P2**, **P3**, **P4**, and **P5**. The membership functions of the fuzzy sets used to describe preference strength are shown in *Figure 2.56*. Preference strength will be indicated by a preference index, **PI**, which lies between 0 and 1, where a decrease in the preference index means a decrease in the "strength" of the dispatcher's decision to assign a certain transportation request to a certain type of vehicle.

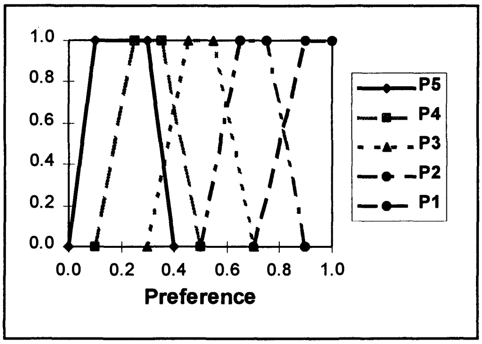

Figure 2.56. Membership functions of fuzzy sets: **P1** is very strong, **P2** is strong, **P3** is medium, **P4** is weak, **P5** is very weak preference

For every type of vehicle, a corresponding approximate reasoning algorithm is developed to determine the dispatcher's preference strength in terms of meeting a specific transportation request with the type of vehicle in question. The approximate reasoning algorithms for each type of vehicle differ from each other in terms of the number of rules they contain and the shapes of the membership functions of individual fuzzy sets. For example, for a vehicle with a capacity of 14t, the approximate reasoning algorithm reads as shown in *Table 2.17*.

Table 2.17. Approximate reasoning algorithm for a vehicle with a capacity of 14t

		Capacity utilization		
		LCU	MCU	HCU
Suitability	LS	P5	P4	P3
	MS	P3	P2	P2
	HS	P2	P1	P1

Using the approximate reasoning by max-min composition, every preference index value is assigned a corresponding grade of membership. Let us denote this value by P_{ij}. This value expresses the "strength" of the dispatcher's preference that the *i*th transportation request be met by vehicle type *j*.

Similar approximate reasoning algorithms were developed for the other types of vehicle.

2.13.4. Calculating the dispatcher's preference when several types of vehicle are involved in meeting requests

Up until now, we have only considered the vehicle assignment problem when one type of vehicle is used to meet every transportation request. Some transportation companies often use several different types of vehicle to meet a specific transportation request. When meeting requests with several different types of vehicle, every request can be met in one or several different ways. For example, if the amount of freight in the *i*th request equals $Q_i = 18$t and if we have two types of vehicle whose capacities are 5t and 7t, respectively, there are four possible alternatives to meeting the ith request shown in *Table 2.18*.

Table 2.18. Alternatives to meeting the *i*th request

Alternative number	Number of vehicles in services	
	7t	5t
1	3	0
2	2	1
3	1	3
4	0	4

The first of the possible alternatives to meet any transportation request is the one in which only one type of vehicle is used, the vehicle with the greatest capacity. Every other alternative differs from the previous to the effect that there is a smaller share of vehicles with a higher capacity and a greater share of vehicles with a smaller capacity. The last possible alternative uses vehicles with the smallest capacity.

Let us denote the following:

Q_{ijk} is the amount of freight from the *i*th request transported by vehicle type *j* when request T_i is met using alternative *k*
N_{ijk} is the number of type *j* vehicles that participate in meeting request T_i when request T_i uses alternative *k*.

It is clear that the total freight Q_i from transportation request T_i that is met using transportation alternative *k* equals the sum of the amount of freight of request T_i transported by individual types of vehicles that is,

$$\sum_{j=1}^{n} Q_{ijk} = Q_i \tag{2.105}$$

The capacity utilization (expressed as a percentage) λ_{ijk} of vehicle type j that takes part in meeting transportation request T_i using alternative k can be defined as

$$\lambda_{ijk} = \frac{Q_{ijk}}{C_j\, N_{ijk}\, n_i} 100\,[\%] \tag{2.106}$$

Let us denote by P_k the dispatcher's preference to use service alternative k to meet transportation request T_i. It is clear that

$$P_k = \frac{\sum_{j=1}^{n} N_{ijk}\, C_j\, n_i\, P_{ij}}{\sum_{j=1}^{n} N_{ijk}\, C_j\, n_i} \tag{2.107}$$

Corresponding dispatcher preference P_{ij} must be calculated for every type of vehicle j taking part in meeting transportation request T_i. Preference values P_{ij} are calculated based on approximate reasoning algorithms.

Based on relation (2.107), dispatcher preference to meet transportation request T_i with any of the possible service alternatives k can be calculated.

2.13.5. Heuristic algorithm to assign vehicles to transportation requests

The basic characteristics of every transportation request are the amount of freight that is to be transported and the distance to be traveled. Therefore, requests differ in terms of the volume of transportation work (expressed in ton-kilometers) to be executed, and in terms of the revenues and profits that every transportation request brings to the transportation company. It was also emphasized in our previous remarks that a company might have long-term cooperation with some clients, while other clients request the transportation company's services from time to time. Therefore, some transportation requests can be treated as being "more important," or "specially important requests," having "absolute priority in being carried out," and so on. All of this indicates that before assigning vehicles to transportation requests, the requests must first be sorted. The requests can be sorted in descending order by number of ton-kilometers that would be realized if the request were carried out, in descending order of the amount of freight in each request, in descending order of the requests' "importance" or in some other way. The manner in which the requests are sorted depends on the company's overall transportation policy. Milosavljevic et al. (1996) assumed that sorting of the transportation requests is made before vehicles are assigned to transportation requests.

The heuristic algorithm of assigning vehicles to transportation requests developed by Milosavljevic et al. (1996) consists of the following steps:

Step 1: Denote by i the index of transportation requests. Let $i = 1$.

Step 2: Generate all possible alternatives to meet transportation request T_i.

Step 3: Denote by $k(i)$ the index of possible alternatives to meet transportation request T_i. Let $k(i) = 1$.

Step 4: Analyze alternative $k(i)$. If available resources (number of available vehicles of a specific type) allow for alternative $k(i)$, go to Step 5. Otherwise go to Step 7.

Step 5: Determine the preference for every type of vehicle that takes part in implementing alternative $k(i)$ using an approximate reasoning by max-min composition.

Step 6: Calculate the dispatcher's preference to use alternative $k(i)$ to meet transportation request T_i. Use relation (2.107) to calculate this preference.

Step 7: Should there be any uninvestigated alternatives, increase the index alternative value by 1 ($k(i) = k(i)+1$) and go to Step 4. Otherwise, go to Step 8.

Step 8: Should none of the potential alternatives be possible owing to a lack of resources, transportation request T_i cannot be met. The final value of the dispatcher's preference (when there is at least one alternative possible) equals the maximum value of the calculated preferences of the considered alternatives. In this case, transportation request T_i is met by the alternative that corresponds to the maximum preference value.

Step 9: Decrease the number of available vehicles for the types of vehicle that took part in meeting transportation request T_i by the number of vehicles engaged in meeting the request.

Step 10: If any transportation requests have not been considered, increase the index by i ($i = i+1$) and return to Step 2.

2.13.6. Numerical example

The developed algorithm was tested on a fleet of vehicles containing three different types of vehicle. Capacity per type of vehicle and their respective number in the fleet are: Q_1 = 4.4t (N_1 = 48 vehicles), Q_2 =7.0t (N_2 = 49 vehicles), Q_3 = 14t (N_3 = 42 vehicles).

Table 2.19 presents the characteristics of the set of seventy eight transport requests to be met.

As can be seen from *Table 2.19*, each of the seventy eight transportation requests is characterized by amount of freight Q_i and distance D_i. The transportation work undertaken by the transportation company could be expressed in ton-kilometers (tkm). Based on the characteristics of the transportation requests, it is easy to calculate that the total number of the ton-kilometers to be carried out by the transportation company equals

$$\sum_{i=1}^{78} Q_i\ D_i = 177{,}570.3 \text{ tkm} \tag{2.108}$$

The quality of the solution obtained can be measured as the percentage of realized transportation requests and the percentage of realized ton-kilometers. Since the transportation company's profit directly depends on the number of effected ton-kilometers, it was decided that the quality of the solution obtained should be judged on the basis of the total number of realized ton-kilometers. The solutions obtained from the developed model were compared with those reached by an experienced dispatcher. Let us consider the following four ways of assigning vehicles to transportation requests:

1. An experienced dispatcher assigned vehicles to the transportation requests. The dispatcher was not given any instructions regarding the manner in which the assignments should be made.
2. An experienced dispatcher assigned vehicles to the transportation requests. The dispatcher was asked to assign only one type of vehicle to each transportation request.
3. Vehicles were assigned to transportation requests based on the developed algorithm, with only one type of vehicle being assigned to each transportation request.
4. Before assigning vehicles, the transportation requests were sorted by descending order of ton-kilometers. Vehicles were assigned to transportation requests using the developed algorithm, to the effect that

one or several different types of vehicle took part in meeting each request.

Table 2.19. Characteristics of 78 transport requests to be met

Request number	Request amount of freight (tons)	The distance (km)	Daily number of trips by one vehicle	Request number	Request amount of freight (tons)	The distance (km)	Daily number of trips by one vehicle
1.	22.0	42.0	2	40.	11.0	180.0	1
2.	3.0	25.0	4	41.	13.0	12.0	5
3.	7.0	138.0	1	42.	28.0	198.0	1
4.	39.0	280.0	1	43.	34.0	265.0	1
5.	6.0	75.0	2	44.	52.0	140.0	1
6.	17.0	189.0	1	45.	2.0	180.0	1
7.	5.0	45.0	2	46.	1.5	17.0	5
8.	21.0	110.0	1	47.	3.0	29.0	3
9.	8.0	180.0	1	48.	67.0	270.0	1
10.	27.0	42.0	2	49.	1.0	87.0	2
11.	43.0	197.0	1	50.	1.7	195.0	1
12.	2.0	317.0	1	51.	5.0	49.0	2
13.	6.0	180.0	1	52.	8.0	165.0	1
14.	16.0	78.0	2	53.	12.0	87.0	2
15.	25.0	78.0	2	54.	28.0	65.0	2
16.	34.0	57.0	2	55.	24.0	29.0	3
17.	23.0	57.0	2	56.	21.0	12.0	5
18.	12.0	129.0	1	57.	17.0	369.0	1
19.	9.0	32.0	3	58.	19.0	100.0	2
20.	21.0	21.0	4	59.	17.0	120.0	1
21.	7.0	180.0	1	60.	18.0	140.0	1
22.	7.0	87.0	3	61.	31.0	190.0	1
23.	4.0	49.0	2	62.	3.0	120.0	1
24.	26.0	127.0	1	63.	8.0	108.0	2
25.	22.0	240.0	1	64.	4.0	140.0	1
26.	19.0	220.0	1	65.	3.0	17.0	5
27.	14.0	100.0	2	66.	9.0	98.0	2
28.	15.0	121.0	1	67.	4.4	78.0	2
29.	38.0	27.0	4	68.	4.4	78.0	2
30.	41.0	129.0	1	69.	4.2	112.0	1
31.	8.0	160.0	1	70.	3.5	5.0	6
32.	9.0	180.0	1	71.	27.0	15.0	5
33.	16.0	70.0	2	72.	12.0	5.0	6
34.	21.0	161.0	1	73.	7.5	98.0	2
35.	32.0	180.0	1	74.	18.7	210.0	1
36.	42.0	120.0	1	75.	6.5	180.0	1
37.	16.0	132.0	1	76.	21.0	600.0	1
38.	12.0	12.0	5	77.	13.5	120.0	1
39.	9.0	27.0	4	78.	4.9	120.0	1

The results obtained are shown in *Table 2.20.*

Table 2.20. Comparison of the total number of ton-kilometers realized for the four different ways of assigning vehicles to transportation requests

Possible ways of assigning vehicles to transportation requests	The amount of time needed to assign vehicles to planned transportation requests	The total number of realized ton-kilometers	The percentage of realized ton-kilometers
I	2 hr. 30 min.	163,821	92.26 %
II	2 hr. 15 min.	154,866	87.15 %
III	40 sec.	152,727	86.01 %
IV	40 sec.	170,157	95.83 %

The developed model shows indisputable advantages compared to the dispatcher, particularly concerning the amount of time needed to assign vehicles to planned transportation requests. It might also be noted that the model sufficiently imitates the work of an experienced dispatcher. Using the model, it is possible to achieve results that are equal to or greater than the results achieved by an experienced dispatcher. Testing a large number of dispatchers and testing the model on a large number of different examples would confirm whether the model gives better results than the dispatcher in every situation.

Chapter 3.

A Fuzzy Mathematical Programming Approach to Transportation

3.1. THE BASIC PREMISES OF FUZZY MATHEMATICAL PROGRAMMING

In the past three decades, linear, nonlinear, dynamic, integer and multicriteria programming have been used very successfully to solve various traffic and transportation problems. Typical problems include assigning traffic in a network, distributing personnel to jobs, distributing transport facilities to carry out planned schedules, and planning fleet development and transportation-location tasks. Their solutions can be found using one of the mathematical programming methods.

In most of the real-world decision-making problems the input data are not always known precisely or the information is not available regarding certain input parameters that are part of a mathematical model. There is often uncertainty surrounding specific costs, the number of vehicles that can be employed to carry out a transport task, the number of passengers that want to travel between certain pairs of nodes in a network, and so on. The formulation of a linear programming lacks flexibility in dealing with imprecise input data. Therefore, in recent years this type of problem has been approached with fuzzy optimization techniques.

Thus, the possibilities must be considered of resolving certain linear or dynamic programming problems when some of the parameters in the model are fuzzy numbers. Hamacher et al. (1978) claim that in contrast to the traditional model of linear programming fuzzy linear programming is not characterized by a unique model. Depending on the nature of fuziness, fuzzy numbers can represent coefficients in an objective function and/or set of constraints. Fuzziness might also appear in the very formulation of the constraints (for example, the requirement that quantity x is "approximately less than" quantity y is not "precise" enough).

During the decision-making process in a fuzzy environment fuzzy objectives, fuzzy constraints and fuzzy decisions, represented through fuzzy

D. Teodorovič et al., *Traffic Control and Transport Planning:*

sets with corresponding membership functions, are considered. The decision is simply defined as a selection of sets of feasible solutions or merely one solution, which simultaneously satisfies the fuzzy objectives and fuzzy constraints. The logical "and" by which the fuzzy sets are connected corresponds to the operation of their intersection (Bellman and Zadeh, 1970). When the concept of a decision is defined in this way, objectives and constraints are of equal importance. That is the basis for the statement claiming the identity of the roles of objectives and constraints in the formulation of a decision making process in a fuzzy environment.

Let us assume that in the maximization problem there is one or more objective functions. We denote by **G** the set that represents the fuzzy domain of the set of objective functions. We denote by **C** the fuzzy set that represents the fuzzy domain of the set of constraints. Let fuzzy sets **G** and **C** be defined over set X. According to Bellman and Zadeh, the fuzzy domain of the solution is characterized by fuzzy set **D** whose membership function reads

$$\mu_{\mathbf{D}}(x) = \min \{\mu_{\mathbf{G}}(x), \mu_{\mathbf{C}}(x)\} \tag{3.1}$$

We note that fuzzy set **D** is the intersection of fuzzy sets **G** and **C**. In other words, fuzzy set **D** is the set of solutions to the problem that satisfy both the set of objective functions and the set of constraints.

The objective function that is, the goal we have set is expressed by the fuzzy set **G**. Let this goal be expressed by the desire that "x is considerably greater than b." We also assume that the model has a constraint of the type "x is considerably less than a." This constraint is quantified by fuzzy set C (*Figure 3.1*). When choosing the final solution to this problem, the set of solutions that has the highest grade of membership in set **D** is most often chosen. In other words, the set of final solutions $\mathbf{A_f}$ is defined as

$$\mathbf{A_f} = \{x_f \mid \mu_{\mathbf{D}}(x_f) > \mu_{\mathbf{D}}(x)\} \tag{3.2}$$

In the example shown in *Figure 3.1*, the set of final solutions has only one solution, x_f.

The membership function of fuzzy set **D** is noted in *Figure 3.1* by a solid line. As already mentioned, fuzzy set **D** is the set of all solutions to the problem that satisfy both the objective function and constraint that have been set. The optimal solution is that with the maximum degree of the membership function.

When solving a linear programming transportation problem precise information on transportation costs, travel time, amount of load, demand, and supply often do not exist (Teodorovic, 1994). The input data are most commonly obtained by prediction or estimation. A satisficing solution need not necessarily imply minimum transportation costs. In fact, in the majority of cases it is sufficient if costs remain within a reasonable range that is, if they are "reasonable." This type of problem is solved by fuzzy optimization techniques.

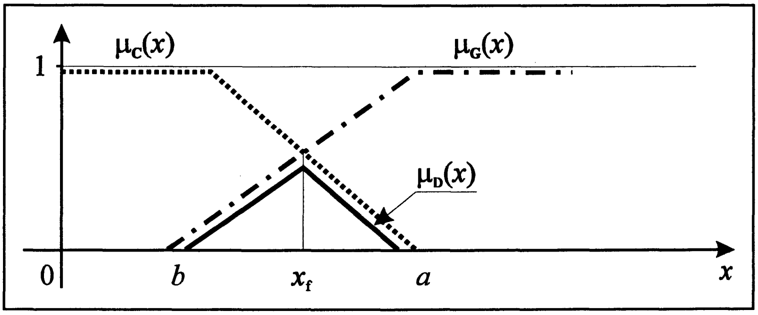

Figure 3.1. Membership functions of fuzzy sets **G**, **C**, and **D**

Most commonly, in dealing with such type of problems the following procedure is employed: first, the conventional transportation problem is solved; then, by introducing maximum levels of tolerated violations fuzziness is described along with the importance of the objectives and constraints.

Perincherry and Kikuchi (1990), when solving a transshipment problem, describe demand and supply through fuzzy numbers. To satisfy the fuzzy demand of "approximately D tons of load," they define a fuzzy set as "more than D tons of load," which is equivalent to the expression "satisfied demand." They consider the supply "adequate" and the obtained costs "reasonable." They claim that the traditional programming formulation lacks flexibility in dealing with imprecise input information.

Kikuchi et al. (1991) formulated and solved fuzzy linear programming transportation problems with vagueness and ambiguity. In the former case they considered fuzziness in the objective, fuzziness in the quantity of supply, and fuzziness in demand. The authors have done a thorough sensitivity analysis. They concluded that the fuzzy approach allows a very useful sensitivity analysis of different levels of "minimum" cost objective and suitable levels of supply and demand. In the latter case they considered fuzziness in the unit transportation cost mainly due to the uncertainty of travel time. They used the algorithm proposed by Tanaka and Asai (1984) to solve the problem. An example for minimizing the cost of transporting soil during highway construction is presented as a numerical example.

Tzeng and Teng (1990) claimed that in an increasingly complicated society investment planning for transportation infrastructure is conducted in a fuzzy environment. They applied a fuzzy multiobjective nonlinear method in a multistage hierarchy structure.

Bit et al. (1992) presented an application of fuzzy linear programming to a multiobjective transportation problem. Using the fuzzy programming method to solve the multiobjective transportation problem with K objective functions they obtained K nondominated solutions and an optimal compromise solution. Since the existing interactive algorithms give more than K nondominated solutions and the decision maker has to determine a compromise solution from the set of nondominated solutions, they concluded that the fuzzy linear programming method is a more suitable method for the multiobjective transportation problem.

Bhattacharya et al. (1992) used a fuzzy goal programming to locate a single facility on a plane bounded by convex region. They simultaneously considered three objectives: maximization of the minimum distances from the facility to the demand points, minimization of the maximum distances from the facility to the demand points, and minimization of the sum of all transportation costs. They supposed that the aspiration levels for objectives are not precisely given that is, they are fuzzy in nature.

Czyzak and Zak (1995) formulated a model of an urban transportation system using fuzzy multiobjective mathematical programming technique. The imprecision of data about riding time, the number of trams or buses available in the system and their capacities, the number of passengers to carry in 1 hour on a certain route, the cost for vehicle-kilometer for a certain mode, is modeled by L-R type fuzzy numbers. The model generates various assignments of available vehicles to the particular routes of transportation network. It allows changes in vehicle allocation that result in cost savings, better utilization of the fleet, and a higher standard for passengers' trips.

3.2. FUZZY LINEAR PROGRAMMING

In linear programming formulation, both the objective functions and the constraints are linear functions of decision variables. The general linear programming problem is most often formulated in the following way:

$$\max \mathrm{F} = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{3.3}$$

subject to

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1 \tag{3.4}$$

$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq b_2 \tag{3.5}$$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq b_m \tag{3.6}$$

Our task is to find the optimal solution (x_1^*, x_2^*, ..., x_n^*) that satisfies all the constraints and whose objective function F reaches a maximum value. Depending of the present fuzziness, various types of fuzzy linear programming problems can be formulated and solved. For example, the precise enough information is often not available regarding the coefficients appearing in the objective functions and/or constraints. For example, coefficient c_1 could be "approximately 5," coefficient b_2 "approximately 7," a_{31} "approximately 10," and so on. Likewise, the constraints are not necessarily "hard" or crisp.

The application of fuzzy optimization techniques to linear programming problems with one or more objective functions is first demonstrated by Zimmermann (1976, 1978). As a basic advantage of the fuzzy optimization technique Zimmermann emphasizes the fact that the decision maker does not have to precisely formulate the problem. This type of fuzzy programming, that is, the introduction of "fuzzy constraints" and "fuzzy objectives" into the conventional mathematical programming, is called "mathematical programming with vagueness" or "flexible programming" (Iniuguchi et al. 1990). Fuzzy objectives and fuzzy constraints represent the decision maker's satisficing set and their membership grades correspond to his/her degree of satisfaction. Inuiguchi et al. call a mathematical programming with fuzzy coefficients a "mathematical programming with ambiguity" or "possibilistic programming." The fuzzy coefficients represent the fuzzy regions where the coefficients possibly take place, and these regions are regarded as the possibility distributions. Tanaka and Asai (1984) solved a fuzzy linear programming problem under consideration of the ambiguity of parameters. They showed that the solution can be obtained through iterative use of a linear programming technique. Tanaka et al. (1986) noticed that the solution of a fuzzy linear programming problem depends strongly on the fuzziness of coefficients: the smaller the fuzziness of coefficient becomes, the more satisfactory solution is expected to be obtained. They discussed the tradeoff between a cost for investigating a fuzzy coefficient and the increase of satisfaction for the decision maker.

Kabbara (1982) offers various possible modeling of a linear programming problem without a solution by fuzzy sets. He claims: "The fuzzy optimization model provides a method, with the aid of fuzzy set theory, to formulate optimization problems without a solution. This can be done by accepting tolerances in the realization of the objective function and/or one or all of the constraints." When a multiobjective problem is being solved, it is usually assumed that all the objectives are equally important, which is rather unrealistic. To express the relative importance attached to the objectives each objective can be assigned a weight coefficient. The importance of an objective in fuzzy multiobjective decision making problems is determined by maximum tolerances in violations. According to Kabbara (1982) maximum levels of tolerances describe relative importance of the objectives.

The following sections will explain possible approaches to solving the fuzzy linear programming problems. Detailed analysis can be found in Zimmermann (1987).

3.3. SOLVING A FUZZY LINEAR PROGRAMMING PROBLEM WHEN CONSTRAINT COEFFICIENTS ARE FUZZY

We will use the following example to show the solution to a fuzzy linear programming problem when the constraint coefficients are fuzzy. Let us consider the following linear programming problem:

$$\max \mathrm{F} = c_1x_1 + c_2x_2 \tag{3.7}$$

subject to

$$a_1x_1 + a_2x_2 \leq b \tag{3.8}$$

$$e_1x_1 + e_2x_2 \leq r \tag{3.9}$$

Let the values of coefficients a_1, a_2 and b be only approximately known, that is, they are uncertain. In classical linear programming formulation, coefficients a_1, a_2, and b are real numbers. In our case, the nature of the problem makes these coefficients fuzzy numbers. Our constraint reads:

$$\mathbf{A}_1x_1 + \mathbf{A}_2x_2 \leq \mathbf{B} \tag{3.10}$$

Let us also assume that $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{B}$ are triangular fuzzy numbers that is, $\mathbf{A}_1 = (a_1{}^1, a_2{}^1, a_3{}^1)$, $\mathbf{A}_2 = (a_1{}^2, a_2{}^2, a_3{}^2)$, and $\mathbf{B} = (b_1, b_2, b_3)$. These fuzzy numbers are shown in *Figure 3.2*.

Multiplying fuzzy number $\mathbf{A}_1$ by constant x_1 gives triangular fuzzy number $\mathbf{A}_1x_1 = (a_1{}^1x_1, a_2{}^1x_1, a_3{}^1x_1)$. In the same vein, multiplying triangular fuzzy number $\mathbf{A}_2$ by constant x_2 gives triangular fuzzy number $\mathbf{A}_1 x_2 = (a_1{}^2 x_2, a_2{}^2 x_2, a_3{}^2 x_2)$.

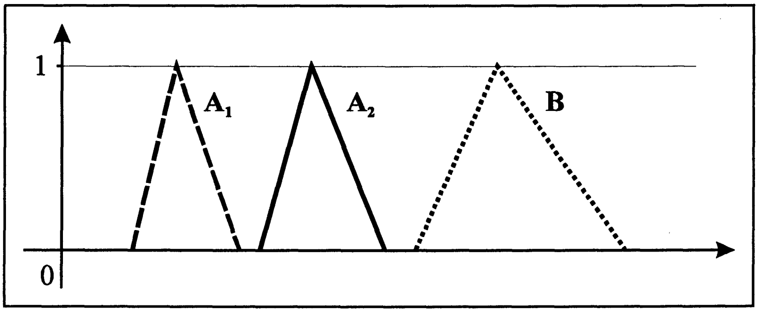

Figure 3.2. Fuzzy numbers $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{B}$

We denote by $\mathbf{D} = (d_1, d_2, d_3)$ the triangular fuzzy number obtained as the result of adding triangular fuzzy numbers $\mathbf{A}_1 x_1$ and $\mathbf{A}_2 x_2$. The left boundary, average value, and right boundary of fuzzy number $\mathbf{D}$ equal $d_1 = a_1{}^1 x_1 + a_1{}^2 x_2$, $d_2 = a_2{}^1 x_1 + a_2{}^2 x_2$, and $d_3 = a_3{}^1 x_1 + a_3{}^2 x_2$.

Figure 3.3 shows the membership functions of fuzzy number $\mathbf{D}$ and fuzzy number < $\mathbf{B}$ ("less than B"). Since b_2 is the average value of fuzzy number $\mathbf{B}$, the expression "less than B" means "less than approximately b_2."

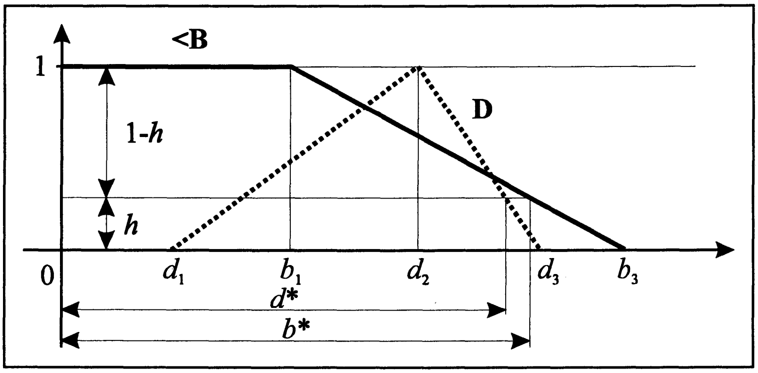

Figure 3.3. Membership functions of fuzzy numbers $\mathbf{D}$ and < $\mathbf{B}$

The other shapes of the membership function of the fuzzy number < $\mathbf{B}$ that are commonly used are shown in *Figure 3.4.*

The *Figure 3.3* also shows on the ordinate axis the level of satisfaction h that we wish to achieve. This level of satisfying the constraint can be achieved when the highest possible value of fuzzy number $\mathbf{D}$ for this level of satisfaction is less than or equal to the highest value of fuzzy number < $\mathbf{B}$ for this level of satisfaction. We denote by d^* and b^*, respectively, the highest possible values of fuzzy numbers **D** and **B** for level of satisfaction h. Based on the similarities of corresponding triangles, it is easy to show that $d^*=d_2+(1-h)(d_3-d_2)$ and $b^*=b_1+(1-h)(b_3-b_1)$.

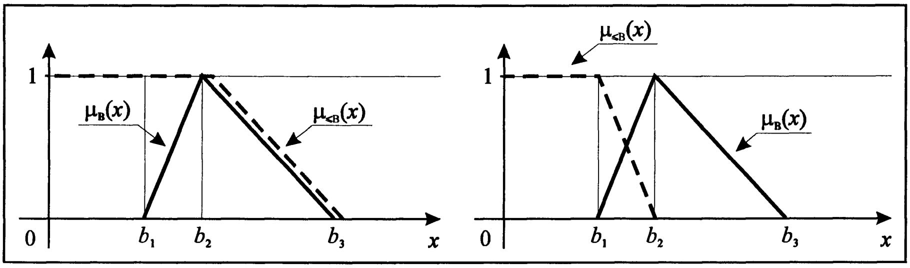

Figure 3.4. The other shapes of membership function of the fuzzy number < **B**

In order to satisfy the constraint with at least a level of satisfaction h, the following relation must be fulfilled:

$$d_2 + (1-h)\,(d_3-d_2) \leq b_1 + (1-h)\,(b_3-b_1) \tag{3.11}$$

Let us assume in our example that F $= c_1x_1 + c_2x_2$ represents the profit we wish to maximize. Let us also assume that we can estimate a "satisfactory profit." This estimate is approximate that is, let "satisfactory profit" be expressed by the triangular fuzzy number $\mathbf{F} = (f_1, f_2, f_3)$.

Triangular fuzzy number **F** and triangular fuzzy number > **F** are shown in *Figure 3.5*.

It is clear from *Figure 3.5* that a "profit greater than satisfactory" will be achieved with a level of satisfaction at least equal to h, if

$$c_1x_1 + c_2x_2 \geq f_1 + h\,(f_3-f_1) \tag{3.12}$$

In other words, our objective function has become a constraint, which agrees completely with Bellman and Zadeh (1970) whereby both objective functions and constraints in a fuzzy environment are treated in the same way. Since we have transformed the objective function into a constraint, the question arises of defining a new objective function.

We will naturally try to find a solution that maximizes the level of satisfying the objective function and constraint, h. In another words, in order to elicit the optimal decision that is, to determine the optimal solution (x_1^*, x_2^*) that satisfies both the objective and constraints by the maximum possible degree h, a fuzzy optimization principle is applied by which h is maximized (Zimmermann, 1976):

$$\max h \tag{3.13}$$

subject to

$$c_1x_1 + c_2x_2 \geq f_1 + h\,(f_3 - f_1) \tag{3.14}$$

$$d_2 + (1-h)\,(d_3 - d_2) \leq b_1 + (1-h)\,(b_3 - b_1) \tag{3.15}$$

$$e_1x_1 + e_2x_2 \leq r \tag{3.16}$$

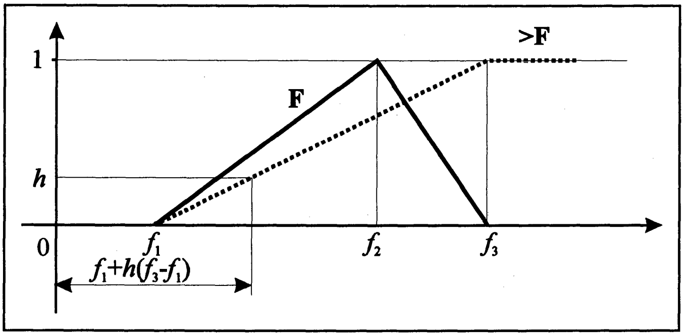

Figure 3.5. Triangular fuzzy number **F** representing "satisfactory profit" and triangular fuzzy number > **F** denoting the expression "profit greater than satisfactory"

Example 3.1. Consider the following fuzzy linear programming problem:

$$\max F = 2x_1 + 3x_2 \tag{3.17}$$

subject to

$$x_1 + 2x_2 \leq 8 \tag{3.18}$$

$$2x_1 + x_2 \leq 10 \tag{3.19}$$

$$-x_1 + x_2 \leq 2 \tag{3.20}$$

$$x_2 \leq 3,\ x_1 \geq 0,\ x_2 \geq 0 \tag{3.21}$$

We can see that some of the coefficients in the second constraint are fuzzy numbers. Let **2** and **10** be triangular fuzzy numbers that are subjectively expressed as **2** = (1.5, 2, 2.5), **10** = (8, 10, 12).

Applying the rules of fuzzy arithmetic, we can state that $\mathbf{2}x_1 + x_2$ is also a triangular fuzzy number. This fuzzy number equals

$$\mathbf{2}x_1 + x_2 = (1.5x_1 + x_2, 2x_1 + x_2, 2.5x_1 + x_2) \tag{3.22}$$

The membership function of fuzzy numbers $\mathbf{2}x_1 + x_2$, **10** and ≤ **10** are shown in *Figure 3.6.*

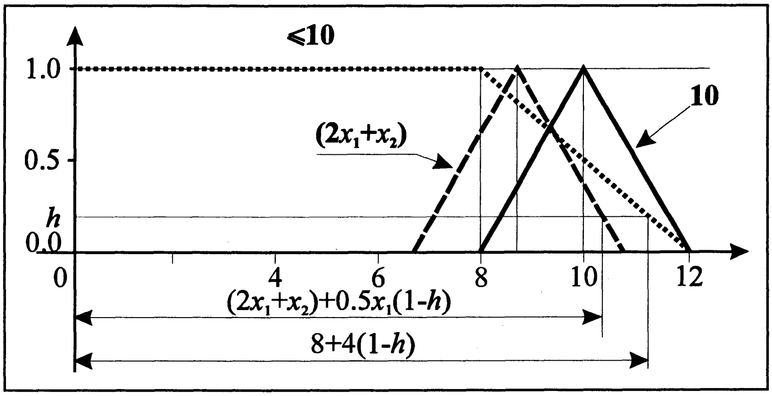

Figure 3.6. Membership functions of fuzzy numbers $\mathbf{2}x_1 + x_2$, **10**, and ≤ **10**

Bearing the previous discussion in mind, it is easily shown that the highest allowed value of fuzzy number $\mathbf{2}x_1 + x_2$ for level of satisfaction h equals $(2x_1 + x_2) + 0.5\,x_1(1 - h)$. At the same time, the highest allowed value of fuzzy number **10** for level of satisfaction h equals $8 + 4(1 - h)$. The second constraint reads

$$(2x_1 + x_2) + 0.5\,x_1(1 - h) \leq 8 + 4(1 - h) \tag{3.23}$$

Let our objective function represent profit that we wish to maximize. For randomly chosen values $x_1 = 3.5$ and $x_2 = 2$ that satisfy the given constraints, profit equals 13. We arbitrarily present "satisfactory profit" as triangular fuzzy number **13** = (12, 13, 14).

Figure 3.7 presents the membership functions of fuzzy number **13** and fuzzy number ≥ **13** referring to profit that is "greater than satisfactory profit."

Our original objective function is transformed into the following constraint:

$$2x_1 + 3x_2 \geq 12 + 2h \tag{3.24}$$

The nonlinear programming problem to be solved reads

$$\max h \tag{3.25}$$

subject to

$$2x_1 + 3x_2 \geq 12 + 2h \tag{3.26}$$

$$x_1 + 2x_2 \leq 8 \tag{3.27}$$

$$(2x_1 + x_2) + 0.5\, x_1(1 - h) \leq 8 + 4(1 - h) \tag{3.28}$$

$$-x_1 + x_2 \leq 2 \tag{3.29}$$

$$x_2 \leq 3, x_1 \geq 0, x_2 \geq 0 \tag{3.30}$$

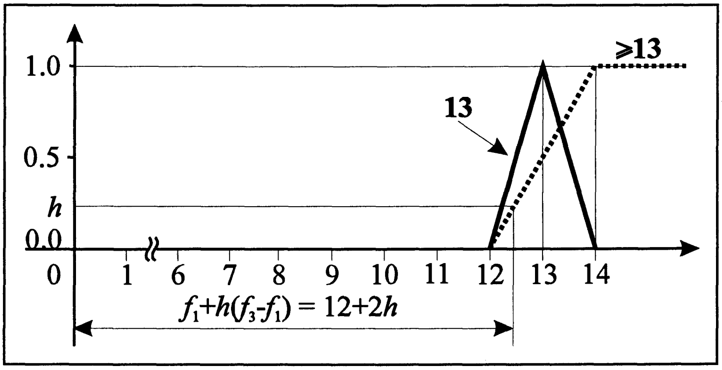

Figure 3.7. Membership functions of fuzzy numbers **13** and ≥ **13**

To solve this problem we used the iterative algorithm proposed by Tanaka and Asai (1984). The main idea is to determine a set of feasible solutions for an assumed initial value of h and then increase the value of h until feasible solutions exist. In order to chose the optimal solution (x_1^*, x_2^*) among the small feasible set (where h takes the maximal value) we maximized the original objective function subject to other constraints. The following solution was found $x_1^*= 3.08$, $x_2^*= 2.46$, which achieved a level of satisfaction $h = 0.75$. We increased an initial value h by 0.05.

3.4. SOLVING A FUZZY LINEAR PROGRAMMING PROBLEM WHEN OBJECTIVE FUNCTION COEFFICIENTS ARE FUZZY

In some linear programming problems, "fuzziness" appears in the objective function coefficients. As in the previous cases, this is most often due to insufficiently precise information regarding the values of some parameters. Let us note the following objective function:

$$\max \mathrm{F} = \mathbf{C}_1 x_1 + \mathbf{C}_2 x_2 + \dots + \mathbf{C}_n x_n$$

Let coefficients $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_n$ be triangular fuzzy numbers. Expression $\mathbf{C}_1 x_1 + \mathbf{C}_2 x_2 + \dots + \mathbf{C}_n x_n$ also represents a triangular fuzzy number. We will denote this number by $\mathbf{F} = (f_1, f_2, f_3)$. Values f_1, f_2, f_3 are determined based on the known left boundaries, c_{1i}, average values, c_{2i}, and right boundaries, c_{3i}, of fuzzy numbers $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_n$ using the rules of fuzzy arithmetic. These values equal

$$f_1 = \sum_{i=1}^{n} c_{1i} x_i \quad f_2 = \sum_{i=1}^{n} c_{2i} x_i \quad f_3 = \sum_{i=1}^{n} c_{3i} x_i$$

Figure 3.8 presents the membership function of fuzzy number **F**.

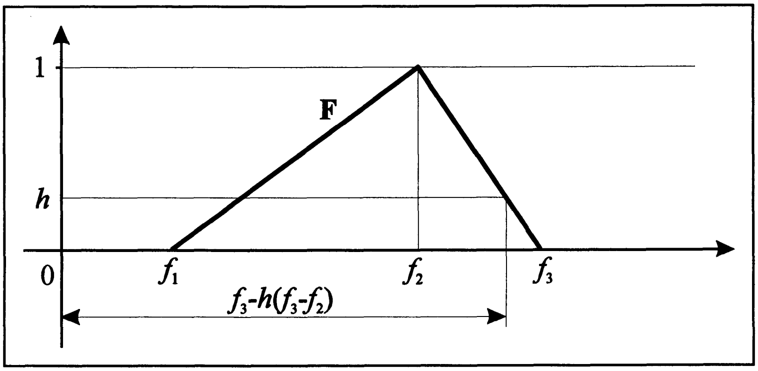

Figure 3.8. Membership function of fuzzy number **F** representing the objective function

Based on the similarities of corresponding triangles, it is easy to show that the highest value of the objective function for level of satisfaction h equals $f_3 - (f_3 - f_2)h$. It would be logical to try to “shift” this value as much to the right as possible. In other words, the initial objective function is transformed into the following objective function:

$$\max f_3 - (f_3 - f_2)h$$

In this case, the solution is obtained by previously defining the value of level of satisfaction h.

Example 3.2. Solve the following fuzzy linear programming problem:

$$\max \mathrm{F} = \mathbf{5}x_1 + \mathbf{6}x_2 \tag{3.31}$$

subject to

$$x_1 - 2x_2 \leq 4 \tag{3.32}$$

$$6x_1 + 7x_2 \leq 42 \tag{3.33}$$

$$x_1 \geq 0, x_2 \geq 0 \tag{3.34}$$

Note that fuzzy numbers **5** and **6** are coefficients of the objective function. Let these triangular fuzzy numbers equal $\mathbf{5} = (4, 5, 6)$ and $\mathbf{6} = (5, 6, 7)$.

Fuzzy number **F** equals

$$\mathbf{F} = (f_1, f_2, f_3) = \mathbf{5}x_1 + \mathbf{6}x_2 = (4x_1 + 5x_2,\ 5x_1 + 6x_2,\ 6x_1 + 7x_2) \tag{3.35}$$

The new nonlinear objective function reads

$$\max f_3 - (f_3 - f_2)h = \max 6x_1 + 7x_2 - (x_1 + x_2)h \tag{3.36}$$

Let us assume that we want to achieve level of satisfaction $h = 0.9$. The linear programming problem to be solved reads

$$\max \mathrm{G} = 5.1x_1 + 6.1x_2 \tag{3.37}$$

subject to

$$x_1 - 2x_2 \leq 4 \tag{3.38}$$

$$6x_1 + 7x_2 \leq 42 \tag{3.39}$$

$$x_1 \geq 0, x_2 \geq 0 \tag{3.40}$$

The solution was obtained: $x_1 = 5.895, x_2 = 0.946$, G = 34.895.

3.5. SOLVING A FUZZY LINEAR PROGRAMMING PROBLEM WHEN THE CONSTRAINTS ARE FUZZY

Constraints appear in linear programming problems due to the context of the problem being considered and require that one value be "greater than," "greater than or equal to," "equal to," "less than or equal to," or "less than" some other value. On the other hand, when an analyst is analyzing a problem and building a model, conversations with the user-decision maker often indicate that a certain value should be "approximately greater than," "about equal to," or "approximately less than" some other value. Such "relaxed" constraints result from the subjectivity that is present in the modeling process and from uncertainty regarding the values of some parameters.

Example 3.3. Let some fuzzy linear programming problem include a constraint of the form:

$$2x_1 + 3x_2 \underset{\sim}{<} 8 \tag{3.41}$$

This constraint results from the desire for the expression on the left side to be "approximately less than 8". An essential distinction should be made between the expression "approximately less than 8" and the expression less than "approximately 8." In the expression "approximately less than 8", the fuzziness appears in the constraint itself, while in the expression less than "approximately 8" the fuzziness is present in the constraint coefficient.

Example 3.4. Let us assume we have a constraint of the type

$$3x_1 < 18 \tag{3.42}$$

The linguistic interpretation of this constraint reads that $3x_1$ is "approximately less than 18". The expression "approximately less than 18" denotes the fact that $3x_1$ is essentially less than 18, which does not exclude the possibility that $3x_1$, in certain cases, could be a little bit greater than 18. Our expression "approximately less than" contains the readiness to accept a slight excess of the constraint, within a certain threshold of tolerance. The expression "$3x_1$ is approximately less than 18" can be represented by a fuzzy set. We will denote this set by **A**. When $x_1 = 2$, the grade of membership of number 2 in fuzzy set **A** is 1, for $3 \times 2 = 6 < 18$. Number 5's grade of membership in fuzzy set **A** is also 1, for $3 \times 5 = 15 < 18$. It is clear that with a rise in the value of variable x_1 (over number 6) there is a drop in the grade of membership in set **A**. *Figure 3.9* presents the membership function of set **A**, which is subjectively defined.

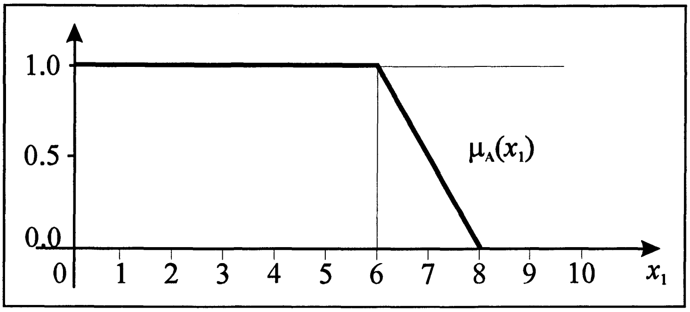

Figure 3.9. Membership function of fuzzy set **A**

It is clear from the picture that we are ready to accept for x_1 all values from the interval (6,8). This means that there is a certain tolerance interval that expresses our readiness to breach the given constraint. The membership function of fuzzy set **A** equals

$$\mu_A(3x_1) = \begin{cases} 1, & 3x_1 \le 18 \\ 1 - \dfrac{3x_1 - 18}{6}, & 18 \le 3x_1 \le 24 \\ 0, & 3x_1 \ge 24 \end{cases} \tag{3.43}$$

Let us assume that we have a constraint of the form:

$$\mathbf{A}x \underset{\sim}{<} \mathrm{B} \tag{3.44}$$

where $\mathbf{A} = (a_1, a_2, a_3)$ is a triangular fuzzy number.

The constraint represents the requirement for fuzzy number **A**x to be approximately less than number B. The expression "approximately less than" means that there is a tolerance interval (B, B + C) within which the set constraint can be breached. The value of constant C is defined subjectively. *Figure 3.10* presents this tolerance interval and fuzzy number **A**x. Based on the similarity of corresponding triangles, it can easily be shown that

$$a_2x + (1 - h)(a_3 - a_2) \leq \mathrm{B} + (1 - h)\mathrm{C} \qquad (3.45)$$

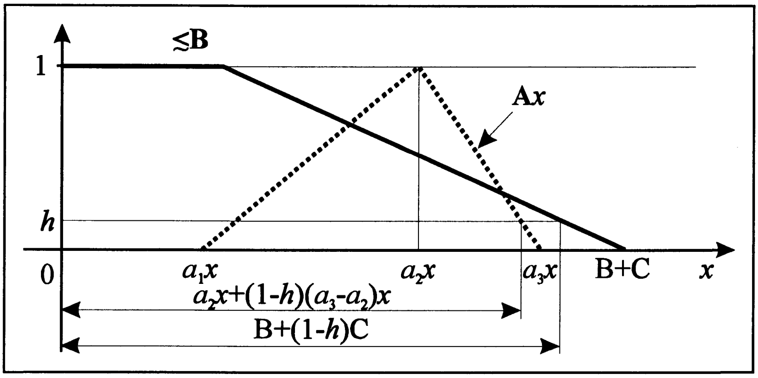

Figure 3.10. Constraint of the type **A**x < B

This inequality replaces considered constraint when solving the fuzzy linear programming problem.

3.6. AIRLINE NETWORK DESIGN BY FUZZY LINEAR PROGRAMMING

Designing an airline network is an extremely complex planning task. The chosen network shape and the flight frequency on individual links directly effect the business results of the air carrier and the quality of service provided to passengers. Thus, when choosing the network shape, both the interests of the airline carrier and those of the service users must be taken into consideration. The airline carrier is interested in carrying out the planned volume of traffic with the least possible number of aircraft. The carrier also wishes to generate the lowest possible operating costs and the highest annual aircraft flight time and load factor. On the other hand, passengers are interested in high flight frequencies, a large number of nonstop flights, small waiting time between flights at transfer points, and so on.

The uncertainty surrounding some input parameters complicates the choice of the airline network's shape. For example, often only approximate passenger demand is known between pairs of cities in a future period or only approximate operating costs are known on certain routes. Teodorovic et al. (1994) developed a model to design a network of airline routes and determine flight frequencies. The first part of their model uses the generalized Floyd algorithm and fuzzy logic to determine route candidates. The second part of the model, based on fuzzy linear programming, determines flight frequencies on the route candidates.

3.6.1. Statement of the problem

Many authors have considered the problem of designing a network of air routes and determining flight frequencies on an airline network. Different aspects of this problem were treated in the papers of Gordon and de Neufville (1973), Carter and Morlok (1975), Kirby (1980), Kanafani (1981), Kanafani and Ghobrial (1982, 1985), Pollack (1982), Ghobrial (1983), Schwieterman and Spencer (1986), Ghafouri and Lam (1986), Flynn and Ratick (1988), Teodorovic and Krcmar-Nozic (1989), and Chou (1990).

The basic input data when choosing an airline network's shape are the estimated passenger flows between individual pairs of cities. As mentioned earlier, it is assumed that we only know the approximate passenger flows between individual pairs of cities. In other words, all that we have available is an estimate that the annual number of passengers between city A and city B is, for example, "approximately 100,000 passengers," "between 100,000 and 120,000 passengers," and so on. The uncertainty regarding the number of passengers is clearly more pronounced when the estimated number refers to several years in advance.

Figure 3.11 presents n cities that are to be connected by air transportation.

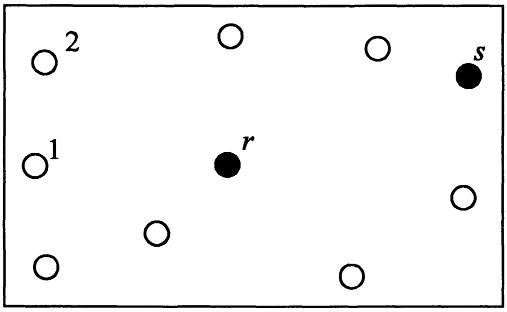

Figure 3.11. Cities that are to be connected by air transportation

The cities to be connected by air transportation are denoted by numbers 1, 2, 3, ..., *n*. Air transportation between cities i and j can be made by nonstop flights and/or by flights with one or more intermediate stops. Let the estimated annual passenger flows between individual pairs of cities be as follows:

$$F = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \cdots \\ n \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & \dots & n \end{array} \\ \begin{bmatrix} 0 & \mathbf{F}_{12} & \mathbf{F}_{13} & \cdots & \mathbf{F}_{1n} \\ \mathbf{F}_{21} & 0 & \mathbf{F}_{23} & \cdots & \mathbf{F}_{2n} \\ \mathbf{F}_{31} & \mathbf{F}_{32} & 0 & \cdots & \mathbf{F}_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{F}_{n1} & \mathbf{F}_{n2} & \mathbf{F}_{n3} & \cdots & 0 \end{bmatrix} \end{array} \tag{3.46}$$

where $\mathbf{F}_{ij}$ is the estimated annual number of passengers from city *i* to city *j* ($i = 1, 2, ..., n; j = 1, 2, ..., n$).

The estimated numbers of passengers are denoted in bold letters. If, for example, the estimated number of passengers from city *i* to city *j* is **100,000**, this means that we estimate that the number of passengers from city *i* to city *j* is “approximately 100,000 passengers.”

The problem discussed by Teodorovic et al. (1994) paper can be defined as follows: For known estimated annual numbers of passengers between individual pairs of cities, determine the shape of a network of nonstop flights and flights with intermediate stops and their corresponding flight frequencies.

3.6.2. Proposed solution to the problem

The problem of determining the shape of air transportation routes and corresponding flight frequencies is combinatorial by nature. It should also be underlined that when air transportation is to be established among a large number of cities, the dimensions of the problem can become very large. In order to decrease these dimensions, Teodorovic et al. (1994) decided to first choose from among the very large number of possible routes, a group of routes to be the route candidates for air traffic. In the second step, flight frequencies are determined exclusively for the set of route candidates.

Trips between two cities may be made by direct nonstop flights or by flights with one or more intermediate stops. When there are intermediate stops, a distinction must be made between through service and connecting service. Connecting service is when the passenger makes a physical change of plane. Direct nonstop flights are naturally preferable from the passengers' point of view. On the other hand, in order to achieve the best economic results, the carrier is forced to combine passengers from several markets, and to include flights with intermediate cities in the network. All direct nonstop flights should of course belong to the set of route candidates. A question naturally arises regarding how to determine the route candidates that contain one or more intermediate stops. Every potential route is characterized by total distance, total travel time, and total number of intermediate stops. It was decided to evaluate or compare route candidates primarily according to the total distance of the routes. Both carriers and passengers are interested in the shortest possible routes. The carrier's interest is to decrease direct operating costs, while the passengers' interest is based on the desire for the highest level of service. From the passengers' viewpoint, total travel time might be an even better criterion on which to compare different routes. However, in the airline network design phase, the future routing plan and airline schedule are not known. In other words, in the global airline network design phase when we do not know the future airline schedule, it is not possible to know whether passengers who have an intermediate stop will physically change planes or not. Thus, in this phase of work, the total travel time of individual routes is not known with enough precision. This fact influenced the choice of total distance as the prevailing criterion with which to compare different routes. The number of intermediate stops on a route is certainly an extremely important criterion when evaluating individual routes. The following types of question logically arise: From the carrier's and/or passengers' viewpoint, is it "better" to have a route that is 1,000 miles with one intermediate stop or 800 miles with two intermediate stops? In considering route length and the number of intermediate stops, Swan (1979) notes that "both stops and extra miles have significant costs, so practical options for a path neither stop too often nor go too far around." The fuzziness present in such a statement indicates that the fuzzy logic might be applied to further evaluate routes based on route length and the number of intermediate stops.

Teodorovic et al. (1994) chose the route candidates in two steps. In the first step, the "shortest" routes are chosen from the set of all possible routes. Choosing the set of "shortest" routes means that for every pair of cities with transport demand between them, *k* shortest routes are first determined. The *k* shortest routes are determined using the generalized Floyd algorithm. In the second step, the set of route candidates is narrowed even further. The route candidates obtained using the generalized Floyd algorithm are ranked based on their length and number of intermediate stops. This ranking is done using a model based on the principles of approximate reasoning.

3.6.3. Choosing air traffic route candidates

The "classical" Floyd algorithm enables the shortest distance to be determined between all pairs of nodes in the network. The generalized Floyd algorithm is a generalization of the "classical" algorithm; for a previously specified k, it enables the k shortest paths to be found between all pairs of nodes. By applying the generalized Floyd algorithm to every pair of cities which have transport demand between them, the k shortest route candidates are determined. Clearly, the first shortest path between any two cities will always be a direct nonstop flight between the two cities. The second, third, fourth, ... shortest paths are route candidates with one or more intermediate points. As already pointed out, the route candidates obtained from the generalized Floyd algorithm are ranked in the next step based on total length and total number of intermediate points. It is intuitively clear to us that it would be "much better" to choose "shorter" routes with a "smaller" number of intermediate points. In other words, we have a "higher" preference to choose a "shorter" route with a "smaller" number of intermediate points. A question logically arises about determining the parameters that influence the strength of our preference. Before answering this question, let us introduce into the discussion the route length index β_{rs} which is defined as follows:

$$\beta_{rs} = \frac{d_{rs}}{D_{rs}} \tag{3.47}$$

where d_{rs} is route length from city r to city s for a direct nonstop flight, and D_{rs} is actual length of the route considered from city r to city s.

It is clear that "shorter" routes correspond to "larger" route length index values and vice versa. The route length index (β) and total number of intermediate points along some route (n) could be the basic parameters that influence the strength of our preference regarding the choice of individual routes. The strength of our preference is indicated by the preference index that is found within interval [0,1], to the effect that a rise in our preference is accompanied by a rise in the preference index number.

Let us denote by **RB**, **RM**, and **RS**, respectively, fuzzy sets "big route length coefficient," "medium route length coefficient," and "small route length coefficient," and by **NB**, **NM**, and **NS** fuzzy sets "big number of intermediate points," "medium number of intermediate points," and "small number of intermediate points."

An approximate reasoning algorithm was developed to determine the strength of the preference linked to the choice of route candidates. The approximate reasoning algorithm that determines the preference index of individual route candidates is comprised of the following rules:

Rule 1: If route length coefficient β is **RB** and number of intermediate points n is **NB**, then preference index p is **Medium**,
or
Rule 2: If route length coefficient β is **RB** and number of intermediate points n is **NM** or **NS**, then preference index p is **High**,
or
Rule 3: If route length coefficient β is **RM** and number of intermediate points n is **NB**, then preference index p is **Low**,
or
Rule 4: If route length coefficient β is **RM** and number of intermediate points n is **NM**, then preference index p is **Medium**,
or
Rule 5: If route length coefficient β is **RM** and number of intermediate points n is **NS**, then preference index p is **High**,
or
Rule 6: If route length coefficient β is **RS** and number of intermediate points n is **ANY**, then preference index p is **Low**.

The membership functions of fuzzy sets **RB**, **RM**, **RS**, **NB**, **NM**, and **NS** are shown in *Figure 3.12*.

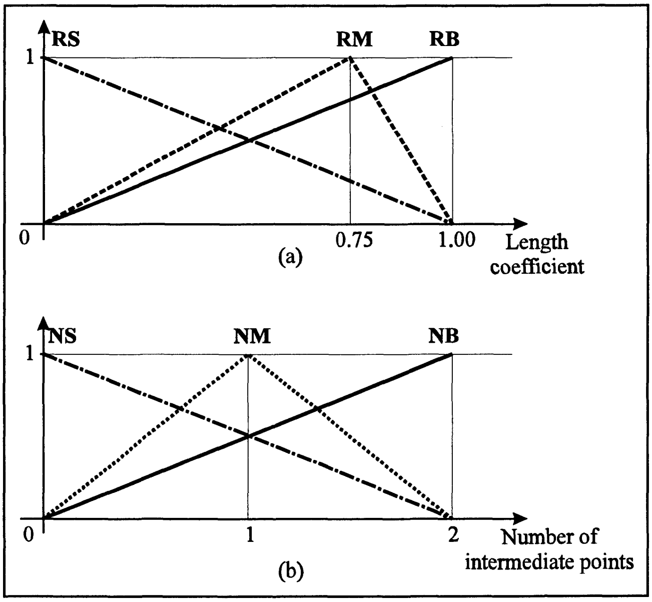

Figure 3.12. (a) Membership functions of fuzzy sets **RB**, **RM**, and **RS**, (b) Membership functions of fuzzy sets **RS**, **NB**, **NM**, and **NS**

The membership functions of the fuzzy sets "Low preference," "Medium preference," and "High preference" are shown in *Figure 3.13*.

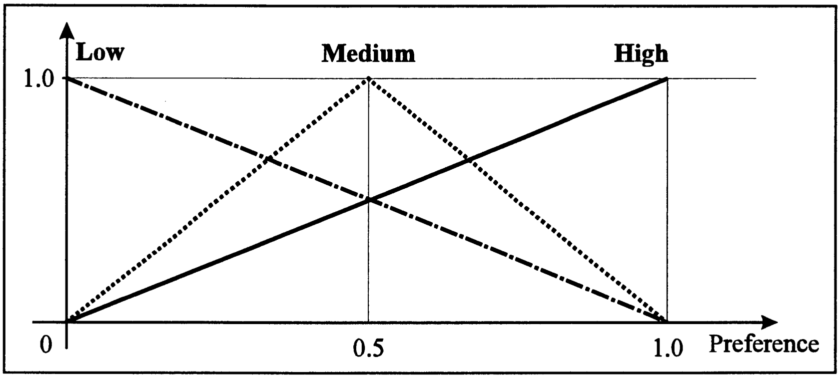

Figure 3.13. Membership functions of fuzzy sets "low," "medium," and "high" preference

In this manner, using approximate reasoning by max-min composition, all considered routes were characterized by a corresponding preference index value.

3.6.4. Determining flight frequencies on route candidates

The higher the preference index value, the better the route's chance of becoming a final route candidate. Our further discussion will only consider those route candidates that satisfy the inequality

$$p_i \geq p^* \tag{3.48}$$

where p_i is the calculated preference index value along the ith route, and p^* is an arbitrarily preassigned boundary preference index value.

The set of chosen route candidates comprises the airline network. This airline network G(N, A) could be represented by a direct graph. This graph includes a set of consecutively numbered nodes N, as well as a set of consecutively numbered links A.

Let us denote by R the set of origin cities ($R \subseteq N$). Let us also denote by S the set of destination cities ($S \subseteq N$). These two sets are not mutually exclusive ($R \cap S \neq \varnothing$).

Each origin-destination city pair r - s is connected by a set of paths (route - candidates) through the network. Let us denote this set by P_{rs} ($r \in R$, $s \in S$). The set of all paths in the network is denoted by P. Let us also introduce the following notation:

f_{rs} total annual number of passengers from city r to city s
f_{rsp} annual number of passengers traveling from city r to city s along path p ($p \in P_{rs}$)
f_a total annual number of passengers on link a
d_a length of link a in miles
m_{rsp} number of intermediate points along path p from city r to city s
N_{rsp} annual flight frequency from city r to city s along path p
N_a annual flight frequency on link a
t_{rsp} average tariff to transport one passenger from city r to city s along path p.

Let us also introduce into the discussion binary variable $\delta_{a,p}^{r,s}$ defined as follows:

$$\delta_{a,p}^{r,s} = \begin{cases} 1, & \text{if link } a \text{ is a part of path } p \text{ from city } r \text{ to city } s \\ 0, & \text{otherwise} \end{cases} \tag{3.49}$$

It is clear that

$$f_a = \sum_r \sum_s \sum_p \delta_{a,p}^{r,s} f_{rsp} \tag{3.50}$$

In other words, the flow on each link is the sum of the flows on all paths going through that link.

In order to be able to define the objective function that represents the air carrier's profit on the network, we must first take a look at the air carrier's costs. We will consider air carrier costs by using the methodology proposed by Kanafani and Ghobrial. Kanafani and Ghobrial divided air carrier costs into direct operating costs (DOC) and indirect operation costs (IOC). Direct operating costs for a flight over a link with length d_a is given by

$$\mathrm{DOC}_a' = \alpha + \beta d_a \tag{3.51}$$

where is DOC_a' is the total direct operating cost in dollars, d_a is the stage length in miles, α, and β are constant parameters specific to an aircraft type.

Total annual direct operating costs on link a equal

$$\mathrm{DOC}_a = (\alpha + \beta d_a) N_a \tag{3.52}$$

Kanafani and Ghobrial assumed “that a unit of indirect operating cost per passenger can be used as a constant in the cost model.” In other words, total annual indirect operating costs equal

$$\mathrm{IOC}_a = h\, f_a \tag{3.53}$$

where IOC_a are total annual indirect operating costs on link a, and h is the unit “handling” cost per passengers in dollars.

Kanafani and Ghobrial also divided the cost of passenger travel time into two components. The first is the line haul travel time component. The line haul travel time component L equals

$$L = V_1 f(\gamma + \rho R + \Delta) \tag{3.54}$$

where V_1 is a unit of time-cost transformation reflecting the perceived money cost of line haul travel time, f is the number of passengers carried during a given time period, R is the stage length in miles, γ, and ρ are travel time function parameters, and Δ is a component of travel time referring to airport time.

For a nonstop flight consisting of only one link a, the component of travel time referring to airport time is represented by ground time, which we will denote by Δ_1. For a flight consisting of several links, the component of travel time referring to airport time is represented by stopover time or transfer time. It has already been mentioned that during the global design of the airline network, the future routing plan and airline schedule are not known. Thus, in this phase it is not known whether passengers will stay in the aircraft at some intermediate stop or will physically change planes to continue their trip. Although there are differences in average stopover and transfer times, it was decided for Δ_2 to represent the component of travel time referring to airport time in the multi-link case. In other words, for a link that represents part of some route (and is not the first link on the route), the component of travel time referring to airport time equals Δ_2, regardless of whether the passengers stayed in the same plane or physically changed aircraft before flying that link.

The second component of passenger travel time is the schedule delay component. In order to simplify the model to determine flight frequencies on route candidates, Teodorovic et al. (1994) decided for the cost of passenger travel time to consist of only line haul travel time.

Total annual costs of the air carrier flying airline network G(N,A) equal

$$C_1 = \sum_a (\alpha + \beta d_a) N_a + \sum_a h\, f_a \tag{3.55}$$

Let us now introduce into the discussion binary variable y_{rsp}, which is defined as follows:

$$y_{rsp} = \begin{cases} 1, & \text{if } m_{rsp} = 0 \\ 0, & \text{otherwise} \end{cases} \qquad (3.56)$$

When a route has no intermediate points (a direct nonstop flight), binary variable y_{rsp} equals 1. When a route has one or more intermediate points, the corresponding binary variable y_{rsp} equals zero.

Let us also introduce into the discussion binary variable $\rho_{a,p}^{r,s}$ defined as follows:

$$\rho_{a,p}^{r,s} = \begin{cases} 1, & \text{if the link is the first link on path} \\ & p \text{ leading from node } r \text{ to node } s \\ 0, & \text{otherwise} \end{cases} \qquad (3.57)$$

Total annual passenger travel time costs on airline network G(N,A) equal

$$C_2 = \sum_a V_1 f_a(\gamma + \rho d_a + L_a) \qquad (3.58)$$

where

$$L_a = \sum_r \sum_s \sum_p \delta_{a,p}^{r,s}\left(\left(y_{rsp} + \left(1 - y_{rsp}\right)\rho_{a,p}^{r,s}\right)\Delta_1 + \left(1 - y_{rsp}\right)\left(1 - \rho_{a,p}^{r,s}\right)\Delta_2\right) \qquad (3.59)$$

Total annual costs on airline network G(N,A) equal

$$\mathrm{C} = \mathrm{C}_1 + \mathrm{C}_2 \qquad (3.60)$$

It was assumed that the average tariff to transport one passenger between city r and city s the on path p is constant and independent of path p. In other words, it is assumed that

$$t_{rst} = t_{rs}, \qquad p \in \mathrm{P}_{rs} \qquad (r \in \mathrm{R},\ s \in \mathrm{S}) \qquad (3.61)$$

where t_{rs} is the average tariff to transport one passenger from city r to city s.

We will also introduce the assumption that enough transport capacities are available to satisfy passenger travel demand and the assumption that travel demand is inelastic as long as there are transport capacities to satisfy it. Thus, since demand is inelastic, and since the average transport tariff between any two cities is independent of the path taken, it can be concluded that the total revenues that the carrier earns on the airline network are constant. In this case, the problem of determining flight frequencies that maximize air carrier profit becomes the problem of determining flight frequencies that minimize costs.

Let us denote by η_a the load factor on link a. This load factor equals

$$\eta_a = \frac{f_a}{nN_a} \tag{3.62}$$

where n is the number of seats on the aircraft making the flight.

Flight frequencies on the route candidates that minimize the total costs are obtained by solving the following problem (P_1):

$$\min F = \sum_a (\alpha + \beta d_a) N_a + \sum_a h n \eta_a N_a + \sum_a V_1 n \eta_a N_a (\gamma + \rho d_a + L_a) \tag{3.63}$$

$$n\eta_a N_a - \sum_r \sum_s \sum_p \delta_{a,p}^{r,s} f_{rsp} \geq 0, \qquad \forall a \tag{3.64}$$

$$\sum_p f_{rsp} = f_{rs}, \qquad \forall (r,s), \qquad p \in P_{rs} \tag{3.65}$$

$$N_a = \sum_r \sum_s \sum_p \delta_{a,p}^{r,s} N_{rsp}, \qquad \forall a \tag{3.66}$$

$$\sum_p y_{rsp} N_{rsp} \geq L'_{rs}, \qquad \forall (r,s), \qquad p \in P_{rs} \tag{3.67}$$

$$\sum_p N_{rsp} \geq L''_{rs}, \qquad \forall (r,s), \qquad p \in P_{rs} \tag{3.68}$$

$$\text{all} \quad N_a, N_{rsp}, f_{rsp} \geq 0 \tag{3.69}$$

where L_{rs}' is least number of direct flights that must be made from city r to city s during the year, and L_{rs}'' is least number of flights that must be made from city r to city s during the year.

Let us explain in more detail the constraints defined by relations (3.64) to (3.69). Based on relation (3.64), we conclude that the offered number of seats on each link must be at least equal to the annual number of passengers on all paths that contain that link. Relation (3.65) defines the condition that the sum of the passengers on individual flights from some city r to some city s equals the total number of passengers traveling between these two cities. This condition must be met for all pairs of cities in the network where air traffic is planned to be established. Relation (3.66) indicates that flight frequency on any link equals the sum of the flight frequencies on the paths that the link is part of. Relation (3.67) determines the least allowed flight frequencies for direct nonstop flights between any two cities in the network. From the passengers' viewpoint, the "highest-quality" routes are those without intermediate stops (direct nonstop flights). Relation (3.67) defines the least number of flights that must be made on a "highest-quality" route. Relation (368) determines the least allowed flight frequencies between any two cities in the network. For example, relation (3.68) might require that at least 500 flights a year must be flown between some two cities, while at the same time relation (3.67) might require that at least 100 of these 500 be direct nonstop flights. Relations (3.67) and (3.68) build the quality of air traffic service offered to passengers into the model. Relation (3.69) requires all variables to be nonnegative. Since relations (3.63) to (3.69) attempt to determine annual flight frequencies, the model does not have any constraints built into it indicating that flight frequencies must be integer variables.

The problem defined in relations (3.63) to (3.69) is a linear programming problem and can be successfully solved using a program package designed to solve linear programming problems. On the other hand, we are most often unable to precisely determine the basic input data - the number of passengers between individual pairs of cities. In other words, there is uncertainty when determining of the number of passengers between individual pairs of cities. Uncertainty is also present when determining air carrier costs for one or two years in advance. Teodorovic et al. (1994) treated the estimated numbers of passengers between individual pairs of cities, as well as the carrier's operating costs as fuzzy numbers. The minimum annual flight frequencies L_{rs}' and L_{rs}'' between pairs of cities are also fuzzy numbers. We would note that the fuzziness present in the number of passengers and carrier costs primarily arise from the fact that it is impossible to determine with absolute precision the number of passengers and carrier costs for one or two years in advance. The minimum annual flight frequencies L_{rs}' and L_{rs}'' can also be treated as deterministic values. When the constraints defined by relations (3.64) and (3.69) are built into the model, the analyst-decision maker often wants the minimum annual flight frequency to be "at least somewhere around 200 flights," for example, or "approximately 200 flights," rather than exactly "200 flights." In other words, there is often no extremely firm position regarding certain constraints in the model. It would be better to say that we have a certain feeling about the value of the minimum annual flight frequencies. It can be noted that in the problem defined by relations (3.64) to (3.69), fuzziness appears due to either the infeasibility of exactly predicting certain values, or by the lack of a firm position regarding some other values.

Let us consider the following constraint:

$$\sum_p f_{rsp} = \mathbf{f}_{rs} \tag{3.70}$$

The estimated number of passengers $\mathbf{f}_{rs}$ from city r to city s is the triangular fuzzy number shown in *Figure 3.14*.

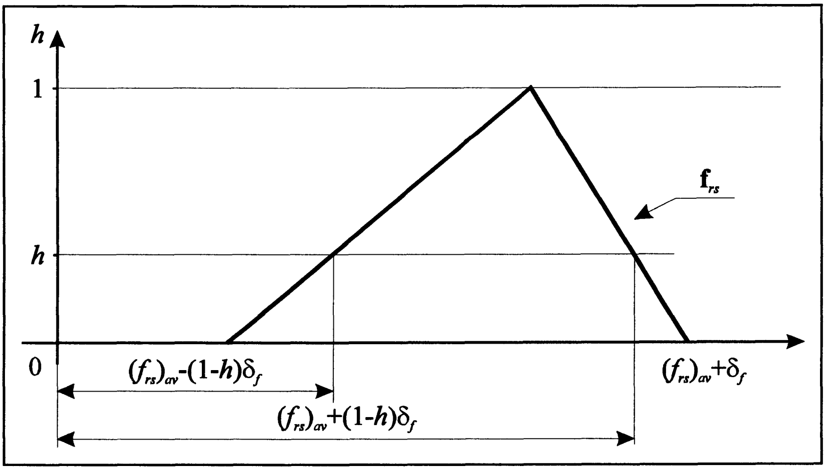

Figure 3.14. Estimated number of passengers $\mathbf{f}_{rs}$ from city r to city s

It is clear from the figure (based on the similarity of the triangles) that $\sum_p f_{rsp}$ will be equal to $\mathbf{f}_{rs}$ with a level of satisfaction at least equal to h if the following is satisfied:

$$(f_{rs})_{av} - (1-h)\delta_f \le \sum_p f_{rsp} \le (f_{rs})_{av} + (1-h)\delta_f \tag{3.71}$$

Let us also consider the following constraint:

$$\sum_{p} N_{rsp} \geq \mathbf{L}''_{rs} \tag{3.72}$$

Fuzzy numbers $\mathbf{L}_{rs}''$ and $\geq \mathbf{L}_{rs}''$ are shown in *Figure 3.15*.

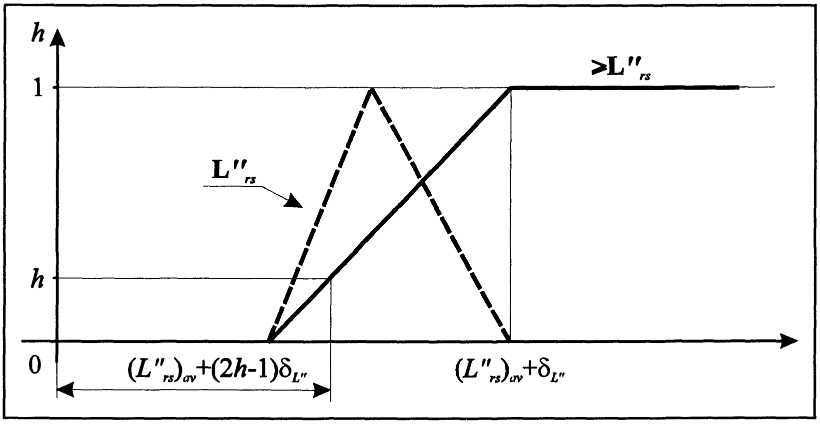

Figure 3.15. Fuzzy numbers $\mathbf{L}_{rs}''$ and $\geq \mathbf{L}_{rs}'$

It is easily shown (similarity of the triangles) that $\sum_{p} N_{rsp}$ will be greater or equal to $\mathbf{L}_{rs}''$ if the following is satisfied:

$$\sum_{p} N_{rsp} \geq \left(L''_{rs}\right)_{av} + (2h-1)\delta_{L''} \tag{3.73}$$

In the same manner, $\sum_{p} y_{rsp} N_{rsp}$ will be greater than or equal to $\mathbf{L}_{rs}'$ if the following is satisfied:

$$\sum_{p} y_{rsp} N_{rsp} \geq \left(L'_{rs}\right)_{av} + (2h-1)\delta_{L'} \tag{3.74}$$

Let us introduce "acceptable costs" into the discussion. In other words, instead of minimizing costs, we will try to generate acceptable costs with a level of satisfaction at least equal to h. We define *acceptable costs* as the triangular fuzzy number **T** (*Figure 3.16*).

"Acceptable costs" are arbitrarily defined. The value corresponding to a grade of membership equal to one, T_{av}, denotes the costs obtained by solving problem P_1. The value of δ_T is also arbitrarily defined. Since objective functions and constraints are treated in the same manner in a fuzzy environment, the objective function (minimizing costs) is transformed into the following constraint:

$$\sum_a \left[\alpha + \beta d_a + hn\eta_a + V_1 n\eta_a\left(\gamma + \rho d_a + L_a\right)\right] \le T - \left(2h - 1\right)\delta_T \tag{3.75}$$

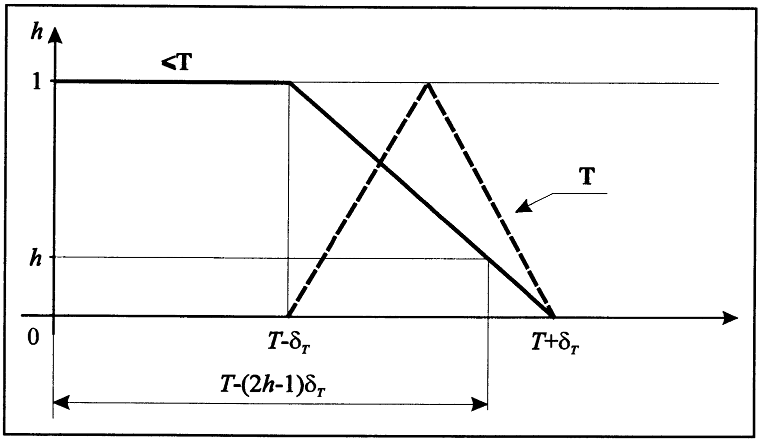

Figure 3.16. "Acceptable costs" **T** and costs equal to or less than "acceptable costs" (≤ **T**)

The new objective function reflects our desire to maximize the level of satisfaction h. The problem (P_1) can be transformed into the following problem (P_2):

$$\max h \tag{3.76}$$

subject to

$$\sum_a \left[\alpha + \beta d_a + hn\eta_a + V_1 n\eta_a\left(\gamma + \rho d_a + L_a\right)\right] N_a \le T - \left(2h - 1\right)\delta_T \tag{3.77}$$

$$n\eta_a N_a - \sum_r \sum_s \sum_p \delta_{a,p}^{r,s} f_{rsp} \ge 0, \qquad \forall a \tag{3.78}$$

$$\sum_p f_{rsp} \le \left(f_{rs}\right)_{av} + \left(1 - h\right)\delta_f, \qquad \forall\,(r, s), \quad p \in P_{rs} \tag{3.79}$$

$$\sum_{p} f_{rsp} \geq \left(f_{rs}\right)_{av} - (1-h)\delta_{f}, \qquad \forall\,(r,s), \quad p \in \mathrm{P}_{rs} \tag{3.80}$$

$$N_{a} = \sum_{r}\sum_{s}\sum_{p} \delta_{a,p}^{r,s} N_{rsp}, \qquad \forall\,(r,s) \tag{3.81}$$

$$\sum_{p} N_{rsp} \geq \left(L_{rs}^{''}\right)_{av} + (2h-1)\delta_{L}^{''}, \qquad \forall (r,s), \qquad p \in \mathrm{P}_{rs} \tag{3.82}$$

$$\sum_{p} y_{rsp} N_{rsp} \geq \left(L_{rs}^{'}\right)_{av} + (2h-1)\delta_{L}^{'}, \qquad \forall\,(r,s), \qquad p \in \mathrm{P}_{rs} \tag{3.83}$$

$$\text{all} \quad N_{a}, N_{rsp}, f_{rsp} \geq 0 \tag{3.84}$$

The developed model was tested on several different numerical examples. After applying the first part of the model (based on fuzzy logic), the route candidate network was obtained. The second part of the model (based on fuzzy linear programming) enabled the calculation of corresponding flight frequencies.

As already mentioned, the model was tested on several different numerical examples. We will illustrate the results on a hypothetical numerical example. The cities to be linked by air traffic are shown in *Figure 3.17*:

We assume that the hypothetical aircraft to be used has a capacity $n = 200$ seats and we wish to achieve an average load factor $\eta_p = 0.7$. The effective capacity of the aircraft is $n\eta_p = 140$. *Table 3.1* gives the list of nonstop flights, costs per nonstop flight, estimated number of passengers (in thousands), minimum annual flight frequencies and minimum number of nonstop flights. The estimated number of passengers, minimum annual flight frequencies, and minimum number of nonstop flights are given in the form of triangular fuzzy numbers.

By applying the model of approximate reasoning, Teodorovic et al. (1994) got the route candidates shown in *Table 3.2*. These route candidates correspond to preference index values that are greater than 0.5. All route candidates whose preference index value was less than 0.5 were rejected. The preference index value of 0.5 used to choose the route candidates was chosen arbitrarily.

Table 3.1. Pairs of cities, costs, estimated number of passengers, minimum annual flight frequencies, and minimum number of nonstop flights

Link *a*	Costs per flight (× 100 $)	City pair	Estimated number of passengers (× 1000)	Minimum annual flight frequencies	Minimum number of nonstop flights
1-2	(70,77,84)	1-2	(90,100,110)	(630,700,770)	(315,350,385)
1-3	(88,97,106)	1-3	(144,160,176)	(900,1000,1100)	(630,700,770)
1-4	(128,138,148)	1-4	(342,380,418)	(1800,2000,2200)	(1350,1500,1650)
1-5	(119,129,139)	1-5	(63,70,77)	(315,350,385)	(315,350,385)
1-6	(158,178,198)	1-6	(450,500,550)	(2250,2500,2750)	(1620,1800,1980)
1-7	(195,215,235)	1-7	(180,200,220)	(900,1000,1100)	(630,700,770)
2-1	(73,83,93)	2-1	(90,100,110)	(630,700,770)	(315,350,385)
2-3	(60,68,76)	2-3	(81,90,99)	(630,700,770)	(315,350,385)
2-4	(70,78,86)	2-4	(405,450,495)	(2250,2500,2750)	(1350,1500,1650)
2-5	(64,71,78)	2-5	(360,400,440)	(2250,2500,2750)	(1350,1500,1650)
2-6	(108,118,128)	2-6	(540,600,660)	(2970,3300,3630)	(1620,1800,1980)
2-7	(153,163,173)	2-7	(423,470,517)	(2250,2500,2750)	(1620,1800,1980)
3-1	(92,102,112)	3-1	(144,160,176)	(900,1000,1100)	(630,700,770)
3-2	(60,68,76)	3-2	(81,90,99)	(630,700,770)	(315,350,385)
3-4	(95,105,115)	3-4	(67,75,83)	(630,700,770)	(315,350,385)
3-5	(58,64,70)	3-5	(85,95,105)	(630,700,770)	(315,350,385)
3-6	(118,128,138)	3-6	(90,100,110)	(630,700,770)	(315,350,385)
3-7	(159,169,179)	3-7	(468,520,572)	(2700,3000,3300)	(1620,1800,1980)
4-1	(133,143,153)	4-1	(342,380,418)	(1800,2000,2200)	(1350,1500,1650)
4-2	(69,79,89)	4-2	(405,450,495)	(2250,2500,2750)	(1350,1500,1650)
4-3	(100,110,120)	4-3	(67,75,83)	(630,700,770)	(315,350,385)
4-5	(62,69,76)	4-5	(351,390,429)	(1800,2000,2200)	(1350,1500,1650)
4-6	(55,61,67)	4-6	(225,250,275)	(1350,1500,1650)	(630,700,770)
4-7	(94,104,114)	4-7	(252,280,308)	(1350,1500,1650)	(900,1000,1100)
5-1	(126,136,146)	5-1	(63,70,77)	(315,350,385)	(315,350,385)
5-2	(63,73,83)	5-2	(360,400,440)	(2250,2500,2750)	(1350,1500,1650)
5-3	(61,71,81)	5-3	(85,95,105)	(630,700,770)	(315,350,385)
5-4	(62,70,78)	5-4	(351,390,429)	(1800,2000,2200)	(1350,1500,1650)
5-6	(72,82,92)	5-6	(261,290,319)	(1350,1500,1650)	(900,1000,1100)
5-7	(118,128,138)	5-7	(180,200,220)	(900,1000,1100)	(630,700,770)
6-1	(166,186,206)	6-1	(450,500,550)	(2250,2500,2750)	(1620,1800,1980)
6-2	(108,118,128)	6-2	(540,600,660)	(2970,3300,3630)	(1620,1800,1980)
6-3	(125,135,145)	6-3	(90,100,110)	(630,700,770)	(315,350,365)
6-4	(54,60,66)	6-4	(225,250,275)	(1350,1500,1650)	(630,700,770)
6-5	(72,82,92)	2-5	(261,290,319)	(1350,1500,1650)	(900,1000,1100)
6-7	(54,60,66)	6-7	(270,300,330)	(1620,1800,1980)	(900,1000,1100)
7-1	(190,210,230)	7-1	(180,200,220)	(900,1000,1100)	(630,700,770)
7-2	(147,157,167)	7-2	(423,470,517)	(2250,2500,2750)	(1620,1800,1980)
7-3	(158,168,178)	7-3	(468,520,572)	(2700,3000,3300)	(1620,1800,1980)
7-4	(88,98,108)	7-4	(342,380,418)	(1350,1500,1650)	(900,1000,1100)
7-5	(110,120,130)	7-5	(180,200,220)	(900,1000,1100)	(630,700,770)
7-6	(50,56,62)	7-6	(270,300,330)	(1620,1800,1980)	(900,1000,1100)

After solving problem (P_1), total annual costs (in million dollars) equal $T_{av} = 986$. Value $\delta_T = 100$ was arbitrarily determined. "Acceptable" annual costs **T** (in million dollars) are **T** = (886, 986, 1086).

Solving the fuzzy linear programming problem, annual flight frequencies were obtained (*Table 3.2*).

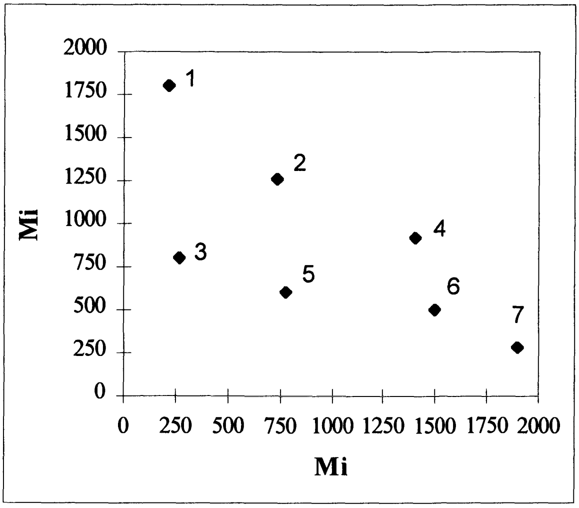

Figure 3.17. Cities to be linked by air routes

These annual frequencies correspond to a level of satisfaction of $h = 0.65$. By shifting the "acceptable costs" to the left, the level of satisfaction is decreased. *Figure 3.18* shows the dependence of the level of satisfaction h on the value of T_{av}.

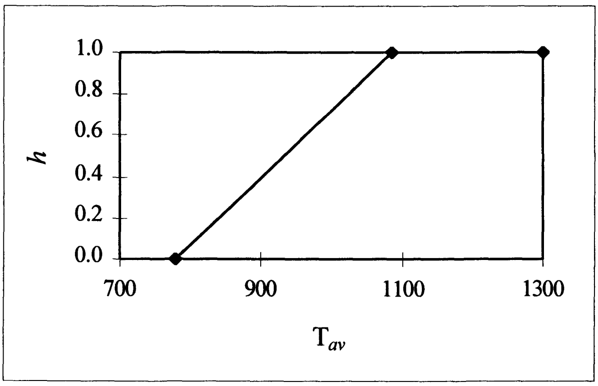

Figure 3.18. Dependence of the level of satisfaction h on T_{av}

It can be concluded that values T_{av} found within interval [789, 1092] correspond to the level of satisfaction values from interval [0.5, 1]. If we are prepared to accept the fact that all constraints are not completely satisfied, we can considerably decrease carrier costs.

Table 3.2. Annual flight frequencies

Route	Annual flight frequencies	Route	Annual flight frequencies	Route	Annual flight frequencies
1-2	690	5-2-1	0	3-7	3150
1-3-2	32	5-3-1	0	3-5-7	0
1-3	1072	5-2	2433	3-6-7	0
1-2-3	0	5-3-2	0	3-5-6-7	0
1-5-3	0	5-4-2	145	4-1	2621
1-4	2475	5-3	722	4-2-1	0
1-2-4	0	5-4	2062	4-5-1	0
1-5-4	0	5-6-4	0	4-2	2577
1-5	483	5-6	2000	4-5-2	0
1-2-5	0	5-4-6	0	4-3	518
1-3-5	0	5-7	1343	4-5-3	0
1-6	3235	5-6-7	0	6-3	690
1-2-6	0	5-4-7	0	6-5-3	32
1-4-6	146	6-1	3448	6-4-3	0
1-2-4-6	0	6-2-1	0	6-4	1547
1-7	1380	6-4-1	0	6-5	1755
1-2-7	0	6-4-2-1	0	6-4-5	0
1-6-7	214	6-2	4138	6-7	1856
1-2-6-7	0	6-4-2	178	7-1	1380
2-1	690	6-5-2	0	7-2-1	0
2-3-1	32	2-4-6	0	7-6-1	0
2-3	385	2-5-6	0	7-6-2-1	0
2-5-3	338	2-7	3241	7-2	3241
2-4	2577	2-6-7	0	7-6-2	0
2-5-4	0	2-4-7	178	7-4-2	0
2-5	2421	3-1	1072	7-3	3150
2-3-5	0	3-2-1	0	7-5-3	0
2-4-5	156	3-5-1	0	7-6-3	0
2-6	4138	3-2	397	7-6-5-3	0
4-2-3	205	3-5-2	326	7-4	1931
4-6-3	0	3-4	486	7-6-4	0
4-5	2062	3-5-4	44	7-5-4	441
4-6-5	0	3-2-4	193	7-5	1375
4-6	1547	3-6-4	0	7-6-5	214
4-7	1753	3-5	722	7-4-5	0
4-6-7	0	3-6	690	7-6	1856
4-5-7	473	3-5-6	0		
5-1	483	3-4-6	32		

Every pair (h, T_{av}) corresponds to a certain frequency plan. In this manner, a large number of different frequency plans are generated for the decision maker. The choice of a specific frequency plan is conditioned by the decision maker's willingness to accept the fact that some constraints are not completely satisfied, with a view to decreasing total costs.

3.7. AIRLINE NETWORK SEAT INVENTORY CONTROL: A FUZZY MATHEMATICAL PROGRAMMING APPROACH

Demand for airline services stems from different activities: business trips (planned in advance or sudden departures to important business meetings), vacations, visits to friends, and numerous other private reasons result in passengers making certain demands of the airline carrier. Passengers choose a flight and tariff and inform the air carrier (either by telephone or by going to a travel agent). Requests are usually made several days or weeks before the departure of the flight chosen by the passenger. In a smaller number of cases, passengers appear at the airport right before departure and make a last-minute request ("go-show" passengers).

When a request is accepted, a seat for the passenger is reserved on the desired flight. In most cases, making a reservation does not mean purchasing the ticket as well. A certain number of passengers cancel their reservations. Also, a certain number of passengers with valid reservations and purchased tickets do not appear for departure. These are "no-show" passengers. The reasons for no-show passengers might be subjective (a last-minute change of plans) or objective (the flights that passengers used prior to the flight in question were canceled or delayed). In most cases, passengers with valid reservations and a purchased ticket appear for departure. In certain instances, it happens that some of these passengers cannot get a seat on their planned flight. This is the problem of overbooking, which creates considerable problems for the air carrier. The air carrier gives some sort of compensation to overbooked passengers, We would note, nonetheless, that most passengers with a purchased ticket make their trips as planned.

Since air traffic in the United States was deregulated in 1978, launching a wave of deregulation throughout the world, air carriers have offered passengers on the same flight a large number of different tariffs. This is due to the air carriers' desire to achieve the greatest possible load factor and total revenue. It is clear that an air carrier's total revenue depends both on the tariffs that are offered and on the manner in which available seats are allocated to passenger itinerary-fare class combinations. Efforts to develop and implement seat inventory control models have their economic justification. As noted by Rothstein (1985), a 1% increase in load factor has the result of increasing the air carrier's annual revenue by millions of dollars. In the past three decades, a large number of models have been developed that treat different aspects of the seat inventory control problem.

The assumption that total demand for some future flight has a normal distribution was used by many authors. On the other hand, Mak (1992) stresses, "there is no guarantee, however, on the accuracy of using historical data as a forecasting tool." Belobaba (1987a, 1987b) underscores the need for "human intervention" when solving airline seat inventory problems.

Teodorovic (1998) decided to estimate future demand using statistical data from the past, the experience and intuition of the analyst, and the number of already accepted reservations in different passenger itinerary-fare class combinations. Future demand in different passenger itinerary-fare class combinations was treated in his paper as a fuzzy number.

The seat inventory control problem can be solved for multiple-fare classes on a single leg, multiple- fare classes for flights with several legs, for a single-fare class on multiple itineraries, or for multiple-fare classes and multiple itineraries.

3.7.1. Statement of the problem

Let us briefly show Teodorovic's model (1998) to solve the airline network seat inventory control problem. The airline carrier's network can be represented by graph G(N,A), whereby N is the set of airports in the network, and A is the set of flight legs (a leg is a nonstop flight between two cities or a link flown by an aircraft at a specific time). Let us introduce sets R and S, which are defined as follows:

$R \subseteq N$ is set of origins
$S \subseteq N$ is set of destinations

It is clear that $R \cap S \neq \varnothing$. The air carrier designs an airline schedule that is characterized by flight frequencies and departure times. Passengers adjust their travel plans to the proposed airline schedule. This results in the creation of passenger itineraries through the network.

Let us denote by PI the set of passenger itineraries. Passengers can pay various fares for traveling on their itinerary. The most expensive tariff is usually paid by business class passengers, who as a rule make their travel plans much later than passengers making private trips.

Let us denote by I the set of passenger itinerary-fare class combinations. Every passenger itinerary-fare class combination $i \in I$ is characterized by departure airport $r_i \in R$, landing airport $s_i \in S$, path taken by trip $p_i \in PI$, departure time t_{d_i}, arrival time t_{a_i}, and fare paid f_i. In other words, the group of six parameters (r_i, s_i, p_i, t_{d_i}, t_{a_i}, f_i) completely describes passenger itinerary-fare class combination $i \in \mathrm{I}$.

Every flight leg is also characterized by a departure airport, landing airport, departure time, landing time, and remaining number of available seats on the leg. Let C_a denote the available seats on leg $a \in \mathrm{A}$.

Let us introduce binary variable δ_{ia}, which is defined as follows:

$$\delta_{ia} = \begin{cases} 1, & \text{if leg } a \in \mathrm{A} \text{ is part of passenger itinerery} \\ & \text{fare} - \text{class combination } i \in \mathrm{I} \\ 0, & \text{otherwise} \end{cases} \tag{3.85}$$

One flight leg can be part of a large number of different passenger itinerary-fare class combinations (*Figure 3.19*).

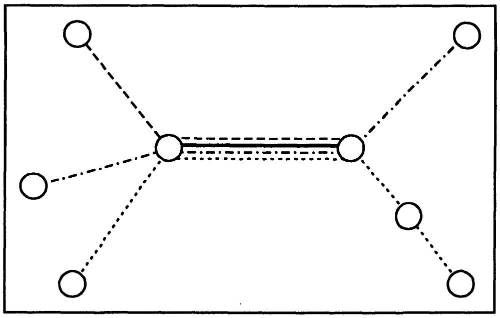

Figure 3.19. Flight leg which is part of a large number of different passenger itinerary-fare class combinations

As previously mentioned, every flight and every passenger itinerary has a variety of fare classes. The seats available in different fare classes can be allocated in a variety of ways. The simplest and most researched case is when there are distinct and separate inventories for each fare class. For example, if we have 100 seats available on a leg and four different fare classes, we can decide to allocate fourty seats to the first fare class, twenty five to the second, twenty five to the third, and ten to the fourth fare class. The nested reservation system is a more complex case; its basic characteristic is that passenger requests that bring in higher revenue are not refused as long as there are seats available in fare classes that bring in lower revenue. Nested reservation systems were studied in detail by Belobaba (1987a, 1987b), who indicated their advantages compared to the reservation system with separate inventories for each fare class. We would note that Belobaba's paper considered primarily the problem of seat allocation in the case of multiple-fare classes on a single leg. Nested reservation system were also studied by Curry (1990) and Smith et al. (1992). As opposed to classical leg based nested seat inventory control system, Curry (1990) defines nest as "a collection of fare-classes, all for the same O-D, which are ordered in a strict hierarchy. A higher ranked fare may take a seat from the inventory of any lower ranked fares in the same nest, but not from the inventory of the fares outside the nest." It surely is possible to have single, several, or all fare classes for any O-D within the certain nest. The development and implementation of certain nested seat inventory control system depends on the computer reservation system of the carrier as well as on the whole array of real-life constraints. In specific cases upper and/or lower bounds on available seats are established due to the existence of international agreements between the carriers and/or governments of some countries.

Wong et al. (1993) developed the "flexible assignment approach," which "assigns some seats exclusively to each single or multi-leg itinerary as in fixed assignment and assigns the remaining seats to groups of itineraries as in bucket control." When considering control strategy used by the airline following questions come to mind: Which control strategy the airline should use when solving seat allocation problem in case of multiple-fare classes and multiple itineraries? Is it possible to figure out an analytic basis that indicates the superiority of one control strategy over the others? How to define the nest? Is it always possible to apply nested hierarchical structure, which is often used for the multiple-fare class, single itinerary problem? What is the impact of the specific control strategy on passenger demand (original passenger demands are usually changed depending on applied strategy), number of denied passengers, number of no-show passengers, number of cancellations, behavior of competitors, and so on?

There are no straigthforward answers to these questions. Current practice, current computer reservation systems, existence of international agreements between carriers and/or countries, as well as existence of other factors, condition that different carriers still use different control strategies (separate inventories or different forms of nested reservation systems). It is obvious that the quality of a specific control strategy need be measured considering several factors that are not yet thoroughly explored. Wong et al. (1993) recently pointed out that "the rules for bucket control are heuristic in nature and although they are generally believed to make an important contribution to revenue, there is no analytic basis for establishing their optimality among all seat allocation rules." The mathematical modeling of different control strategies is necessary in order to better understand and solve the complex problem of seat allocation in case of multiple-fare class and multiple itineraries.

Teodorovic's research (1998) considers the problem of seat allocation in the case of multiple-fare classes and multiple itineraries. We call this the *airline network seat inventory control* (ANSIC) problem.

Let us denote by S_i the seats allocated to passenger itinerary-fare class combination $i \in$ I. Based in *Figure 3.19*, we conclude that the following relation must be satisfied:

$$\sum_{i \in I} \delta_{ia} S_i \le C_a, \quad \forall a \in \mathrm{A} \tag{3.86}$$

In other words, the sum of the seats assigned to different itinerary-fare class combinations that include the considered leg must be less than or equal to the available number of seats on the leg. Relation (3.86) must be fulfilled for all legs in the network.

Itinerary-fare class combinations are comprised of one (a nonstop flight) or more flight legs. Let us note *Figure 3.20*.

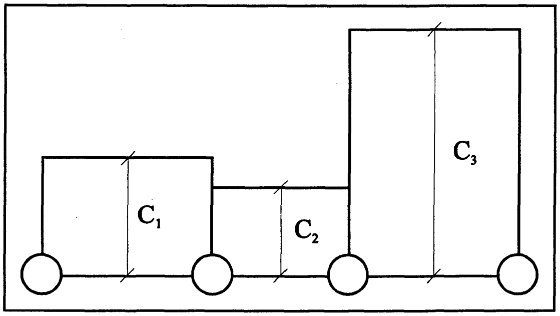

Figure 3.20. A passenger itinerary-fare class combination containing three flight legs

Figure 3.20 shows the number of seats available (C_1, C_2, C_3) on the legs that are part of the considered passenger itinerary-fare class combination. Let us denote this passenger itinerary-fare class combination by *b*. It is clear that the number of seats S_b assigned to combination b must satisfy the following relation:

$$S_b \le \min\left(\mathrm{C}_1, \mathrm{C}_2, \mathrm{C}_3\right) \tag{3.87}$$

In other words, the maximum number of seats assigned to any passenger itinerary-fare class combination may not be greater than the smallest number of seats available on the legs that are part of the considered combination. Let us define set $\mathrm{N}_i = \{a \mid \delta_{ia} = 1\}$. In the general case, relation (3.87) reads

$$S_i \le C_a \quad \forall a \in \mathrm{N}_i \ , \ \forall i \in \mathrm{I} \tag{3.88}$$

Let us denote by $\mathbf{D}_i$ future demand on passenger itinerary-fare class combination i. As previously noted, it was assumed that future demand on a passenger itinerary-fare class combination could be treated as a fuzzy number (*Figure 3.21*).

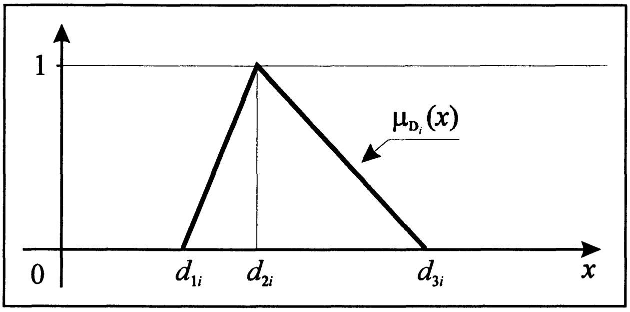

Figure 3.21. Membership function $\mu_{D_i}(x)$ of fuzzy number $\mathbf{D}_i$

The probability density function of future demand is obtained as the result of "measuring" demand. In other words, based on statistical data and the verification of statistical hypotheses, the conclusion is made that future demand has a certain probability density function. Models developed to date have been based on the assumption of a specific probability density function (most often Gaussian). In certain situations, local demand on some itineraries is unexpectedly high or unexpectedly low, through demand is unexpectedly extremely high, and so on. As we can see, the very expressions "high," "low," and "extremely high" contain fuzziness. In such cases, "human intervention" that is, a correction of the existing probability density function is needed. Future demand can be subjectively estimated based on the experience and intuition of the decision maker by considering historical data and present demand. In other words, all objective information must be used when estimating future demand, but certain subjective estimates must be built into it as well. Future demand on all passenger itinerary-fare class combinations could be represented by triangular fuzzy numbers. Using the triangular form of the membership function is conditioned by the fact that fuzzy arithmetic operations are the simplest for triangular fuzzy numbers. (We would note that the form of the membership function can in some cases be determined exclusively on the basis of objective information. For the case of some other passenger itinerary-fare class combination, objective information can be corrected by the analyst's subjective evaluation). The Teodorovic's model (1998) does not exclude the possibility of using fuzzy numbers that are not triangular. It is clear that seat allocation S_i to passenger itinerary-fare class combination i must satisfy the following inequality:

$$S_i \leq \mathbf{D}_i \quad , \forall i \in \mathrm{I} \tag{3.89}$$

In other words, the number of seats allocated to any passenger itinerary-fare class combination must be less than or equal to demand for that combination.

The ANSIC problem can be defined as follows: For known passenger itinerary fare-class combinations, known number of available seats on individual flight legs, known fares for individual passenger itinerary fare-class combinations, and approximately known future demand in different combinations (expressed by fuzzy numbers), allocate available seats to individual combinations in order to maximize the air carrier's total revenue. The formally denoted problem (P_1) can be expressed as

$$\max \mathrm{F} = \sum_{i \in \mathrm{I}} f_i \, S_i \tag{3.90}$$

$$\sum_{i \in \mathrm{I}} \delta_{ia} \, S_i \leq C_a \qquad \forall a \in \mathrm{A} \tag{3.91}$$

$$S_i \leq C_a \qquad \forall a \in \mathrm{N}_i, \ \forall i \in \mathrm{I} \tag{3.92}$$

$$S_i \leq \mathbf{D}_i \qquad \forall i \in \mathrm{I} \tag{3.93}$$

$$S_i \geq 0 \qquad \forall i \in \mathrm{I} \tag{3.94}$$

$$S_i \text{ is an integer}, \quad \forall i \in \mathrm{I} \tag{3.95}$$

Problem P_1 as defined by relations (3.90) to (3.95) is a mathematical programming problem. Subjectively estimated future demand $\mathbf{D}_i$ is denoted in bold letters in order to stress that $\mathbf{D}_i$ is a fuzzy number.

Glover et al. (1982) developed a network-based seat allocation model to solve the seat allocation problem for an airline network with a large number of passenger itinerary-fare class combinations. The model they developed considered demand to be a deterministic variable: the model did not have a variability of demand or uncertainty linked to future demand.

3.7.2. Heuristic algorithm to solve the ANSIC problem

Consider the constraint

$$S_i \leq \mathbf{D}_i \quad \forall i \in \mathrm{I} \tag{3.96}$$

As noted above, future demand of the *i*th passenger itinerary-fare class combination, $\mathbf{D}_i$, is a fuzzy number. This fuzzy number is an attempt to express the uncertainty regarding future demand and the estimate of the overbooking percentage. The overbooking percentage is dependent on both the rate of reservation cancellations and the no-show probability at departure. Let us call a fuzzy number $\leq \mathbf{D}_i$ "demand equal to or less then $\mathbf{D}_i$." The membership function of the fuzzy number $\leq \mathbf{D}_i$ is defined in a manner proposed by Kikuchi and Perrinchery (1990) (*Figure 3.22*). The membership function $\mu_{\leq \mathbf{D}_i}(x)$ gives us the information about the truth of the statement that the number x is less then or equal to the fuzzy number $\mathbf{D}_i$.

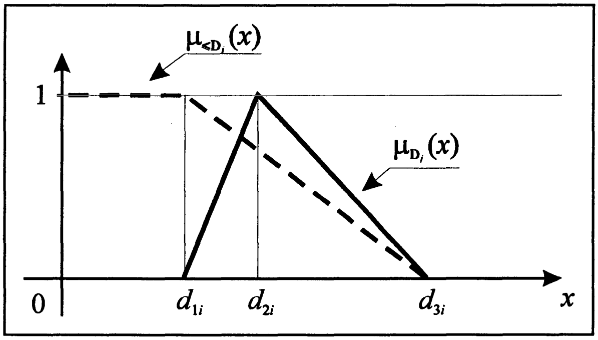

Figure 3.22. The membersip function of the fuzzy number $\leq \mathbf{D}_i$

The choice of a specific shape of the membership function of the fuzzy number has no crucial impact on generality of the model developed by Teodorovic (1998).

It is clear from *Figure 3.22* (based on the similarity of the triangles) that S_i will be less than or equal to $\mathbf{D}_i$ with a level of satisfaction at least equal to h that is,

$$\mu_{\leq \mathbf{D}_i}(S_i) \geq h \tag{3.97}$$

if the following is satisfied:

$$S_i \leq d_{1i} + (1-h)(d_{3i} - d_{1i}) \tag{3.98}$$

Let us introduce acceptable revenue into the discussion. In other words, instead of maximizing revenue, we will try to generate revenue greater than or equal to the acceptable revenue with a level of satisfaction at least equal to h. Acceptable revenue is defined as the triangular fuzzy number **R** (*Figure 3.23*).

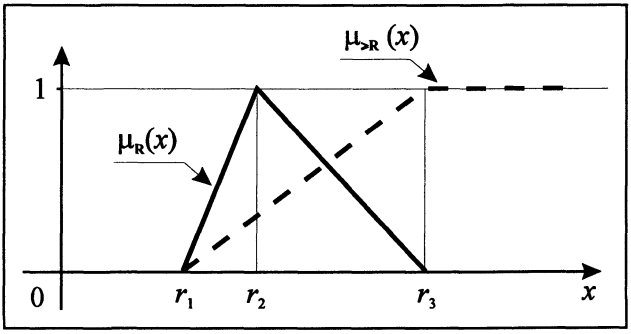

Figure 3.23. Triangular fuzzy number **R** representing acceptable revenue and revenue equal to or greater than acceptable revenue

Acceptable revenue is arbitrarily defined. In reality, acceptable revenue would be defined by the airline company's analyst-decision maker. Since objective functions and constraints are treated in the same manner in a fuzzy environment, the objective function (maximizing revenue) is transformed into the following constraint (*Figure 3.23*):

$$\sum_{i \in I} f_i \, S_i \geq r_1 + h(r_3 - r_1) \tag{3.99}$$

In other words, if relation (3.99) is satisfied, we can maintain that greater than satisfactory revenue will be achieved with a satisfaction level at least equal to h.

The new objective function reflects our desire to maximize level of satisfaction h. In this manner, problem (P_1) can be transformed into the following problem (P_2):

$$\max h \tag{3.100}$$

$$\sum_{i \in I} f_i \, S_i \geq r_1 + h(r_3 - r_1) \tag{3.101}$$

$$\sum_{i \in I} \delta_{ia} S_i \le C_a \quad \forall a \in A \tag{3.102}$$

$$S_i \le C_a \quad \forall a \in N_i \ , \ \forall i \in I \tag{3.103}$$

$$S_i \le d_{1i} + (1-h)(d_{3i} - d_{1i}) \quad \forall i \in I \tag{3.104}$$

$$S_i \ge 0 \quad \forall i \in I \tag{3.105}$$

$$S_i \text{ is an integer}, \quad \forall i \in I \tag{3.106}$$

The solution technique used for the problem (P_2) depends on the size of the problem instance at hand. Problem (P_2) can be classified as an integer programming problem. Existing methods developed in the field of integer programming cannot be successfully applied in solving the (P_2) problem. Large world air carriers serve, for example, 1,000, 2,000, or 2,500 origin-destination markets. Carriers have five, six, seven or eight different fare classes on some flights. Thus, the number of possible passenger itinerary-fare class combinations on the air carrier's entire network is enormous. Some companies also have an extremely large total number of flight legs.

The dimensions of problem (P_2) and the desire to enable interactive work when solving it (that is, to enable "human expertise") have led to solving problem (P_2) using a heuristic algorithm. Teodorovic (1998) developed a heuristic algorithm to solve problem (P_2). The algorithm, which belongs to the category of heuristic "greedy" algorithms, is comprised of the following algorithmic steps:

Step 1: Sort the passenger itinerary fare-class combinations in decreasing order of fares f_i. This creates the "fares list" and also establishes the priority when serving individual passenger itinerary fare-class combinations. Sort the legs in increasing order of capacity C_a. The "capacities list" indicates the order in which to examine constraints regarding the capacity of different legs.

Step 2: Arbitrarily choose level of satisfaction h. Also, arbitrarily determine "acceptable revenue" (define fuzzy number **R**).

Step 3: *The control of fulfillment of capacity constraints along all passenger itinerary-fare class combinations.* Determine the maximum number of seats S_i ($\forall i \in \mathrm{I}$) that can be allocated to individual passenger itinerary-fare class combinations. Denote this number of seats as S_i^*. It is clear that

$$S_i^* = \min \left\{ \min_{a \in \mathrm{N}_i} \{C_a\}, \left(d_{1i} + (1-h)\left(d_{3i} - d_{1i}\right)\right) \right\} \tag{3.107}$$

Step 4: *The control of fulfillment of capacity constraints along all legs.* Process the "capacities list" beginning with the smallest capacity. Process the "fares list" beginning with the largest fare. For passenger itinerary-fare class combination i under consideration, check whether the allocation of S_i^* seats to combination i violates the capacity constraints of leg a through which the considered combination passes. If it does not violate the capacity constraints of leg a, allocate S_i^* seats to combination i. Go to the next combination. If the fares list has been exhausted, begin examining the next leg. When all legs have been examined, go to Step 5. Should the allocation of S_i^* seats to combination i violate the capacity constraints of leg a, decrease the value of S_i^*. The new, decreased value of S_i^* equals

$$S_i^* = C_a - \sum_{k=1}^{i-1} \delta_{ka} S_k^* \tag{3.108}$$

All passenger itinerary-fare class combinations for which $\delta_{ka} = 1$, where $k > \mathrm{i}$ should not be allocated a single seat ($S_k^* = 0$). Go to the next leg. When all legs have been examined, go to Step 5.

Step 5: *The control of constraint pertaining to achieving the defined level of satisfaction h and also previously defined "acceptable revenue."* Check whether the following relation is satisfied:

$$\sum_{i \in \mathrm{I}} f_i S_i \geq r_1 + h\left(r_3 - r_1\right) \tag{3.109}$$

If the relation is not satisfied, then the greedy algorithm failed to find a feasible solution for the defined "acceptable revenue" **R** and level of satisfaction h. In order to find a feasible solution, we must return to Step 2 and change either the "acceptable revenue" **R** or the level of satisfaction h, or both.

The lowering of the "acceptable revenue" value (moving the fuzzy number **R** to the left) also means that the analyst wants to comply with the assigned constraints which have a level of satisfaction at least equal to h, even at the cost of the somewhat lowered total revenue. The lowering of the level of satisfaction h points to the desire of the analyst to accomplish the assigned "acceptable revenue" with the lowered value of h that is, with the higher risk that some of the constraints will not be fulfilled. Simultaneous lowering of both "acceptable revenue" **R** and the level of satisfaction h describes the situation when both lowered total revenue and higher risk of some unfulfilled constraints are acceptable.

Should the above relation be satisfied, the solution has been reached and available seats have been allocated to individual passenger itinerary-fare class combinations. This allocation provides the air carrier with revenue greater than or equal to "acceptable revenue" **R** for a level of satisfaction at least equal to h. Should the analyst-decision maker be dissatisfied with the solution for any reason whatsoever, he should return to Step 2 and change "acceptable revenue" **R** and/or level of satisfaction h.

Step 3 of the algorithm checks up (examines) the fulfillment of the constraints defined by the relations (3.103) and (3.104). In other words, this step inspects the capacity constraints along all passenger itinerary-fare class combinations. The control is done along all itineraries for all flight legs through which all the itineraries pass.

Step 4 of the algorithm relates to the examination of the fulfillment (achievement) of the capacity constraints along all legs. In other words, Step 4 checks up the constraint defined in the relation (3.102). For every leg in the network it is checked up whether the total sum of the seats assigned to the itineraries which pass through the leg is smaller then the leg capacity.

Step 5 of the algorithm pertains to the examination of the fulfillment of the constraint relating to the defined level of satisfaction and defined acceptable revenue.

3.7.3. Numerical example

The above heuristic algorithm was tested by Teodorovic (1998) on a large number of different hypothetical examples. All computer experiments were made on a personal computer (486 processor, 50 MHz). In cases when future demand on a passenger itinerary-fare class combination is treated as a deterministic variable, the fuzziness present in constraint (3.93) of the problem (P_1) disappears. Then the problem (P_1) becomes a classical integer programming problem that can be solved using commercial linear/integer programming software. It is surely very important to establish a relationship between the heuristic results and *linear programming* (LP) and *integer programming* (IP) results. The comparison of the results obtained by the proposed heuristic algorithm and the results obtained by the BLP programming package (*bounded linear programming*) is done in a following way. Twenty-four hypothetical numerical examples are generated. Numerical examples differed from each other in a number of passenger itinerary-fare class combinations, number of flight legs, and values of demand of different passenger itinerary-fare class combinations. Each of the considered numerical examples was first solved using the BLP programming package. Then the same example was solved again using the proposed greedy heuristic algorithm. Since the demand of each of the passenger itinerary-fare class combinations is a deterministic variable, then all the fuzzy numbers $\mathbf{D}_i$=(d_{1i}, d_{2i}, d_{3i}), ($\forall i \in$ I) that represented the demands in the problem (P_2) got transformed into the crisp numbers d_{2i} ($\forall i \in$ I). In other words, considered numerical examples were solved using both LP and heuristic algorithm with the assumption that the demand of the ith passenger itinerary-fare class combination equals d_{2i}. The optimal allocation plan and the maximum value of the total revenue R^* which is realized by the carrier were obtained by using the linear programming. In order to apply the proposed heuristic algorithm, it was necessary to define the "acceptable revenue" **R**. As the "acceptable revenue" **R** the number $\mathbf{R} = (R^*, R^*, R^*)$ was taken. In other words, the "acceptable revenue" **R** was treated as a crisp number which equals R^*. Considering that the future demand and "acceptable revenue" were treated as deterministic variables, thus the value of the level of satisfaction h that is assigned in the second step of the heuristic algorithm had no impact on the solution obtained (This is easily noticed by analyzing relations (3.101) and (3.104)). *Table 3.3* shows the results obtained by using linear programming and the results obtained by applying the greedy heuristic algorithm.

As it can be seen from *Table 3.3*, the heuristic results and the LP results are very close to each other in all the tested numerical examples. In all twenty four tested numerical examples, the relative error considerably lower than 1% was accomplished, which points to the quality of the proposed greedy heuristic algorithm. This algorithm provides quality solutions, very easily solves the problems of larger dimensions, and provides the decision maker with the interactive work.

It is certainly of great importance to determine the impact of uncertainty of future demand of passenger itinerary-fare class combinations on the total revenue of a carrier. It is also very significant to determine the impact of the subjectively determined "acceptable revenue" **R** and the level of satisfaction h on the quality of the obtained results. Let us do this analysis at one of the tested numerical examples. Besides other numerical examples, the smaller numerical example comprising 250 different passenger itinerary-fare class combinations and sixty flight legs was analyzed in more detail by Teodorovic (example 1 from *Table 3.3*). While treating this example, it was assumed that all demands of passenger itinerary-fare class combinations were deterministic variables. For known values of all the variables f_i, C_a, and D_i

($\forall i \in \mathrm{I}$, $\forall a \in \mathrm{A}$) the values of all the variables S_i ($\forall i \in \mathrm{I}$) and the total revenue of a carrier were calculated using the BLP programming package. The total revenue in this case was 1,134,391.

Table 3.3. Results obtained by using linear programming and by applying greedy heuristic algorithm

Problem number	Number. of combinations × number of legs	Results obtained: by using linear programming (LP)	Results obtained: by applying heuristic algorithm (H)	Relative error (LP-H)*100/LP [%]
1.	250x60	1134391	1134391	0.00
2.	250x60	1142631	1140468	0.19
3.	250x60	997753	997315	0.04
4.	250x60	993909	986330	0.76
5.	800x200	3539441	3527957	0.32
6.	800x200	3260472	3238017	0.69
7.	800x200	3388177	3383330	0.14
8.	800x200	3156390	3132997	0.74
9.	1500x350	6095960	6079288	0.27
10.	1500x350	6301445	6273103	0.45
11.	1500x350	6362435	6339143	0.37
12.	1500x350	6106820	6077484	0.48
13.	2000x400	7713805	7681961	0.41
14.	2000x400	8440372	8412475	0.33
15.	2000x400	8144743	8118101	0.33
16.	2000x400	8216806	8187581	0.36
17.	2500x500	9691895	9626035	0.68
18.	2500x500	9940985	9858168	0.83
19.	2500x500	10512164	10459719	0.50
20.	2500x500	10119047	10080102	0.38
21.	3000x600	12476995	12419341	0.46
22.	3000x600	12128785	12077841	0.42
23.	3000x600	12542591	12444988	0.78
24.	3000x600	12529924	12455588	0.59

In the next step of the analysis of this numerical example, all the demands were treated as fuzzy numbers $\mathbf{D}_i = (d_{1i}, d_{2i}, d_{3i})$, ($\forall i \in \mathrm{I}$). Values d_{2i} were equal to the values of deterministic demand and values d_{1i} and d_{3i} were calculated as

$$d_{1i} = 0.9\, d_{2i} \qquad d_{3i} = 1.1\, d_{2i} \qquad \forall i \in \mathrm{I} \tag{3.110}$$

In other words, the uncertainty in regard to the future demand was considered.

Table 3.4 presents the results which show the link between desired "acceptable revenue" **R** and desired "level of satisfaction" h for a smaller numerical example comprising 250 different passenger itinerary-fare class combinations and sixty flight legs.

Table 3.4. Total revenue for different combinations of level of satisfaction and acceptable revenue (network with 250 combinations and 60 flight legs)

Level of satisfaction (h)	Acceptable Revenue					
	R_1	R_2	R_3	...	R_{13}	R_{14}
0.00	1231711	1231711	1231711	...	1231711	*
0.05	1212656	1212656	1212656	...	*	*
0.10	1202000	1202000	1202000	...	*	*
0.15	1193646	1193646	1193646	...	*	*
0.20	1181271	1181271	1181271	...	*	*
0.25	1176302	1176302	1176302	...	*	*
0.30	1163179	1163179	1163179	...	*	*
0.35	1155130	1155130	1155130	...	*	*
0.40	1142230	1142230	1142230	...	*	*
0.45	1135453	1135453	1135453	...	*	*
0.50	1129301	1129301	1129301	...	*	*
0.55	1114060	1114060	1114060	...	*	*
0.60	1102921	1102921	1102921	...	*	*
0.65	1094960	1094960	1094960	...	*	*
0.70	1086271	1086271	1086271	...	*	*
0.75	1077499	1077499	1077499	...	*	*
0.80	1063494	1063494	1063494	...	*	*
0.85	1054544	1054544	*	...	*	*
0.90	1046326	1046326	*	...	*	*
0.95	1035231	*	*	...	*	*
1.00	1034611	*	*	...	*	*

* The greedy algorithm failed to find a feasible solution

$R_1 = (925, 975, 1025)$ ... $R_{13} = (1225, 1275, 1325)$

$R_2 = (950, 1000, 1050)$ ... $R_{14} = (1250, 1300, 1350)$

$R_3 = (975, 1025, 1075)$ ...

The analyst-decision maker arbitrarily assigns "acceptable revenue" **R** and "level of satisfaction" h. The basic idea of the model developed by Teodorovic (1998) is to find the allocation of seats to combinations that earns revenues that are greater than or equal to satisfactory, and has a satisfaction level of at least h.

As can be seen from *Table 3.4*, a higher level of satisfaction can be achieved by simultaneously decreasing the amount of acceptable revenue that is, the revenue we want to realize. The opposite also holds true. Greater passenger revenues can be achieved by decreasing the level of satisfaction. By alternately assigning acceptable revenue and/or level of satisfaction, a large number of different seat allocations are generated. The choice of a specific seat allocation depends on the analyst's readiness to accept a certain risk (somewhat lower level of satisfaction h) in order to achieve larger total revenue.

The basic assumption behind the analyst's interactive work, generation of a larger number of different seat allocations, and final choice of seat allocation to individual passenger itinerary-fare class combinations is short computer time. In order to establish the length of computer time, a large number of computer experiments were made (PC computer, 486 processor, 50 MHz). *Table 3.5* presents the computer times achieved. We would note that computer time does not include the time needed to sort fares in decreasing order and to sort capacities in increasing order.

The solution times shown in *Table 3.5* are given for one value of **R** and one value of h.

As can be seen from *Table 3.5*, satisfactory computer times were reached. This allows the analyst to work interactively, generate a large number of seat allocation plans and choose the final seat allocation.

Table 3.5. Computer time for different dimensions of the airline network seat inventory problem

Number of different passenger itinerary fare-class combinations	Total number of flight legs	CPU times
1,000	500	21 sec
2,000	500	41 sec
3,000	500	61 sec
10,000	500	3 min 19 sec
20,000	500	6 min 30 sec
2,000	1,000	1 min 22 sec
3,000	1,000	2 min 3 sec
4,000	1,000	2 min 43 sec
10,000	1,000	6 min 43 sec
20,000	1,000	13 min 15 sec

Chapter 4.

Applications of Artificial Neural Networks in Transportation

In the 1990s over fifty papers were published applying neural network models to transportation problems (Dougherty, 1995). Most of these papers were concerned with road transportation. Yang et al. (1992) and Dougherty and Joint (1992) modeled the driver's behavior when making strategic and instinctive decisions. These authors use neural networks to analyze data gathered in the following way: volunteer drivers who took part in the experiments chose the routes by comparing the values of different criteria. The collected data were used to train neural networks. The trained networks successfully reproduced the drivers' decisions. When the developed networks were applied to new data, they produced good results more quickly and accurately than alternative techniques like logit models. Many authors have studied the simulation of the driver's behavior while driving a vehicle. For example, Hunt and Lyons (1994) use neural networks to model the driver's behavior while changing the speed and lane on a highway. A developed neural network is regarded as a "neuro driver" who maneuvers the chosen vehicle following input signals (relative position of the vehicle surrounded by other vehicles in particular time intervals) from the near vicinity. The basic disadvantage of this procedure is the problem of collecting real data necessary to train the network. If the data are recorded from one driver only, the neural network displays driving characteristics similar to that driver. A survey should be conducted to determine the set of representative drivers and their driving should be recorded.

Ritchie and Cheu (1993) use neural networks to detect long-lasting traffic jams (traffic accidents, broken vehicles, dropped cargo, maintenance, constructions). The output of a neural network is the classification of transportation on a particular section: (1) "no congestion" and (2) "long congestion." By changing the condition from (1) to (2) an "alarm" is activated. Discovering such problems at the moment they occur is of significant importance. In training the network 200 congestions were simulated, the duration of one simulation being 25 minutes within which fifty input vectors were generated. Thus, the training set contained 10,000

D. Teodorovič et al., *Traffic Control and Transport Planning:*

data. In comparison with the currently used algorithms, the authors demonstrated that neural networks have a potential to achieve a significant improvement in discovering long term congestions on a travel network.

Neural networks are used to analyze congestions of urban travel networks and seasonal fluctuations in the vehicle flow (Hua and Faghri, 1993), to estimate origin-destination matrices (Kikuchi et al., 1993), and to evaluate travel network improvement (Wei and Schonfeld, 1993). In predicting the vehicle flow (over a long period or immediately after the observed moment) a number of authors present very good results (Chin et al., 1992; Doughety et al., 1994).

The applications of neural networks in transportation have reached the stage in which all these important investigations should be applied to real transportation systems.

4.1. BASIC CONCEPTS OF ARTIFICIAL NEURAL NETWORKS

People relatively easily perform a variety of complex tasks that are highly difficult to solve by computational techniques of traditional algorithms. The brain architecture largely differs from common serial computers and the researchers of artificial neural networks seek to endow these machines with the abilities of data processing similar to those of a human. Artificial neural networks are inspired by biology that is, they are composed of the elements that function similarly to a biological neuron. These elements are organized in a way that is reminiscent of the anatomy of a brain. In addition to this superficial similarity, artificial neural networks display a striking number of the brain's properties. For example, they are able to learn from experience, to apply to new cases generalizations derived from previous instances, and to abstract essential characteristics of input data that often contain irrelevant information.

Despite these functional similarities, even the most optimistic advocates could not claim that artificial neural networks will soon completely mimic the functions of a human brain. Real "intelligence" displayed by the most sophisticated artificial neural networks does not exceed the intelligence level of a worm (Wassermann, 1989); the enthusiasm should therefore be balanced to accommodate to the real situation. However, it is equally wrong to neglect the performance of certain artificial neural networks, which staggeringly resembles that of a brain. These abilities, however limited they may be, indicate that a deeper understanding of human intelligence may be at hand accompanied by a number of revolutionary changes.

Conceptually, feedforward neural networks approximate unknown functions that is, they can be considered as universal approximators. The theorem proved by Hornik et al. (1989) and Cybenko (1989) states that a multilayered feedforward neural network with one hidden layer can approximate any continuous function up to a desired degree of accuracy provided it contains a sufficient number of nodes in the hidden layer.

4.1.1. Neurons

4.1.1.1. Biological neuron

The human brain contains roughly 10^{11} neurons that apart from the characteristics that they share with other cells have unique capabilities to receive, process and transmit electrochemical signals to other neurons. Neurons communicate between themselves and are organized in the form of a neural network that is regarded as the brain's system of communication. *Figure 4.1* illustrates the structure of a biological neuron. Neurons consist of the cell body (*soma*) and several branches. The branches conducting information to the cell (*stimulus*) are called *dendrites*, and the branches conducting information out of the cell (*response*) are called *axons*. The emitted signals differ in frequencies, duration and amplitudes. The interaction between the neurons occurs at specific connection points called *synapses*.

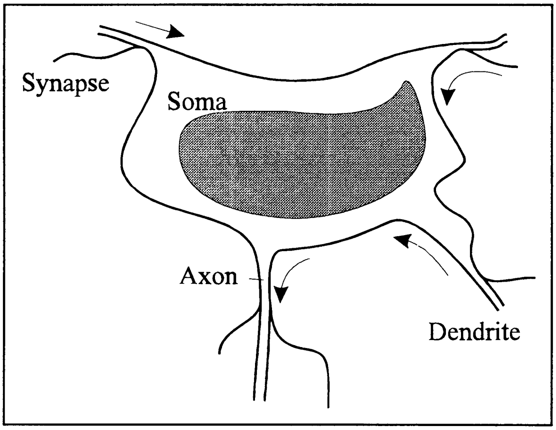

Figure 4.1. Structure of a biological neuron

Since the connections between neurons are of different strength, a stimulus is considered to be a weighted sum of the received inputs. If a stimulus exceeds a threshold, the neuron transmits the signals down the axon to other neurons. The brain contains over fifty different kinds of neurons relative to their shape and specialized functions. The brain is a complex communication system. The overall length of a network's branches approximates 10^{14} m. The presence of such a large number of connections establishes a high level of "massive parallelism." While a single neuron relatively slowly responds to outside stimuli, the brain as a whole solves complex problems in a remarkably short time - in a fraction of a second or in a couple of seconds. The brain can analyze complex problems and adequately cope with unforeseen situations due to the knowledge stored in synapses and the ability to adapt to new situations. The human brain can learn and generalize the acquired knowledge. The generalization of knowledge refers to similar stimuli inducing similar responses.

4.1.1.2. An artificial neuron

The first model of an artificial neuron that was proposed by McCulloch and Pitts (1943) was a binary device with a binary input, binary output, and fixed activation threshold.

The later model of an artificial neuron (*Figure 4.2*) copied the properties of a biological neuron. The input signals x_1, x_2, ..., x_n, representing the output signals of other neurons, are multiplied by associated connection strengths, w_1, w_2, ..., w_n. Each connection strength (weight) corresponds to the strength of a biological synaptic connection. The output signal (*NET*) is equal to the weighted sum of input signals.

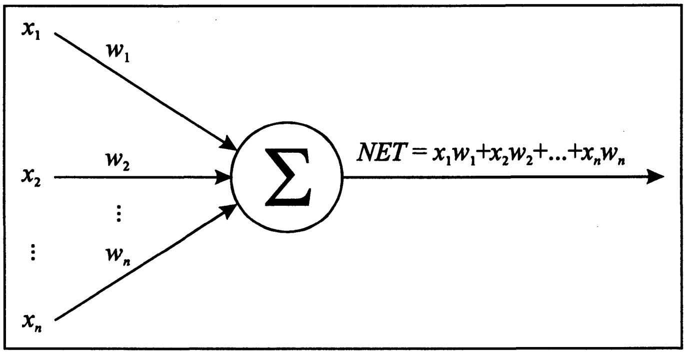

Figure 4.2. Structure of an artificial neuron

Figure 4.3 represents an artificial neuron with an associated activation function that is, a typical processing element. The presented activation function is most commonly used: nonlinear, continuous, monotonously increasing, bounded, differentiable logistic function. The range of the weighted sum of input signals, *NET*, is compressed by an "S" curve such that the value of the output signal, *OUT*, never exceeds a relatively low level regardless of the value of *NET*. The most commonly used activation functions are step function, sigmoid function, hypertangent function, and identity function (*Figure 4.4*).

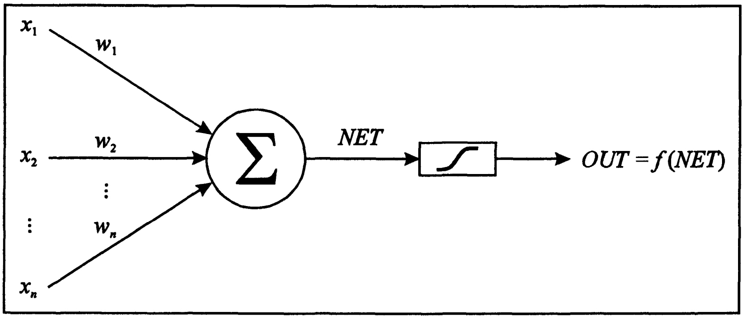

Figure 4.3. Processing element: artificial neuron with activation function

The transformation of input signals by a logistic curve enables the receiving and processing of very weak and very strong signals. When input signals are very strong (positive or negative), the curve slope is very small. When input signals are weak, the curve slope is large, which means that a usable output signal can be produced.

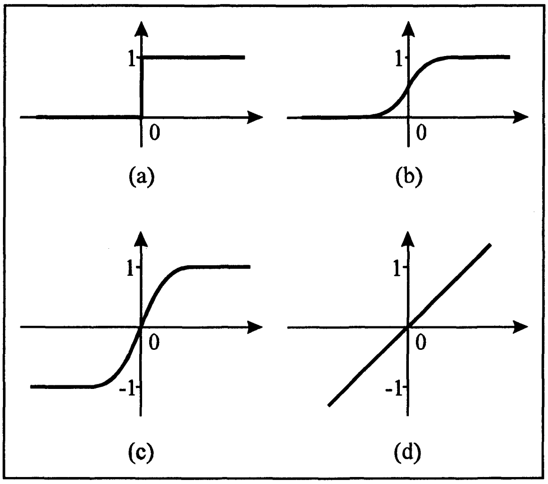

Figure 4.4. The most commonly used activation functions: (a) step function, (b) sigmoid function, (c) hypertangent function, and (d) identity function

4.1.2. Characteristics of neural networks

The present neural network architecture is based on a simplified model of the brain, the processing task being distributed over numerous neurons (nodes, units, or processing elements). Although a single neuron is able to perform simple data processing, the strength of a neural network is obtained as the result of the connectivity and collective behavior of these simple nodes. Neural computation differs from conventional algorithmic computation, although it can also be presented by an algorithm and incorporated into a serial machine. These apparently contradictory statements can be reconciled if we agree that this is an approach that distinguishes two techniques rather than an ultimate implementation. As far as computers are concerned, there is no difference between data stored in a long-term memory and active data in the working memory. Unlike computers, where information is stored for a later use, neural networks store the strength of synaptic connections (weights of the network's branches) by which the required data are reproduced. If a neural network is used in control and if the required knowledge is stored in weights, learning refers to estimating the appropriate weights in order to reproduce certain decisions under particular conditions. In other words, neural networks can be trained by adjusting the connection strengths in order to abstract the relations between the presented input/output data.

In describing the characteristics of neural networks the works of the following authors were used: Rumelhart and McClelland (1986), Wassermann (1989), Brown and Harris (1994), Jang and Sun (1995), Lin and Lee (1995), and Kasabov (1996). A neural network is characterized by

1. A set of processing elements
2. Connectivity of those elements
3. The rule of signal propagation through the network
4. Activation or transfer functions
5. Training algorithms (learning rules or learning algorithms)
6. Environment in which the network functions

A neural network configuration begins by defining a set of processing elements. Processing elements (nodes, neurons, units) are simple elements performing relatively simple signal processing in order to determine the output signal. In other words, each node receives input values from its nearest neighbors according to which it computes and transmits a single output value. Thus, each node has a local parameter set. A neural network is inherently parallel in the sense that a large number of processing elements make simultaneous computations. It can be said that a neural network performs parallel distributed data processing.

A neural network contains three types of nodes: input, output, and hidden. Input nodes receive input signals from outside sources that is, sources outside the network. Output nodes transmit signals that is, output values outside the network. Input and output signals of hidden nodes reside within the network and are not "visible" from the outside. Each node transmits signals of different strengths to its neighbors (the nodes to which it is connected). In certain neural network models each node is at each instant associated with activation state and the output function f by which the current activation state is mapped into the output signal. Function f can be a presupposed threshold that has to be exceeded for the node's output signal to be transmitted. It means that the observed node cannot affect other nodes as long as its activation does not exceed a particular threshold.

Many of the currently used neural models have a fixed network structure. A network structure that is, the number of nodes and their connection types can also be very flexible. An artificial neural network is completely determined (capabilities of modeling, representation, and generalization) once we have established the network's structure, activation function of each node, and learning rule.

How a network's nodes are connected is of vital importance. The synaptic connectivity of a network can be total or partial (for example, only connections between different layers are allowed). Also, if for example a neural network is intended to support control of a traffic dispatcher, its structure can mimic the dispatcher's decision process (special connectivity). Each branch of the network is associated with a weight (positive or negative value) modifying the strength of a signal. The absolute value of a branch's weight represents the connection strength. In order to calculate the output signal of a node, the weighted sum of input signals is modified by an activation function.

4.1.3. Classification of neural networks

Classification according to the network's structure. Depending on the number of input and output sets of nodes, neural networks can be classified in the following way:

- Autoassocciative, that is, input nodes are simultaneously output nodes; Hopfield network (1982) represents an example of an autoassocciative network.
- Heteroassociative, when there are different sets of input and output nodes; for example, Perceptron (Rosenblatt, 1958), multilayered perceptron (Rumelhart et al., 1986), Kohonen's network (1982).

Classification according to the presence (or absence) of feedback. Depending on the signal transmission through the network, the two basic types are

- Neural networks with feedforward signal propagation, where there are no connections between output and input nodes, for example, perceptrons.
- Neural networks with a feedback, called *reccurent neural networks,* where there are connections between output and input nodes. Output signals are determined by current input signals and previous output signals (Hopfield network).

Classification according to the network training. Three basic ways of training a neural network are

- *Supervised training*: Training is performed while the network "learns" to associate each input vector x to its corresponding targeted or desired output vector y. For example, a neural network can be trained to approximate function $y = f(x)$ represented by a set of training pairs (x, y).
- *Unsupervised training*: The training algorithm was developed by Kohonen (1982) while trying to describe the learning model of a biological system. During the process of learning a neural network "learns" only from a sample of input vectors. During the training, statistical properties of the samples are estimated, and similar input vectors are grouped into classes. The training algorithm modifies the weights of the network's branches in order to determine the consistent output vectors.
- *Reinforcement learning*: Training by criticizing (rewarding or punishing) as a combination of the two previous training algorithms is performed in the following way. For an input vector an output value is estimated by the network. If the value is viewed as "good," the neural network is "rewarded" by increasing the existing weights of the branches; otherwise, the network is "punished" by regarding the connection weights inadequate and they are, accordingly, decreased.

4.1.4. A multilayered feedforward neural network

A multilayered feedforward neural network is a network in which the input signal extends forward through several layers, while it is being processed to estimate the network's output signal. Each layer contains a certain number of nodes. Each node is a processing element associated with the corresponding activation function by which the weighted sum of input values is transformed to determine the output value. To each node's input only the outputs of nodes from a previous layer are supplied and the output signal is transmitted to the nodes of the next layer *Figure 4.5*. Each node is associated with a weight vector by which the input vector is linearly transformed. The node connectivity is generally total except in some transformations when it is partial. Nodes representing additional "noise" can also be included in neural network models. For example, Wei and Schonfeld (1993) added such a node in advance to the input layer of a two-layered feedforward neural network and connected it with all other nodes in the hidden and output layer. Although the output value of the additional node is equal to one, the authors showed by experiments that such additional "noise" can accelerate a network's training.

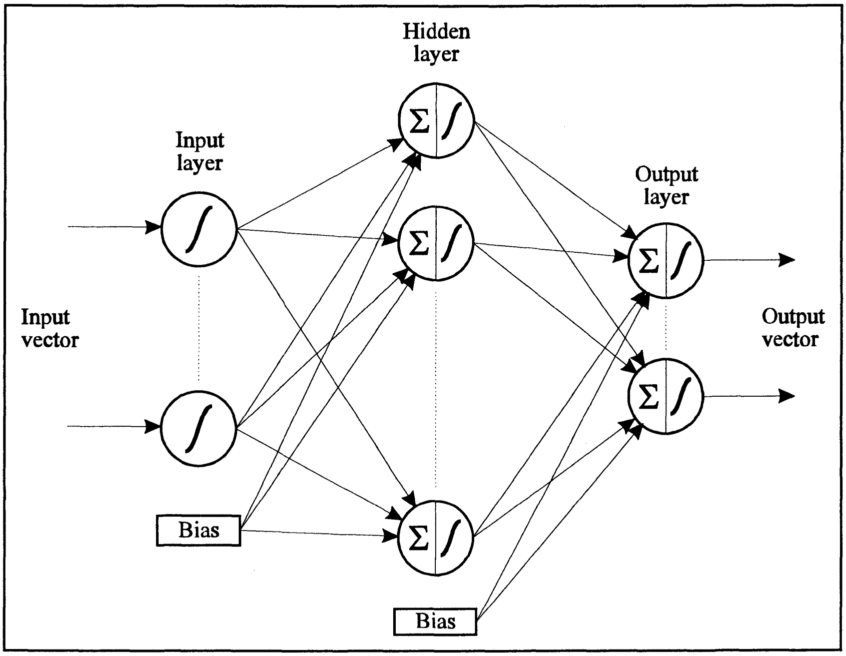

Figure 4.5. Two-layered feedforward neural network

4.1.5. Training of a neural network

Artificial neural networks are capable of modifying their behavior in response to the environment. This factor is, more that any other, responsible for the interest these networks have aroused. When presented with a set of input data (most frequently with the desired output data) they self-adapt to incite appropriate responses. A wide range of training algorithms has been developed, each having its own advantages and disadvantages. Most of the training algorithms can be classified according to their modeling, learning, and validation properties. The modeling abilities of an algorithm determine the range of nonlinear functions that it is able to precisely reproduce. The chosen structure of a neural network model can influence the convergence rate of a training algorithm and even determine the type of learning to be used. Finally, any practical application of a training algorithm requires examination of convergence, stability and accuracy that would verify the knowledge acquired. If an algorithm is able to learn, it is also able to forget, and it must be verified that the stored knowledge is sufficient.

Training algorithms are very simple mechanisms for adapting the weights of a network's branches, requiring for each branch only locally available data. Their implementation generally does not involve complex computations and consequently no powerful computation configurations are needed. Training algorithms function in ill-defined time-varying environments, with a minimal amount of human intervention.

The multilayered neural networks have come into use after the development of an error backpropagation algorithm, which was used for training a network. Various researches have independently developed a suitable and currently most popular algorithm for training a multilayered feedforward neural network (Le Cun, 1985; Parker, 1985; Rumelhart and McLelland, 1986). The proposed backpropagation algorithm is a gradient procedure. The activation functions of nodes are bounded, continuous, monotonously increasing, nonlinear, differentiable functions. The output function of the network is a continuous, differentiable weight function enabling the search of the extremum by the "gradient descent" algorithm.

The optimal weights, w_{ij}, are determined by the rule of gradient descent (*delta rule*, *generalized delta rule*) minimizing the criterion function or error. Each iteration of the algorithm (cycle or epoch defined as the process of transmission of one or a few training pairs through the network whereby the error is calculated) contains two passes (*Figure 4.6*):

- Propagation of one or a set of input signals forward to the output layer (in the original algorithm input signals were brought to the network individually)
- Backward pass where the computed error extends backward in order to calculate the changes of parameters (weight of the network's branches).

The procedure is performed in numerous iterations using the same training pairs until the error becomes "sufficiently" small.

One of the important problems concerning the training algorithm of a gradient type is that of the local minimum that is, the local minimum can prevent the algorithm from reaching the global minimum.

The second issue in practical applications concerns the problem of a reasonable training time, as the application of this algorithm often appears to be long-term. Furthermore, using this algorithm in training a network can result in the "overestimation" of output results. Some researchers have experienced the effect of an "excessive" network training (Croall and Mason, 1992). A prolonged period of training did not yield better values of the criterion function; the network seemed to have lost some of its knowledge.

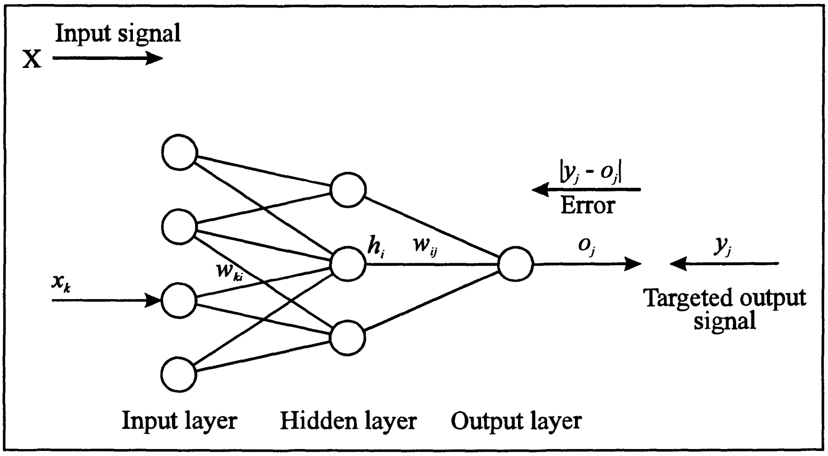

Figure 4.6. Taxonomy of training a multilayered perceptron: the input signal extends forward and the computed error backward

4.1.5.1. Generalization (of the acquired knowledge)

Central to the problem of learning is the ability of an algorithm to generalize correctly from a limited number of samples, which means that the algorithm has to interpolate and locally extrapolate rather precisely. The adaptively designed linear models and controllers imply a global, linear relation. If a miminum amount of *a priori* data on the structure of the desired function is assumed, the algorithm must draw the relevant knowledge from a set of data, which usually requires a large number of examples.

Once a neural network has been designed, its reaction can, to a certain extent, be insensitive to minor variations in the input set of data. This ability to differentiate in the presence of noise and distortion of shapes is of key importance. It is worthwhile to note that the artificial neural network automatically makes generalizations due to its structure, without the use of a human intelligence that would be embedded in it in the form of *ad hoc* computer programs.

A neural network's modeling abilities should be sufficiently flexible to achieve the sought goals, but if a model is too flexible, it can exceed (overestimate) the data and the generalization will be poor. Learning from a set of data can be a task of function approximation, as it is not enough to simply learn the reproduction of data; the structure underlying the function must also be learned from samples or else the network cannot correctly generalize to the neighboring input data. The adaptive neural network must be well configured since the possibility of generalization requires not only a good model structure but the convergence of parameters as well.

If a network is poorly configured, the designer may be inclined to think that the network is functioning properly even though the prediction or forecasting errors are small. However, the convergence of parameters in a badly configured network is slow, and a premature termination of a learning process means that the network is incapable of making proper generalizations (Brown and Harris, 1994).

4.1.5.2. Consideration of a network's reliability and stability

Prior to applying artificial neural networks in situations that may jeopardize human lives or material values, the issue of their reliability is of primary importance. Like humans, whose brain structure they mimic, artificial neural networks retain a certain degree of unpredictability. Unless all possible input values are tried out, there is not a reliable way of predicting the exact output values. In a large network such thorough testing is impractical, and we must be satisfied with a statistical assessment of the performance. In some circumstances such a procedure is unacceptable. The problem lies in the fact that a computer is expected to be free from any mistakes. According to a number of experts, the fact that artificial neural networks are sometimes prone to mistakes even when functioning correctly is considered unacceptable as far as machines are concerned.

A network is stable if the learned weights minimally change as the range of the training set of data increases. This is an extremely important property of adaptive models that are trained and applied in real time (for example, in system control). If the data from a training set belong to certain regions of the input space, over a long period only relevant data should be collected. When the learned weights are not in the vicinity of globally optimal weights, the training should be frequently repeated. Thereafter, it is necessary to examine the convergence of parameters. When the training set of data is a sufficiently large sample, there is a strong belief that the trained network will produce useful solutions and that the frequency of the repeated training will be small.

In assessing a trained neural network it is desirable to understand the influence of each branch's weight on output results of the network. Since the values of these adaptive parameters directly affect the output results of a network, determining individual influences would allow us to search the region in which the network is insufficiently trained and help us assess the network.

4.1.5.3. Validation of a neural model

The most important part of any model design procedure is validation of the given model. The tests are performed to estimate the extent to which the network has learnt the training data and how able it is to generalize (interpolate and extrapolate) to unforeseen cases. Most training algorithms are able to successfully learn a set of training data given a sufficiently flexible model structure, although the question of whether they have the ability to generalize correctly (or sufficiently accurately) remains unresolved. For safety reasons, the performance of the trained network must be fully understood; a network cannot simply be regarded as a black box about which nothing can be proved.

There are a number of ways in which the performance of a trained network can be assessed, the simplest one is the assessment of the network's performance in reproducing the training data. A better approach is to divide the available data into a training set and a test set and to use the testing data to evaluate the final model. In other words, any set of data presented to a learning algorithm must be split into a part that is used to train the network and another part used for estimating the network's performance by testing its ability to generalize correctly. However, if there is lack of the available data, it is generally difficult to determine how much should be used for training and how much should be used for testing. One heuristic that works well in practice is to use two-thirds for training and one-third for testing. This approach is reasonable since a learning algorithm cannot perform well if there are insufficient training data, but the output results cannot be assessed accurately if the test set is too small.

4.1.5.4. Network transparency

A certain degree of doubt about neural network models expressed by some researchers is probably due to the traditional artificial neural networks' inability to "explain" the way in which they cope with problems. Many neural networks are vague; the stored knowledge is distributed over a number of parameters in a complicated way that is not easy to understand.

The internal representation resulting from training is often so complex that it resists any analysis in all except the most trivial cases. This is closely connected with our inability to explain how we recognize a person despite the difference in distance, viewpoint, illumination, and passage of time.

4.2. TRANSPORTATION DEMAND FORECASTING WITH NEURAL NETWORK MODELS

As already known, the transportation demand between two regions depends on socioeconomic characteristics of the given regions, as well as the characteristics of the connecting transportation system. The output data of transportation demand models usually consist of an estimated number of the daily vehicle miles of travel (in individual road traffic), number of miles of travel (in air, railway and international coach travel), used telephone impulses (in communication traffic), and so on.

The input data for planning transportation networks and transportation facilities and for building new and reconstructing the existing traffic and transportation facilities include the data characterizing transportation demand. The transportation demand forecasting is of outstanding importance for the functioning of different traffic and transportation systems, as well as their further development.

Over the last few decades, in a number of transportation studies different regression travel demand forecasting models have been used. In recent years, neural networks have come to be used in the travel demand forecasting. The basic reason lies in the fact that neural networks are able to capture complex relationships and learn from examples and also able to adapt when new data become available.

Among other authors, neural networks have been used in the transportation demand forecasting by Chin et al. (1992) and Kalic et al. (1992). Chin et al. (1992) used a multilayered, feedforward neural network. Their model is based on socioeconomic and highway operation data for 339 urban areas in the United States from 1982 to 1988. One of the neural networks proposed by Chin et al. (1992) is presented in *Figure 4.7.*

As seen from *Figure 4.7*, the authors used a three-layered neural network. Between input and hidden layers there is a full connectivity. The authors used the proposed network to estimate the daily vehicle miles of travel (DVMT). The data supplied to the input layer are traffic-generation-related, traffic-attraction-related, and network-supply-related. As Chin et al. (1992) noted, "to account for the three above-mentioned explicit relationships as well as two unknown potential relationships, a total of five nodes are used in the hidden layer." As already pointed out, the author experimented with several different neural networks. These neural networks differed in the number of hidden layers, as well as the number of nodes in the hidden layers. The authors demonstrated that the average absolute relative errors between the neural network results and real data are within 4.77% and 7.09%, which can certainly be considered satisfactory.

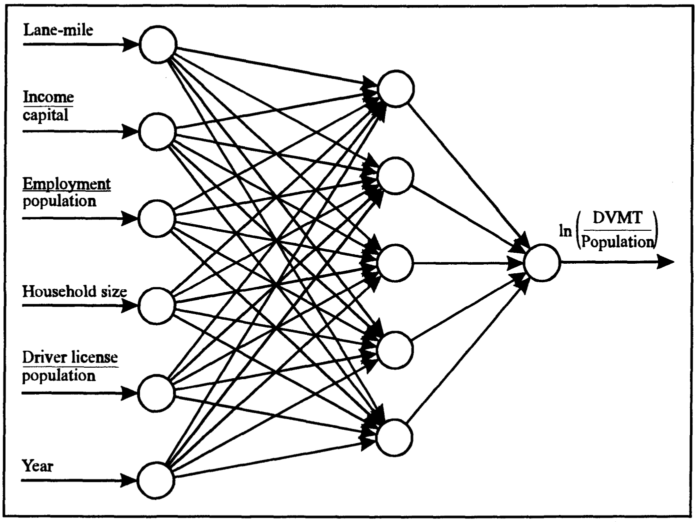

Figure 4.7. Neural network daily vehicle miles of travel (DVMT) forecasting model

Kalic et al. (1992) used a neural network model to estimate the number of used telephone impulses in Yugoslav towns. The number of telephone impulses was calculated according to the number of inhabitants, number of telephones, and national income.

The neural network used by Kalic et al. (1992) is shown in *Figure 4.8*.

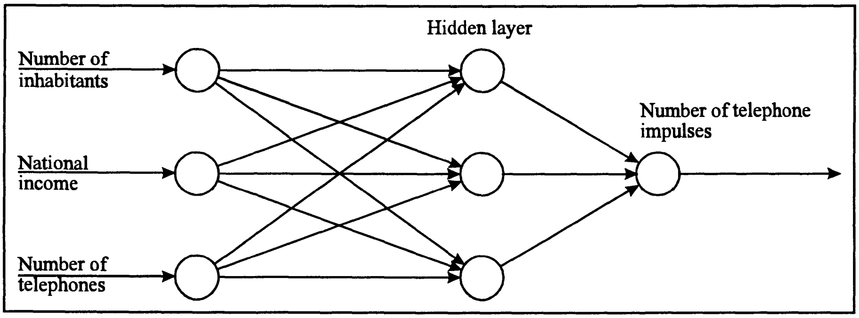

Figure 4.8. Neural network for estimating the number of telephone impulses

In training the network, Kalic et al. (1992) used the simulated annealing technique, which in the context of artificial neural networks belongs to the group of statistical training methods. The main procedures of the simulated annealing technique were presented in one of the previous sections of this book. The data used in training the network referred to forty eight towns in Yugoslavia. The average relative error in the set of training towns equaled 10%. In the control set of nine towns the average relative error equaled 8.5%.

4.3. BACKPROPAGATION NEURAL NETWORK FOR ESTIMATION OF AN ORIGIN-DESTINATION MATRIX

The estimation of the origin-destination matrix from traffic counts is one of the most significant problems encountered by traffic planners. This problem is found in determining the number of car trips between urban and/or suburban zones, in examining the functioning of transit lines in urban public transportation, in studying the suburban or international railway traffic, and so on. Observe *Figure 4.9*.

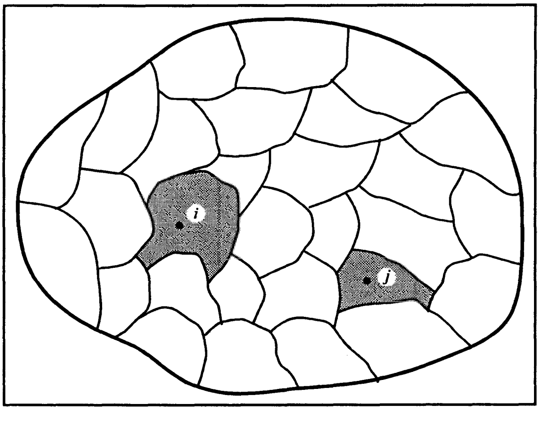

Figure 4.9. Urban area divided into zones

Figure 4.9 represents an urban area divided into zones. Each zone is associated one point (centroid) representing the center of the zone. According to the usual traffic-planning procedure, after the total number of trips that can be generated by a particular zone has been determined (trip generation models), trips are distributed to particular zones. The trip distribution involves determining the number of trips between particular zones. Let us denote by n the total number of zones:

$$\begin{array}{cc|ccccccc|c} & & & & & \text{Destinations} & & & & \\ & & 1 & 2 & 3 & \cdots\ j & \cdots & n & & \\ & 1 & 0 & f_{12} & f_{13} & \cdots\ f_{1j} & \cdots & f_{1n} & & P_1 \\ & 2 & f_{21} & 0 & f_{23} & \cdots\ f_{2j} & \cdots & f_{2n} & & P_2 \\ & 3 & f_{31} & f_{32} & 0 & \cdots\ f_{3j} & \cdots & f_{3n} & & P_3 \\ \text{Origins} & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & & \vdots \\ & i & f_{i1} & f_{i2} & f_{i3} & \cdots\ f_{ij} & \cdots & f_{in} & & P_i \\ & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & & \vdots \\ & n & f_{n1} & f_{n2} & f_{n3} & \cdots\ f_{nj} & \cdots & 0 & & P_n \\ & & A_1 & A_2 & A_3 & \cdots\ A_j & \cdots & A_n & & T \end{array} \quad (4.1)$$

We denote by f_{ij} ($i = 1, 2, \ldots, n, j = 1, 2, \ldots, n$) the number of car trips from trip origin i to trip destination j. We also denote by P_i the number of trips produced in zone (node) i. It is clear that

$$P_i = \sum_{j=1}^{n} f_{ij}, \qquad i = 1, 2, \ldots, n \quad (4.2)$$

The number of trips attracted by node j is denoted by A_j. The following relation is also fulfilled:

$$A_j = \sum_{i=1}^{n} f_{ij}, \qquad j = 1, 2, \ldots, n \quad (4.3)$$

The total number of trips T made from all origins to all destinations is equal to

$$T = \sum_{i=1}^{n} P_i = \sum_{j=1}^{n} A_j \quad (4.4)$$

The estimation of origin-destination patterns involves the estimations of the number of trips f_{ij} ($i = 1, 2, \ldots, n, j = 1, 2, \ldots, n$) between particular zones when the values of trip productions P_i ($i = 1, 2, \ldots, n$) and trip attractions A_j ($j = 1, 2, \ldots, n$) are known.

A similar problem concerning the estimation of origin-destination patterns occurs in examining the operations on one transit line (*Figure 4.10*).

In *Figure 4.10* are shown a number of travel patterns within one transit line. Let us denote by f_{ij} the number of passengers boarding at station i and alighting at station j. Let us, also, denote by B_i (i = 1, 2, ..., n) the total boarding at station i and by A_j (j = 1, 2, ..., n) the total alighting at station j.

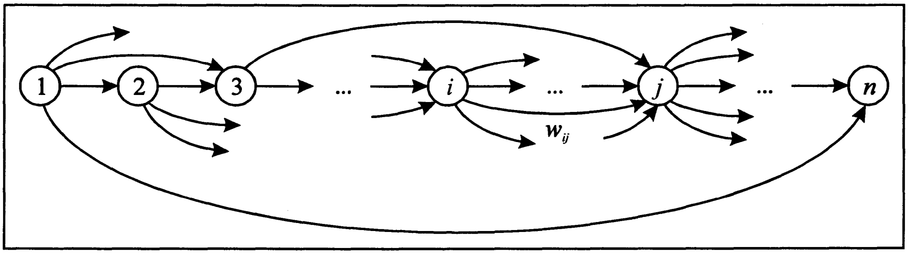

Figure 4.10. Travel patterns within one transit line having a total of n stations

We are again faced with the problem of trip distribution involving the determination of the number of trips between particular stations:

$$\text{Origin stations}\quad \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ i \\ \vdots \\ n \\ \\ \end{array} \begin{array}{c} \text{Destination stations} \\ \begin{array}{ccccccc} 1 & 2 & 3 & \cdots & j & \cdots & n \end{array} \\ \left[\begin{array}{ccccccc} 0 & f_{12} & f_{13} & \cdots & f_{1j} & \cdots & f_{1n} \\ f_{21} & 0 & f_{23} & \cdots & f_{2j} & \cdots & f_{2n} \\ f_{31} & f_{32} & 0 & \cdots & f_{3j} & \cdots & f_{3n} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ f_{i1} & f_{i2} & f_{i3} & \cdots & f_{ij} & \cdots & f_{in} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ f_{n1} & f_{n2} & f_{n3} & \cdots & f_{nj} & \cdots & 0 \end{array}\right] \\ \begin{array}{ccccccc} A_1 & A_2 & A_3 & \cdots & A_j & \cdots & A_n \end{array} \end{array} \begin{array}{c} \\ B_1 \\ B_2 \\ B_3 \\ \vdots \\ B_i \\ \vdots \\ B_n \\ \\ \end{array} \tag{4.5}$$

We denote by w_{ij} the ratio of the number of passengers boarding at station i and alighting at station j to the total number of boardings at station i that is,

$$w_{ij} = \frac{f_{ij}}{B_i} \tag{4.6}$$

It is clear that

$$\sum_j w_{ij} = 1, \quad \forall i = 1 \ to \ n-1 \tag{4.7}$$

In other words, the passenger boarding at the *i*th station will have alighting at one of the subsequent stations. The following relation is also fulfilled:

$$\sum_i f_{ij} = A_j, \ \forall j = 2 \ \ to \ \ n \tag{4.8}$$

that is,

$$\sum_i w_{ij} \ B_i = A_j, \ \forall j = 2 \ \ to \ \ n \tag{4.9}$$

Kikuchi et al. (1993) investigated the problem of finding the values of w_{ij}'s when B and A are given. (B and A represent vectors (B_1, B_2, ..., B_n) and (A_1, A_2, ..., A_n), respectively). Kikuchi et al. (1993) tried to determine the travel pattern (that is, the values of w_{ij}'s) in the case of a large number of sets for B and A. These authors departed from the basic idea that "although the values of A and B may vary for each data set, the pattern which characterizes the relationship between B and A can be captured if many sets of values for A and B exist."

Assuming that there is a consistent origin-destination travel pattern, Kikuchi et al. (1993) searched for the origin-destination matrix "which satisfies the data as much as possible." In the first part of their paper Kikuchi et al. (1993) studied the problem of the existence of a pattern, whereas the second part was devoted to the identification of the pattern.

Let us present the results achieved by Kikuchi et al. (1993) in solving the problem of pattern identification by a simple backpropagation neural network. Kikuchi et al. (1993) illustrated the application of the network by a simple example.

Note a two-way transit line with six stations (*Figure 4.11*). The values of w_{ij} are assumed to be known. These values are given in *Table 4.1*.

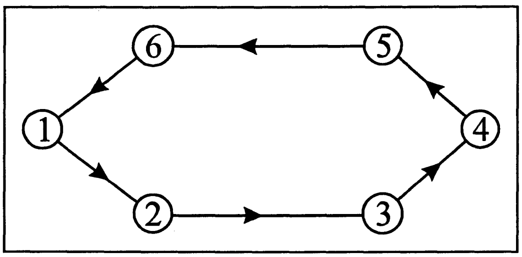

Figure 4.11. Two-way transit line with six stations

Table 4.1. The values of w_{ij}

Origin station	Destination stations 1	2	3	4	5	6
1	0	0.2	0.3	0.1	0.3	0.1
2	0.4	0	0.1	0.1	0.2	0.2
3	0.1	0.3	0	0.2	0.3	0.1
4	0.2	0.2	0.1	0	0.2	0.3
5	0.1	0.2	0.1	0.3	0	0.3
6	0.2	0.4	0.1	0.2	0.1	0

According to the values given in *Table 4.1*, we can write

$$\begin{aligned}
A_1 &= 0.4\ B_2 + 0.1\ B_3 + 0.2\ B_4 + 0.1\ B_5 + 0.2\ B_6 \\
A_2 &= 0.2\ B_1 + 0.3\ B_3 + 0.2\ B_4 + 0.2 B_5 + 0.4\ B_6 \\
A_3 &= 0.3\ B_1 + 0.1\ B_2 + 0.1\ B_4 + 0.1\ B_5 + 0.1\ B_6 \\
A_4 &= 0.1\ B_1 + 0.1\ B_2 + 0.2\ B_3 + 03\ B_5 + 0.2\ B_6 \\
A_5 &= 0.3\ B_1 + 0.2\ B_2 + 0.3\ B_3 + 02\ B_4 + 0.1\ B_6 \\
A_6 &= 0.1\ B_1 + 0.2\ B_2 + 0.1\ B_3 + 0.3\ B_4 + 0.3\ B_5
\end{aligned} \tag{4.10}$$

For the purpose of generating a large number of various data, it is assumed that quantities B_i (i = 1, 2, ..., 6) are random variables having a normal distribution with the following parameters:

$B_1 \sim$ N (3000, 50); $B_2 \sim$ N (450, 50); $B_3 \sim$ N (1800, 50);
$B_4 \sim$ N (400, 50); $B_5 \sim$ N (1800, 50); $B_6 \sim$ N (1500, 50)

There have been generated m sets of the input data for B ($B = (B_1, B_2, ..., B_6)$). For the generated values of B_i (i = 1, 2, ..., 6), according to relation (4.10) the values of A_i (i = 1, 2, ..., 6) are easily calculated. Hence, each data set describing the boarding is corresponded by a set of data describing the alighting.

Let us note a simple backpropagation network presented in *Figure 4.12*.

Using the backpropagation neural network from *Figure 4.12*, it is possible to determine the values of w_{21}, w_{31}, w_{41}, w_{51}, and w_{61}. These values provide the data on the travel patterns whose destination is station No. 1.

The network presented in *Figure 4.12* is trained by m different sets of values B_2, B_3, B_4, B_5, B_6, and A_1.

Figure 4.13 represents the backpropagation neural network used to determine the values of the travel patterns whose destination is station No. 2.

In the same manner, the values of the travel patterns whose destinations are stations No. 3, No. 4, No. 5, and No. 6 are determined.

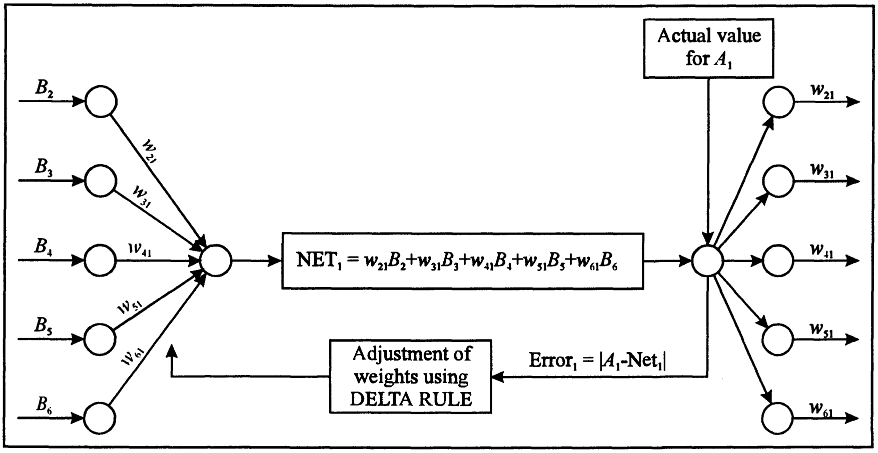

Figure 4.12. Backpropagation neural network used to determine the values w_{ij}

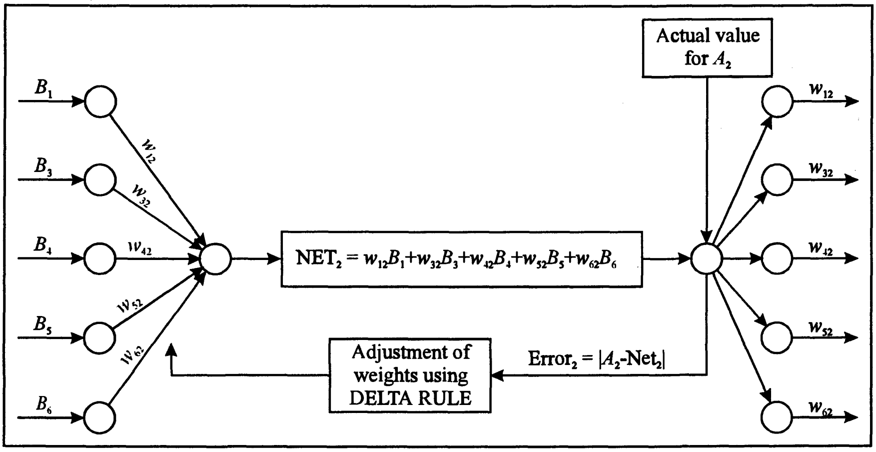

Figure 4.13. Backpropagation neural network used to determine the values of travel patterns with destination station No. 2

To obtain these values, six neural networks have to be trained (on the considered two-way transit line there are six stations). In this way, the values of w_{ij}'s were obtained satisfying the following relation:

$$\sum_i w_{ij}\, B_i = A_j \tag{4.11}$$

The following relations must also be satisfied:

$$\sum_j w_{ij} = 1, \; i = 1 \; to \; n \tag{4.12}$$

Note the backpropagation neural network shown in *Figure 4.14*.

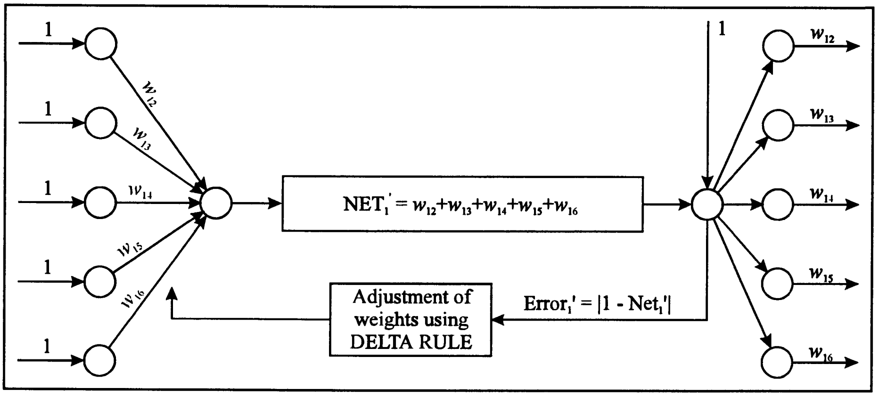

Figure 4.14. Backpropagation neural network used to determine the values of travel patterns whose origin station is No. 1

Figure 4.14 represents the backpropagation neural network used to determine the values of the travel patterns whose origin station is No. 1. The input values to this neural network equal 1. The target value is also 1, since the following relation must be satisfied:

$$1 \cdot w_{12} + 1 \cdot w_{13} + 1 \cdot w_{14} + 1 \cdot w_{15} + 1 \cdot w_{16} = 1 \tag{4.13}$$

The initial values of weights w_{12}, w_{13}, w_{14}, w_{15}, and w_{16} were obtained during the training of the six neural networks, by which the values of the travel patterns with particular destination stations were determined. Using the backpropagation neural networks similar to that in *Figure 4.14*, the values of the travel patterns whose origins are station No. 2, station No. 3, station No. 4, station No. 5 and station No. 6 are determined.

Kikuchi et al. (1993) repeatedly trained the two sets of neural networks until the prespecified convergence criterion was satisfied. The performed numerical experiments indicate that, generally, as the number of data sets increases, so does the agreement of w_{ij}'s obtained by the neural networks with the originally assumed w_{ij}'s.

It can be concluded that substantial agreements were achieved between the originally assumed data and those obtained by the backpropagation neural networks.

4.4. BACKPROPAGATION NEURAL NETWORK FOR ESTIMATING A REAL-TIME ORIGIN-DESTINATION MATRIX FROM TRAFFIC COUNTS

It has been shown in the previous discussions that a backpropagation neural network can successfully be used to estimate the origin-destination patterns in the case of one transit line. The origin-destination matrix has to be estimated in other traffic problems as well. The problem of estimating Real time origin-destination flows from traffic counts is of particular interest. A good estimation of real-time origin-destination flows in a road network or through intersections can largely influence the efficient on line traffic management and control. Yang et al. (1992) used a back-propagation neural network for the identification of real-time origin-destination flows from traffic counts. These authors made estimations of real-time origin-destination flows in the case of a typical four-way intersection, as well as in the case of a freeway section.

Let us present the basic results of Yang et al. (1992) on the example of a four-way intersection, keeping to the original notation of the authors.

A typical four-way intersection is shown in *Figure 4.15*.

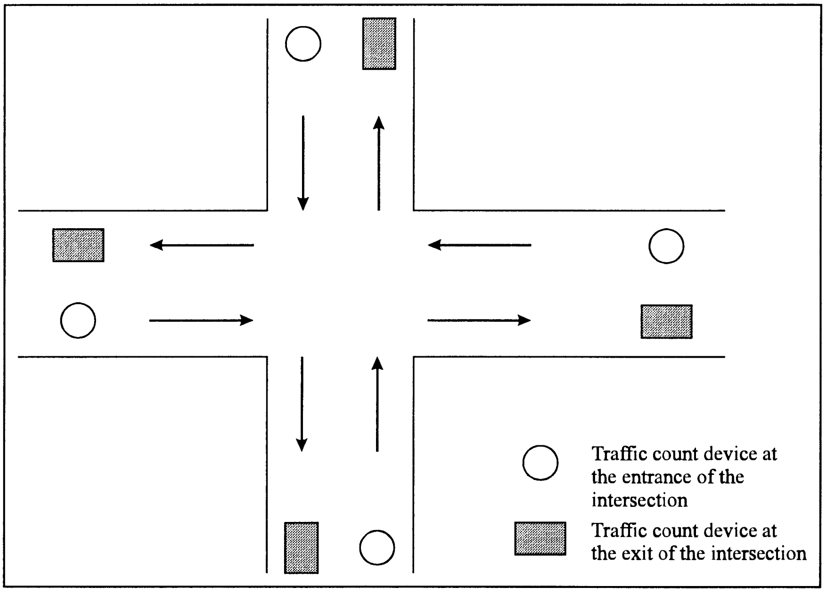

Figure 4.15. A four-way intersection

Figure 4.15 also shows the traffic count devices placed at each entrance and exit of the intersection. The time interval within which the intersection is observed is divided into k smaller time intervals (Yang et al. suggest a small time interval of 5 minutes). The traffic count devices provide the count over each small interval.

Let us assume that the transport facility has M entrances and N exits (In the case of the four-way intersection shown in *Figure 4.15* we have $M = N = 4$).

Let us denote by $q_m(k)$ the count at entrance m during interval k. Let us also denote by $y_n(k)$ the count at exit n during interval k and by $f_{mn}(k)$ the number of vehicles from entrance m to exit n during interval k. *Figure 4.16* presents the counts at particular entrances and exits as well as the number of vehicles between certain entrances and exits during interval k.

Let us denote by $b_{mn}(k)$ the percental participation of vehicles that during interval k pass through entrance M going to exit N in the total number of vehicles passing through entrance m during interval k. It is clear that

$$b_{mn}(k) = \frac{f_{mn}(k)}{q_m(k)}, \quad m = 1, 2, ..., M, n = 1, 2, ..., N \tag{4.14}$$

We also have

$$\sum_{n=1}^{N} b_{mn}(k) = 1, \quad m = 1, 2, \ldots, M \tag{4.15}$$

In other words, each vehicle entering the intersection at one of the entrances must leave the intersection at one of the exits.

The percentage of $b_{mn}(k)$ has to be nonnegative that is,

$$b_{mn}(k) \geq 0, m = 1, 2, ..., M\ n = 1, 2, ..., N \tag{4.16}$$

In some transport facilities a traffic flow between certain O-D pairs cannot occur. Let us denote by Ω the set of O-D pairs between which a traffic flow cannot occur. Clearly, we have

$$b_{mn}(k) = 0 \text{ for } (m,n) \in \Omega \tag{4.17}$$

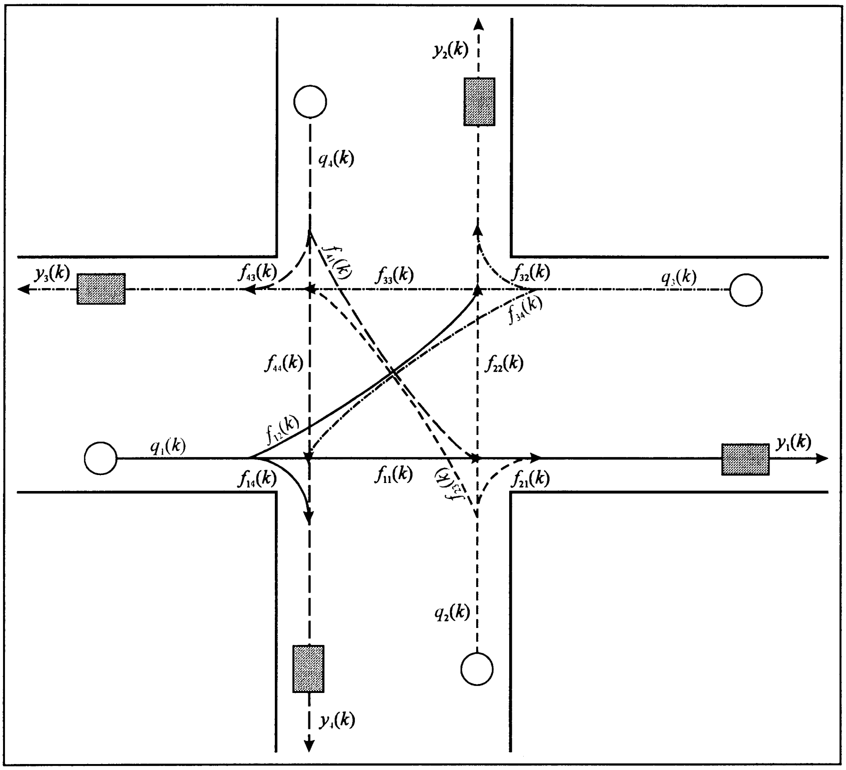

Figure 4.16. Counts at particular entrances and exits and number of vehicles between particular entrances and exits.

In a typical four-way intersection U turns are usually prohibited. It means that in our case set Ω consists of the following elements:

$$\Omega = \{(1,3), (2,4), (3,1), (4,2)\} \tag{4.18}$$

The following relation is also satisfied:

$$y_n(k) = \sum_{m=1}^{M} q_m(k) \cdot b_{mn}(k), \quad n = 1, 2, \ldots, N \tag{4.19}$$

In other words, the count at exit n during interval k is equal to the summation of the number of vehicles coming from all entrances to exit n during interval k.

As Yang et al. (1992) note, the above relation is met in the following form:

$$y_n(k) = \sum_{m=1}^{M} q_m(k) \cdot b_{mn}(k) + \zeta_n(k), \quad n = 1, 2, \ldots, N \tag{4.20}$$

where $\xi_n(k)$ denotes a random term resulting mainly from the count errors.

Yang et al. (1992) proposed that the estimation of split parameters b_{mn} be made by a simple two-layered neural network. This neural network, consisting only of the input and output layer, is shown in *Figure 4.17.* (The two-layered neural network refers to the intersection shown in *Figure 4.16*).

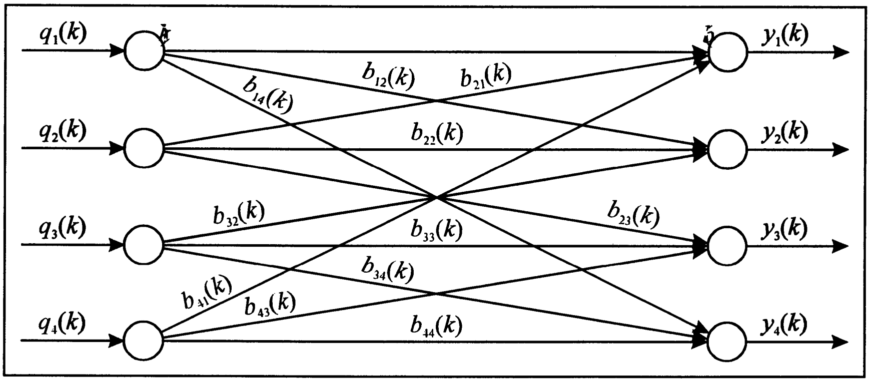

Figure 4.17. Neural network used to determine split parameters for a four-way intersection

The neural network shown in *Figure 4.17* consists only of the input and output layer. The input to the *m*th unit in the input layer is $q_m(k)$. In other words, the input to the *m*th unit in the input layer is equal to the count at entrance m during interval *k*. The connection weights from the *m*th unit in the input layer to the *n*th unit in the output layer are equal to the split parameters $b_{mn}(k)$ that we wish to estimate. Thus, upon completing the training of the neural network that is, on determining the connection weights, the values of the split parameters of the four-way intersection will be obtained.

We denote by $u_n(k)$ the input signal to the *n*th unit in the output layer that is,

$$u_n(k) = \sum_{m=1}^{M} b_{mn}(k) \cdot q_m(k), \quad n = 1, 2, \ldots, N \tag{4.21}$$

Yang et al. (1992) assumed that the distribution of traffic counts over time intervals is an approximately normal distribution that is,

$$u_n(k) \sim \mathrm{N}(h_n, w_n), \qquad n = 1, 2, \ldots, N \tag{4.22}$$

where h_n is the mean of traffic counts over time intervals at exit n, and w_n is the standard deviation of traffic counts over time intervals at exit n.

The standard normal distribution of traffic counts over time intervals $x_n(k)$ is calculated as

$$x_n(k) = \frac{u_n(k) - h_n}{w_n}, \qquad n = 1, 2, \ldots, N \tag{4.23}$$

We denote by $O_n(k)$ the output from the nth unit in the output layer. The authors calculated the values of $O_n(k)$ by a logistic function that is,

$$O_n(k) = \frac{1}{1 + e^{-x_n(k)}} \tag{4.24}$$

In a similar way, the traffic count $y_n(k)$ at exit n was normalized and the appropriate standard normal distribution $s_n(k)$ was performed:

$$s_n(k) = \frac{y_n(k) - h_n}{w_n}, \qquad n = 1, 2, \ldots, N \tag{4.25}$$

The target value $Z_n(k)$ of the nth unit in the output layer was also calculated by the logistic function:

$$Z_n(k) = \frac{1}{1 + e^{-s_n(k)}} \tag{4.26}$$

The error function E to be minimized is given by

$$E = \frac{1}{2} \sum_{k=1}^{K} \sum_{n=1}^{N} \left[O_n(k) - Z_n(k) \right]^2 \tag{4.27}$$

To determine the split parameters $b_{mn}(k)$ the fundamental learning algorithm (the error backpropagation method) was used. According to the steepest descent method, parameters $b_{mn}(k)$ should be modified proportionally to

$$\frac{\partial E}{\partial b_{mn}(k)}, \text{ that is, } \Delta b_{mn}(k) = -\varepsilon \frac{\partial E}{\partial b_{mn}(k)} \tag{4.28}$$

where ε is a learning rate coefficient between 0 and 1.

After finding the appropriate partial derivations, we obtain

$$\Delta b_{mn}(k) = -\varepsilon \frac{O_n(k)\left[1 - O_n(k)\right]\left[O_n(k) - Z_n(k)\right]q_m(k)}{w_n} \tag{4.29}$$

The process of modifying the values of parameters $b_{mn}(k)$ can be described by the following recursive formula:

$$\bar{b}_{mn}(k+1) = \bar{b}_{mn}(k) + \Delta b_{mn}(k) \tag{4.30}$$

The following recursive formula is also used

$$\bar{b}_{mn}(k+1) = \bar{b}_{mn}(k) + \Delta b_{mn}(k) + \alpha b_{mn}(k-1) \tag{4.31}$$

where α is a stabilizing coefficient between 0 and 1.

It has already been pointed out that split parameters $b_{mn}(k)$ must satisfy the following constraint:

$$\sum_{n=1}^{N} b_{mn}(k) = 1, \qquad m = 1, 2, \ldots, M \tag{4.32}$$

To that purpose, it has been suggested that the split parameters be normalized after each updating in the following way:

$$\hat{b}_{mn}(k) = \frac{\bar{b}_{mn}(k)}{\sum_{n=1}^{N} \bar{b}_{mn}(k)} \tag{4.33}$$

Yang et al. (1992) have tested the proposed model when turning movement ratios are fixed and when the split parameters are time-varying. Considering the configuration of the given intersection, it was assumed that the initial values of the split parameters were equal to

$$b_{mn} = \frac{1}{3}, \qquad m \neq n \tag{4.34}$$

The authors generated the inflows $q_m(k)$ through each intersection entrance (m=4) over the 100 time intervals (k = 1, 2, ..., 100). The inflows $q_m(k)$ are generated as random integers of the defined normal distribution.

The authors demonstrated "that the estimates of the split parameters converge to the neighborhood of their true values after about 20 iterations."

The proposed model was also tested by estimating the origin-destination flows on a freeway section using real collected data. The results were exceptionally good, so practical applications of the proposed model are to be expected.

4.5. NEURAL NETWORK APPROACH TO TRANSPORTATION NETWORK IMPROVEMENT PROBLEM

The transportation network improvement problem is undoubtedly one of the most important problems encountered by traffic planners. In order to satisfy a growing traffic demand, new links must be added to transportation networks or the capacity of the existing links must be expanded. The expansion is usually done in the circumstances of a limited budget. A transportation budget is related to a particular time period within which the improvements on the network are to be made.

Note the simple transportation network shown in *Figure 4.18*. Let us denote by v and c, respectively, the volume and capacity associated with particular links.

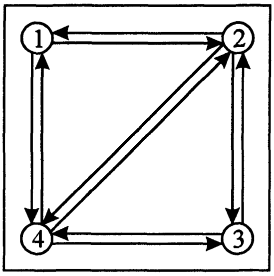

Figure 4.18. Transportation network on which particular improvements are to be made

Let us assume that the values of the volume-to-capacity ratio (v/c) on each link are high, so improvements within the network have to be made. The purpose of the improvements is, basically, to increase the capacity of particular links. Certainly, the improvements can also be made by building new links. Thus, whether a new lane has been added to an existing link, or a new link has been built, an improvement within the transportation network is assumed to have been made. Let us also assume that the transportation network improvement is to be made within the incoming three years. The branch (1,2), for instance, can be added one lane during the first year or during the second year or not before the third year. Similarly, a link (3,1) can be built during the first, second, or third year. It should be pointed out that the realization of an improvement project significantly affects the driver's behavior, route choice, traffic assignment, and total transportation costs. Are we to add lanes to branches (1,2) (2,4) and (3,2) in the second year or build a link (1,3) in the first year, and add a lane to branch (1,2) in the second year and to branch (3,4) in the third year? A multiperiod transportation network design problem is combinatorial by nature. When solving this type of problem, it is necessary to determine the combination of the improvement projects and schedule of the improvement projects that will minimize the total transportation costs in the given time period.

Various aspects of the network design problem have been considered by numerous authors (Le Blanc, 1975; Johnson et al., 1978; Poorzahedy and Turnquist, 1982; Magnanti and Wong, 1984). The approaches to solving the network design problem suggested by these authors are generally based on mathematical programming methods. It should be noted that Johnson et al. (1978) demonstrated that the incapacitated problem is a NP-hard problem.

Wei and Schonfeld (1993) developed a neural network model to solve the multiperiod network design problem. Wei and Schonfeld (1993) showed the neural network approach to be particularly good for solving the multiperiod network design problem. Let us briefly present the results achieved by Wei and Schonfeld (1993).

Each combination of the improvement projects and schedule of the improvement projects is corresponded by a particular value of the total travel time in the network. The computation of the total travel time for different combinations of projects and schedules is very time-consuming. Namely, to compute the total travel time for a particular combination, a traffic assignment procedure has to be performed. When considering a transportation network of a considerable size during an extended time period a number of traffic assignment procedures have to be performed.

Every feasible solution representing a particular improvement is characterized by the ordinal number of the link on which the improvement is being made and the year in which it is being made. After performing a particular traffic assignment procedure, for each of the feasible solutions the total travel time is calculated. The basic idea of Wei and Schonfeld (1993) consisted in choosing the subset of the solutions for which the values of the total travel time have been calculated. Wei and Schonfeld (1993) trained the proposed neural network using the subset of feasible solutions as input data and the corresponding values of the total travel time as target values. In this way, a tool was obtained by which the total travel time for any combination of projects and schedules is easily calculated.

Let us assume that traffic demand is given in terms of the peak-hour origin-destination matrix for each year. Let us also assume that traffic demand is independent of network improvements. We denote by M the total number of projects, and by T the considered time period. By keeping to the original notation of Wei and Schonfeld (1993), let us introduce in the consideration decision variable v_m for each project m (m = 1, 2, ..., M) defined in the following manner:

Vector $V = (v_1, v_2, ..., v_M)$ is the decision vector representing various solutions to the considered problem. Note the transportation network shown in *Figure 4.19*.

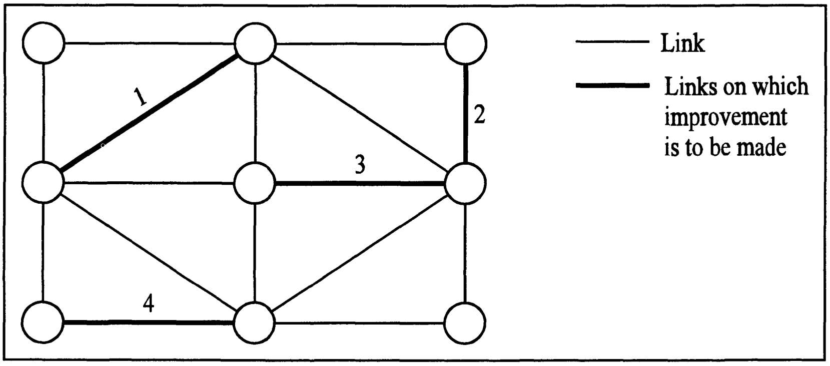

Figure 4.19. Transportation network with links on which improvement is to be made

Let us assume that improvements have to be made on links 1, 2, 3, and 4 marked in *Figure 4.19* by a heavy line. The improvements have to be made within time period of T = 5 years. Decision vector V provides full data on when the improvements on particular links are made. Thus, for example, the four (3, 1, 4, 2) means that on link No. 1 the improvement is made in the third year, on link No. 2 in the first year, on link No. 3 in the fourth year and on link No. 4 in the second year. The four (1, 1, 1, 1) denotes the situation when the improvements on all four links are made in the first year. The four (6, 6, 6, 6) denotes the case when during the five years no improvement has been made on any link.

Wei and Schonfeld (1993) proposed the following type of neural network to calculate the total travel time in the transportation network.

As seen from *Figure 4.20*, the number of neurons in the input layer is equal to the number of links in the transportation network on which the improvements are to be made.

It is clear that vector (1, 1, 1, 1) is corresponded by the smallest value of the total travel time. This vector corresponds to the so-called strategy of implement-all. All other vectors are corresponded by different values of the total travel time. Wei and Schonfeld (1993) proposed as the output value the difference between total travel time corresponding to the input vector and total travel time corresponding to vector (1, 1, 1, 1). In this manner, a better accuracy is achieved than in the case of the output value being corresponded by the value of the total travel time.

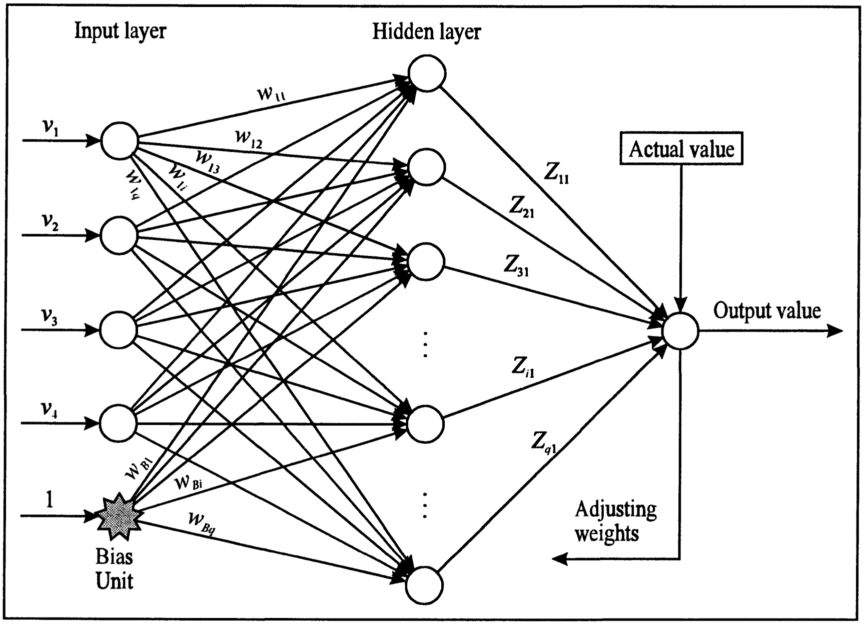

Figure 4.20. Neural network used to calculate total travel time

Let us denote by N a total number of solution used to train the neural network. Most of the solutions were generated randomly. Let us also denote by O_i, T_i, the output and target values when the input vector is V_i. The squared error is

$$SQE = \sum_{i=1}^{N} (O_i - T_i)^2 \tag{4.35}$$

The average percent error (*APE*) is

$$APE = \frac{1}{N} \sum_{i=1}^{N} \frac{|O_i - T_i|}{T_i} \cdot 100\% \tag{4.36}$$

When training a neural network, we tend to minimize the values of *SQE* that is, the values of *APE*.

Wei and Schonfeld (1993) demonstrated that a very simple artificial neural network can be a powerful tool for predicting the total travel time in the transportation network. In this way, the solving of a complex multiperiod network design problem is facilitated to a considerable extent.

4.6. ROUTING AND SCHEDULING ALGORITHM FOR THE DYNAMIC DIAL-A-RIDE PROBLEM BASED ON NEURAL NETWORKS

We have considered the dial-a-ride problem in the context of fuzzy relations as well as in the context of fuzzy arithmetic rules. We have presented in detail the scheduling algorithm for the static dial-a-ride problem based on fuzzy arithmetic rules, which was developed by Kikuchi (1992). Different versions of the dial-a-ride problem share the following characteristics:

- Each ride request is characterized by the origin and destination of movement and the time moment or time interval within which the client is to be picked up.
- The service is usually done by vehicles of the same capacity.
- In vehicle route designing the organizer seeks to carry out the planned rides by the smallest possible number of vehicles, to design the set of routes having the shortest length, to achieve the smallest possible delays, and so on.

As already pointed out, in the dynamic version of the dial-a-ride problem the routes are designed in real time. Having received the call of a new client demanding the service, the dispatcher immediately redesigns the routes that is, attempts to "insert" the new request into one of the existing routes. The dynamic version of the dial-a-ride problem is called by some authors the vehicle dispatching problem. In considering the transportation of the elderly and handicapped persons, police, ambulance, and fire department services, different delivery services, parcel pickup service, and courier service we are faced with different vehicle dispatching tasks.

Vehicle dispatching is performed by dispatchers who supervise the operations and who primarily rely on their experience and intuition. The dispatchers usually have little time to make their decisions and often work under a lot of pressure.

Shen et al. (1993) used neural networks to solve the vehicle dispatching problem in the case of a courier service company.

Let us present the results achieved by Shen et al. (1993). A specialized courier service company receives calls from clients who demand that a priority mail be picked-up and then distributed at a particular location. The clients also specify the time moments at which the pickup and delivery are to be done. Let us note that these times are not considered to be the so-called hard constraints but rather preferences.

When a new customer calls the dispatch office, the decision whether the new demand is to be accepted or not has to be made. The dispatcher has to make his decision in real time. He is given the information concerning the location of particular vehicles, the customers who have already been served and the planned serving of certain customers by particular vehicles. (*Figure 4.21*).

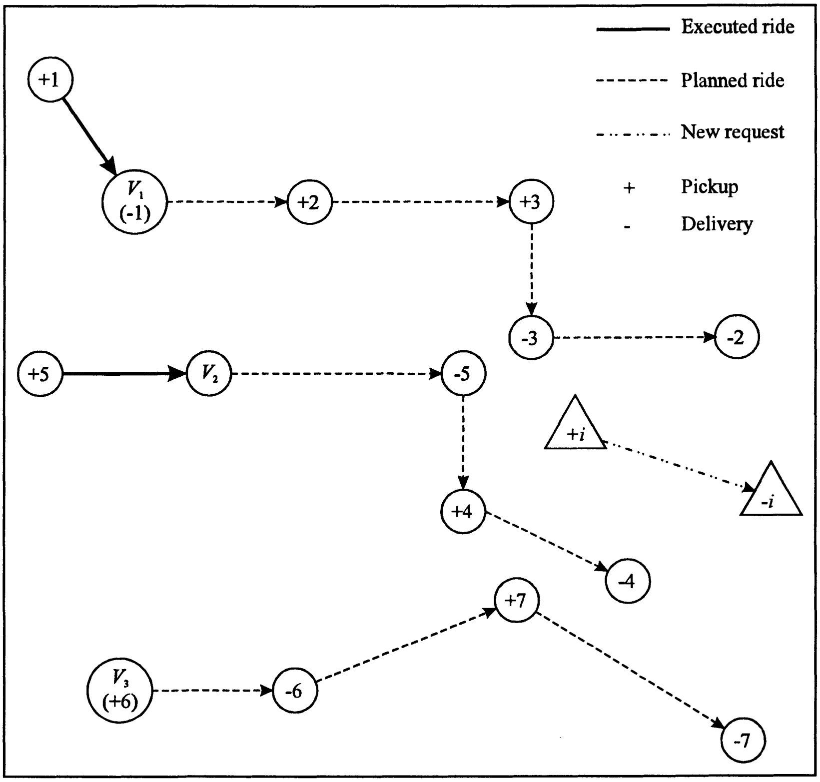

Figure 4.21. Executed and planned rides at the moment of *i*th client making a service request

As shown in *Figure 4.21*, vehicle v_1 has taken over the priority mail from user No. 1, delivered the priority mail from user No. 1 and, at the moment of the new request, is about to make for user No. 2, then user No. 3, and so on. At the moment of the new request, vehicle v_2, having taken over the priority mail from user No. 5, is on the way to deliver this mail and take over the mail from user No. 4. At the moment of the new request, vehicle v_3 is taking over the priority mail from user No. 6. It should be pointed out that at the moment of the new request (request i designated by triangular symbols), the routes that is, the working assignments for all the three vehicles shown in *Figure 4.21* have already been planned. The dispatcher has to assign the new request to one of the existing vehicles. In other words, request i has to be inserted into one of the existing routes. When solving this problem, minimizing of transportation cost as well as maximizing of customers' satisfaction have to be considered.

To solve the presented problem, Shen et al. (1993) developed a neural network model capable of simulating the decision process of the dispatcher.

Shen et al. (1993) started by developing modules for request localization, estimation of travel distances, and estimation of travel times. When estimating the travel times, meteorological conditions as well as the traffic congestion and the drivers' productivity were considered.

Let, among others, nodes i, k, l, and m be planned within a route (*Figure 4.22*).

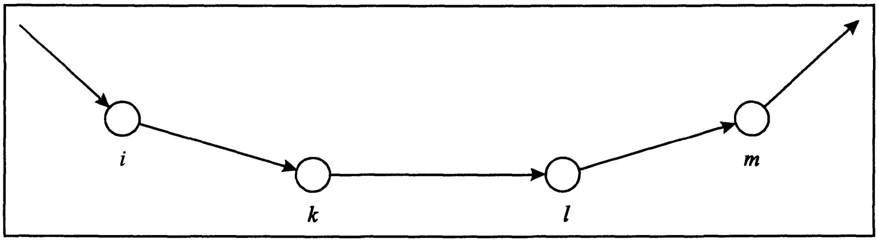

Figure 4.22. Nodes i, k, l, and m, planned within a route

It is assumed that we wish to insert pickup point $+j$ of request j between nodes i and k and delivery point $-j$ of request j between nodes l and m.

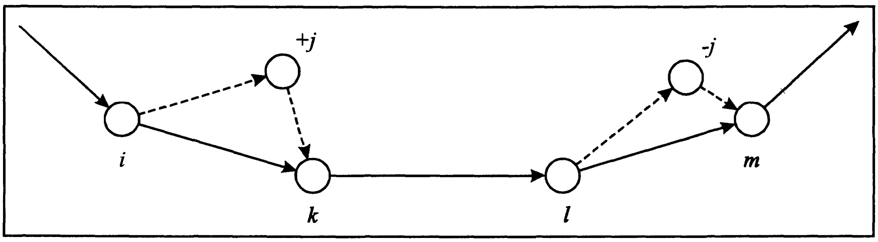

Figure 4.23. Insertion of new request j into the existing route (i, k, l, m)

The detour in time DEP for the new pickup point $+j$ equals

$$\text{DEP} = t(i, +j) + t(+j, k) - t(i, k)$$

Analogously, the detour in time DED for the new delivery point $-j$ equals

$$DED = t(l, -j) + t(-j, m) - t(l, m)$$

The detour in time DE to serve the new request j is DE = DEP + DED that is,

$$DE = [t(i, +j) + t(+j, k) - t(i, k)] + [t(l, -j) + t(-j, m) - t(l, m)]$$

Shen et al. (1993) obtained the latest pickup time of request j by adding 30 minutes to the time of call of customer j. The latest delivery time of request j was obtained by adding 90 minutes to the time of call (*Figure 4.24*).

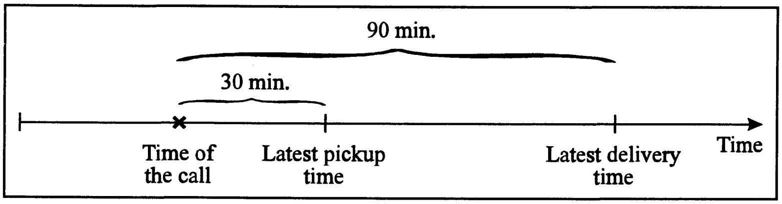

Figure 4.24. The time of call, the latest pickup time, and the latest delivery time

The lateness at pickup node $+j$ is calculated as

$$\text{lateness}_{+j} = \max\{0, \text{ service time}_{+j} - (\text{time of the call} + 30)\} \qquad (4.37)$$

The lateness at pickup node $+j$ is shown in *Figure 4.25*.

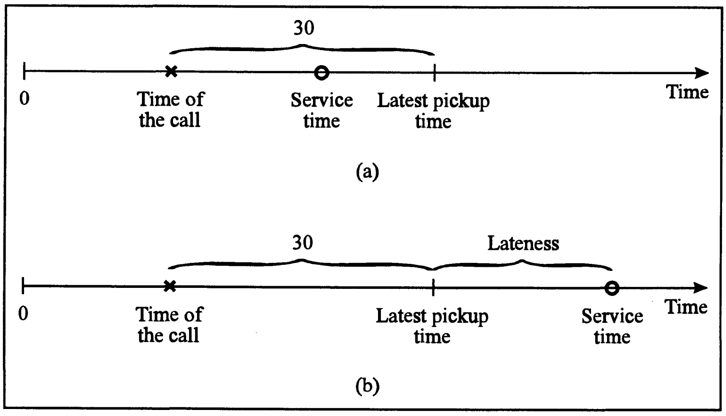

Figure 4.25. Lateness at pickup node

In the case shown in *Figure 4.25*(a) the lateness at pickup node $+j$ equals zero, as the service is carried out within the defined time interval. The case shown in *Figure 4.25*(b) is characterized by lateness, the service being carried out outside the defined time interval.

The lateness at delivery node $-j$ is calculated as

$$\text{lateness}_{-j} = \max\{0,\ \text{service time}_{-j} - (\text{time of the call} + 90)\} \qquad (4.38)$$

The formula proposed by Shen et al. (1993) to compute the insertion cost when pickup node $+j$ of request j is inserted between nodes i and k and when delivery node $-j$ of request j is inserted between nodes l and m is given by

$$c(+j, -j, i, k, l, m) = \text{DE} + \sum_u \text{lateness}_u \qquad (4.39)$$

The second term in the above relation sums up the additional lateness over $+j$, $-j$ and all the service nodes located in the route after node $+j$.

The assignment of new request to a particular route (that is, to a particular driver) produces a smaller or larger detour in time (that is, a smaller or larger lateness). The description of a dispatching situation for a single driver can be made by a vector consisting of a number of attributes. Thus, the input vector of driver j for new request i is

$$V_{ij} = (V_{ij1}, V_{ij2}, \ldots, V_{ijm})$$

where V_{ijp} is pth attribute value of driver j for request i, and m is number of attributes.

In their paper Shen et al. (1993) opted for the following nine attributes (m=9): DE is detour in time to serve the new request, PT is pickup time at the new request, DT is delivery time at the new request, APL is average additional lateness at pickup points due to the insertion of the new request, ADL is average additional lateness at delivery points due to the insertion of the new request, NPL is number of additional late services at pickup points due to the insertion of the new request, NDL is number of additional late services at delivery points due to the insertion of the new request, NR is number of requests in the route, and ET is ratio of empty travel time on total travel time.

The values of all attributes V_{ij1}, V_{ij2}, ..., V_{ijm} associated with a particular driver have to be transformed so as to be compared with the values associated with other drivers. (Indeed, a detour in time of 20 minutes cannot be estimated unless we know the detour in time of other drivers.)

Shen et al. (1993) proposed that in the first step the so-called translation be made that is,

$$V'_{ijp} = V_{ijp} - \min_{k=1,2,\ldots,n} V_{ikp}, \quad j = 1,2,\ldots,n, \; p = 1,2,\ldots,m \tag{4.40}$$

where n is the number of drivers, and m is the number of attributes.

In the next step, the obtained values are normalized between 0 and 1:

$$V''_{ijp} = \frac{V'_{ijp}}{\max\limits_{k=1,2,\ldots,n} V'_{ikp}}, \quad j = 1,2,\ldots,n, \; p = 1,2,\ldots,m \tag{4.41}$$

Shen et al. (1993) developed a learning module, the purpose of which was to suggest the best driver to serve the new request. The learning module was, in fact, a backpropagation neural network.

The neural network consisted of input layer, hidden layer, and output layer. Each layer was fully connected to the next layer via weighted connections. The input values were propagated through the weighted connections to the hidden layer where they were processed. They were then further propagated from the hidden layer to the output layer (*Figure 4.26*). The configuration proposed by Shen et al. (1993) was not typical of the standard backpropagation networks (*Figure 4.26*).

In the standard backpropagation neural network the input and output units are usually not directly connected. The network shown in *Figure 4.26* was trained with a classical backpropagation algorithm. The authors used ninety requests to train the network and fifty requests to test it.

The work of twelve drivers was observed. Hence, there were $90 \cdot 12 = 1080$ input vectors and $50 \cdot 12 = 600$ vectors for testing. Each vector consisted of nine elements. The dispatcher's decisions were collected coming from a variety of dispatching situations.

The output from neural network O_{ij} takes a value between 0 and 1. The larger the value, the greater the suitability of ith request to be assigned to jth driver. The characteristics of the new request were twelve times (there were twelve drivers) passed through the trained neural network. On sorting the twelve output values in a declining order, the appropriate rank of the drivers was obtained. The driver with the largest output value represented the proposition of the neural network. The decisions of the neural network were compared with the dispatcher's decisions. It was observed that 80% of the drivers chosen by the dispatcher were simultaneously ranked first by the neural network. It was, also, observed that the trained neural network never made an "irrational" choice of driver.

It is particularly interesting to compare the results obtained by the neural network with those obtained from the dispatcher. Shen et al. (1993) compared the productivity, the ratio of empty travel time on total travel time, the sum of lateness at the pickup points, and the sum of lateness at the delivery points. Except in the case of total lateness at the delivery points, the neural network outperformed the dispatcher.

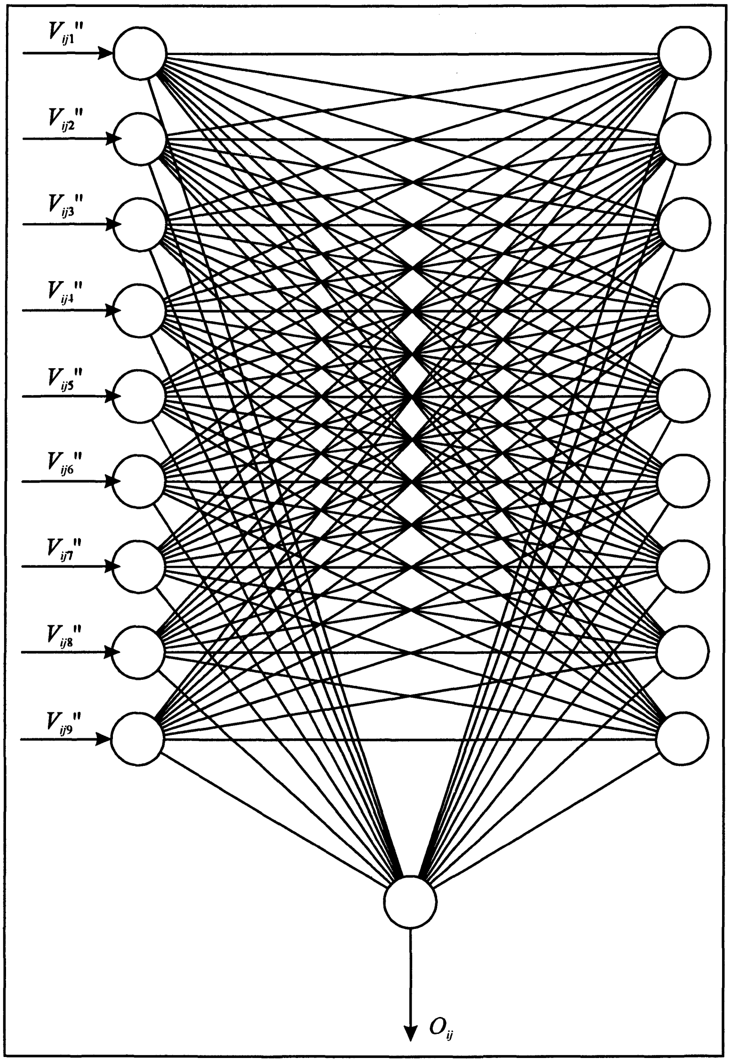

Figure 4.26. Neural network model by which suitability of driver *j* for serving *i*th request is established

4.7. A NEURAL NETWORK APPROACH TO MITIGATION OF VEHICLE SCHEDULE DISTURBANCES

Disturbances of planned schedules that the carrier is not able to anticipate appear in all transportation systems. Reasons for disturbances are various. Vehicle schedules can change daily due to delays or peaks of demands, failures of one or more vehicles, breakdowns of some transportation facilities, meteorological reasons, crew delays or absences, or traffic accidents at terminals or along the way. The carrier's policy for responding to a disturbance depends on a mode of transportation (highway, air, rail, water), distance traveled (urban transportation, suburban transportation, continental flights, intercontinental flights), and regularity of operations (regular, charter).

Whenever a disturbance in a vehicle schedule appears, the dispatcher in charge of traffic control urgently tries to minimize negative effects (trip cancellations, delays) resulting from the disturbance. Most often it is not possible to completely eliminate schedule disturbances. In other words, the dispatcher attempts to mitigate schedule disturbances. The dispatcher most often wishes to minimize the number of canceled trips if some trips must be canceled, total delay of passengers on transportation network, total vehicles delay or engagement in terminal area, and so on.

The basic objective of the research done by Vukadinovic et al. (1996) was to study the potential of using neural network approach to mitigation of vehicle schedule disturbances using a suburban railway system as an example. For example, the following situation is very common in train stations: When a planned schedule of a suburban railway system is disturbed due to a train failure, and there are no other trains available, passengers have to be moved by a reduced motive power availabilities. Based on personal experience and judgement, the dispatcher in charge of traffic control can reassign arrival and departure times of trains at stations and specify new train itineraries. During interviews with experienced dispatchers it was observed that they usually make decisions intuitively and can hardly describe motives. The dispatcher needs to be very skilled and trained properly to adequately perform the task that is, to make decisions that should considerably reduce users' losses. A need to formalize the dispatcher's judgement process arises due to a lack of experienced dispatchers and their long training process. Also, traffic control can hardly be improved without described heuristic procedure. Knowing that the consequences of wrong decisions are increased costs due to longer waiting times of users, developing a reliable automatic decision support system is essential. The dispatcher may accept or reject the guidance of the system.

The main objective of research done by Vukadinovic et al. (1996) is to examine the possibility of developing a decision support system that could decrease the workload for the dispatcher and improve the quality of decisions. Formalization of dispatcher's knowledge in terms of structured heuristic rules can be very difficult. A neural network that has the ability to adapt or learn without any knowledge of the transportation system and any logic such as if-then operations is developed. As a result of this study, it was shown that the proposed feedforward adaptive neural network with supervised learning capability can be used to simulate a dispatcher's decision-making process.

4.7.1. Statement of the problem

All transportation industries characterize vehicle schedule disturbances. Dispatchers in charge of traffic control usually are allowed a very short time to make decisions that should mitigate negative effects resulting from the disturbance. The combinatorial nature of the problem, its complexity, as well as the short available time for making decisions cause an excessive amount of stress for the dispatcher. The dispatcher is very often forced to answer the following questions:

- Can all planned trips be made with the available number of vehicles ?
- Which trips should be canceled if some trips must be canceled?
- Can total passenger delay be minimized and how?
- Does the new modified schedule satisfy all operational constraints (limited working times of labor and terminals, technical maintenance requirements)?

One possible approach to developing models to mitigate vehicle schedule disturbances is through mathematical programming that implies formulation of one or more adequate objective functions and a set of constraints. Possible objective functions that have to be minimized are total number of canceled trips, total number of passengers whose trips are canceled, total passengers delay, and total carrier's costs. The main disadvantage of this approach based on mathematical programming is that it is hard to enumerate and formulate all relevant parameters, variables, as well as "hard" constraints. Also, these traditional approaches cannot reflect the comprehensive knowledge and the intelligence of a dispatcher, particularly his or her ability to deal with uncertainty and imprecision. Teodorovic and Guberinic (1984) and Teodorovic and Stojkovic (1990, 1995) considered the problem of airline schedule disturbances. The combinatorial nature of the problem, its large dimensions, and its complexity required the use of various heuristic algorithms to obtain satisfying solutions for a relatively short time. Sforza (1989) considered the determination of trains' departure times when some disruption modifies a planned schedule. He described a multicriteria optimization module for insertion in a prototype version of decision support for railway traffic control. The problem is formulated as a mixed integer program that is solved with a branch and bound algorithm. The objective functions considered were the minimization of total delay expressed by the weighted sum of train delays, the minimization of the weighted sum of the squares of train delays, and the minimization of the total trains' engagement in the station area.

Another approach to mitigating vehicle schedule disturbances is extraction of explicit dispatcher's heuristic decision rules. It implies long and elaborate communication with a number of experienced dispatchers, which could be very difficult. It is important to mention the approximate reasoning procedures used to solve some real-time vehicle dispatching problems. Jia and Zhang (1994) considered the problem of modification of operational schedules that is, reassignment of arrival and departure times of trains at stations, which is done by a dispatcher. They developed an intelligent traffic control algorithm where the imprecise knowledge of dispatcher's decision-making behavior is realized through fuzzy decision making. Vukadinovic and Teodorovic (1994) considered the process of loading, transport, and unloading of gravel by inland water transportation. They developed a decision support system for providing assistance to a dispatcher in decisions regarding the number of picked-up and dropped-off barges at river ports. Fuzzy logic was used as a tool to transform the dispatcher's heuristic rules into an automatic strategy.

Another approach to reducing vehicle schedule disturbances is based on learning how to emulate the dispatcher's behavior. Shen et al. (1993) considered real-time dispatching of important letters in a courier service company. The task of the dispatcher is to assign each new request to one of the available drivers. Customers' preferences are stated in terms of approximate pickup and delivery times. In solving this problem, the dispatcher must find a compromise between two conflicting objectives: minimizing the operations cost and maximizing customer's satisfaction. Shen et al. developed an adaptive neural network capable of simulating the decision process of an expert dispatcher, after a training phase with this expert.

The paper written by Vukadinovic et al. (1996) deals with the Belgrade suburban railway system whose network configuration involves rays emanating from a Central Business District (CBD). Thus, all incoming trains have a CBD station (Belgrade) as destination. Following operational schedule or predetermined time diagram, inbound trains arrive and remain at CBD station for stop-times. After that, following each other, they leave CBD and depart to serve three lines of stations. Each line is served by only one electric multiple unit train (EMU), which has to travel to the end of the line before returning. Last stops or end stations are Batajnica, Pancevo and Resnik (*Figure 4.27*). When a planned schedule is disturbed due to a train failure, and there are no other trains available, only two trains can be dispatched from the CBD station. Based on personal experience and judgement, the dispatcher in charge of traffic control chooses two lines that have to be served.

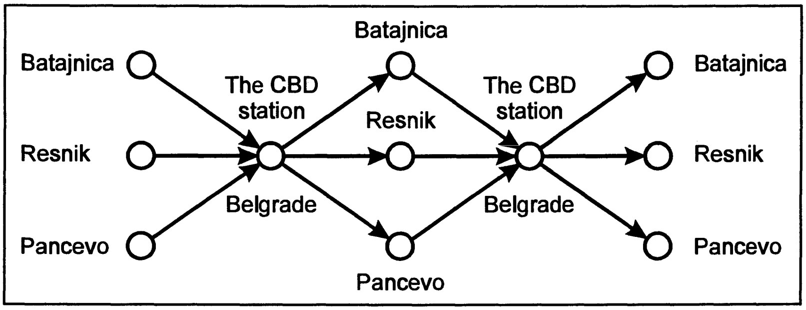

Figure 4.27. The Belgrade suburban railway system

The dispatcher's basic objective is to minimize users' losses, for example, to minimize the number of delayed passengers and total passengers' delay. Besides, the dispatcher considers the importance of serving end stations because in some specific situations one end station or a whole line may have priority over the other line or end station. For example, the schedule of trains destined to and from Pancevo is compatible with the complete railway network's schedule. Lack of trains serving that line would provide a great deal of inconvenience for passengers other than suburban, too. Therefore, the dispatcher ranks all lines that have to be served as alternatives. The worst performing line is selected as dispensable. The decision-making strategy of the dispatcher may be expressed in the rule form: as the number of passengers on one line increases compared to other two lines, as the waiting time to the next train departure on that line is longer, and the importance of serving an end station on that line is greater, the preference of a dispatcher to serve the considered line is stronger. Approaches to reproducing the dispatcher's decision strategy would be extraction of his or her explicit heuristic rules, or learning how to simulate his or her behavior in various situations. Evidently, it is easier to collect examples of the dispatcher's decisions than to formalize his or her knowledge in the form of rules.

4.7.2. Proposed solution to the problem

A need to develop a practical decision support system able to capture a dispatcher's decision-making behavior and to be implemented whenever a real-time request comes in arises due to a desire to improve the quality of decisions. A neural network is chosen due to its ability to learn from examples of collected decisions and generalize acquired knowledge. The character of adaptive neural networks that is, network structures with modifiable parameters provides the interpretation of behavior that has been attributed to the application of specific rules of thought. According to Másson and Wang (1990), "The neural network approach to solving combinatorial problems is appealing, because there is no need for an explicit formulation of the search strategy in the solution space." The greatest potential of neural networks remains in the high-speed processing, the learning capabilities and their ability to generalize from examples. The goal of supervised learning is acquisition of connection strengths that is, modifiable parameters which allows a network to behave as though it knew the dispatcher's exact rules.

The description of a proposed neural network will be proceeded after a simple question that is asked about all types of function approximations. How well does a feedforward neural network approximate an unknown function? Cybenko (1989) demonstrated that a feedforward neural network is a universal approximator. He proved that using a backpropagation least mean square error learning algorithm, layered neural networks with only one hidden layer and using sigmoid function units can closely approximate any continuous function.

4.7.2.1. Network configuration

Specifying the set of processing units connected through directed links and what they represent is typically the first stage of specifying a neural network. Wang (1994) pointed out: "To date in the literature there is no established procedure to determine the necessary and sufficient number of units for a specific application. The common practice is trial and error in nature." That is, as Wang and Archer (1994) claim, "In principle there is no need for more than one hidden layer in order to generate an arbitrary function."

Proposed neural network with a restricted connectivity structure (*Figure 4.28*) is referred to as a two-layered network because only two layers perform calculations. It is configured based on the authors' knowledge and understanding of the problem.

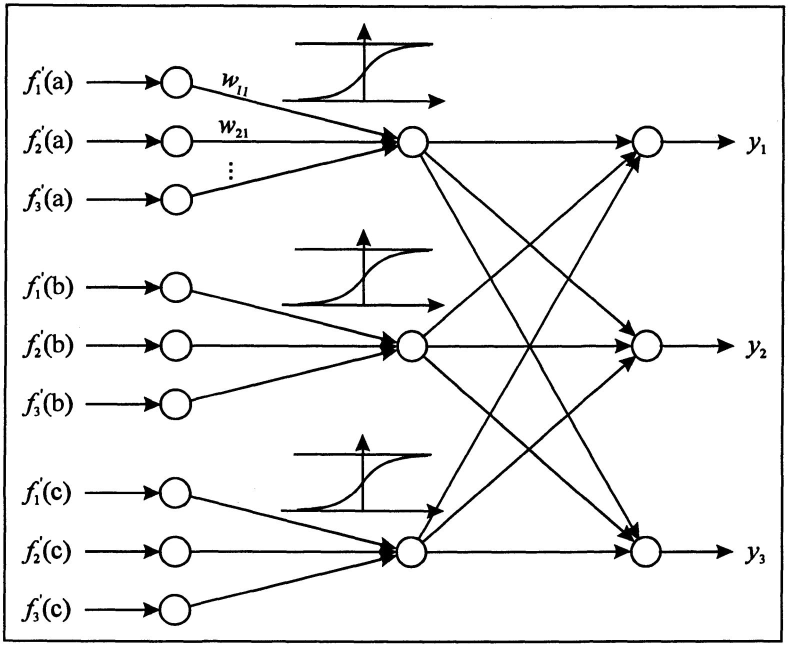

Figure 4.28. A two layered feedforward adaptive network

The adaptive network shown in *Figure 4.28* is a feedforward layered network because the output of each unit propagates from the input side (left) to the output side (right).

4.7.2.2. Pattern of connectivity

Neural networks are able to establish and compute a function from input space to output space. Thus, the examples of a dispatcher's dispatching policy are expressed as input-output vectors and considered as a training set. The dispatcher's problem may be considered as a multicriteria decision-making problem:

$$\mathrm{Max}\{f_1(x), f_2(x), f_3(x) | \ x \in \mathrm{A}\} \tag{4.42}$$

where

$$f_1(x) = f_1^*(x) + f_1^{**}(x) \tag{4.43}$$

$$f_2(x) = \max\left(f_2^*(x), f_2^{**}(x)\right) \tag{4.44}$$

where A = $\{a, b, c\}$ is the set of alternatives (that is, three lines that have to be served), $f_1^*(x)$ is the number of passengers along a line that travel toward the end station x, $f_1^{**}(x)$ is the number of passengers along a line that travel from the station x, $f_1(x)$ is the total number of passengers along a line (x is last stop) that travel in both directions, $f_2^*(x)$ is the time until departure of next train that travels toward the end station x, $f_2^{**}(x)$ is the time until departure of next train that travels from the station x, and $f_3(x)$ is the importance of serving an end station x.

The dispatcher ranks all lines that have to be served. The worst-performing line will not be served by the trains dispatched from the CBD station. Thus, a neural network able to simulate the dispatcher's ranking process is configured.

Input layer consists of the nine units that simply transfer the values of three criteria for each alternative: $f_1'(a)$, $f_2'(a)$, $f_3'(a)$, $f_1'(b)$, $f_2'(b)$, $f_3'(b)$, $f_1'(c)$, $f_2'(c)$, $f_3'(c)$. The three units in the hidden layer represent the aggregate measures of alternatives. Consider the first unit in the hidden layer. The input values $f_j'(a)$, $j = 1, 2, 3$, are multiplied by weights that is, strengths of connections between this unit and the first three units in the input layer. Those products are summed. Simultaneously, weighted sums of inputs to the hidden layer are calculated for each unit in the hidden layer. In order to obtain outputs from units in the hidden layer, weighted sums are modified by sigmoid function, which is called an *activation function*. Output layer is fully connected with the hidden layer. The network's output values y_i, $i = 1, 2, 3$, calculated as weighted sums of inputs to the second layer, represent ranks of alternatives.

Research on potential of neural networks for solving multiple-criteria decision-making problems appears often in a recent literature. For example, under assumption that decision maker's preference structure is unknown, Wang and Malakooti (1992) claim that neural network is capable of representing it. Wang (1994) proposed a neural network able to capture decision maker's fuzzy preference structure.

4.7.2.3. Learning (Least mean sum of absolute errors learning algorithm)

The goal of supervised learning is acquisition of connection strengths that is, modifiable parameters which allows a network to behave as though it knew the dispatcher's explicit heuristic rules. When known inputs are applied, a trained neural network with modified connection strengths would give outputs similar to targeted. An adaptive network's performance is measured as the deviation between the targeted output and the network's output. This deviation or the error measure is considered as the objective function and heuristic simulated annealing is used to minimize it.

The simulated annealing algorithm consists of the following steps:

Step 1: Develop a proper annealing schedule $\{t_1, t_2, ..., t_M\}$ defined as a sequence of temperatures, $t_1 > t_2 > ... > t_M$, and the amount of time at each to reach equilibrium for that temperature. Set $i = 1$.

Step 2: Generate a random set of initial connection strengths. They are uniformly distributed on an interval. For all input vectors in a training set obtain the network outputs and calculate the objective function value (OFold). Objective function minimizes the mean sum of absolute errors between the targeted outputs and the network's outputs.

Step 3: Generate a new set of strengths by small perturbation. Obtain the network's outputs with the same input vectors (for the complete training set) and calculate the new objective function value (OFnew). Evaluate the change in the objective function (δ = OFnew - OFold). If $\delta < 0$, go to Step 5. Otherwise go to Step 4.

Step 4: ($\delta \geq 0$) Compare a random variable, r, drawn from an uniform distribution on the [0,1] interval, with the probability of accepting the new set of weights $P(\delta) = \exp(-\delta/t_i)$. If $r < P(\delta)$, go to Step 5. Otherwise keep the old set of weights, and go to Step 3.

Step 5: ($\delta < 0$ or $r < P(\delta)$) Memorize the new set of weights and the new objective function value.

Step 6: If the thermal equilibrium has been reached at the temperature t_i, set $i = i + 1$. Steady state or equilibrium is reached when we observe that an improvement of the objective function is highly unlikely. An epoch is an interval between checking if the equilibrium is reached. The epoch implies λ exchanges of all connection strengths, where λ is a predefined number. The best solution that is, the least mean sum of absolute errors between the network's outputs and the targeted outputs, obtained through λ exchanges of strengths represents the epoch. Consider the case where k epochs have already been generated. After the next epoch, the equilibrium is reached if

$$\frac{\left|\mathrm{OFV}_{k+1} - \mathrm{OFV}_p\right|}{\mathrm{OFV}_{k+1}} < \varepsilon, \qquad k = 1,2,\ldots,N-1 \tag{4.45}$$

where OFV_{k+1} is the objective function value that represents the k+1-st epoch, OFV_p is the least objective function value of all previous epochs' solutions, ε is the predefined constant, and N is the maximum number of generated epochs at one temperature. If $i > M$, the algorithm is completed.

The convergence of obtained values for connection strengths is inherited from simulated annealing technique's properties.

4.7.3. Numerical example

Neural network is trained on sixty different examples of the dispatcher's decisions and tested on twenty examples. The example of one decision is shown in *Table 4.2.*

Table 4.2. The example of an arbitrary dispatcher's multicriteria decision

	$f_1(x)$	$f_2(x)$	$f_3(x)$	The rank of the alternative
Batajnica	100	65'	0.5	3
Pancevo	90	75'	0.8	1
Resnik	70	100'	0.4	2

For each given example, the number of passengers along a whole line and time until the next train departure was known for each line. A poll was conducted among experienced dispatchers in order to get the importance of serving end stations. Assigned importance belonged to the [0,1] interval. A higher value implied a greater importance of serving an end station.

For neural network training purposes, the inputs are normalized through linear transformation:

$$f_j'(x) = \frac{f_j(x) - f_{j\min}(x)}{f_{j\max}(x) - f_{j\min}(x)} \tag{4.46}$$

where $f_{j\max}(x)$ and $f_{j\min}(x)$ are the maximum and the minimum of the possible values of the criterion $f_j(x)$ across all examples.

The activation function applied to the weighted sums of the inputs to the hidden layer is in the following form:

$$\frac{1}{1 + \exp\left(-0.1 \cdot \sum_{j=1}^{3} f_j'(x) \cdot w_{jk}\right)}, \quad \forall x, x = a, b, c; \ \forall k, k = 1, 2, 3 \tag{4.47}$$

The application of simulated annealing requires a large number of experiments. The initial temperature and the number of temperatures are varied during numerous experiments. The best result is obtained when an array of 150 temperatures ($t_{i+1} = 0.95t_i$) and an initial temperature $t_1 = 30$ are used.

Initial connection strengths are arbitrarily chosen from the [0, 0.05] interval. The maximum number of generated epochs at one temperature was seventy. In other words, seventy epochs are generated if thermal equilibrium is not reached in the meantime. The epoch implied twenty seven exchanges of all connection strengths. The value of ε used to check if the equilibrium is reached was 0.02.

Regarding training pairs, the trained network completely predicts the dispatcher's preference structure in 67% of the cases. However, the lines were ranked in order to determine a dispensable line. In 85% of the cases the trained network forecasts a line that will not be served by trains dispatched from the CBD station. The network is tested on twenty examples, and the following results are obtained: in 55% of the cases the network predicts the dispatcher's preference structure, and in 75% of the cases the network indicates the worst performing line. Based on obtained results it may be concluded that proposed neural network can be used to rank given alternatives on the dispatcher's behalf. However, it cannot be used to replace human decision making.

In the considered problem, the values of three criteria for each alternative were crisp numbers. In some situations, it is too costly to precisely determine the values of criteria for each alternative, particularly the number of passengers along a whole line of stations, and the importance of serving end stations. The number of passenger along a line is usually an approximate number, so it can be represented by a fuzzy number. Besides, importances of serving end stations are subjectively determined by dispatchers. Also, dispatchers more easily handle linguistic categories such as "it is very important to serve an end station" and "it is of a little importance to serve a particular line" than crisp numbers representing importances of service. Thus, dispatchers should be allowed to express some parameters linguistically. Future research regarding mitigation of vehicle schedule disturbances should include fuzzy neural network models. A need to represent some input values to a neural network by fuzzy numbers arises due to either approximity or subjectivity of information.

The results obtained in the research done by Vukadinovic et al. (1996) may be considered encouraging. They offer possibilities for modeling complex dispatching problems in all modes of transportation. The potential of valuable returns from even small percentage of savings motivates a thoughtful interest in such models.

4.8. A NEURAL NETWORK APPROACH TO VESSEL DISPATCHING PROBLEM

Management personnel in all transportation industries faces a complex decision-making environment where a large number of planning and operational tasks have to be solved. This complexity calls for decision support systems. A decision support system could assist and guide or even completely replace a human decision maker whose reasoning is heavily dependent on his or her own personalized perceptions.

In their research Vukadinovic et al. (1997) focused on the process of loading, transport, and unloading of gravel by inland water transportation. Gravel is loaded by a suction dredger into barges. Load barges are grouped and transported by pusher tugs upstream to the ports where the unloading is carried out by the existing unloading facilities. Empty tows, each consisting of a pusher tug and assigned group of barges, navigate downstream to be loaded at the same spot.

The entire process is controlled by a dispatcher who represents an essential part of the system. Using personal experience and information at his or her disposal, the dispatcher chooses the number of barges in each tow, as well as their dispatching times. Ideally, as Jaikumar and Solomon claim (1987): "given a rank ordering of outgoing barges at every port, such as one say, based on due dates, an incoming tug into a port will dropoff a certain number of barges, say, *m*, then pickup the first *m* barges in that port's rank ordering of outgoing barges." However, instead of a desired number of barges, a tug is often assigned an *adequate* number of barges that is smaller or larger than the desired.

The decision as to what number of barges should be grouped and assigned to a tug is termed the *assignment policy*. Given the highly combinatorial nature of the problem, the determination of an optimum assignment policy is a complex issue. Knowing that the consequences of wrong assignment are increased costs due to the longer waiting times of tugs and barges or poor utilization of tugs' operating characteristics, it is essential to find an optimum assignment policy.

The main objective of the mentioned paper was to research the possibility of developing a decision support system that could decrease the workload for the dispatcher and improve the quality of decisions. During interviews with experienced dispatchers it was observed that they usually make decisions intuitively and can hardly describe motives. Since the dispatcher needs to be very experienced and trained properly to adequately perform the task, the development of an automatic system that could assist him or her seemed to be a good idea. Hoping to achieve good performance, a neural network that has the ability to adapt or learn without any knowledge of the system and any logic such as if-then operations is developed. As a result of a study, it was shown that the proposed feedforward adaptive neural network with supervised learning capability can be used to imitate the actual dispatcher's policy. The dispatcher could accept or reject the guidance of the system.

4.8.1. Statement of the problem

At a loading port, gravel is loaded into the barges by a suction dredger. The random instances of time when full barges and outgoing pusher tugs are ready to depart from the port will be termed as their release times. Information about release times is available from the daily schedule that is produced by upper-level management. At loading port, the level of management activities can be classified into operational control and strategic planning that are performed by dispatcher and upper-level management, respectively. In general, the dispatcher deals more with structured problems affected by internal factors while upper-level management is concerned with unstructured problems affected by external factors.

The key operational problem that needs to be solved by the dispatcher and the proposed decision support system is the assignment of load barges to pusher tugs. A group of barges and a tug form a tow that is departed upstream as soon as it is formed. The formed tows consist of a different number of barges due to varying operating characteristics of pusher tugs (that is, power and cost structure). Thus, for each pusher tug there is an optimum or desired number of barges to be pushed at optimum speed.

The described problem seems trivial. The dispatcher has the information regarding a daily number of load barges and the number of desired barges by pusher tugs, as well as their release times for the next 24 hours. His or her main task is to partition the total number of load barges and meet the tugs' requirements. His or her rank ordering of barges is based on their waiting times. However, the dispatcher needs to satisfy the goals of a transport company that include increased sale of gravel with minimum transport costs resulting in higher profits. This is achieved by better utilization and shorter waiting times of vessels, as well as navigation at optimum speed.

In general, disturbances of planned schedules that management personnel is not able to anticipate appear in all transportation systems. Reasons that cause disturbances are various. Vehicle schedules can change daily due to unexpected delays or peaks of demands, failures of one or more vehicles, breakdowns of some transportation facilities, meteorological reasons, crew delays or absences, or traffic accidents at terminals or along way. At the loading port, deviations from the daily schedule are common. An incoming tug often has to do some technical operations of approximate duration (change of crew, maintenance or major repair, maneuvering, preparation of barges for loading, formation of loaded tows) before departure. It implies that a planned release time of the tug may be delayed. Besides, gravel is loaded into the barges by a suction dredger. The loading times cannot be determined precisely, thus causing changes in the release times of barges.

Whenever a disturbance in a daily schedule appears, the dispatcher urgently tries to minimize negative effects resulting from the disturbance. Most often it is not possible to completely eliminate schedule disturbances. In other words, the dispatcher attempts to mitigate schedule disturbances by reassigning departure times of tows, changing the assignment of barges to tugs, forming some other tows if a tug becomes unavailable, and so on. The dispatcher most often wishes to minimize the total tugs' delay or engagement at port, total delay of barges, and so on.

As the number of load barges and tugs increases, the number of possible formed tows also increases. The dispatcher has an impression that he or she is facing a new combinatorial problem each time when making a decision. In order to properly solve the problem, he or she needs to be very experienced and skilled. In their work, Vukadinovic et al. (1997) wanted to formulate the dispatcher's assignment policy and develop an automatic system that could guide him or her in making the adequate decision. Although they were certain that a good decision requires balance between utilization of tugs' and barges' time loss in waiting, they had certain difficulties trying to extract the dispatcher's assignment policy and formulate an adequate objective function and reasonable constraints.

The dispatcher's knowledge can be considered as experience-based heuristic knowledge. That knowledge is implicit, and dispatchers that apply it neither think about it nor are aware of having it. Experienced dispatchers most often properly solve problems but they are not able to describe their motives. During structured, time-consuming interviews the authors tried to get from dispatchers information about their behavior and general strategy. "Personality and cognitive style can influence individual decision styles" (Er, 1988). Some dispatchers who may be considered conservative most often try to precisely meet the tugs' requirements. Others prefer to satisfy objectives of a transport company that include shorter waiting times of tugs and barges: the consequence is that they most often assign to each tug a little larger or a little smaller number of barges than desired. Their strategies certainly provide long-term advantage for a company.

The simplest form of knowledge representation is in the form of rules. Rules consist of two parts: the first part starts with IF and the second part continues from THEN. The first part represents the condition preceding the second part. During construction of automatic systems based on rules the trial and error procedure is most often used to update a rule base when an old set of rules does not solve problems sufficiently well. During the research done by Vukadinovic et al. (1997) it was very difficult to elicit explicit dispatcher's decision rules. Therefore, a model with the ability to learn and generalize from examples of dispatcher's decisions was needed.

4.8.1.1. Analysis of the characteristics of the dispatcher's assignment policy

At the loading port, the problem that is solved by the dispatcher is the assignment of load barges to pusher tugs. The dispatcher has the information regarding a scheduled number of load barges and the number of desired barges by pusher tugs, as well as their release times for the next 24 hours. He or she partitions the total number of load barges in order to meet the tugs' requirements as fully as possible. The problem may be considered as a multicriteria decision-making problem since the dispatcher attempts to minimize waiting times of pusher tugs and barges and maximize utilization of tugs' operating characteristics.

While talking with experienced dispatchers and monitoring their decision-making process, it is noticed that they assign different priorities to tugs depending on their power and size. Priorities are expressed in terms of allowable waiting times of tugs and their utilization. For example, for each pusher tug, the dispatcher has to answer the following questions:

- How many barges are assigned to a tug?
- Is it possible for a tug to wait for loading of a barge and, in that case, does it wait for one or more barges to be loaded?

Through examples of decisions it is realized that dispatchers consider and constrain waiting times of tugs differently (that is, allowable waiting times for bigger tugs are shorter than waiting times for smaller tugs). The consequence is that bigger tugs are very often assigned a lower number of load barges than desired due to the lack of already loaded barges or prolonged waiting times for their loading.

The dispatcher wishes to dispatch each tug with a required (complete) or an adequate (smaller or a little larger) number of barges. For example, it is not justifiable to dispatch a tug capable of handling a large number of barges with a small number of barges, but it can be reasonable to dispatch it with a medium number of barges. It implies that, at release time of a tug, if there are enough load barges at port, the required number of barges is assigned to the tug. If there is a lack of load barges, the dispatcher estimates released times of succeeding barges. If a tug's time loss in waiting for loading is estimated to be very long, he or she dispatches the tug with an incomplete but in the circumstances an adequate tow. While thinking about the considered tug, the dispatcher simultaneously anticipates decisions required for the succeeding tugs.

The dispatcher wishes to minimize waiting times of load barges. The consequence is that a somewhat bigger number of barges is assigned to a tug to avoid long waiting times of the remaining barges that are delayed by the availability of succeeding tugs.

Figure 4.29 shows an example of a daily dispatcher's assignment policy. Load barges and pusher tugs are drawn when they are released. The numbers written in parentheses represent the barges desired by the tugs that is, the tugs' requirements. Their sum is smaller than the total number of load barges. It can be seen that each tug is assigned a little larger or smaller number of barges than the desired. A group of barges and a tug form a tow. Tows are drawn when they are formed and dispatched to the unloading ports. It can be observed that the departure times of tows, dt_j, j=1, 2, 3, 4, are different from the release times of pusher tugs.

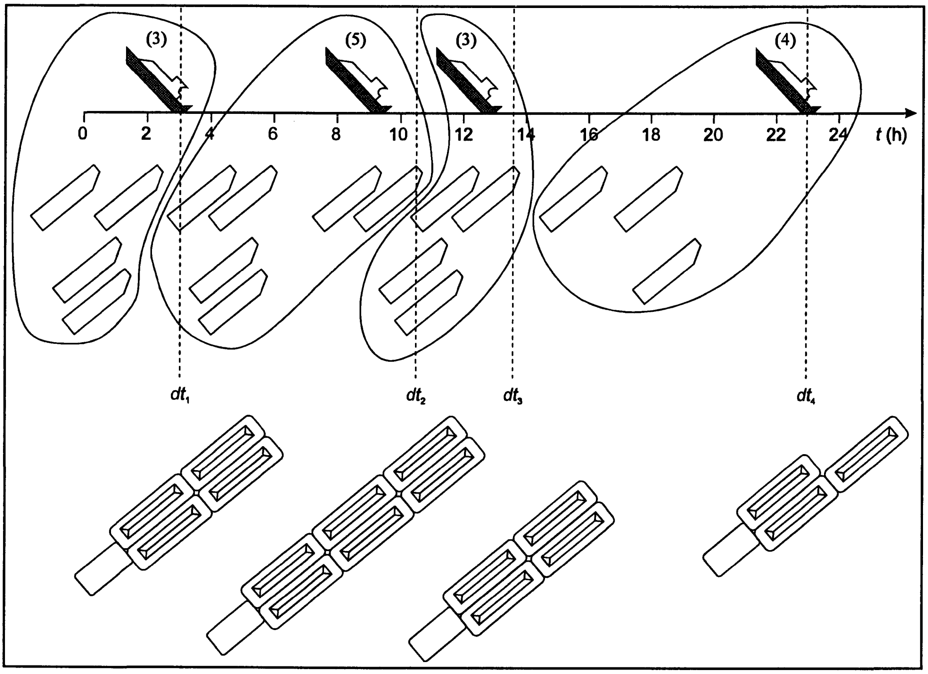

Figure 4.29. The example of the dispatcher's assignment policy

4.8.2. Proposed solution to the problem

The daily assignment of barges to tugs belongs to a class of assignment problems. The classical approach to solving these types of problems is through mathematical programming. The objective is to design a set of barge assignments to pusher tugs that incurs minimum costs, while ensuring that constraints describing proper utilization and minimum waiting times of vessels are not violated. However, there are certain disadvantages incorporated in the mathematical programming approach. The main one is the fact that it is hard to quantify the goals that motivate dispatchers that is, to formulate a reasonable objective function and hard constraints. Also, mathematical formulation leads to a very large number of variables if at a given port all possible groupings are considered. Further, the optimum solutions obtained from mathematical programming models are no longer valid in a case of unexpected delays caused by bad weather conditions (such as strong wind or fog) or breakdowns. In reality, the deviations from the daily schedule are common. Pusher tugs and barges are operated during the night (around the clock) and do not have planned idle periods that absorb delays in operations.

In real-life situations the dispatcher controls the process. This is the reason that a practical real-time decision support system with the ability to simulate the dispatcher's behavior is developed. The character of adaptive neural networks that is, network structures with modifiable parameters provides the interpretation of behavior that has been attributed to the application of specific rules of thought. In this application, the goal of supervised learning is acquisition of connection strengths that is, modifiable parameters which allows a network to behave as though it knew the dispatcher's exact rules.

4.8.2.1. Network configuration

Specifying the set of processing units connected through directed links and what they represent is typically the first stage of specifying a neural network. The size of the network is of great importance.

The adaptive neural network shown in *Figure 4.30* is a feedforward layered network because the output of each unit propagates from the input side (left) to the output side (right). The configured feedforward neural network performs a nonlinear transformation of input data in order to approximate output data. The number of processing units is determined by the nature of the dispatcher's decision-making problem.

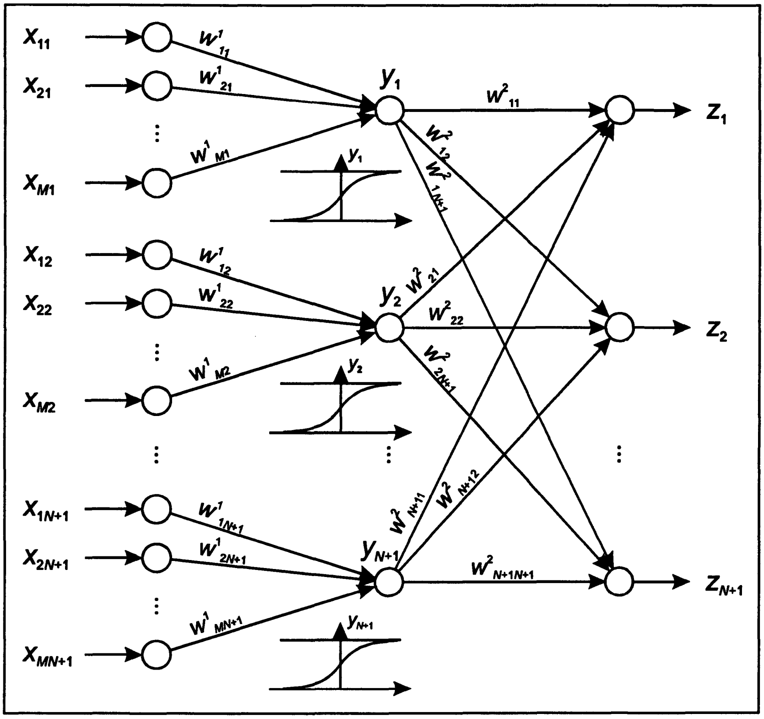

Figure 4.30. A two-layered feedforward adaptive network that is configured based on the analysis of the dispatcher's decision-making process

4.8.2.2. Pattern of connectivity

Neural networks are designed to establish and compute a function from input space to output space. The examples of the dispatcher's assignment policy are expressed as input-output vectors and training and testing pairs are formed. The raw input data are preprocessed (the term used by Croall and Mason, 1992) that is, prepared prior to their presentation to a neural network. The level of preprocessing is determined by understanding of the dispatcher's decision making process. An input vector $x = \{x_{ij} | i = 1, ..., M, j = 1, ..., N+1\}$, where M is the maximum daily number of load barges, and N is the maximum number of pusher tugs, consists of $M \times (N+1)$ elements. An element x_{ij} is defined as suitability of a barge i to be assigned to a tug j. Considering dispatcher's criteria, it is calculated as a function of a tug's j desired number of barges and a deviation between release times of a barge i and a tug j.

Although the dispatcher wishes to assign all load barges to the tugs, it may happen that he or she is forced to leave some of them unassigned. It may happen particularly when the number of load barges is much greater than the number of the barges desired by the tugs, or if tugs' time losses in waiting for the barges loaded at the end of the day are very high. This is the reason that a dummy pusher tug, $(N+1)$, being released at midnight, is introduced to pick up the unassigned barges.

The network is configured based on the analysis of the dispatcher's decision-making process. Input layer consists of $M \times (N+1)$ units. It simply transfers inputs further via the interconnections to a hidden layer. The first M inputs x_{i1}, $i = 1, \ldots, M$, represent suitabilities of all barges to be assigned to the first outgoing tug. The second M inputs x_{i2}, $i = 1, \ldots, M$, represent suitabilities of all barges to be assigned to the second outgoing tug, and so on. Hidden layer consists of N+1 units standing for the maximum daily number of pusher tugs (dummy tug included). The first M units in the input layer are connected with the first unit in the hidden layer that is related to the first outgoing tug. The second M units in the input layer are connected with the second unit in the hidden layer that is related to the second outgoing tug, and so on. Since inputs to the hidden layer represent suitabilities of all barges to be assigned to each tug, by connecting them as described, the aggregate measures of tugs' participations in the amount of load (number of barges) are obtained. Thus, each unit in the hidden layer serves as an aggregate measure of a tug's share. Each interconnection has associated with it a strength that modifies the intensity of the signal. In order to obtain outputs from units in the hidden layer, y_k, $k = 1, \ldots, N+1$, weighted sums of inputs to the hidden layer are modified by sigmoid function, that is called an activation function. According to Wasserman (1989), "Single-layer networks are severely limited in their computational capabilities. Thus, the nonlinear activation functions are vital to the expansion of the network capability beyond that of a single-layer network." Hidden layer is fully connected with output layer that also has N+1 units. The network outputs z_l, $l = 1, \ldots, N+1$ are calculated as weighted sums of inputs to the second layer. Elements of output vector represent the numbers of barges assigned to each tug.

4.8.2.3. Learning (least mean sum of absolute errors learning algorithm)

In order to recover and emulate the dispatcher's assignment policy, the aim of learning is to set connection strengths to some adequate values. When known inputs are applied, a trained neural network with modified connection strengths would give outputs similar to targeted. An adaptive network's performance is measured as the deviation between the targeted output and the network's output. This discrepancy or the error measure is considered as the objective function and the heuristic simulated annealing is used to minimize it.

Rumelhart et al. (1986) presented a popular backpropagation algorithm for supervised learning of neural networks. The algorithm they have produced is a gradient descent algorithm: it follows the slope of the error surface downward, toward the minimum. However, one of the major unsolved problems in a gradient-based learning algorithm is the problem of local minima that is, a local minimum could prohibit the algorithm to reach the global minimum. Another problem in practical applications is the problem of reasonable period of learning time.

Wang (1994) claims, "Since an error function associated with multiple training samples is usually defined as the sum of error functions of single training samples and an error function involving a single training sample corresponds to an attractor, the number of local minima (attractors) increases as the number of training samples increases. In other words, a learning process with fewer training samples has less chance to get stuck at a local minimum than that with more training samples, because the former usually has less local minima than the latter."

The objective function that has to be minimized is calculated as a mean sum of absolute differences between the network outputs and the targeted outputs over all training pairs. The heuristic simulated annealing is used to minimize the objective function. Statistical training method such as the simulated annealing requires the definition of an energy function (objective function) depending on the parameters of the neural network. Whenever a new set of strengths is generated randomly, the resulting energy is determined. If the obtained energy is improved, then the new set of strengths is memorized; otherwise, acceptance or rejection of the change is decided according to a given probability distribution. The possibility that the change that worsens (increases) the energy is retained implies that the algorithm would hardly be trapped in a local energy minima.

Some researchers experienced the effect of overtraining of networks (Croall and Mason, 1992). It happened that longer learning time did not lead to better values of the objective function; it even seemed that the network lost some of its knowledge. The possible explanation could be that the learning process got stuck at a local minimum. The fact that algorithm would hardly be trapped in a local minima that is, it will converge to a global minima was the main reason to chose this method as a learning rule. The disadvantage of using the simulated annealing algorithm as a learning rule of the neural network is a very long training period.

The simulated annealing algorithm consists of the following steps:

Step 1: Develop a proper annealing schedule $\{t_1, t_2, ..., t_K\}$ consisting of a sequence of temperatures (control parameters), $t_1 > t_2 > ... > t_K$, and the amount of time required to reach equilibrium at each temperature. Set $i = 1$.

Step 2: Generate a random set of initial connection strengths. They are uniformly distributed on an interval. For all input vectors in a training set obtain the network outputs and calculate the objective function value (OF_{old}). Objective function minimizes the mean sum of absolute errors between the targeted outputs and the network outputs.

Step 3: Generate a new set of strengths by small perturbation. Obtain network outputs with the same input vectors (for the complete training set) and calculate the new objective function value (OF_{new}). Evaluate the change in the objective function ($\delta = OF_{\text{new}} - OF_{\text{old}}$). If $\delta < 0$, go to Step 5. Otherwise go to Step 4.

Step 4: ($\delta \geq 0$) Compare a random variable, r, drawn from a uniform distribution on the [0, 1] interval, with the probability of accepting the new set of strengths $P(\delta) = \exp(-\delta/t_i)$.
If $r < P(\delta)$, go to Step 5
else keep the old set of strengths, and go to Step 3.

Step 5: ($\delta < 0$ or $r < P(\delta)$) Memorize the new set of connection strengths and the new objective function value.

Step 6: If the thermal equilibrium has been reached at the temperature t_i, set $i = i+1$. Steady state or equilibrium is reached when we observe that an improvement of the objective function is highly unlikely. An epoch is an interval between checking if the equilibrium is reached. The epoch implies λ exchanges of all connection strengths, where λ is a predefined number. The best solution that is, the least mean sum of absolute errors between network outputs and targeted outputs, obtained through λ exchanges of strengths represents the epoch. Consider the case where k epochs have already been generated. After the next epoch, the equilibrium is reached if

$$\frac{|OFV_{k+1} - OFV_p|}{OFV_{k+1}} < \varepsilon, \ k = 1, \ldots, MEP - 1 \tag{4.48}$$

where OFV_{k+1} is the objective function value that represents the k+1-st epoch, OFV_p is the least objective function value of all previous epochs' solutions, ε is the predefined constant, and MEP is the maximum number of generated epochs at one temperature. If $i > K$, the algorithm is completed.

Solutions obtained by simulated annealing do not depend on the initial solution and usually approximate the optimal solution. However, the annealing schedule (that is, the way the temperature gradually decreases) and the initial temperature influence the performance of the algorithm. Initially, a temperature is given a high value; then it is slowly reduced until some small value, for which no deteriorations are accepted any more, is reached. Thus, convergence of obtained values for connection strengths is inherited from the convergence of the simulated annealing algorithm.

The implementation of neural network results is very simple. The reliance on the obtained result will be enhanced if sufficient and representative training data are available. A good set of data is a basic requirement for working with neural networks. Some authors suggest that as many as 100 data points is needed for each unit in a hidden layer (Hall and Smith, 1992). However, many researchers report quite acceptable results from relatively small training data set. In general, the more data the better the solution. The purpose of testing a neural network behavior on real (unseen) data is to provide an evidence about adequacy of the network. As Dougherty (1995) claims: "It is important to remember that much more data are often required for training then testing; therefore where data are limited it may be better to reserve what real data are available for the testing phase, as this allows more credible results to be produced."

4.8.2.4. Network application in the case of disturbance in a daily schedule

At the loading port, the deviations from the daily schedule are common. The planned release time of a tug may be delayed due to its late arrival into the port or some technical operations of approximate duration before departure. A tug may even suddenly become unavailable if some kind of maintenance or major repair is necessary to be done. Besides, released times of barges often cannot be determined precisely due to their uncertain loading times.

Whenever a disturbance in a daily schedule appears, the dispatcher urgently attempts to mitigate schedule disturbances by reassigning departure times of tows, changing the assignment of barges to tugs, and forming some other tows if any tug becomes unavailable that is, by updating his or her assignment policy. The trained neural network has the ability to emulate the dispatcher's decision-making process even in those situations that is, to provide an adequate support. Once a network has been trained on a given set of data, there is a justified belief that it will perform reliably on new data of similar nature. If planned released times of tugs or/and barges are delayed, the trained network can be used easily to solve the problem: the raw input data will be preprocessed and presented to the network. The network will provide a result in a very short period of time. Some possible disturbances in a daily schedule are defined, and the assignments of barges to the tugs done by the dispatcher and the trained network are compared.

The network is structured to simulate the dispatcher's daily assignment policy in the case when there are maximum of M barges that have to be grouped and assigned to the maximum of N pusher tugs. However, the network can be successfully applied to any problem of a smaller size: if there are fewer than M planned barges, M_1, and fewer than N pusher tugs, N_1, during the day, inputs related to the remaining (nonexistent) barges, $M - M_1$, and tugs, $N - N_1$, are equal to 0.

Due to scarcity of real examples of the dispatcher's decisions the network was trained, and its performance was tested on few examples of smaller size.

Even if through a planning period (a day) a tug or a barge becomes unavailable, the network can be immediately used to propose a solution to a new problem of smaller size (from a certain moment until the end of the day). Therefore, the network can incorporate any unavailability of tugs or barges.

4.8.3. Numerical example

The proposed neural network is trained on fifty six examples of dispatcher's assignment policy that is, training pairs and tested on sixteen examples. The experience gained during this research indicated the importance of getting enough representative training and testing data. However, due to the high cost of information (in time and money) the number of available data was small. More real data would certainly yield a more reliable solution.

In order to form a training pair, an input vector element, x_{ij}, is calculated as a function of a tug's j desired number of barges, D_j, and a deviation between release times of a barge i, r_i and a tug j, r_j, using the following equation:

$$x_{ij} = \begin{cases} D_j^4 e^{-\frac{r_j - r_i}{144}}, r_j \geq r_i \\ D_j^4 e^{\frac{D_j(r_j - r_i)}{144}}, r_j < r_i \end{cases} \quad i = 1, \ldots, M, j = 1, \ldots, N \tag{4.49}$$

The common daily practice at port is that the number of present barges and tugs is recorded every 10 minutes. Therefore, release times, r_i, r_j, are expressed in units of 10 minutes that is, $r_i, r_j \in [0, 144]$. The suitability of a barge i to be assigned to a tug j, x_{ij}, is directly proportional to a tug's j

required number of barges meaning that there is a higher chance for a barge to be assigned to a tug when tug's requirement is larger. An exponent 4 resulted from numerous experiments: the best results are achieved with this exponent. The two exponential curves that represent suitability of a barge to be assigned to a tug are shown in *Figure 4.31.*

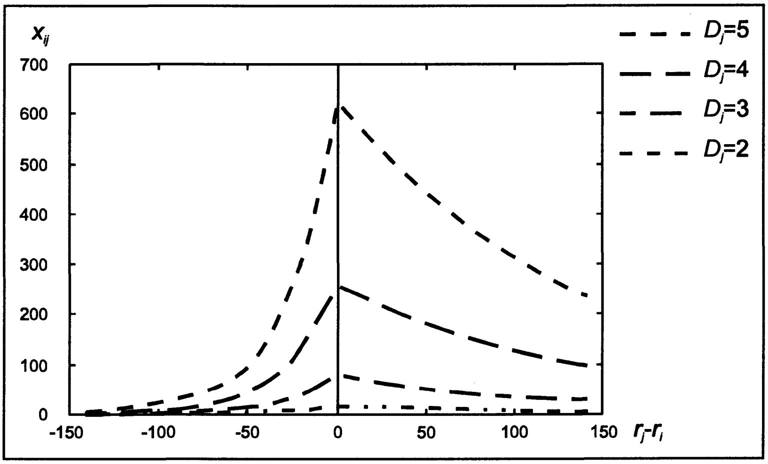

Figure 4.31. Suitability of a barge to be assigned to a pusher tug

It can be seen that as a deviation between release times of a barge and a tug, $(r_j - r_i)$, increases, the suitability of a barge to be assigned to a tug, x_{ij}, decreases. Different slopes emphasize that x_{ij} decreases much faster when a tug is waiting for a barge to be loaded than vice versa.

The suitability of a barge i to be assigned to a dummy tug N+1 is shown in *Figure 4.32* and calculated as

$$x_{iN+1} = \frac{1}{D_{N+1}^4} e^{-\frac{D_{N+1}^4 (r_{N+1} - r_i)}{144}}, i = 1, \ldots, M \tag{4.50}$$

Compared to other pusher tugs, suitability of a barge to be assigned to a dummy tug should be the least. A deviation between release times of a barge and a dummy tug is positive because a tug is released at midnight. The dummy pusher tug, *(N+1)*, is introduced to pick up the unassigned barges.

According to historical data, the daily maximum number of barges was twenty and maximum number of pusher tugs was five. Therefore, the input vector consists of 20·(5+1)=120 elements. The proposed number of units in the neural network for this application is 132. There are 156 modifiable connection strengths.

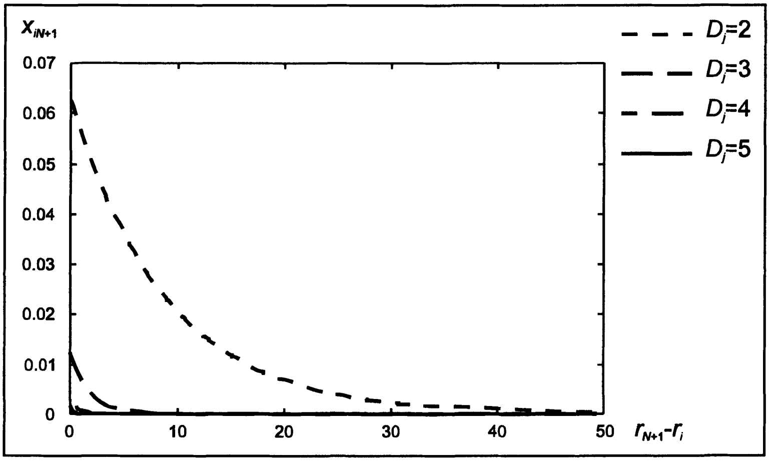

Figure 4.32. Suitability of a barge to be assigned to a dummy pusher tug

The activation function applied to the weighted sums of the inputs to the hidden layer is in the following form:

$$\frac{1}{1+\exp\left(-\sum_{i=1}^{M} w_{ij} \cdot x_{ij}\right)}, \quad \forall j, j = 1, \ldots, N+1 \tag{4.51}$$

The application of simulated annealing requires a large number of experiments. The initial temperature and the number of temperatures are varied during numerous experiments. The best result is obtained when an array of sixty temperatures (t_{i+1}=0.9t_i) and an initial temperature t_1 = 10 are used.

Initial connection strengths are arbitrarily chosen from the [0, 0.05] interval. The maximum number of generated epochs at one temperature was seventy. In other words, seventy epochs are generated if thermal equilibrium is not reached in the meantime. The epoch implied 360 exchanges of all connection strengths. The value of ε used to check if the equilibrium is reached was 0.05.

4.8.3.1. Analysis of the trained network solution

The comparison of results obtained by the dispatcher and the trained network is a multicriteria decision making problem. *Table 4.3* presents the values of the criteria that we consider to be important.

Table 4.3. Comparison of the results obtained by the dispatcher and the trained neural network

	Training pairs		Testing pairs	
	The results obtained by the dispatcher	The results obtained by the network	The results obtained by the dispatcher	The results obtained by the network
Mean waiting time of a barge (min)	180	190	290	310
Mean waiting time of a pusher tug (min)	60	70	45	35
The total number of barges left unassigned	61	66	16	23
Mean number of barges assigned to a tug asking for 5 barges	5.21	4.71	5.09	4.45
Mean number of barges assigned to a tug asking for 4 barges	4.38	4.12	4.00	3.89
Mean number of barges assigned to a tug asking for 3 barges	3.58	3.68	3.44	3.33
Mean number of barges assigned to a tug asking for 2 barges	2.64	3	2.77	3

As regards training pairs, although the trained network did not completely recover the dispatcher's assignment policy, the difference between the obtained results does not seem significant. Mean waiting times of a barge and a pusher tug, as well as the total number of barges left unassigned are similar. When a serious analysis of the dispatcher's assignment policy is done, it is concluded that his or her main concern is to assign all load barges to pusher tugs. He or she preferred to assign more barges than desired to the tugs in order not to leave some of them at the port. The consequence is that the mean numbers of barges assigned to the tugs are always larger than the desired numbers. Compared to the results obtained by the dispatcher, mean numbers of barges assigned to the tugs by the network are a little lower but may be considered satisfactory and acceptable.

It can be noted that mean waiting time of a tug is nearly three times lower than mean waiting time of a barge. In real-life situations, allowable waiting times for tugs are much shorter than waiting times for barges.

As regards testing pairs, compared to the dispatcher's results, the network gives lower mean waiting time of a pusher tug. The consequence is that the total number of barges left unassigned is larger when the problem is solved using the trained neural network. Mean waiting time of a tug is nearly nine times lower than mean waiting time of a barge when results are obtained by the network.

A proper comparison of the results obtained by the dispatcher and the trained network should include a thoughtful analysis of different importances of the criteria.

In order to check the network's performance in the case of disturbances in the planned daily schedule the example shown in *Figure 4.29* is considered and analyzed. The scheduled release times of pusher tugs requiring 3, 5, 3, and 4 barges are 180th, 560th, 760th, and 1390th minute. The barges are planned to be ready to depart from the port in 10th, 60th, 70th, 130th, 280th, 320th, 340th, 360th, 550th, 630th, 680th, 710th, 740th, 820th, 980th, 1130th, and 1160th minute. *Figure 4.29* shows the dispatcher's assignment decision. It can be seen that each tug is assigned a little larger or smaller number of barges than desired: the first tug is assigned four barges, the second tug is assigned six barges, the third tug is assigned four barges, and the fourth tug is assigned three barges. For the sake of analysis, the artificially created disturbances that can potentially influence the change in the dispatcher's decision are defined and shown in *Table 4.4*.

For the results obtained by the dispatcher and the trained network, mean waiting times of a barge and a pusher tug, as well as the total number of barges left unassigned are similar. It is certain that the decision support system in the form of the trained network needs shorter time to achieve solutions than the dispatcher. The mean sum of absolute differences between the network outputs and the dispatcher's outputs over all these examples (the error) is 0.5. It is interesting to notice how the dispatcher and the network meet the tugs' requirements: the error between the tugs' requirements and the dispatcher's assignments is 3.375, and the error between the tugs' requirements and the network's assignments is 2.875. Thus, the network solution can be considered better than that of the dispatcher's in terms of satisfaction of the tugs' requirements.

Apparently, based on the obtained results it may be concluded that the proposed neural network can be used to assign load barges to pusher tugs under the authority of the dispatcher. However, it cannot be used to replace human decision making.

One of the widely quoted potential of neural networks is high-speed processing. However, for large problems with many training pairs a lot of CPU time is needed during learning before sufficient convergence is attained. In this application, the simulated annealing algorithm that requires a very long training period is used as a learning rule of the neural network. "Whether speed can be exploited as an advantage or whether it will act as a disadvantage therefore depends on the particular application. Is it a real-time application? Does it need frequent retraining? Is the network of a size practical to be implemented in hardware and yet will save significant time?" (Croall and Mason, 1992).

Table 4.4. Comparison of the results obtained by the dispatcher and the trained neural network in the case of artificially created delays

Artificially created delays	The numbers of barges required by pusher tugs	The dispatcher's solution	The network solution
1st tug is delayed 60 minutes	3, 5, 3, 4,	4, 6, 4, 3	4, 6, 4, 3
3rd tug is delayed 80 minutes	3, 5, 3, 4,	4, 6, 4, 3	4, 6, 4, 3
1st tug became unavailable and 2nd tug is delayed 100 minutes	5, 3, 4,	6, 4, 5	6, 4, 5
4th barges is delayed 70 minutes	3, 5, 3, 4,	4, 6, 4, 3	4, 5, 4, 4
10th barge is delayed 40 minutes	3, 5, 3, 4,	4, 5, 4, 4	4, 5, 4, 4
9th barge is delayed 50 minutes	3, 5, 3, 4,	4, 6, 4, 3	4, 6, 4, 3
14th barge is delayed 100 minutes	3, 5, 3, 4,	4, 5, 4, 4	4, 5, 4, 4
8th barge is delayed 60 minutes, and 13th barge is delayed 40 minutes	3, 5, 3, 4,	4, 6, 4, 3	4, 5, 4, 4
Mean waiting time of a barge (min)	/	213	216
Mean waiting time of a pusher tug (min)	/	15	9
The total number of barges left unassigned	/	2	2

In the research done by Vukadinovic et al. (1997) the decision support system in the form of the proposed neural network that could assist the dispatcher is developed. The operational problem that needs to be solved by the developed network is the assignment of load barges to pusher tugs. Although the network solution is part of the operational control, it cannot be considered as a real-time control since it is done for the planning period of 24 hours. However, disturbances of planned schedules are very common.

Whenever a disturbance in a daily schedule appears, the dispatcher urgently attempts to mitigate negative effects resulting from the disturbance. In those situations the trained neural network has to be used and its solution implemented immediately. For such real-time dispatcher's tasks speed of obtaining a proper solution is very important. Thus, in those real-time control operations speed has proven to be a distinct advantage for neural networks.

Once a network has been trained on a large set of representative data, there is a justified belief that it will perform reliably without frequent retraining when installed in the field.

At loading port, according to a given historical data the daily maximum number of barges was twenty and the maximum number of pusher tugs was five. Therefore, in this numerical example the proposed network is structured to simulate the dispatcher's daily assignment policy in the case when there is a maximum of twenty barges that have to be grouped and assigned to the maximum of five pusher tugs. The quality of network solutions and very short time necessary to achieve them have shown that the trained network can be successfully applied to any problem of the same or smaller size. For real problems of larger size the basic structure of the network would be the same; the number of units would increase depending on the number of planned load barges and pusher tugs during the day.

While the network is applied only to a small problem, there is room for extension to problems of larger size. The information about release times of pusher tugs and barges, available from the daily schedule, was precise. However, the information at the dispatcher's disposal is most often given in a qualitative form and is not precise: "barges need from 2 to 3 hours to be loaded; a tug will be ready to depart from the port around 4 p.m." Future research regarding dispatching of tows from the loading port, consisting of a pusher tug and assigned group of barges, should include fuzzy neural network models. A need to represent some input values to a neural network by fuzzy numbers arises due to either approximity or subjectivity of information.

Chapter 5.

Generating and Tuning the Fuzzy Logic Systems Developed in Transportation Applications

A successful application of fuzzy logic implies prior determination of shapes of membership functions of input and output variables as well as generation of a fuzzy rule base. In some applications, the final set of fuzzy rules and the choice of membership functions are defined by trial and error. Mendel (1995) claims: "Prior to 1992, all fuzzy logic systems reported in the open literature fixed the parameters of the membership functions somewhat arbitrarily, e.g., the locations and spreads of the membership functions were chosen by the designer independent of the numerical training data. Then, at the first IEEE Conference on Fuzzy Systems, held in San Diego, three different groups of researchers presented the same idea: *tune the parameters of a fuzzy logic system using the numerical training data.*"

In the following sections some possible membership value assignments will be discussed.

5.1. MEMBERSHIP FUNCTIONS DESIGN

Fuzzy sets are used to describe fuzziness characterizing particular variables. Our idea of a particular fuzzy set is greatly influenced by the shape (triangular, trapezoidal, Gaussian, and so on) of its membership function. In other words, conclusions drawn about a certain phenomenon characterized by fuzziness are determined largely by the shape of its membership function.

By using membership functions we are able to determine an element's membership grade in a particular fuzzy set. When reading that is, determining the grade of membership, two cases are to be distinguished (see *Figure 5.1*).

Figure 5.1 illustrates the membership function of the fuzzy set "short travel time." Case (a) refers to the case of determining the membership grade of the value of 40 minutes in the fuzzy set "short travel time." The obtained value of 0.33 can also be interpreted as the truth-value contained in

D. Teodorović et al., *Traffic Control and Transport Planning:*

the claim that the time of 40 minutes is "short travel time." Case (b) describes the situation characterized by imprecision in the value of the element whose grade of membership we wish to determine. As shown in *Figure 5.1* case (b) refers to the situation of determining the grade of membership of the value of "about 30 minutes" that is, the truth value that "travel time of about 30 minutes" is "short travel time." The obtained membership grade of 0.57 indicates that, with the corresponding truth value of 0.57, travel time of about 30 minutes is claimed to be "short travel time."

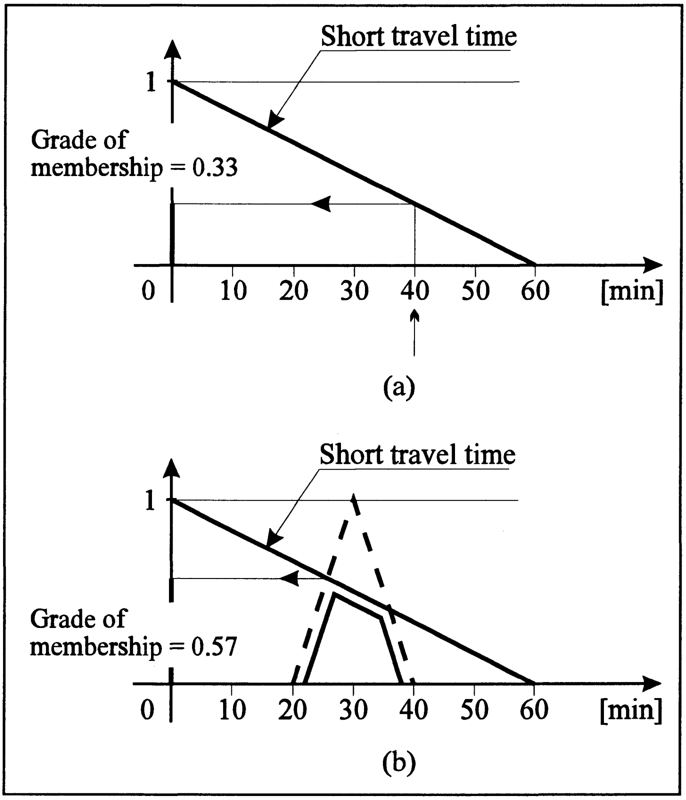

Figure 5.1. (a) Fuzzy set "short travel time" and crisp reading, (b) Fuzzy set "short travel time" and fuzzy reading

The most common shapes of membership functions are found to be triangular, trapezoidal, and Gaussian (see *Figure 5.2*).

How are membership functions determined in the first place that is, how do we arrive at the shape of a membership function? The process of membership value assigning may be based either on experience and intuition or on a particular algorithm.

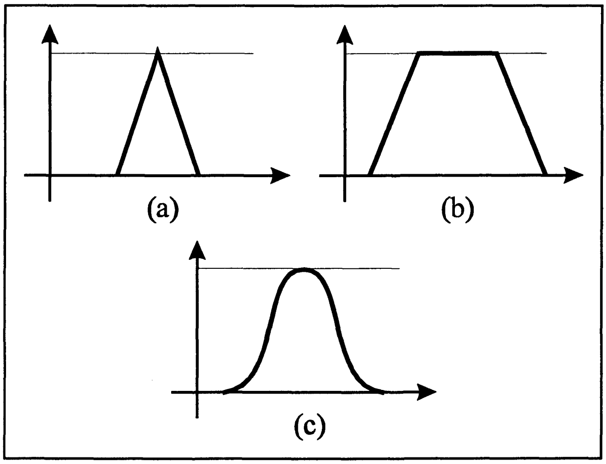

Figure 5.2. Triangular (a), trapezoidal, (b) and Gaussian (c) shapes of membership functions

5.2. MEMBERSHIP VALUE ASSIGNMENT PROCESS BASED ON INTUITION

In a number of cases, membership functions are determined subjectively by an expert, decision maker, or analyst. The subjective way of determining membership functions is based on experience, intuition, and knowledge of particular events. It is worthy of note that membership value assignments made subjectively depend on the given situation as well as on the person making the assignments. *Figure 5.3* represents "short travel time in urban transportation" and "short travel time in air transportation." As can be seen, the context in which a particular expression ("short travel time") is used considerably influences the choice of a particular membership function. Also, different individuals make membership value assignments in different ways. *Figure 5.4* represents two different membership value assignments based on two different subjective feelings about the notion of "short travel time in urban transportation."

Since "everything is a matter of degree" (Kosko, 1993), the membership value assignment calls for "tolerance" that is, inexclusion. In other words, certain membership functions are bound to overlap (*Figure 5.5*).

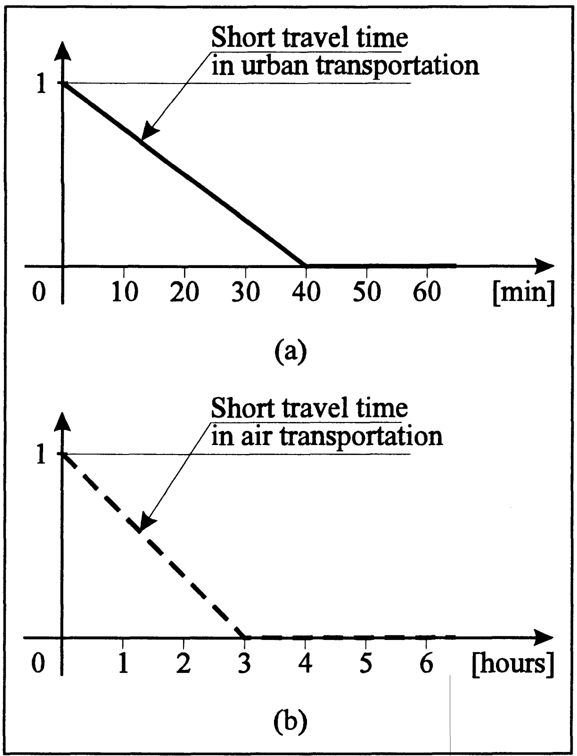

Figure 5.3. (a) Short travel time in urban transportation, (b) short travel time in air transportation

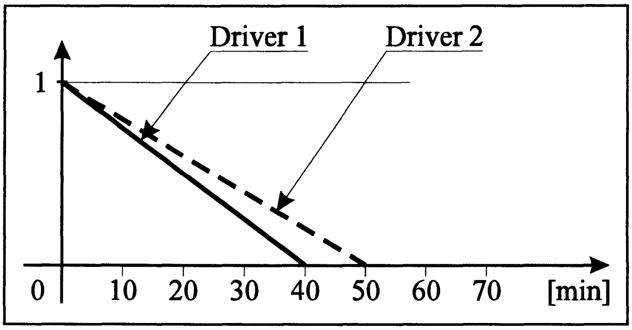

Figure 5.4. Two membership value assignments for "short travel time in urban transportation" obtained by two different drivers

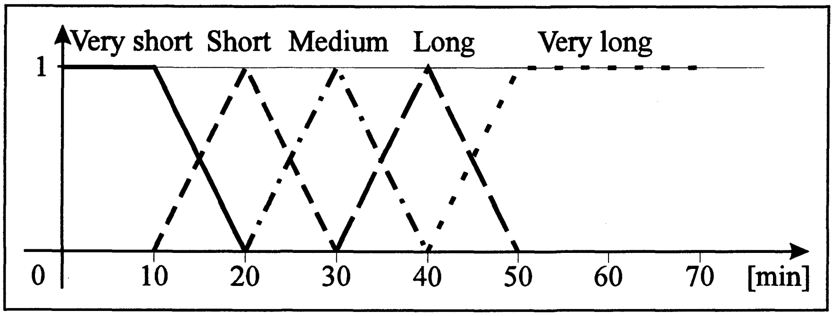

Figure 5.5. Overlapping membership functions related to travel time [min] in urban transport

As shown in *Figure 5.5,* a particular travel time can be both short and medium having, however, different grades of membership. Overlapping between certain membership functions reflects the way in which particular events are most commonly observed as well as the very nature of the theory of fuzzy sets.

5.3. MEMBERSHIP VALUE ASSIGNMENT PROCESS BASED ON FACTS AND KNOWLEDGE

In the previous discussion we pointed out various aspects of the membership value assignment process based on intuition. In certain situations, however, in making the appropriate membership value assignments we often draw on our knowledge and the facts available. Let us illustrate this case by the following example:

Figure 5.6 depicts the so-called grid network in part of a town. For measuring distance in such type of a network the so-called Manhattan distance is applied. Note nodes *I* and *J,* as shown in *Figure 5.6.* Let us denote by m(*I*, *J*), the "Manhattan" distance between node *I* and node *J.* This distance is found to be $m(I, J) = |x_i - x_j| + |y_i - y_j|$.

Suppose we wish to define the membership function of a fuzzy set referred to as "short distance between nodes." We can rely on our knowledge and the available facts. We know that through a "grid" network we cannot move in a straight line. We are also acquainted with a way of measuring distances in such a network ("Manhattan" distance). If nodes *I* and *J* coincide, the distance between them equals zero and the grade of membership in the fuzzy set called "short distance between nodes" is found to be 1. As the "Manhattan" distance between the nodes increases, the corresponding grade of membership will decrease. Let us denote by $\mu_{SD}(I, J)$ the membership function of the fuzzy set called "short distance between nodes." Considering the stated facts, one way in which the desired membership function can be represented is as follows:

$$\mu_{SD}(I,\ J) = \frac{1}{1 + |\, x_i - x_j \,| + |\, y_i - y_j \,|} \tag{5.1}$$

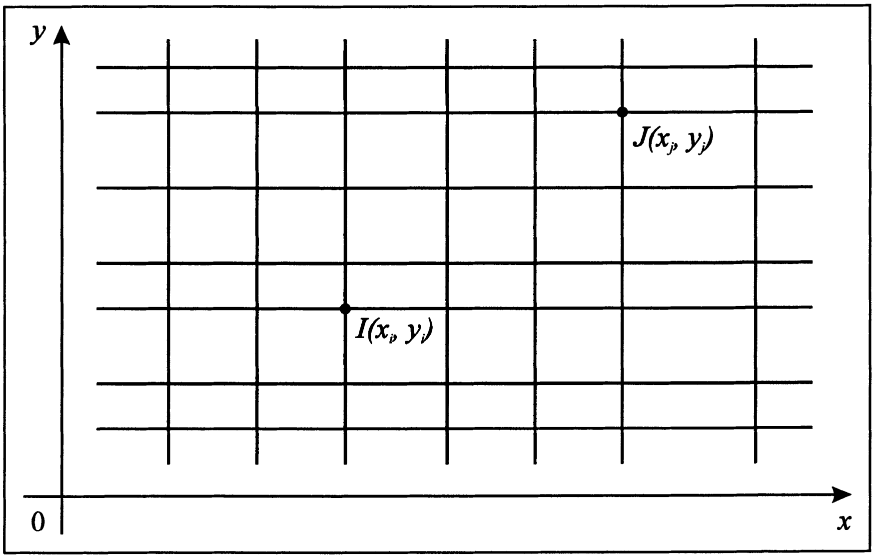

Figure 5.6. "Grid" network in part of a town

5.4. MEMBERSHIP VALUE ASSIGNMENTS AS A COMBINATORIAL OPTIMIZATION PROBLEM

When the membership value assignment process is treated as a combinatorial optimization problem, the role and importance of an expert in the stage of membership functions design is reduced. Most of the combinatorial optimization problems are difficult to solve. These are, most commonly, found to be NP-complete problems that cannot be solved exactly in polynomial time. A typical representative of this type of problems is the *traveling salesman problem* (TSP). In recent years, the so-called metaheuristic algorithms have increasingly been used in solving difficult combinatorial optimization problems. These include the *simulated annealing* (Metropolis et al., 1953; Kirkpatrick et al., 1983; Cerny, 1985), *genetic algorithms* (Holland, 1975; Goldberg, 1989), and *tabu search* (Glover, 1986; Glover and Laguna, 1993).

Fuzzy logic systems estimate sampled functions from input to output. In order to fix shapes and locations of membership functions using genetic algorithms, simulated annealing, and tabu search, a representative numerical input/output data set is needed. For different categories of input/output variables the initial shapes and locations of membership functions are assumed. The objective function that is minimized is usually calculated as the difference between output data and the results obtained by the developed fuzzy logic system using the same input data. In other words, when solving a combinatorial optimization problem, the goal is to achieve the best possible fitness of the considered membership functions of input/output variables.

5.4.1. Tuning membership functions using simulated annealing

The simulated annealing technique is one of the methods increasingly used in the last few years in solving complex combinatorial problems. Kirkpatrick et al. (1983) and separately Cerny (1985) were the first to suggest the simulated annealing technique in solving combinatorial optimization problems. This method is based on the analogy with certain problems in the field of statistical mechanics found in the studies of Metropolis et al. (1953). The name of the simulated annealing comes from the analogy with physical processes. The process of annealing consists in decreasing the temperature of a material, which in the beginning of the process is in the molten state, until the lowest state of energy is attained. At certain points during the process the so-called thermal equilibrium is reached. In the case of physical systems we seek to establish the order of particles that has the lowest state of energy. This process requires that the temperatures at which the material remains for a while be previously specified.

The basic idea of simulated annealing consists in performing small perturbations (small alterations in the positions of particles) in a random fashion and computing the energy changes between the new and the old configurations of particles, ΔE. In the case when $\Delta E < 0$, it can be concluded that the new configuration of particles has lower energy. The new configuration then becomes a new initial configuration for performing small perturbations. The case when $\Delta E > 0$ means that the new configuration has higher energy. However, in this case the new configuration is not to be automatically excluded from the possibility of becoming a new initial configuration. In physical systems, "jumps" from lower to higher energy levels are possible. The higher the temperatures, the higher the probability that the system will "jump" to a higher energy state. As the temperature decreases, the probability that such a "jump" will occur diminishes. Metropolis et al. (1953) used Boltzmann's distribution of probabilities to compute the probability of a "jump." Probability P that at temperature T the energy will increase by ΔE equals

$$P = e^{-\frac{\Delta E}{T}} \tag{5.2}$$

The decision whether a new configuration of particles for which $\Delta E > 0$ should be accepted as a new initial configuration is made upon the generation of a random number r from the interval [0, 1]. (Generated random number is uniformly distributed). If $r < \mathrm{P}$, the new configuration is accepted as a new initial configuration. In the opposite case, the generated configuration of particles is excluded from consideration.

In this manner, a successful simulation of attaining thermal equilibrium at a particular temperature is accomplished. Thermal equilibrium is considered to be attained when, after a number of random perturbations, a significant decrease in energy is not possible. Once thermal equilibrium has been reached, the temperature is decreased, and the described process is repeated at a new temperature.

The described procedure can also be used in solving combinatorial optimization problems. A particular configuration of particles can be interpreted as one feasible solution. Likewise, the energy of a physical system can be interpreted as the value of an objective function, while temperature assumes the role of a control parameter.

5.4.1.1. Tuning membership functions using simulated annealing: Traffic signal coordination by fuzzy logic

Let us consider a one-way street with two lanes. Suppose that along the observed section of the street there are no side streets.

Pavkovic (1993) demonstrated the possibility of using fuzzy logic to determine a *green time offset*. Green time offset refers to the time interval from the moment the green phase begins at one signal to the moment it begins at the next signal. In the *simple progressive system*, all offsets should be arranged in order to let a vehicle entering the system, in the direction of progression just after the green initiation of the first signal, to arrive at all other signals just after their green initiations. The offset between two signals can be calculated by dividing the signal spacing by speed of progressing vehicles. It is usually computed by some of the widely used computer programs, such as TRANSYT, MAXBAND, PASSER II, MULTIBAND, and SCOOT. Pavkovic (1993) developed a simple approximate reasoning algorithm for determining a speed of progression. This algorithm is based on the fact that under the uninterrupted flow condition, up to the point of critical density, the average overall speed of traffic decreases as the traffic volume increases.

Consider the following fuzzy sets: "small volume-to-capacity ratio" (**SR**), "medium *v/c* ratio" (**MR**), "large *v/c* ratio" (**LR**), "very large *v/c* ratio" (**VLR**), "small speed of traffic" (**SS**), "medium speed of traffic" (**MS**), "large speed of traffic" (**LS**), and "very large speed of traffic" (**VLS**).

The approximate reasoning algorithm reads:

If volume-to-capacity ratio = **SR**, then speed = **VLS**,
or
If volume-to-capacity ratio = **MR**, then speed = **LS**,
or
If volume-to-capacity ratio = **LR**, then speed = **MS**,
or
If volume-to-capacity ratio = **VLR**, then speed = **SS**.

The green time offset is easily computed by dividing the signal spacing, L, by the obtained speed of traffic, u.

Pavkovic (1993) arbitrarily defined the membership functions of the introduced fuzzy sets (triangular and trapezoidal fuzzy numbers), applied different values of *v/c* ratios through the approximate reasoning algorithm, and computed the appropriate speeds and green time offsets. After analyzing the obtained results, certain membership functions were modified, a new testing of the approximate reasoning algorithm was performed, and so on. The final results were found to be in substantial agreement with those obtained by the computer program PASSER II. It can be concluded that the computation of the membership functions was performed by trial and error. Continuing work of Pavkovic (1993), let us show how these membership functions can be determined with the aid of the simulated annealing technique.

Table 5.1 represents different values of traffic volumes, v, corresponding volume-to-capacity ratios and appropriate speeds obtained by the given approximate reasoning algorithm. The distance between the traffic signals is assumed to be $L = 300$ m. Due to simplicity, the street capacity is considered to be crisp, $c = 3200$ vehicles/h, although it could be described by a fuzzy number.

Table 5.1. Values of traffic volume, v/c ratio and corresponding speed

Traffic volume v (vehicles/h)	v/c ratio	Speed u (km/h)
160	0.05	57
320	0.10	57
480	0.15	57
640	0.20	57
800	0.25	51
960	0.30	48
1120	0.35	47
1280	0.40	45
1440	0.45	42
1600	0.50	40
1760	0.55	39
1920	0.60	35
2080	0.65	26
2240	0.70	22
2400	0.75	20
2560	0.80	13
2720	0.85	13
2880	0.90	13
3040	0.95	13

The shapes of the membership functions that produced results given in *Table 5.1* were obtained by trial and error. Let us assume that values in *Table 5.1* are obtained by measurements ("recorded" values).

As can be seen, an observed mapping from the input (volume-to-capacity ratio) to the output (speed of traffic) is expressed through the developed approximate reasoning algorithm. The numerical input/output data set is shown in *Table 5.1*. The shapes of membership functions of the input and output variable are assumed to be trapezoidal. The choice of their locations is based on the related input/output data intervals that are known from experience (*Table 5.2*).

Table 5.2. Related intervals containing v/c and u values

Interval containing v/c values	Interval containing speed values, u (km/h)
$0 \leq v/c < 0.3$	$50 \leq u < 60$
$0.3 \leq v/c < 05$	$40 \leq u < 50$
$0.5 \leq v/c < 0.7$	$30 \leq u < 40$
$0.7 \leq v/c < 1$	$u \leq 20$

The initial shapes of the membership functions are marked in *Figure 5.7* by a solid line. The dashed lines denote the "boundary" shapes of the membership functions. By varying the parameters (intersections or cross points of the trapezoid sides and the x axis), different membership functions are obtained. Our task consists in finding the set of membership functions of input and output variable such that the speed values obtained by the approximate reasoning algorithm are "as close as possible" to the "recorded" values. Each parameter is assigned an interval within which it is likely to occur. Thus, for example, the initial intersection of the left trapezoid side (representing the membership function of fuzzy set **LR**) and the x axis is point (35,0). This parameter can be found between points (20,0) and (40,0). To solve the given problem more easily, let us perform the discretization of the intervals within which a parameter can occur. Each interval will be divided into small intervals whose width makes 5% of the initial width. Let us assume that parameters can be found only at the points representing the ends of the small intervals.

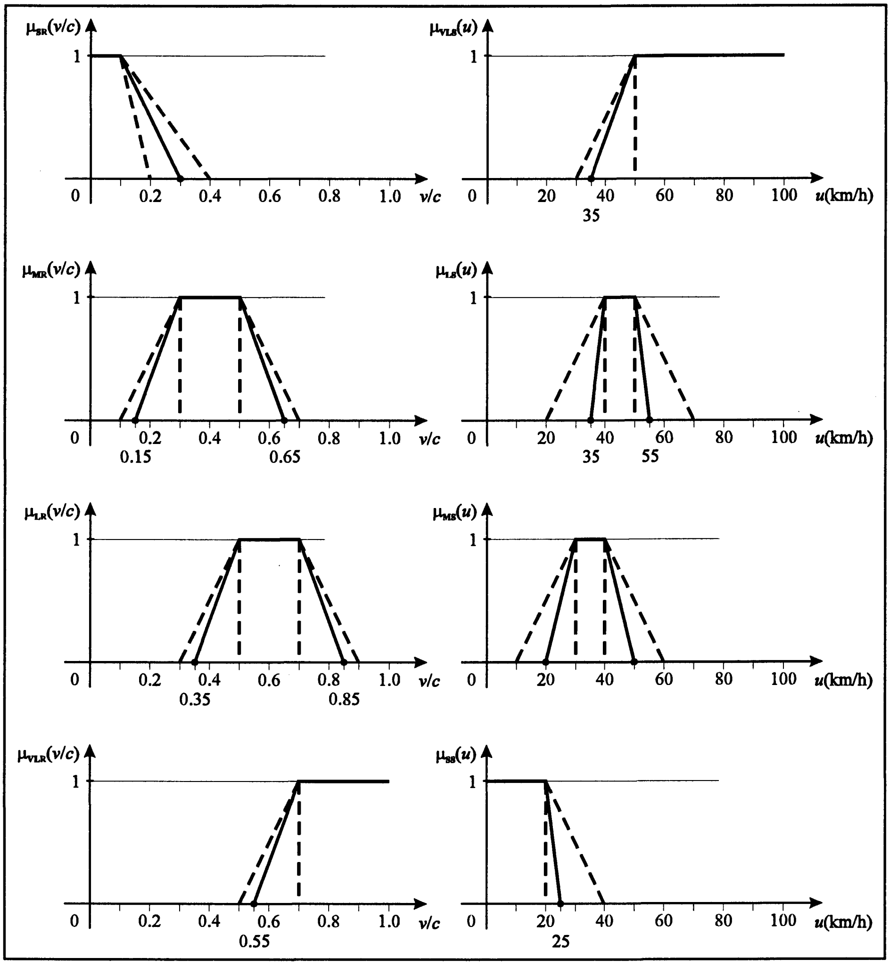

Figure 5.7. Initial membership functions of fuzzy sets: (a) **SR, MR, LR, VLR** (b) **SS, MS, LS, VLS**

It can be seen from *Figure 5.7* that the positions of twelve parameters have to be determined (membership functions of fuzzy sets **SR**, **VLS**, **VLR**, and **SS** have one, while membership functions of fuzzy sets **MR**, **LS**, **LR**, and **MS** have two intersections each). We assign to the parameters of the membership functions shown in *Figure 5.7* the following ordinal numbers: **SR** (1), **VLS** (2), **MR** left intersection (3), **MR** right intersection (4), **LS** left intersection (5), **LS** right intersection (6), **LR** left intersection (7), **LR** right intersection (8), **MS** left intersection (9), **MS** right intersection (10), **VLR** (11), and **SS** (12).

A move is understood to be a simultaneous movement of the positions of two parameters. The positions of other ten parameters remain unchanged. Let us first change the position of parameter 1. This change is made in a random manner. From the set of parameters {2, 3, ..., 12} we randomly choose the next parameter whose position is to be changed. This change is also performed randomly. When making the next move, we randomly change the position of parameter 2. From the set of parameters {1, 3, ..., 12} we choose a parameter whose position will be randomly changed. This is followed by a change of the position of parameter 3, while the next parameter whose position is to be changed is chosen from the set of parameters {1, 2, 4, ..., 12}, and so on. In this manner, it is possible for each of the parameters to change its position. Let one epoch consist of twenty moves (this number is arbitrarily determined). It has also been decided that at any temperature up to thirty epochs can be executed. A change in temperature without the execution of all thirty epochs may occur if thermal equilibrium is attained. Thermal equilibrium is assumed to be attained if

$$\left| \frac{L_p - L_i}{L_p} \right| \leq \varepsilon \tag{5.3}$$

where L_p is a value of the objective function in the last epoch; i is an index referring to previous epochs at current temperature, i=1, 2, ..., p-1; and ε is an arbitrarily determined constant (in this example, $\varepsilon = 0.05$).

The minimization of the objective function is performed at fifty temperatures. Let the initial temperature be $T_1 = 40$. The temperature is assumed to change according to the following relation: $T_i = 0.9T_{i-1}$, $i = 2, 3, \ldots, 50$. It should be noted that the initial temperature, total number of temperatures and temperature schedule are determined arbitrarily. These values are usually obtained after numerous numerical experiments, taking into account the context of the considered problem.

Using the approximate reasoning by max-min composition, for each of the nineteen values of *v/c* ratio (*Table 5.1*) the corresponding speed value can be calculated. The obtained speed values can be compared with the "recorded" speed values (*Table 5.1*) and the corresponding absolute deviations can be calculated. The objective function used in this procedure represents the sum of absolute deviations estimated values from the "recorded" speed values. Each time a move was made that is, each time the positions of parameters of two membership functions were changed it was necessary to calculate the nineteen speed values using the approximate reasoning algorithm and to determine the corresponding value of the objective function.

Figure 5.8 represents the membership functions before and after the application of simulated annealing.

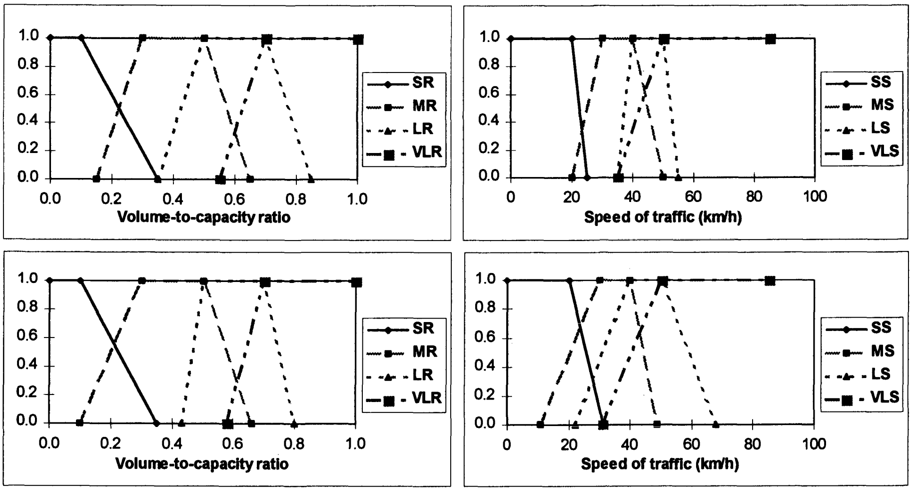

Figure 5.8. The initial (first row) and the final (second row) membership functions of input and output variable

Figure 5.9 represents the change of the objective function relative to the change in temperature.

Table 5.3 shows the "recorded" speed values and the speed values obtained by the approximate reasoning algorithm where the final shapes of the membership functions of input and output variable are determined using the simulated annealing.

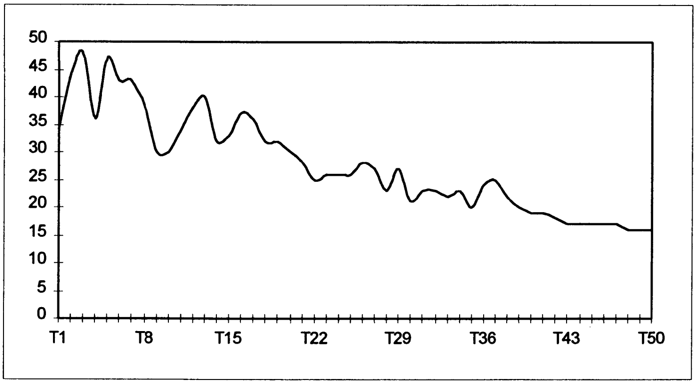

Figure 5.9. Change in the objective function relative to temperature

Table 5.3. "Recorded" speed values and speed values obtained using approximate reasoning algorithm when the membership functions of input and output variable are tuned using simulated annealing

Recorded speed	Obtained speed	Recorded speed	Obtained speed	Recorded speed	Obtained speed
57	62	45	45	20	20
57	62	42	42	13	13
57	60	40	40	13	13
57	57	39	39	13	13
51	52	35	35	13	13
48	48	26	26		
47	45	22	22		

5.4.2. Tuning membership functions using genetic algorithms

Genetic algorithms represent search techniques based on the mechanics of nature selection used in solving complex combinatorial optimization problems. These algorithms were developed by analogy with Darwin's theory of evolution and the basic principle of the "survival of the fittest." The most significant results in the field of genetic algorithms were achieved by Holland (1975) and Goldberg (1989).

In the case of genetic algorithms, as opposed to traditional search techniques, the search is run in parallel from a population of solutions. In the first step, various solutions to the considered maximization (or minimization) problem are generated. In the next step, the evaluation of these solutions that is, the estimation of the objective (cost) function is made. Some of the "good" solutions yielding a better "fitness" (objective function value) are further considered. The remaining solutions are eliminated from consideration. The chosen solutions undergo the phases of *reproduction*, *crossover* and *mutation*. After that, a new generation of solutions is produced to be followed by a new one, and so on. Each new generation is expected to be "better" than the previous one. The production of new generations is stopped when a prespecified stopping condition is satisfied. The final solution of the considered problem is the best solution generated during the search.

In the case of genetic algorithms an encoded parameter set is used. Most frequently, binary coding is used. The set of decision variables for a given problem is encoded into a bit string (chromosome, individual). Let us explain the concept of encoding in the case of finding the maximum value of function $f(x) = x^3$ in the domain interval of x ranging from 0 to 15. By means of binary coding, the observed values of variable x can be presented in strings of the length 4 (since $2^4 = 16$). *Table 5.4* shows sixteen strings with corresponding decoded values.

Table 5.4. Encoded values of variable x

String	Value of variable x	String	Value of variable x
0000	$0 = 0*2^3 + 0*2^2 + 0*2^1 + 0*2^0$	1000	$8 = 1*2^3 + 0*2^2 + 0*2^1 + 0*2^0$
0001	$1 = 0*2^3 + 0*2^2 + 0*2^1 + 1*2^0$	1001	$9 = 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0$
0010	$2 = 0*2^3 + 0*2^2 + 1*2^1 + 0*2^0$	1010	$10 = 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0$
0011	$3 = 0*2^3 + 0*2^2 + 1*2^1 + 1*2^0$	1011	$11 = 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0$
0100	$4 = 0*2^3 + 1*2^2 + 0*2^1 + 0*2^0$	1100	$12 = 1*2^3 + 1*2^2 + 0*2^1 + 0*2^0$
0101	$5 = 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0$	1101	$13 = 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0$
0110	$6 = 0*2^3 + 1*2^2 + 1*2^1 + 0*2^0$	1110	$14 = 1*2^3 + 1*2^2 + 1*2^1 + 0*2^0$
0111	$7 = 0*2^3 + 1*2^2 + 1*2^1 + 1*2^0$	1111	$15 = 1*2^3 + 1*2^2 + 1*2^1 + 1*2^0$

We assume that in the first step the following four strings were randomly generated: 0011, 0110, 1010, and 1100. These four strings form the initial population P(0). In order to make an estimation of the generated strings, it is necessary to decode them. After decoding, we actually obtain the following four values of variable x: 3, 6, 10, and 12. The corresponding values of function $f(x) = x^3$ are equal to $f(3) = 27, f(6) = 216, f(10) = 1000$ and $f(12) = 1728$. As can be seen, string 1100 has the best fitness value.

Genetic algorithms are procedure where the strings with better fitness values are more likely to be selected for mating. Let us denote by f_i the value of the objective function (fitness) of string i. The probability p_i for string i to be selected for mating is equal to the ratio of f_i to the sum of all strings' objective function values in the population:

$$p_i = \frac{f_i}{\sum_j f_j} \tag{5.4}$$

This type of *reproduction* that is, selection for mating represents a proportional selection known as the "roulette wheel selection." (The sections of roulette are in proportion to probabilities p_i). In addition to the "roulette wheel selection," several other ways of selection for mating have been suggested in the literature.

In order to generate the next population P(1), we proceed to apply the other two genetic operators to the strings selected for mating. *Crossover* operator is used to combine the genetic material. At the beginning, pairs of strings (parents) are randomly chosen from a set of previously selected strings. Later, for each selected pair the location for crossover is randomly chosen. Each pair of parents creates two offsprings (*Figure 5.10*).

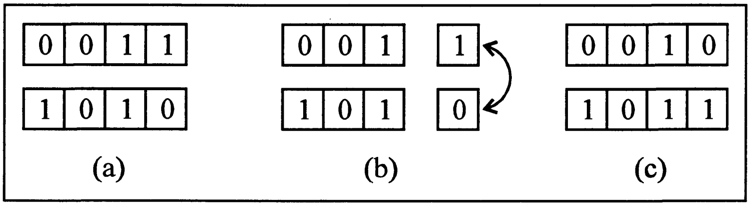

Figure 5.10. A single-point crossover operator: (a) two parents (b) randomly chosen location is before the last bit (c) two offsprings

After completing crossover, the genetic operator *mutation* is used. In the case of binary coding, mutation of a certain number of genes refers to the change in value from 1 to 0 or vice versus. It should be noted that the probability of mutation is very small (of order of magnitude 1/1000). The purpose of mutation is to prevent an irretrievable loss of the genetic material at some point along the string. For example, in the overall population a particularly significant bit of information might be missing (for example, none of the strings have 0 at the seventh location), which can considerably influence the determination of the optimal or near-optimal solution. Without mutation, none of the strings in all future populations could have 0 at the seventh location. Nor could the other two genetic operators help to overcome the given problem.

Having generated population P(1) (which has the same number of members as population P(0)), we proceed to use the operators reproduction, crossover, and mutation to generate a sequence of populations P(2), P(3), and so on.

In spite of modifications that may occur in some genetic algorithms (regarding the manner in which the strings for reproduction are selected, the manner of doing crossover, the size of population that depends on the problem being optimized, and so on), the following steps can be defined within any genetic algorithm:

Step 1: Encode the problem and set the values of parameters (decision variables).

Step 2: Form the initial population P(0) consisting of n strings. (The value of n depends on the problem being optimized). Make an evaluation of the fitness of each string.

Step 3: Considering the fact that the selection probability is proportional to the fitness, select n parents from the current population.

Step 4: Randomly select a pair of parents for mating. Create two offsprings by exchanging strings with the one-point crossover. To each of the created offsprings apply mutation. Apply crossover and mutation operators until n offsprings (new population) are created.

Step 5: Substitute the old population of strings with the new population. Evaluate the fitness of all members in the new population.

Step 6: If the number of generations (populations) is smaller than the maximal prespecified number of generations, go back to Step 3. Otherwise, stop the algorithm. For the final solution chose the best string discovered during the search.

5.4.2.1. Tuning membership functions using genetic algorithms: Traffic signal coordination by fuzzy logic

Consider the problem presented in Section 5.4.1.1. We will show how to tune the membership functions of the input and output variable using genetic algorithms such that the speed values obtained by the approximate reasoning algorithm (presented in Section 5.4.1.1) are "as close as possible" to the "recorded" values.

The initial shapes of the membership functions of the input and output variable are designated in *Figure 5.7* by a solid line. As in the previous case, the dashed lines denote the "boundary" shapes of the membership functions. By varying the intersections of the trapezoid sides and the x axis, different membership functions are obtained. Our task is to tune the initial membership functions of the input and output variable such that the output speed values obtained by the approximate reasoning algorithm are "as close as possible" to the "recorded" speed values. As pointed out in the previous discussions, the intersections or parameters can be found only within certain intervals. In our example there are twelve parameters that is, twelve intervals. Each of the twelve intervals is divided into $2^5 = 32$ smaller intervals (arbitrarily selected division). Binary numbers that are assigned to each interval are joined to make one string in a population (*Table 5.5*).

Table 5.5. Division of 12 intervals into small intervals and their fusion into one large interval

00101	10010	...	010010
1st interval	2nd interval	...	12th interval

These binary numbers denote the positions of parameters. For example, the binary number 00101 corresponds to the decimal number 5, since $0*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 5$. This means that the first parameter belongs to the fifth small interval within the first interval. The binary number 10010 corresponding to the decimal number 18 means that the second parameter belongs to the eighteenth small interval within the second interval, and so on. In this way, any set of membership functions can be determined by a set of parameters encoded into a string. The set of initial membership functions (denoted in *Figure 5.7* by a solid line) is encoded into the basic string. Other strings in the first population are generated by randomly changing the position of each parameter within the appropriate interval.

From a population, the reproduction was performed based on the "roulette wheel selection." The crossover was made by choosing two strings in a random fashion that mutually exchange their genetic material. The position (ordinal number in the string) where the exchange of the genetic material was carried out was chosen randomly (*Figure 5.11*).

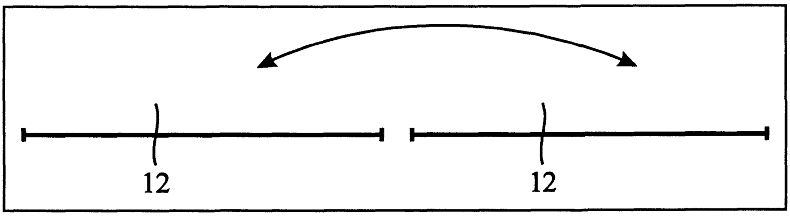

Figure 5.11. Exchange of genetic material

In the first step, the two strings shown in *Figure 5.11* were randomly singled out. In the next step, number 12 was randomly drawn. This means that a mutual exchange of the thirteenth, fourteenth, fifteenth, and other corresponding members of the first and second string will occur. The mutation (change of zero to one or one to zero) of certain members of the string occurred with the probability of 0.001.

The fitness function used for evaluation of strings is defined as follows:

$$f_i = \frac{1}{\sum_{j=1}^{19}\left(u_{fj} - u_j\right)^2} \tag{5.5}$$

where u_{fj} is *j*th speed obtained by the approximate reasoning algorithm, and u_j is "recorded" *j*th speed (*Table 5.1*). Each time the shapes of the membership functions of input and output variable were changed it was necessary to calculate the nineteen speed values by the approximate reasoning algorithm and to calculate the appropriate value of the fitness function.

Figure 5.12 displays the membership functions of input and output variable before and after the application of genetic algorithms.

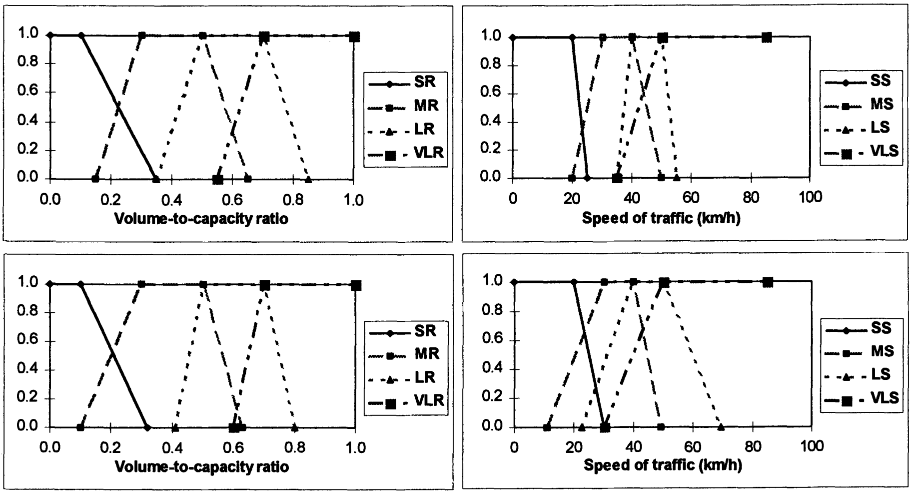

Figure 5.12. The initial (first row) and the final (second row) membership functions of input and output variable

Figure 5.13 represents the change of the fitness function relative to the number of generations.

Table 5.6 shows the "recorded" speed values and the speed values obtained by the approximate reasoning algorithm where the initial shapes of the membership functions of input and output variable are tuned using genetic algorithms.

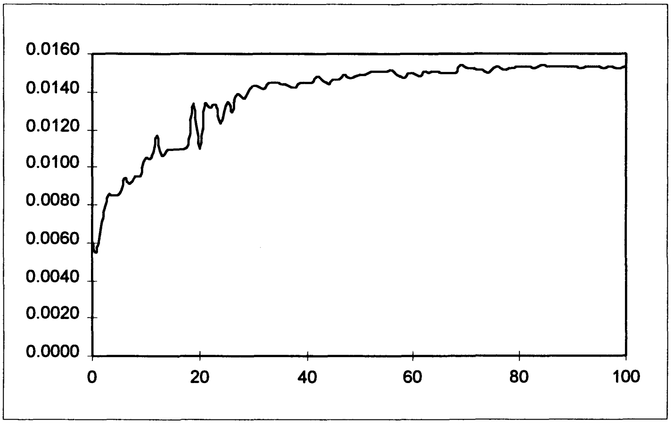

Figure 5.13. The change of the fitness function relative to the number of generations

Table 5.6. "Recorded" speed values and speed values obtained using approximate reasoning algorithm when the membership functions of input and output variables are tuned using genetic algorithms

Recorded speed	Obtained speed	Recorded speed	Obtained speed	Recorded speed	Obtained speed
57	62	45	46	20	20
57	62	42	42	13	13
57	60	40	40	13	13
57	57	39	39	13	13
51	52	35	36	13	13
48	48	26	26		
47	46	22	22		

5.4.3. Tuning membership functions using tabu search

The tabu search technique is exceptionally useful in solving complex combinatorial optimization problems. Glover (1986) proposed a modern formulation of this technique and certain seminal ideas were put forward by Hansen (1986). Let us present the fundamental principles of the tabu search technique by the following illustrative example.

Figure 5.14 presents five nodes to be visited by a traveling salesman. The traveling salesman departs from and returns to a depot designated in *Figure 5.14* by D.

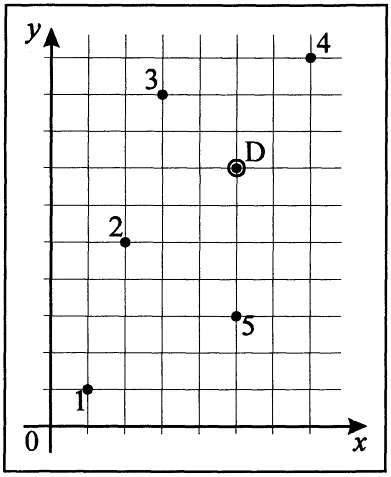

Figure 5.14. Nodes 1, 2, ..., 5 to be visited by traveling salesman located at depot D

The well-known traveling salesman problem consists in finding the shortest route the salesman should take when serving the nodes. The Euclidean distances between certain pairs of nodes are given in the following matrix:

$$\begin{array}{c|cccccc} & D & 1 & 2 & 3 & 4 & 5 \\ D & 0 & 7.2 & 3.6 & 2.8 & 3.6 & 4 \\ 1 & 7.2 & 0 & 4.1 & 8.2 & 10.8 & 4.5 \\ 2 & 3.6 & 4.1 & 0 & 4.1 & 7.1 & 3.6 \\ 3 & 2.8 & 8.2 & 4.1 & 0 & 4.1 & 6.3 \\ 4 & 3.6 & 10.8 & 7.1 & 4.1 & 0 & 7.3 \\ 5 & 4 & 4.5 & 3.6 & 6.3 & 7.3 & 0 \end{array} \tag{5.6}$$

The traveling salesman problem (which has been devoted numerous papers in the literature) is one of the typical permutation problems. Finding the shortest traveling salesman route amounts to finding the appropriate ordering of nodes that is, the appropriate permutation corresponding to the shortest length. Since the traveling salesman starts and finishes his tour at depot D and is to visit five nodes, the total number of permutations, that is, various traveling salesman routes, equals 5! = 120. When there is a larger number of nodes to be visited, the number of possible permutations can be astronomical. Using the tabu search technique (along with other metaheuristic algorithms), we seek to find an optimal or near-optimal solution by examining only a small subset of all possible permutations.

We generate the initial solution to our problem by employing one of the heuristic algorithms for solving the traveling salesman problem. For example, suppose that the initial solution is represented by the following traveling salesman tour (D, 2, 3, 4, 5, 1, D). According to the matrix of Euclidean distances, it is easily computed that the corresponding length of tour of the initial solution is equal to 3.6 + 4.1 +4.1 + 7.3 + 4.5 + 7.2 = 30.8.

The initial solution is shown in *Figure 5.15*.

As Glover and Laguna (1993) note: "Tabu search methods operate under the assumption that a neighborhood can be constructed to identify 'adjacent solutions' that can be reached from any current solution." When permutation problems are concerned, for the purpose of generating a neighborhood the so-called swaps representing pairwise exchanges are most often used. In the case of the traveling salesman problem, a swap exchanges the positions of two nodes in the traveling salesman route. Once a swap has been made, a move has actually been made leading from one solution to another one. If, for example, nodes 3 and 5 swap positions, from the initial solution that read

(D, 2, 3, 4, 5, 1, D)

a move has been made that leads us toward the following solution (D, 2, 5, 4, 3, 1, D).

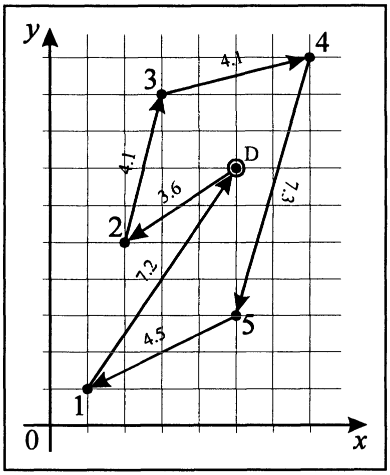

Figure 5.15. Initial solution of the traveling salesman problem

After making a swap, the next solution has a smaller, equal or larger value of the objective function as compared to the previous solution. In the above example, after making a swap of nodes 3 and 5, the objective function takes the following larger value: 3.6 + 3.6 + 7.3 + 4.1 + 8.2 + 7.2 = 34. Since there are five nodes to be visited with two nodes taking part in a swap, the total number of various solutions obtainable by swaps equals 10. In other words, a complete neighborhood of the current solution includes ten different solutions.

The tabu search technique uses the concept of the so-called flexible memory. The basic idea is to pronounce the subset of the moves in a neighborhood forbidden. Which moves are to be forbidden (tabu) is decided according to the recency or frequency that certain moves have participated in generating the previous solutions. In other words, by introducing the tabu moves, we try to avoid the swaps made in recent past. For example, it can be specified that nodes 3 and 5 that have already swaped positions in the traveling salesman route cannot swap positions in the subsequent two, three, or five iterations. Thus, the swap of nodes 3 and 5 is considered to be tabu in the following five iterations. Glover and Laguna (1993) proposed the following tabu data structure to represent the remaining tabu tenure for node pairs:

	2	3	4	5
1				
	2			
		3		
			4	

Each cell in the structure represents the number of iterations remaining until the corresponding nodes are allowed to swap places again. If, for example, the value in cell (2, 4) equals 2, it means that nodes 2 and 4 cannot swap positions during the next two iterations. Similarly, if the value in cell (1, 2) equals zero, nodes 1 and 2 are presently allowed to swamp positions, and so on. Let us note that tabu restrictions do not apply in all situations. If, for example, a tabu move leads to a solution with the shortest route so far discovered, the tabu classification will be disregarded and the move will be allowed.

Let us illustrate the basic points of the tabu search technique by several iterations. Let the initial solution, as already mentioned, be (D, 2, 3, 4, 5, 1, D).

The so-called Iteration 0 equals (traveling salesman route length=30.8):

Current solution

D	2	3	4	5	1	D

Tabu structure

	2	3	4	5
1	0	0	0	0
	2	0	0	0
		3	0	0
			4	0

As pointed out, the total number of different solutions that can be obtained by swaps equals ten. *Table 5.7* presents ten different swaps, ten corresponding new routes, the lengths of the new routes and the gain achieved when a new route replaces the existing one (the gain represents the difference in length between existing route and new route).

Table 5.7. Candidates for swap moves

Swap	New route	New route length	Gain
(1,2)	(*D*, 1, 3, 4, 5, 2, *D*)	34	30.8 – 34 = -3.2
(1,3)	(*D*, 2, 1, 4, 5, 3, D)	36.9	30.8 – 36.9 = - 6.1
(1,4)	(D, 2, 3, 1, 5, 4, D)	31.3	30.8 – 31.3 = - 0.5
(1,5)	(D, 2, 3, 4, 1, 5, D)	31.1	30.8 – 31.3 = - 0.5
(2,3)	(D, 3, 2, 4, 5, 1, D)	33	30.8 - 33 = - 2.2
(2,4)	(D, 4, 3, 2, 5, 1, D)	27.1	30.8 – 27.1 = 3.7
(2,5)	(D, 5, 3, 4, 2, 1, D)	32.8	30.8 – 32.8 = - 2
(3,4)	(D, 2, 4, 3, 5, 1, D)	32.8	30.8 – 32.8 = - 2
(3,5)	(D, 2, 5, 4, 3, 1, D)	34	30.8 – 34 = - 3.2
(4,5)	(D, 2, 3, 5, 4, 1, D)	39.3	30.8 – 39.3 = - 8.5

Let us sort the first five candidates for swap moves in a descending order of gain. (Note that number five is arbitrarily determined. We could have decided to sort the first three candidates or the first six candidates.) The best five candidates are given in *Table 5.8*.

Table 5.8. The best five candidates for swap moves

Swap	Gain
(2,4)	3.7
(1,4)	-0.5
(1,5)	-0.5
(2,5)	-2
(3,4)	-2

The largest gain is achieved if we swap the positions of nodes 2 and 4. Let us swap the positions of nodes 2 and 4. Let us also decide that in the subsequent three iterations we are not allowed to swap the positions of these nodes. In other words, in the subsequent three iterations swap (2, 4) is tabu. (The number of iterations during which the swap is in a tabu state has also been determined arbitrarily. Instead of value 3, we could have chosen value 2, or value 5, and so on.)

Finally, we have obtained a new solution and an altered tabu structure in Iteration 1 (traveling salesman route length = 27.1):

Current solution

D	4	3	2	5	1	D

Tabu structure

	2	3	4	5
1	0	0	0	0
2		0	3	0
3			0	0
4				0

The best five candidates for swap moves are given in *Table 5.9*.

Table 5.9. The best five candidates for swap moves

Swap	New route	New route length	Gain	
(1,5)	(D, 4, 3, 2, 1, 5, D)	24.4	27.1 – 24.4 = 2.7	
(1,2)	(D, 4, 3, 1, 5, 2, D)	27.6	27.1 – 27.6 = -0.5	
(2,5)	(D, 4, 3, 5, 2, 1, D)	28.9	27.1 – 28.9 = -1.8	
(3,4)	(D, 3, 4, 2, 5, 1, D)	29.3	27.1 – 29.3 = -2.2	
(2,4)	(D, 2, 3, 4, 5, 1, D)	30.8	27.1 – 30.8 = -3.7	T

Obviously, the best gain will be achieved if we swap the positions of nodes (1, 5). Notice the letter "T" on the right-hand side of the *Table 5.9* next to the row referring to swap (2, 4). It indicates that swap (2, 4) is not allowed, since its classification is tabu. Even if swap (2, 4) was to have the largest gain, it would not be allowed due to the fact that its classification is tabu. The only exception is the situation in which the tabu move (2, 4) would produce a solution better than any discovered so far. In that case, move (2, 4) would be allowed.

The new solution and the altered tabu structure are shown in Iteration 2:

Current solution

D	4	3	2	1	5	D

Tabu structure

	2	3	4	5
1	0	0	0	3
2		0	2	0
3			0	0
4				0

Due to the tabu structure, swap (1, 5) cannot be made during the subsequent three iterations, whereas swap (2, 4) cannot be made during the subsequent two iterations. The other swaps do not have a tabu classification.

The solution obtained after two iterations is shown in *Figure 5.16.*

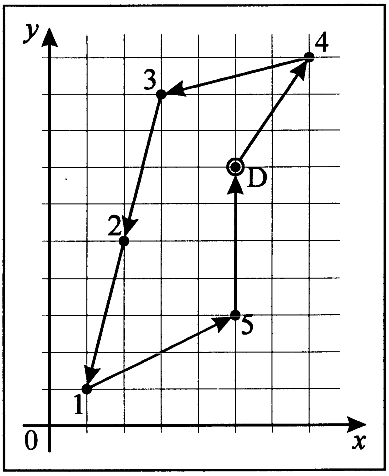

Figure 5.16. Solution of traveling salesman problem obtained after two iterations of using tabu search technique

Following the tabu status of certain swaps refers to the "recent past." In this sense, it can be said that the tabu search technique is characterized by a "recency-based memory." Glover and Laguna (1993) proposed that the recency-based memory be followed by the so-called frequency-based memory. This results in a more efficient search that involves the need both to diversify the search and (in certain cases) the need to intensify the search. Let us explain the concept of the "frequency-based memory" referring to the methodology of Glover and Laguna (1993).

Figure 5.17 presents in a matrix form the recency-based memory and the frequency-based memory.

Let us assume that we have already made seventeen iterations. Let us also assume that the tenure during which a certain swap is in a tabu status equals 2. The recency-based memory includes the cells above the main diagonal. The last two swaps that have been made are (2, 3) and (3, 5). The frequency-based memory includes the cells below the main diagonal. Each of the cells contains a frequency indicating the number of times that the particular swap has already been made. Thus, for example, we can say that swap (2, 4) has been made 5 times, swap (3, 4) four times, swap (1, 5) once, and so on. The data contained in the frequency-based memory enable us to follow the "search history" from the very beginning. This can serve to diversify the search by traveling to new regions. The diversification is made by penalizing nonimproving moves according to the appropriate frequency information, the swaps with greater frequency being given larger penalty. For example, let the best five candidates for swap moves be as shown in *Table 5.10*.

	1	2	3	4	5
1					
2	3		2		
3		1			1
4		5	4		
5	1		2	1	

Figure 5.17. Recency-based memory and frequency-based memory

Table 5.10. The best five candidates for swap moves

Swap	Gain	Gain after penalization	
(3,5)	1.5	1.5	T
(2,3)	0.5	0.5	T
(2,4)	-1.2	-6.2	
(1,5)	-3.7	-4.7	
(4,5)	-4.9	-5.9	

Swaps (3, 5) and (2, 3) that would lead to improving moves have a tabu status. Swaps (2, 4) (1, 5), and (4, 5) lead to nonimproving moves, since their gain is negative. As swap (2, 4) has the least negative gain, this is the swap to be made unless we consider frequency. If, for example, the penalty for a certain swap equals the frequency of making this swap since the beginning of the procedure, than the gain obtained after penalization for swap (2, 4) equals -1.2 - 5 = -6.2, for swap (1, 5) it equals -3.7 - 1 = -4.7 and finally, for swap (4, 5) it equals -4.9 - 1 = -5.9. Without any penalization, swap (2, 4) would have been made. After penalization, we conclude that swap (1, 5) should be made. By employing penalization, instead to the region in which we often "resided" in the past, we "traveled" to the region we "rarely visited." Clearly, penalization can be done in different ways depending on the analyst and the context of the problem concerned.

The process of intensification consists in following the frequencies of subsets of the elite solutions (corresponded by high quality local optima) and in "traveling" to the regions promising a further improvement of the solution.

5.4.3.1. Tuning membership functions using tabu search: Traffic signal coordination by fuzzy logic

Consider the problem presented in Section 5.4.1.1. We will show how to tune the membership functions of input and output variable using tabu search such that the speed values obtained by the approximate reasoning algorithm (presented in Section 5.4.1.1) are "as close as possible" to the "recorded" values.

The initial shapes of the membership functions of input and output variable are marked in *Figure 5.7* by a solid line. The dashed lines denote the "boundary" shapes of the membership functions. As in the previous cases, by varying the intersections of the trapezoid sides and the x axis various membership functions are obtained. The intersections or parameters can be found only within certain intervals. In our example, there are twelve parameters that is, twelve intervals. To solve the presented problem more easily, let us perform the discretization of the intervals within which the intersections can be found. As in Section 5.4.1.1, each of the intervals is divided into small intervals whose width makes 5% of that of the initial interval. Let us assume that the intersections can occur only at points representing the ends of small intervals. It is assumed that a move leading from the current solution to the next solution has been made when there is a simultaneous movement of the positions of two parameters. The positions of the other ten parameters remain unchanged. The change of the positions of the chosen two points is made in a random fashion within the appropriate intervals. The total number of different solutions (different sets of membership functions) that can be obtained when there is a change of two parameters within a move equals sixty six.

It means that the complete neighborhood of the current solution comprises sixty six different solutions. Some of these sixty six solutions are corresponded by a larger, some by an equal, and some by a smaller value of the objective function. Using the approximate reasoning algorithm for each of the nineteen values of volume-to-capacity ratio we can obtain the speed value, compare it with the "recorded" speed value and calculate the corresponding absolute deviation. In other words, the objective function that is minimized represents the sum of the absolute deviations estimated from the "recorded" speed values. Each time a move was made that is, each time the shapes of any two membership functions were changed it was necessary to estimate the nineteen speed values applying to the approximate reasoning algorithm the values of *v/c* ratio from *Table 5.1* and calculate the corresponding values of the objective function.

When tuning the membership functions of input and output variable, the recency based memory and the frequency based memory were used. For each possible new solution the gain was first calculated. On calculating the gain, penalization was performed. Penalization was done by decreasing the calculated value of the gain by the appropriate value from the frequency based memory.

Figure 5.18 represents the initial and final membership functions of input and output variable that are determined using tabu search.

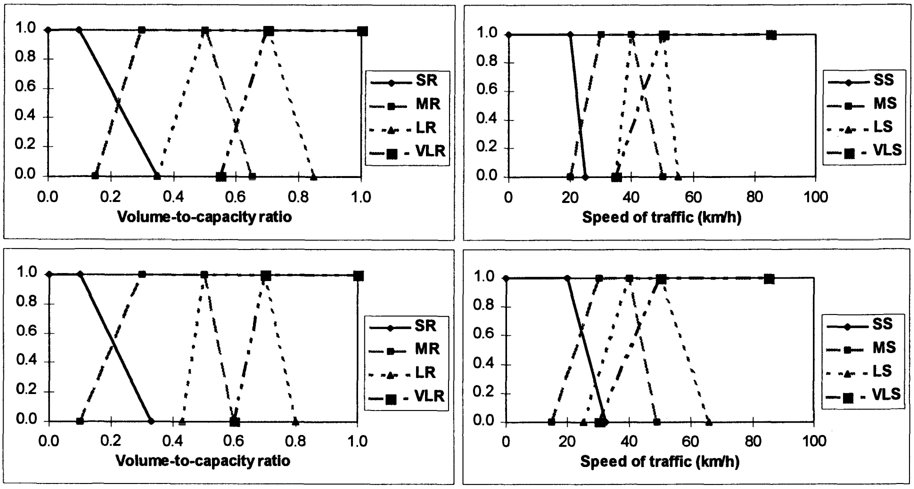

Figure 5.18. The initial (first row) and the final (second row) membership functions of input and output variable

Figure 5.19 represents the change of the objective function relative to the number of iterations.

Table 5.11 represents the "recorded" speed values and the speed values obtained by approximate reasoning by "max-min composition" whereby the final shapes of the membership functions of input and output variable are obtained using the tabu search technique.

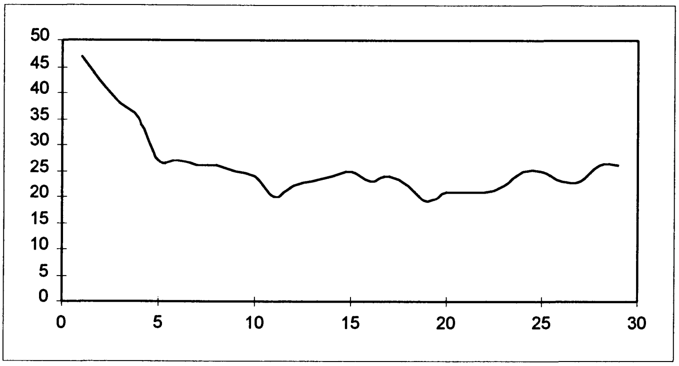

Figure 5.19. Change of the objective function relative to the number of iterations

Table 5.11. "Recorded" speed values and speed values obtained by approximate reasoning algorithm when the membership functions of input and output variables are tuned using tabu search

Recorded speed	Obtained speed	Recorded speed	Obtained speed	Recorded speed	Obtained speed
57	62	45	45	20	20
57	62	42	43	13	13
57	60	40	40	13	13
57	57	39	39	13	13
51	52	35	33	13	13
48	48	26	26		
47	45	22	22		

5.5. METHOD TO GENERATE FUZZY RULES FROM NUMERICAL AND LINGUISTIC INFORMATION

The information used to develop classical expert systems was exclusively obtained from experts. Human experts are generally able to provide certain linguistic information. However, human experts are usually not able to give a full account of their knowledge and experience in terms of linguistic rules alone. In this way, some information is usually lost.

Apart from linguistic information, certain numerical information can also be obtained based on measurements, observation, or statistical analysis of a particular phenomenon. Similarly, sampled input-output data pairs are generally not sufficient to generate fuzzy rules, since they usually do not cover a large variety of situations that can occur in the observed system.

Surely, when generating fuzzy rules it is of interest to consider both numerical and linguistic information. Wang and Mendel (1992c) developed precisely a method suited to generate fuzzy rules from numerical and linguistic information. In the following discussion, we shall briefly present the results achieved by Wang and Mendel (1992c).

5.5.1. Route choice problem in air transportation

Let us present the Wang and Mendel's method (1992c) by the example of the route choice problem in air transportation. We denote by x_1 the input variable representing the difference in the travel time between one route and the other. By x_2 we denote the input variable representing the difference in the weekly flight frequency between one route and the other and by y the output variable representing the percentage of passengers using the first route. Let the set of input-output data pairs contains twenty six elements: $\{(x_1^{(1)}, x_2^{(1)}; y^{(1)}), (x_1^{(2)}, x_2^{(2)}; y^{(2)}), \ldots, (x_1^{(26)}, x_2^{(26)}; y^{(26)})\}$.

We shall observe the passengers' choice of a particular route based on the differences in travel times and weekly flight frequencies with respect to the data on the twenty six pairs of towns. The corresponding data are given in *Table 5.12.*

Table 5.12. Traffic characteristics

Pair of cities	$x_1^{(i)}$	$x_2^{(i)}$	$y^{(i)}$
1	140	24	19
2	-140	-24	81
3	130	41	19
4	-130	-41	81
5	120	43	25
6	-120	-43	75
7	83	0	31
8	-83	0	69
9	90	-1	40
10	-90	1	60
11	150	83	22
12	-150	-83	78
13	60	34	39
14	-60	-34	61
15	50	0	39
16	-50	0	61
17	45	-22	33
18	-45	22	67
19	30	10	43
20	-30	-10	57
21	95	70	34
22	-95	-70	66
23	80	77	36
24	-80	-77	64
25	90	14	39
26	-90	-14	61

Keeping to the notation of Wang and Mendel (1992c), we denote respectively by $[x_1^-, x_1^+]$, $[x_2^-, x_2^+]$, and $[y^-, y^+]$ the domain intervals of x_1, x_2, and y. Considering the values of variables x_1, x_2 and y given in *Table 5.12*, let us arbitrarily determine the domain intervals. The domain intervals are given in *Table 5.13*.

Table 5.13. Domain intervals of x_1, x_2, and y

Variable	Domain interval
x_1	[-200, 200]
x_2	[-100, 100]
y	[0, 100]

In the next step, each domain interval is divided into $2N+1$ regions. Note that N can be different for different variables. It should also be noted that "the lengths of these regions can be equal or unequal."

Figure 5.20 presents the case of variable x_1 whose domain interval is divided into nine regions ($N = 4$).

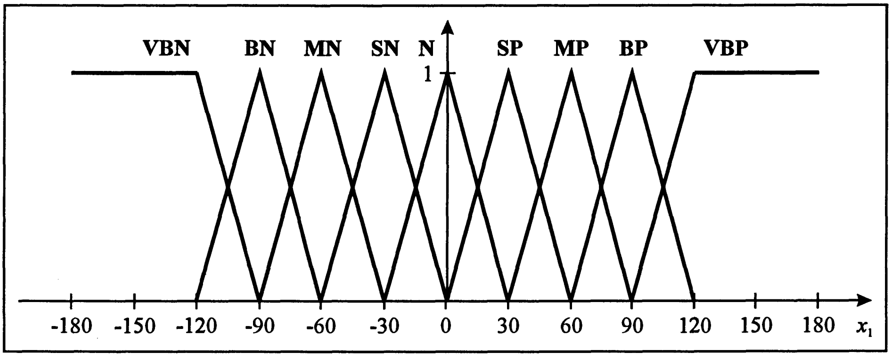

Figure 5.20. Division of domain interval of input variable x_1 and corresponding membership functions

Figure 5.20 presents membership functions of the following fuzzy sets: **VBN** is very big negative difference in travel time, **BN** is big negative difference in travel time, **MN** is medium negative difference in travel time, **SN** is small negative difference in travel time, **N** is negligible difference in travel time, **SP** is small positive difference in travel time, **MP** is medium positive difference in travel time, **BP** is big positive difference in travel time, and **VBP** is very big positive difference in travel time.

Let us divide the domain interval of input variable x_2 into three regions ($N = 1$) (*Figure 5.21*).

Figure 5.21 presents membership functions of the following fuzzy sets: **SF** is smaller flight frequency than on the alternate route, **ASF** is approximately the same flight frequency as on the alternate route, and **BF** is bigger flight frequency than on the alternate route.

The domain interval of output variable y is divided into nine regions ($N = 4$). As can be seen from *Figure 5.22*, the length of intervals in the case of output variable y is unequal. *Figure 5.22* presents the membership functions of the following fuzzy sets: **VVS** is very very small percentage, **VS** is very small percentage, **S** is small percentage, **MS** is medium-small percentage, **M** is medium percentage, **MB** is medium-big percentage, **B** is big percentage, **VB** is very big percentage, and **VVB** is very very big percentage.

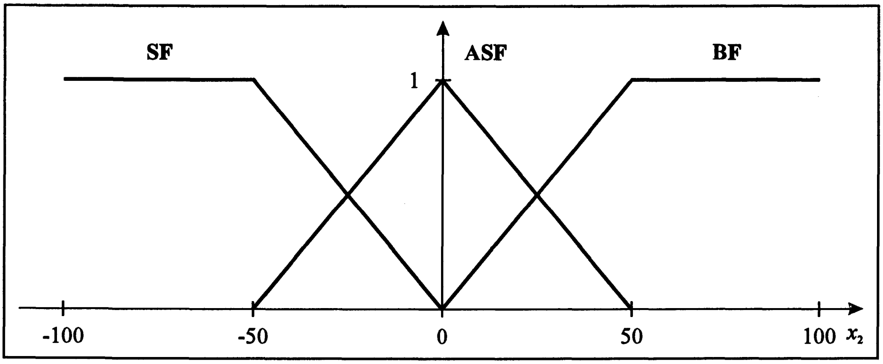

Figure 5.21. Division of domain interval of input variable x_2 and corresponding membership functions

The first step in the proposed method consists of the division of input and output spaces into fuzzy regions. In our case, each variable is corresponded by a different number of regions. The lengths of the regions in the case of input variables x_1 and x_2 are equal, whereas the lengths of the regions in the case of output variable y are unequal. Let us note that the division into regions has been done arbitrarily. In other words, the partition of input and output spaces could have been done in a different way.

In the next step, fuzzy rules have to be generated from the given data pairs. For every input-output data pair ($x_1^{(i)}$, $x_2^{(i)}$; $y^{(i)}$) it is necessary to determine the degrees in different regions. Thus, for example, $x_1^{(1)} = 140$ has degree 1 in **VBP** and zero degrees in all other regions. Similarly, $x_1^{(21)} = 95$ has degree 5/6 in **BP**, degree 1/6 in **VBP** and zero degrees in all other regions. After determining degrees in particular regions, the considered $x_1^{(i)}$, $x_2^{(i)}$ or $y^{(i)}$ are to be assigned to the region with the maximum degree. Keeping to this rule, we can, for example, say that $x_1^{(1)}$ is considered to be

VBP or that $x_1^{(21)}$ is considered to be **BP**. In the next step, one rule is to be obtained from one pair of input-output data. The first pair of input-output data will determine Rule 1, the second pair of input-output data will determine Rule 2, and so on.

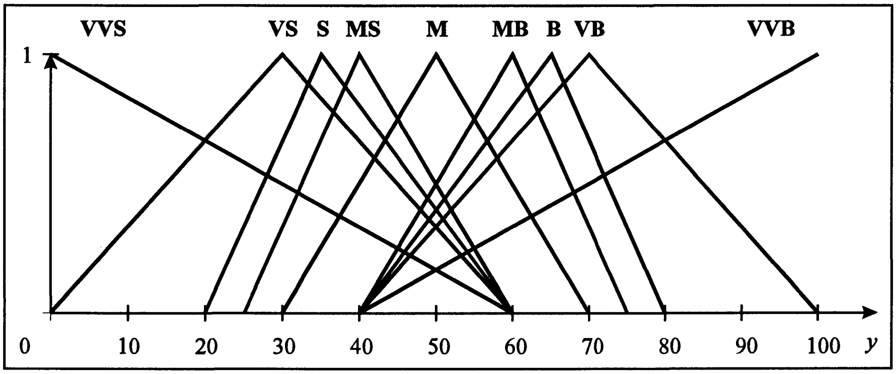

Figure 5.22. Division of domain interval of output y and corresponding membership functions

Let us examine the first pair of input-output data ($x_1^{(1)}$, $x_2^{(1)}$; $y^{(1)}$). By reading the corresponding graphs, it can be said that $x_1^{(1)}$ has the maximum degree in **VBP** (degree equals 1), $x_2^{(1)}$ has the maximum degree in **ASF** (degree equals 26/50), whereas $y^{(1)}$ has the maximum degree in **VVS** (degree equals 41/60). According to the examined values of the first input-output data pair and the established maximum degrees, the first rule can be written as follows:

Rule 1: If x_1 is **VBP** and x_2 is **ASF**, then y is **VVS**.

The second pair of the input-output data ($x_1^{(2)}$, $x_2^{(2)}$; $y^{(2)}$) should help us generate the second rule. According to the graphs, it can be said that $x_1^{(2)}$ has the maximum degree in **VBN** (degree equals 1), $x_2^{(2)}$ has the maximum degree in **ASF** (degree equals 26/50), and $y^{(2)}$ has the maximum degree in **VVB** (degree equals 39/60). The second rule reads as follows:

Rule 2: If x_1 is **VBN** and x_2 is **ASF**, then y is **VVB**.

In this way, one rule is derived from one pair of input-output data. The obtained twenty six rules are given in *Table 5.14*.

When forming a fuzzy rule base, equivalent rules have to be eliminated, while the problem of conflicting is resolved by accepting only one rule from the conflicting group. If the obtained twenty six rules are carefully analyzed, it can be noted that some of them conflict with each other. The conflicting rules are understood to be the rules having the same IF part, but different Then part. To resolve the problem of conflicting, each rule is assigned a degree, leaving in the fuzzy rule base only one rule from the conflicting group having the maximum degree. Wang and Mendel (1992c) suggested that for the rule

Table 5.14. Rules obtained according to input-output data pairs

Input-output data pair	Rule
$x_1^{(1)}, x_2^{(1)}; y^{(1)}$	Rule 1: If x_1 is **VBP** and x_2 is **ASF**, then y is **VVS**
$x_1^{(2)}, x_2^{(2)}; y^{(2)}$	Rule 2: If x_1 is **VBN** and x_2 is **ASF**, then y is **VVB**
$x_1^{(3)}, x_2^{(3)}; y^{(3)}$	Rule 3: If x_1 is **VBP** and x_2 is **BF**, then y is **VVS**
$x_1^{(4)}, x_2^{(4)}; y^{(4)}$	Rule 4: If x_1 is **VBN** and x_2 is **SF**, then y is **VVB**
$x_1^{(5)}, x_2^{(5)}; y^{(5)}$	Rule 5: If x_1 is **VBP** and x_2 is **BF**, then y is **VS**
$x_1^{(6)}, x_2^{(6)}; y^{(6)}$	Rule 6: If x_1 is **VBN** and x_2 is **SF**, then y is **VVB**
$x_1^{(7)}, x_2^{(7)}; y^{(7)}$	Rule 7: If x_1 is **BP** and x_2 is **ASF**, then y is **VS**
$x_1^{(8)}, x_2^{(8)}; y^{(8)}$	Rule 8: If x_1 is **BN** and x_2 is **ASF**, then y is **VB**
$x_1^{(9)}, x_2^{(9)}; y^{(9)}$	Rule 9: If x_1 is **BP** and x_2 is **ASF**, then y is **S**
$x_1^{(10)}, x_2^{(10)}; y^{(10)}$	Rule 10: If x_1 is **BN** and x_2 is **ASF**, then y is **B**
$x_1^{(11)}, x_2^{(11)}; y^{(11)}$	Rule 11: If x_1 is **VBP** and x_2 is **BF**, then y is **VS**
$x_1^{(12)}, x_2^{(12)}; y^{(12)}$	Rule 12: If x_1 is **VBN** and x_2 is **SF**, then y is **VB**
$x_1^{(13)}, x_2^{(13)}; y^{(13)}$	Rule 13: If x_1 is **MP** and x_2 is **BF**, then y is **S**
$x_1^{(14)}, x_2^{(14)}; y^{(14)}$	Rule 14: If x_1 is **MN** and x_2 is **SF**, then y is **B**
$x_1^{(15)}, x_2^{(15)}; y^{(15)}$	Rule 15: If x_1 is **MP** and x_2 is **ASF**, then y is **S**
$x_1^{(16)}, x_2^{(16)}; y^{(16)}$	Rule 16: If x_1 is **MN** and x_2 is **ASF**, then y is **B**
$x_1^{(17)}, x_2^{(17)}; y^{(17)}$	Rule 17: If x_1 is **MP** and x_2 is **ASF**, then y is **VS**
$x_1^{(18)}, x_2^{(18)}; y^{(18)}$	Rule 18: If x_1 is **MN** and x_2 is **ASF**, then y is **VB**
$x_1^{(19)}, x_2^{(19)}; y^{(19)}$	Rule 19: If x_1 is **SP** and x_2 is **ASF**, then y is **MS**
$x_1^{(20)}, x_2^{(20)}; y^{(20)}$	Rule 20: If x_1 is **SN** and x_2 is **ASF**, then y is **MB**
$x_1^{(21)}, x_2^{(21)}; y^{(21)}$	Rule 21: If x_1 is **BP** and x_2 is **BF**, then y is **VS**
$x_1^{(22)}, x_2^{(22)}; y^{(22)}$	Rule 22: If x_1 is **BN** and x_2 is **SF**, then y is **VB**
$x_1^{(23)}, x_2^{(23)}; y^{(23)}$	Rule 23: If x_1 is **BP** and x_2 is **BF**, then y is **S**
$x_1^{(24)}, x_2^{(24)}; y^{(24)}$	Rule 24: If x_1 is **BN** and x_2 is **SF**, then y is **B**
$x_1^{(25)}, x_2^{(25)}; y^{(25)}$	Rule 25: If x_1 is **BP** and x_2 is **ASF**, then y is **S**
$x_1^{(26)}, x_2^{(26)}; y^{(26)}$	Rule 26: If x_1 is **BN** and x_2 is **ASF**, then y is **B**

If x_1 is **A** and x_2 is **B**, then y is **C**

the degree of the rule denoted by D(Rule) be calculated as

$$\mathrm{D(Rule)} = \mu_{\mathbf{A}}(x_1)\, \mu_{\mathbf{B}}(x_2)\, \mu_{\mathbf{C}}(y) \tag{5.7}$$

For example, the degrees of Rules 1 and 2 are equal to

D(Rule1) = $\mu_{VBP}(x_1)\ \mu_{ASF}(x_2)\ \mu_{VVS}(y) = \mu_{VBP}(140)\ \mu_{ASF}(24)\ \mu_{VVS}(19) = 1\ (26/50)\ (41/60) = 0.3553$;

D(Rule 2) = $\mu_{VBN}(x_1)\ \mu_{ASF}(x_2)\ \mu_{VVB}(y) = \mu_{VBN}(-140)\ \mu_{ASF}(-24)\ \mu_{VVB}(81) = 1\ (26/50)\ (39/60) = 0.338$.

In certain situations, after the input-output data pairs have been analyzed, some of them may be considered very important, while some others rarely occur or may even represent measurement errors. Wang and Mendel (1992c) suggested that, if we wish, we can "assign a degree to each data pair that represents our belief of its usefulness." Thus, the data pairs constitute a fuzzy set ("a data pair belongs to this set to a degree assigned by a human expert").

If, for example, the data pair $(x_1^{(1)}, x_2^{(1)}; y^{(1)})$ has the degree $m^{(1)}$, than the degree of Rule 1 is equal to: D(Rule 1) = $\mu_{VBP}(x_1)\ \mu_{ASF}(x_2)\ \mu_{VVS}(y)\ m^{(1)}$.

When the evaluation of "usefulness" of certain input-output data pairs is not made, all degrees $m^{(i)}$ of data pairs are equal to 1. In our example, the usefulness of certain input-output data pairs is not evaluated, so we have $m^{(1)} = m^{(2)} \ldots = m^{(26)} = 1$.

After analyzing the obtained twenty six rules, we observe that certain rules are equivalent (for example, Rule 4 is equivalent to Rule 6). Also, among the obtained rules there are some mutually conflicting rules (for example, Rule 4 and Rule 12 are mutually conflicting). It is convenient to represent a fuzzy rule base in the form presented in *Table 5.15*.

Table 5.15. The form of a fuzzy rule base

	BF									
x_2	**ASF**									
	SF									
		VBN	**BN**	**MN**	**SN**	**N**	**SP**	**MP**	**BP**	**VBP**
						x_1				

Let us insert into the fuzzy rule base the clearly defined rules (rules that are not conflicting). *Table 5.16* presents a partially filled fuzzy rule base.

Table 5.16. Fuzzy rule base in which only unconflicting rules have been inserted

	BF							**S**		
x_2	**ASF**	**VVB**			**MB**		**MS**			**VVS**
	SF			**B**						
		VBN	**BN**	**MN**	**SN**	**N**	**SP**	**MP**	**BP**	**VBP**
						x_1				

In order to completely fill the fuzzy rule base, it is necessary to resolve the problem of conflict. Thus, for example, Rules 3 and 5 are mutually conflicting, whereas Rule 5 is equivalent to Rule 11. The corresponding degrees of the rules are

D(Rule 3) = $\mu_{VBP}(x_1)\,\mu_{BF}(x_2)\,\mu_{VVS}(y) = \mu_{VBP}(130)\,\mu_{BF}(41)\,\mu_{VVS}(19)$
$= 1\,(41/100)\,(41/60)=0.28$

D(Rule 5) = $\mu_{VBP}(x_1)\,\mu_{BF}(x_2)\,\mu_{VS}(y) = \mu_{VBP}(120)\,\mu_{BF}(43)\,\mu_{VS}(25) =$
$1\,(43/100)\,(25/30)=0.358$

D(Rule 11) = $\mu_{VBP}(x_1)\,\mu_{BF}(x_2)\,\mu_{VS}(y) = \mu_{VBP}(150)\,\mu_{BF}(83)\,\mu_{VS}(22) =$
$1\,(83/100)\,(22/30) = 0.61$

As D(Rule 11) > D(Rule 5) > D(Rule 3), only Rule 11 is inserted into the fuzzy rule base. Having calculated degrees of the rules, the fuzzy rule base shown in *Table 5.17* is obtained:

Table 5.17. Fuzzy rule base obtained after resolving the problem of conflicting between rules

	BF							**S**	**S**	**VS**
x_2	ASF	**VVB**	**B**	**B**	**MB**		**MS**	**S**	**S**	**VVS**
	SF	**VB**	**B**	**B**						
		VBN	**BN**	**MN**	**SN**	**N**	**SP**	**MP**	**BP**	**VBP**
						x_1				

The available input-output pairs were not sufficient to cover all possible situations. Clearly, the fuzzy rule base would be more complete if a larger number of different input-output data pairs were available. It can be filled by using the intuition and knowledge of the expert studying the route choice problem. The final fuzzy rule base based on the knowledge of the expert and the available input-output data pairs is shown in *Table 5.18*. In the final fuzzy rule base the rules obtained from the expert are bold and underlined.

Table 5.18. Fuzzy rule base based on expert knowledge and available input-output data pairs

	BF	**VVB**	**VB**	**B**	**B**	**MB**	**MS**	**S**	**S**	**VS**
x_2	ASF	**VVB**	**B**	**B**	**MB**	**M**	**MS**	**S**	**S**	**VVS**
	SF	**VB**	**B**	**B**	**MB**	**MS**	**S**	**S**	**VS**	**VVS**
		VBN	**BN**	**MN**	**SN**	**N**	**SP**	**MP**	**BP**	**VBP**
						x_1				

The role of an expert does not merely consist in creating the rules that are otherwise not obtainable from the available input-output data pairs. As Wang and Mendel (1992c) note: "real numerical data have different reliabilities, for example, some real data can be very bad ('wild data')." Consequently, in certain cases, generating the rules based exclusively on input-output data pairs, without analyzing the usefulness of the data that is, without the data being judged by an expert, can yield unsatisfactory results.

Note that the stated procedure represents a simultaneous generation of fuzzy rules from an expert and from available input-output data pairs. In this manner, numerical and linguistic information is successfully used to generate fuzzy rules. Also, the method presented is characterized by adaptability. Fuzzy rules are generated by learning from examples. In other words, as the number of various input-output data pairs increases, the fuzzy rule base becomes "fuller," and, in some cases, even the "content" of certain boxes of the fuzzy rule base is altered. (If a box contains more than one rule, only the rule that has the maximum degree will eventually remain in the box.)

5.5.2. Solving modal split problem by fuzzy rules generated by learning from examples

Predicting the number of passengers that will choose different competitive modes of transportation is known as the modal split problem. The percentage of passengers choosing a particular mode of transportation depends on transportation costs and travel times of certain modes of transport, daily frequencies, adjustment of the flight schedule and timetables to passengers' demands, reliability of execution of the announced timetables (canceling of certain trips and delays), travel comfort, sense of safety and so on. Over the last five decades a number of models for solving the modal split problem have been developed (Kanafani, 1983). The older models are based on the methods of mathematical statistics (multiple regression). In these models the independent variables are socioeconomic characteristics of the regions between which transportation is carried out (population, rate of employment, national income, and so on), as well as characteristics of transportation modes between the observed regions (travel times, travel costs, number of daily departures, and so on). The dependent variable in these models corresponds to the number of passengers opting for a particular mode of transportation. Among relatively recent models are the so-called choice models. These models are used to estimate the percental participation of certain transportation modes in the total number of travels that is, the probabilities of using particular transportation modes, and they depart from an individual as he is the one who decides on a particular mode of transportation. In addition to frequencies, different travel costs and travel times by different modes of transportation, this choice depends on a variety of factors whose influence cannot be quantified without a survey of the travel population (sense of safety, sense of comfort, and so on).

Teodorovic and Kalic (1996) attempted to model the modal split by fuzzy logic. Let us present the basic results achieved by Teodorovic and Kalic (1996).

When choosing a mode of transportation, the user is usually not acquainted with all information concerning the available timetables, number of departures, departure times, existing fares, and so on. Teodorovic and Kalic (1996) assumed that a passenger decides on a particular transportation mode according to the perceived travel times, perceived sense of safety and comfort as well as the approximately known rates, numbers of daily departures, and departures times. It is to be noted that the perceived travel times and the approximately known rates and numbers of daily departures are most commonly fuzzy. Hence, for a subjective estimation of the travel time by a vehicle (or train or coach) between two towns, expressions such as "time travel is about eight hours" will be used. The claim that "travel time by car is about eight hours" is the result of a subjective feeling that is, an individual's subjective estimation. Also, we often do not have sufficiently precise information on the daily number of departures or the value of another relevant parameter. Thus, for example, passengers may know that on a certain route there are "several" or "five or six" departures every day.

As can be seen, certain parameters used by passengers in choosing an appropriate mode of transportation are characterized by subjectivity, uncertainty and imprecision. The passengers compare the characteristics of certain modes of transportation and opt for particular modes because they are "considerably cheaper" or because "travel time is five or six hours shorter" and so on. These facts suggest the need of an approximate reasoning algorithm for estimating the number of passengers deciding on a particular mode of transportation.

The numerical-linguistic approach to solving the modal split problem developed by Teodorovic and Kalic (1996) is based on the results achieved by Wang and Mendel (1992c).

Consider the problem of passengers' choice of one out of two alternative modes of transportation. Without loss of generality, it is assumed that passengers choose a mode of transportation (A1 or A2) according to differences in travel times and travel costs. Travel times (t_1 and t_2), travel costs (c_1 and c_2), differences in times ($\delta t = t_1 - t_2$) and costs ($\delta c = c_1 - c_2$) and the percentage of passengers (p_1) choosing the first mode of transportation are given in *Table 5.19*.

The data shown in *Table 5.19* were taken over from the work by De Ortuzar and Willumsen (1990). Let us respectively denote by δt, δc, and p difference in travel times, difference in travel costs, and percentage of passengers choosing the first mode of transportation. Obviously, the set of input-output data pairs consists of twelve elements $((\delta t^1, \delta c^1), p_1{}^1)$, $((\delta t^2, \delta c^2), p_1{}^2)$, ..., $((\delta t^{12}, \delta c^{12}), p_1{}^{12})$.

Table 5.19. Characteristics of transportation modes

	A1		A2				
	t_1	c_1	t_2	c_2	δt	δc	p_1 [%]
1	26	160	29	72	-3	88	82
2	23	136	25	64	-2	72	80
3	21	120	24	28	-3	92	88
4	18	108	26	20	-8	88	95
5	30	212	33	104	-3	108	72
6	23	156	27	72	-4	84	90
7	18	120	20	36	-2	84	76
8	16	116	23	28	-7	88	93
9	35	240	35	120	0	120	51
10	25	180	24	92	1	88	56
11	20	144	21	36	-1	108	58
12	15	132	17	24	-2	108	64

It can be said that the values of variables δt, δc, and p_1, respectively, belong to the following intervals [-8, 8], [0, 140], [40, 100].

Within each interval a certain number of fuzzy sets is arbitrarily defined. All chosen membership functions within particular intervals are of a triangular shape having equal lengths of triangle bases.(*Figure 5.23*).

Figure 5.23 lists the names of certain fuzzy sets. Let us note that the number of fuzzy sets within each interval was determined arbitrarily. Wang and Mendel (1992c) suggested that according to each input-output data pair one fuzzy rule be made. Consider, for example, the first input-output data pair $((\delta t^1, \delta c^1), p_1{}^1)$. It can be said that δc^1 with the largest grade of membership belongs to fuzzy set **C3** (membership grades of δc^1 in other fuzzy sets defined within interval [0, 140] are smaller). Likewise, δt^1 with the largest membership grade belongs to fuzzy set **T2**, and $p_1{}^1$ to fuzzy set **P4**. The first fuzzy rule candidate to enter the fuzzy rule base reads as follows:

R1: If δt = **T2** and δc = **C3**, then $p_1{}^1$ = **P4**.

Similarly, based on the second input-output data pair the second fuzzy rule is formed, while the third fuzzy rule is formed based on the third data pair, and so on. As there are twelve input-output data pairs, we shall form a total of twelve rules representing the candidates to be included in the fuzzy rule base. Some of the formulated fuzzy rules may be mutually conflicting. Such, for example, are the rules R5 and R12.

From mutually conflicting rules, the rule with the largest membership grade enters the fuzzy rule base. The calculated degrees of certain rules can be corrected by multiplying them with parameter α (0 < α ≤ 1) representing the degree of our belief that the data used to make the rule are credible and useful.

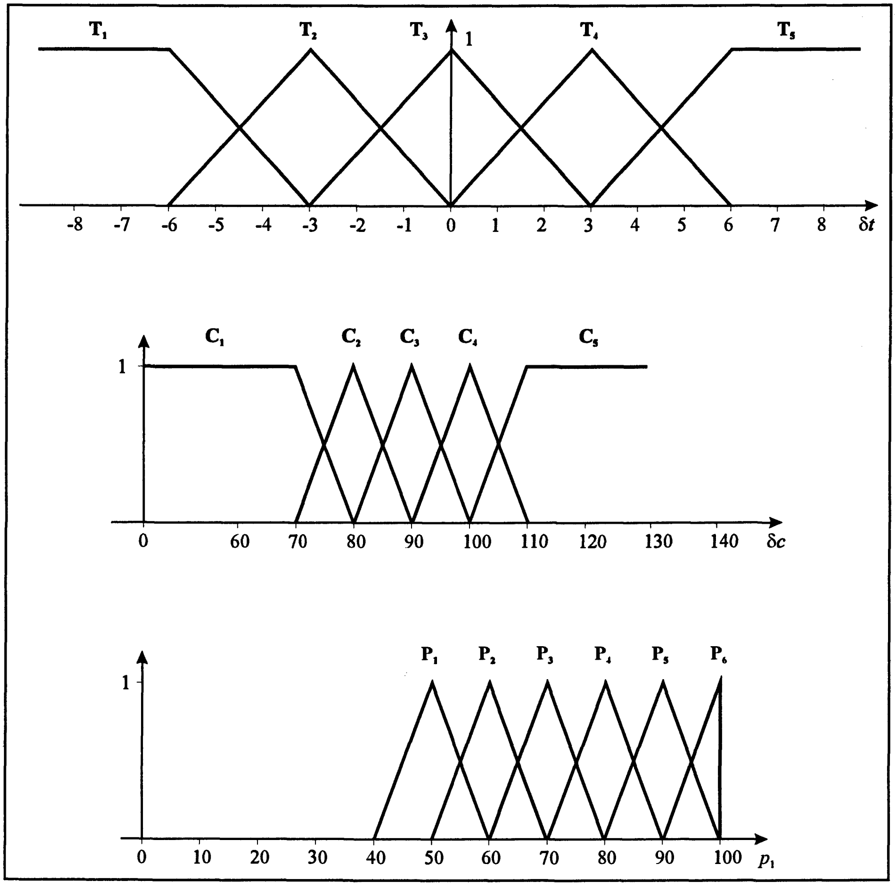

Figure 5.23. Membership functions of fuzzy sets describing input variables δt and δc and output variable p_1

Following the presented procedure suggested by Wang and Mendel (1992c), we formed a fuzzy rule base. This base is represented in *Table 5.20*:

Table 5.20. Fuzzy rule base based on expert knowledge and available input-output data pairs

C5	**P3**	**P3**	**P1**	**P1**	**P1**
C4	**P4**	**P3**	**P1**	**P1**	**P1**
C3	**P6**	**P4**	**P2**	**P1**	**P1**
C2	**P6**	**P4**	**P3**	**P2**	**P1**
C1	**P6**	**P4**	**P3**	**P2**	**P1**
	T1	**T2**	**T3**	**T4**	**T5**

According to the defined rules, for the known values of differences between travel times and those between costs of travel we can calculate the percentage of passengers opting for the first that is, second mode of transportation. *Table 5.21* and *Figure 5.24* show the relation between the real percentage of passengers using the first mode of transportation and the percentage of passengers obtained by the approximate reasoning algorithm.

Table 5.21. Real and calculated percentage of passengers choosing the first mode of transportation

	Real value	Calculated value
1	82	80
2	80	76.36
3	88	77.58
4	95	96.56
5	72	70
6	90	83.92
7	76	71.89
8	93	96.56
9	51	50
10	56	58.84
11	58	57.69
12	64	62.31

Teodorovic and Kalic (1996) developed an approximate reasoning algorithm to solve the modal split problem. Their work represents the first attempt at solving this problem by means of fuzzy logic. The formed fuzzy rule base was obtained according to the available numerical data, with minor subjective corrections performed by the authors. An exceptionally good agreement was obtained between the real numbers of passengers and the values obtained by the model. In further investigations the proposed approximate reasoning algorithm is to be tested on a number of various numerical examples. Also, the comparison of the presented numerical-linguistic approach with possible alternative means of solving the given problem is of considerable importance (classical choice models, artificial neural networks).

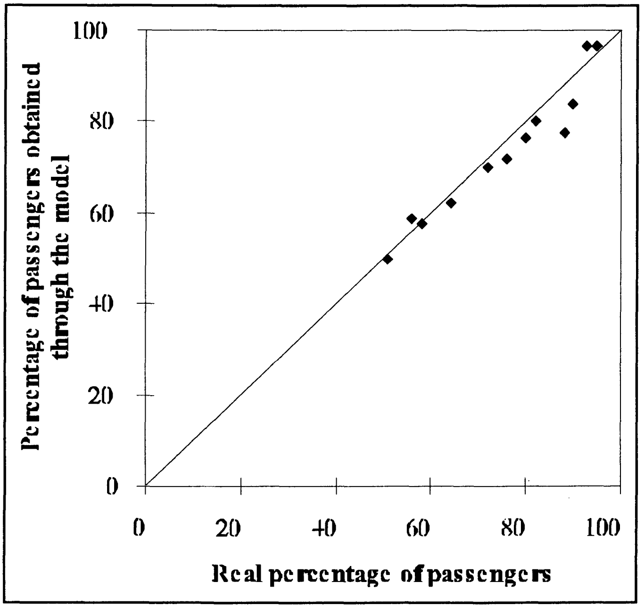

Figure 5.24. Comparison of real percentage of passengers with percentage of passengers obtained through the model

5.6. SIMULTANEOUS DESIGN OF MEMBERSHIP FUNCTIONS AND FUZZY RULES FROM NUMERICAL INFORMATION BY GENETIC ALGORITHMS

As indicated in the previous discussions, the membership value assignment is treated as a combinatorial optimization problem. In this way, the role and importance of an expert in the phase of computing membership functions is diminished. We have specifically indicated the possibilities of using metaheuristic algorithms (simulated annealing, genetic algorithms, tabu search) for membership value assignments. The application of some of the metaheuristic algorithms presupposes the existence of a set of fuzzy rules and the initial shapes of membership functions. When solving a combinatorial optimization problem we seek to attain the best possible fitness of the considered membership functions. In other words, the final shapes of membership functions are the shapes with the best possible fitness between values based on the approximate reasoning model and real values.

Let us note that during the process of finding the final shapes of membership functions the set of the assumed fuzzy rules is not changed.

The two main components of any fuzzy logic controller are the set of membership functions and the set of fuzzy rules. It is clear that between these two sets there are very strong links. Consequently, in designing a fuzzy controller based on numerical information we should simultaneously attempt to design a set of fuzzy rules, as well as a set of membership functions. A number of authors have considered the problem of simultaneous determination of the fuzzy rule set and the set of membership functions (Homafair and McCormick, 1995; Lee and Tagaki, 1993; Homafair and McCormick, 1995). All these authors use genetic algorithms as a tool for simultaneous design of the set of rules and the set of membership functions. The papers by Homafair and McCormick (1995) are particularly important and the results they achieved will be presented in the following discussions.

Homafair and McCormick (1995) have tested a developed model in the case of the cart-centering problem and in the case of the truck backing system. The population size, the maximum number of generations, the probability of crossover, and the probability of mutation were respectively found to be 100, 100, 0.7, and 0.03. In order to select individuals for the next generation, Homafair and McCormick (1995) used the tournament selection. (In the tournament selection at least two members of a population are randomly chosen and the comparison of their fitness is made. The member with the best fitness advances to the next generation.)

Let us suppose that output variable y is dependent on input variables x_1 and x_2. Let us also suppose that the following five fuzzy sets will be used to partition the input and output spaces: **NM** is "negative medium," **NS** is "negative small," **ZE** is "zero," **PS** is "positive small," and **PM** is "positive medium."

Each of the possible sets of rules contains twenty five rules. For example, if we denote **NM** as 1, **NS** as 2, **ZE** as 3, **PS** as 4 and **PM** as 5, possible sets of rules, S_1, S_2, and S_3 can be shown in *Table 5.22*, *Table 5.23* and *Table 5.24*.

Table 5.22. Set of fuzzy rules S_1

				x_2		
		NM	**NS**	**ZE**	**PS**	**PM**
	NM	2	4	2	2	1
	NS	5	3	4	2	3
x_1	**ZE**	2	3	4	5	2
	PS	4	5	2	2	2
	PM	3	3	3	4	5

Table 5.23. Set of fuzzy rules S_2

		x_2 NM	NS	ZE	PS	PM
	NM	3	4	2	1	1
	NS	4	4	4	2	3
x_1	**ZE**	2	3	5	5	2
	PS	4	5	1	1	1
	PM	2	2	2	3	4

A question logically arises as to the "best" set of rules. Is it set S_1 or set S_2 or set S_3 or a possible set S_4, ..., and so on. Let us note that in the considered case there is a total of 5^{25} different sets of rules. Let us assume, for the purpose of simplification, that the membership functions of all fuzzy sets (**NM NS ZE**; **PS** and **PM**) are of a triangular shape. During the process of simultaneous design of membership functions and fuzzy rules, it is important to choose those shapes of membership functions which, along with the corresponding set of fuzzy rules, yield "the best" results. Note the membership function $\mu_{NS}(x)$ of fuzzy set **NS** (*Figure 5.25*):

Table 5.24. Set of fuzzy rules S_3

		x_2 NM	NS	ZE	PS	PM
	NM	3	4	2	2	2
	NS	4	4	3	3	3
x_1	**ZE**	2	3	5	5	1
	PS	3	5	1	2	2
	PM	2	2	3	3	4

Let the initial set **NS** be described by a triangular fuzzy number **NS** = (ns_1, ns_2, ns_3). In the case shown in *Figure 5.25(a)*, during the process of searching the "best" shape of the membership function, the values of ns_1, ns_2, and ns_3 move along the x axis. In the case shown in *Figure 5.25(b)*, the value of ns_2 does not move, therefore only the length of the base of the triangle is changed. For each of the input fuzzy sets **NM, NS, ZE, PS**, and **PM** Homafair and McCormick (1995) defined five possible shapes of the membership functions differing in the length of the base of the triangle. The bases of the triangles that partitioned the output space were fixed for the purpose of decreasing the computing time. Homafair and McCormick used a string to represent a given rule set and membership functions combination. The individual locations which make up a string are called alleles. The string had a total of thirty five alleles. The first twenty five alleles were related to a particular rule set. The following five alleles described the chosen shapes of membership functions for the fuzzy sets of input variable x_1, whereas the last five alleles were related to the chosen shapes of membership functions for the fuzzy sets of input variable x_2. Note, for example, the following string:

3421144423235524511122234	43252	31425
Fuzzy rule set S_2	x_1	x_2

The first twenty five alleles in the string are obtained by reading the rows in

Table 5.23. Based on the last ten values of the alleles, it can be determined which shapes of the fuzzy sets associated with input variables x_1 and x_2 participate in a given rule set and membership functions combination. We can see that Homafair and McCormick used an integer-based instead of a binary-based string to represent a particular fuzzy logic controller. After representing the controller in the form of an integer-based string and by applying genetic algorithms, it is possible to determine the controller (fuzzy set rules and corresponding shapes of membership functions) producing the best results. The process of designing the best controller involves a number of computing experiments as well as varying the population size, the maximum number of generations, the probability of crossover, and the probability of mutation.

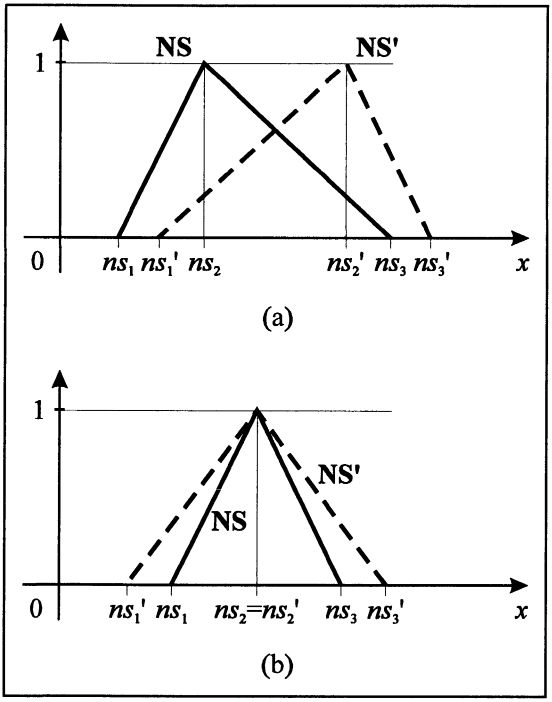

Figure 5.25. Finding the 'best' shape of membership function $\mu_{NS}(x)$

5.7. AN APPLICATION OF NEUROFUZZY MODELING: THE VEHICLE ASSIGNMENT PROBLEM

5.7.1. The basic concepts of neurofuzzy modeling

The fuzzy set theory provides a mathematical framework that can deal with the uncertainties inherent in the process of knowledge acquisition, thinking, and reasoning that is, it provides the quantification of certain linguistic attributes associated with human thinking or knowledge. The idea of computational neural networks came from the realization of incredible capabilities of learning and adaptation of a set of biological neurons. Computational neural networks imitate to a small extent some of the operations perceived in biological neurons: they can be trained to save, recognize, and search the shapes or elements of databases, solve combinatorial optimization problems, manage vaguely defined processes that is, to assess the observed function when its shape is unknown. For example, neural networks can be trained to imitate the behavior of an expert that we are even unable to formulate. In other words, neural networks are able to recognize without defining which characterizes a highly intelligent behavior (Kosko, 1992b). This property enables these systems to make generalizations.

Fuzzy logic systems are of a later date than neural systems, most of the applications coming from Japan. Fuzzy systems contain fuzzy rule bases and reason by a parallel inference. When an input value is brought to the system, all the rules are simultaneously carried out up to a certain degree in order to reach the solution that is, conclusion.

Neural and fuzzy systems estimate functions from samples without requiring a mathematical formulation of the dependence of output on input values and they behave like associative memories (Kosko, 1992b). They learn from examples that is, from samples. They differ in the way of estimating functions (the type of the samples required), the manner of saving and representing the samples and the manner of associative inference (mapping of input into output data). These differences can be simply detected during the development of neural and fuzzy systems.

The choice of a neural or fuzzy system in modelling a problem depends on the nature of the problem and the availability of numerical and linguistic information (*Figure 5.26*). Fuzzy systems estimate functions according to fuzzy set samples ($\mathbf{A}_i$, $\mathbf{B}_i$). Neural systems use numerical samples (x_i, y_i).

In recent years fuzzy systems have estimated the observed functions from numerical samples (examples). An expert can systematically express a rule base or a fuzzy system can be adapted of modified according to a representative numerical sample. In the latter case neuro and fuzzy systems are naturally combined. This combination resembles an adaptive system with sensitive and cognitive properties. Neural networks "blindly" generate and correct fuzzy rules from the training set of data. Such *adaptive fuzzy systems*, *neural fuzzy systems*, or *neurofuzzy models* can potentially include the advantages of both techniques. While neural networks with associative memory are easily trained and have known properties of convergence and stability, their internal representation remains unclear. In contrast, the generated rule base within an adaptive fuzzy algorithm can provide a certain amount of functional transparency through the dependency within each rule.

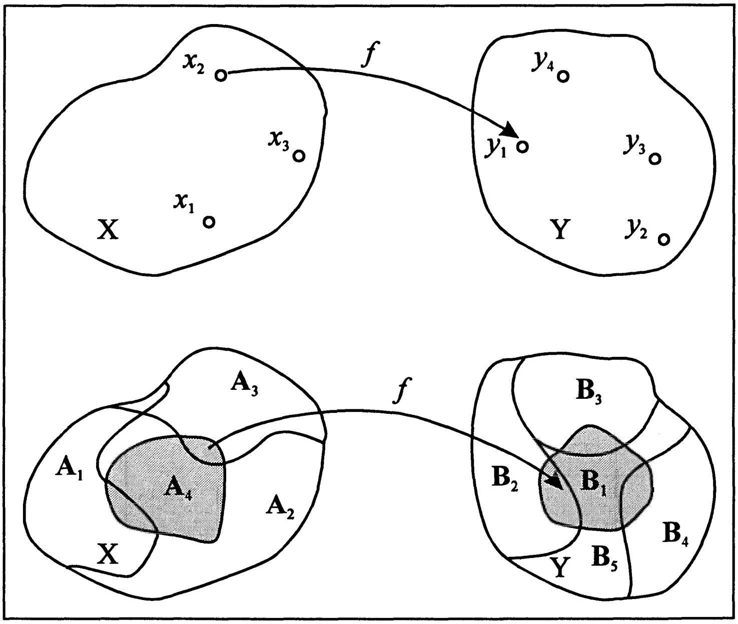

Figure 5.26. Estimation of the mapping *f* by neural and fuzzy systems

The neurofuzzy model is a connective model of application of a set of fuzzy rules by which fuzzy reasoning is performed. The neurofuzzy models developed so far differ in their structure and functioning. For example, the applied type of a fuzzy rule affects the network's structure while the applied inference procedure influences the choice of functions of the network's nodes and interpretation of weights of the network's branches. In addition, the proposed neurofuzzy models can function in three ways:

1. *Fixed functioning* ("fixed membership functions and fixed fuzzy rules"): The network contains a particular set of fuzzy rules, performs fuzzy reasoning, but can neither learn nor adapt.
2. *The neural network is capable of learning*: The neural network is structured such that it can acquire knowledge in a particular form, such as a type of fuzzy rules. The network's structure is most commonly determined in a random fashion. The network is then trained on a set of data. The rules are formulated from the structure of the trained network.
3. *The neural network is capable of adapting*: The neural network is configured according to a set of existing rules. In training the network new rules are formulated from the network's structure and the old ones are changed. In the work of researchers two approaches can be distinguished: "fixed membership functions and adaptable fuzzy rules" and "adaptable membership functions and adaptable fuzzy rules." In both cases the phenomenon of disastrous forgetting must be investigated, which refers to the network's characteristic to forget all it has learned from previous training pairs when they are no longer used in training but are instead replaced by some other pairs.

Putz and Weber (1996) presented a systematic overview of approaches for automatic generation of fuzzy logic systems (neural networks, genetic algorithms, rule-based approaches, and other). The basic principles of neurofuzzy modeling are given in the works of Jang (1997), Lin and Lee (1995), Kasabov (1996), and Brown and Harris (1994).

Jang and Sun (1995) proposed adaptive neurofuzzy systems having the same function as the fuzzy systems of the Sugeno type. In other words, a fuzzy system is mapped into an adaptive multilayered feedforward neural network. The adaptive neural network consists of a set of nodes connected by the network's branches, each node being a process element transforming the input values into a single output value while each branch follows the direction of the signal transmission. Unlike the classical neural network models, the network branches are not associated with the strengths of the nodes connectivity that is, weights. The parameters modified during the training are locally distributed: each node is characterized by a local set of parameters. The union of these sets represents the overall set of the network's adaptive parameters. If the set of parameters describing a node is not empty, such a node is adaptive and is marked by a square. If the set of parameters describing a node is empty, it is a fixed node and is marked by a circle.

Neurofuzzy algorithms are applied in the developed traffic and transportation systems, including the European programs of intelligent vehicles Prometheus and Drive and the American IVHS program.

Kong and Kosko (1992) developed a fuzzy system and a neural network used in managing a truck or a truck and trailer when stopping at an unloading terminal at a parking lot. In developing the fuzzy system the authors used a small number of input/output linguistic examples obtained from the drivers. In developing the neural network as well as the adaptive fuzzy system the authors used a large number of input/output numerical data. When comparing the outcomes, the resulting trajectories of the vehicles were observed. When the fuzzy system was used, the vehicle trajectories were always gentle and short, which is considered desirable. When using the neural network, the vehicle trajectories were occasionally irregular. Furthermore, the training of the network was prolonged as it was performed by a backpropagation algorithm. A sensitivity analysis of the results was also performed by randomly removing some sets of training data during the training and the fuzzy system performance was considered when some of the rules were randomly removed. The fuzzy systems proved to be considerably stable. The neural network performance continued to be good except for the prolonged duration of training. By performing a number of experiments, the authors achieved the following result: When the model is developed in the form of a neural network, with some effort the fuzzy rules approximately describing the behavior of the neural network can be generated. The fuzzy system developed in this way can be improved by adding the rules from experience obtained from the experts or by formulating rules according to new training data.

5.7.2. Tuning the fuzzy logic systems: Neurofuzzy modeling

The considered problem is the daily assignment of a fleet of available vehicles to a certain number of transportation requests. It is stated explicitly in the work of Milosavljevic et al. (1996). A transportation company receives a large number of requests every day from clients wanting to send goods to different destinations. Their fleets of vehicles most often consist of several different types of vehicles. The transportation company has a depot from which the vehicles depart and to which they return after completing their trips. One type or a variety of vehicle types can take part in the delivery to each node. In the research done by Vukadinovic et al. (1998), the case when only one type of vehicle takes part in serving a node is considered.

Each transportation request is characterized by a large number of parameters, including the most important: type of freight, amount of freight (weight and volume), loading and unloading sites, preferred time of loading and/or unloading, and the distance the freight is to be transported. The common practice is that dispatchers assign vehicles to transportation requests. The dispatcher's knowledge can be considered as experience-based heuristic knowledge. Since fuzzy sets can describe qualitative and imprecise information, fuzzy logic is used as a tool to transform the dispatcher's heuristic rules into an automatic strategy. A fuzzy system (fuzzy logic controller) provides an algorithm that converts the linguistic control strategy based on the observed dispatcher's behavior and knowledge into an automatic control strategy.

The main objective of the research done by Vukadinovic et al. (1998) was to examine the possibility of improving the performance of a fuzzy system in order to improve the quality of decisions. The method used for tuning the initial fuzzy system is mapping of a fuzzy system into a feedforward neural network trainable with supervised learning. As a result of the study, it was shown that the proposed fuzzy system equipped with learning capability can be used to imitate the actual dispatcher's assignment policy.

5.7.3. Statement of the problem

The problem analyzed and solved by Milosavljevic et al. (1996) is to determine the vehicle type to meet each transportation request. The classical approach to solving this type of problems is through mathematical programming. The objective is to design a set of vehicle assignments to transportation requests that incurs minimum costs, while assuring that constraints describing proper utilization of vehicles and "coverage" of the planned transportation tasks (satisfaction of transportation requests) are not violated. However, the conventional approach can hardly reflect the knowledge and the intelligence of a dispatcher that is, his or her ability to deal with uncertainty and imprecision.

In addition to the characteristics of the transportation request, when assigning a specific type of vehicle to a specific transportation request, the dispatcher must also bear in mind the total number of available vehicles, the available number of vehicles by vehicle type, the number of vehicles temporarily out of working order, and vehicles undergoing technical examinations or preventive maintenance work. In the research done by Milosavljevic et al. (1996) these constraints are taken care of by the developed heuristic algorithm. The assumption in the research done by Vukadinovic et al. (1998) was that the number of available vehicles by vehicle type is unlimited that is, the model imitates the dispatcher's decision regarding choice of a vehicle type under no constraints.

5.7.3.1. Initial fuzzy logic systems for assignment of vehicles to transportation requests

Milosavljevic et al. (1996) observed and analyzed dispatchers' behavior and concluded that every dispatcher has a pronounced subjective feeling about which type of vehicle corresponds to which transportation request. This subjective feeling concerns both the suitability of the vehicle in terms of the distance to be traveled and the vehicle capacity in terms of the amount of freight to be transported. Milosavljevic et al. (1996) noted that the dispatchers have certain preferences: (1) "very strong" preference is given to a decision that will meet the request with a vehicle type having "high" suitability in terms of distance and "high" capacity utilization, or (2) "very weak" preference is given to a decision that will meet the request with a vehicle that has "low" suitability regarding distance and "low" capacity utilization.

The heuristic algorithm used to assign vehicles to planned transportation requests developed by Milosavljevic et al. (1996) is tested on a fleet of vehicles containing three different types of vehicle with capacities 4.4, 7, and 14 tons. For every type of vehicle, they developed a corresponding fuzzy system (Mamdani's model) to determine the dispatcher's preference strength ("very weak" (**VWP**), "weak" (**WP**), "medium" (**MP**), "strong" (**SP**), "very strong" (**VSP**)) in terms of meeting a specific transportation request with the type of vehicle in question. They noticed that the dispatchers consider the suitability of different types of vehicles as being "low" (**LS**), "medium" (**MS**), and "high" (**HS**) depending on the given distance the freight is to be transported. Also, the capacity utilization (the relationship between the amount of freight and the vehicle's declared capacity, expressed as a percentage) is estimated by the dispatcher as "low" (**LCU**), "medium" (**MCU**), or "high" (**HCU**). The developed fuzzy systems for each type of vehicle are shown in *Table 5.25*. They differ from each other in terms of the number of rules they contain, and the shapes of the membership functions of individual fuzzy sets. The membership functions of the input/output variables are considered as the initial membership functions.

Table 5.25. The initial fuzzy systems that determine the dispatcher's preference in meeting a transportation request with a vehicle of 4.4(a), 7(b) and 14(c) tons capacities

		Capacity utilization		
(a)		**LCU**	**MCU**	**HCU**
	LS	**MP**	**SP**	**SP**
Suitability	**MS**	**WP**	**WP**	**MP**
	HS	**VWP**	**VWP**	**WP**

(b)		Capacity utilization LCU	MCU	HCU
	LS	**VWP**	**VWP**	**WP**
Suitability	MS	**MP**	**SP**	**SP**
	HS	**WP**	**WP**	**MP**

(c)		Capacity utilization LCU	MCU	HCU
	LS	**VWP**	**WP**	**MP**
Suitability	MS	**MP**	**SP**	**SP**
	HS	**SP**	**VSP**	**VSP**

The major problems in the construction of fuzzy systems are the extraction of explicit dispatcher's heuristic control rules and the determination of the appropriate membership functions of the input/output variables. It implies a long and elaborate communication with a number of experienced dispatchers, which could be very difficult. For example, the specification of the membership functions is quite subjective, which means that the membership functions specified for the same concept (for example, low vehicle capacity utilization) by different dispatchers may vary considerably. Thus, the performance of the developed fuzzy system depends on the number of available dispatchers and the ability of an analyst to extract their assignment policy.

The problem considered in the research done by Vukadinovic et al. (1998) is tuning the membership functions of the input/output variables. Based on real numerical training data (input/output pairs), the neural network is used to tune and adapt the initial fuzzy system to achieve better performance.

5.7.4. Tuning the fuzzy logic systems for assignment of vehicles to transportation requests: Neurofuzzy modeling

The term *neurofuzzy modeling* refers to the way of applying various learning rules developed in the neural network literature to fuzzy systems. Compared to black-box modeling techniques like neural networks, fuzzy systems are to a certain degree transparent to interpretation and analysis. Fuzzy systems require that we work out the fuzzy rules operating to different degrees and estimate the parameters of the membership functions (as well as functional forms for membership functions). We can use neural networks to estimate the parameters of fuzzy systems. But neural networks require an accurate (statistically representative) set of numerical training data.

The basic goal of neurofuzzy modeling in the presented transportation application is to decrease the dispatcher's role in the construction of fuzzy systems relying primarily on past numerical examples of the dispatcher's decisions. In the research done by Vukadinovic et al. (1998), the aim of learning is to set the appropriate membership functions of the input/output variables (they are different for each type of vehicle). It is assumed that the fuzzy rules are properly extracted from the dispatcher. The basic assumption is that one type of vehicle is used to meet every transportation request. Considering the developed fuzzy logic systems by Milosavljevic et al. (1996), a neural network is configured individually for every type of vehicle. However, the basic structure of a five layered adaptive neural network that has exactly the same function as the fuzzy logic system (Mamdani's model) is the same.

5.7.4.1. Network configuration

Specifying the set of processing units connected through directed links and what they represent is typically the first stage of specifying a neural network. The initial fuzzy system (for a vehicle with a capacity of 14 tons) is mapped into a five-layered adaptive neural network with a restricted connectivity structure that is shown in *Figure 5.27*. The proposed neural network is referred to as a five-layered network because five layers perform operations. The adaptive network shown in *Figure 5.27* is a feedforward layered network because the output of each unit propagates from the input side (left) to the output side (right).

The number of configured neural networks corresponds to the number of vehicle types.

5.7.4.2. Pattern of connectivity

Neural networks are designed to establish and compute a function from input space to output space. The examples of the dispatcher's assignment policy are expressed as input-output vectors and training and testing pairs are formed.

In the research done by Vukadinovic et al. (1998), the network that has a fixed structure is configured based on the operation of the fuzzy system (Mamdani's model). The input layer consists of two units representing: distance the freight is to be transported, x_1 (needed to determine vehicle suitability in terms of distance), and amount of freight, x_2 (needed to determine vehicle capacity utilization). It simply transfers inputs further via the interconnections to the hidden or first layer. The first unit in the input layer is connected with the first three units in the first layer. The second unit in the input layer is connected with the second three units in the first layer. The strengths of connections between the units in the input layer and the units in the first layer are crisp numbers equal to 1.

The first layer consists of 3 + 3 units representing the number of verbal descriptions quantified by fuzzy sets ("low," "medium," "high") for each input variable. The input variables are vehicle suitability in terms of distance and vehicle capacity utilization. The vehicle capacity utilization is the ratio of the amount of freight transported by a vehicle to the vehicle's capacity. Every unit in the first layer is an adaptive unit with an output being the membership value of the premise part (A fuzzy If-Then rule assumes the form: If premise, then consequence). For example, the output from the first unit (representing the membership function - "low vehicle suitability," **LS**) is the grade of membership of a distance x_1 to a fuzzy set "low suitability," $\mu_{LS}(x_1)$.

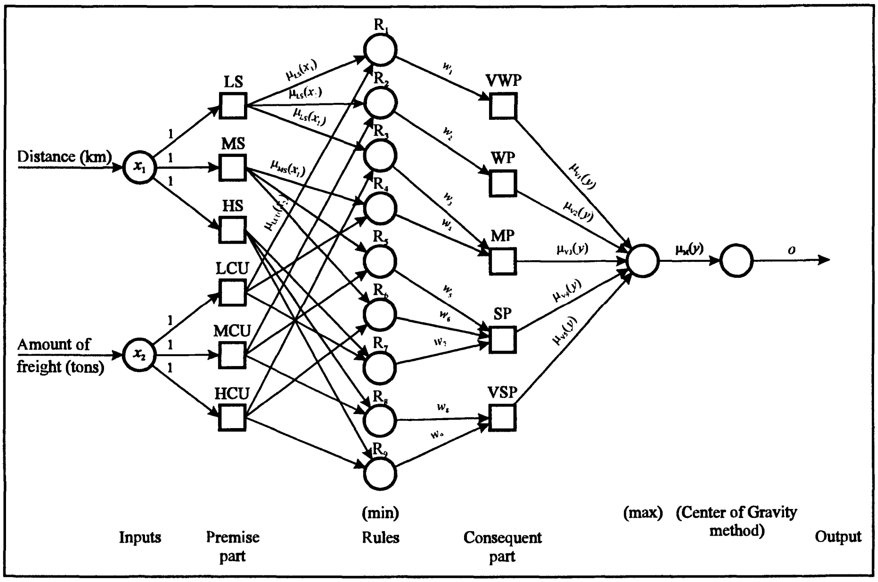

Figure 5.27. A five-layered feedforward adaptive neural network configured for a vehicle with a capacity of 14 tons

The number of units in the second layer equals the number of fuzzy rules. The number of fuzzy rules is equal to the number of cells in the rectangle shown in *Table 5.25*(c). Rule 1 corresponds to the first cell (NW corner), and so on. Every unit in this layer is a fixed unit that calculates the minimum value of incoming two inputs. The outputs from this layer are firing strengths of rules. For example, the output from the first unit in the second layer is $w_1=\min\{\mu_{LS}(x_1), \mu_{LCU}(x_2)\}$.

The third layer has five adaptive units representing strength of the dispatcher's preference ("very weak," "weak," "medium," "strong," "very strong") in meeting a specific transportation request with a type of vehicle in question. Each unit in this layer calculates intersection of a fuzzy set (consequent) with the maximum of incoming rules' firing strengths. For example, the fourth unit calculates intersection of a fuzzy set "strong preference" with the maximum of rules' R_5, R_6, R_7 firing strengths: $\mu_{V4}(y) = \min\{w_i, \mu_{SP}(y)\}$, where $w_i = \max\{w_5, w_6, w_7\}$.

The single unit in the fourth layer is a fixed unit that computes the overall output of the fuzzy system: $\mu_M(y) = \max\{\mu_{V1}(y), \mu_{V2}(y), \mu_{V3}(y), \mu_{V4}(y), \mu_{V5}(y)\}$. The obtained output is then defuzzified in the single unit in the fifth layer. The selection of final crisp value can be made in various ways. In this application the action that is closest to the center of gravity has been computed (center-of-gravity method). The network's output, o, is a real value among 0 and 1. The dispatcher's or targeted output equals either 0 or 1, meaning that the vehicle is either assigned to meet a specific transportation request or not.

5.7.4.3. Supervised learning (A simulated annealing algorithm)

In order to recover and emulate the dispatcher's assignment policy by fuzzy system, the aim of learning is to set the membership functions of the input/output variables to some adequate functions. When a known input, describing a transportation request, is applied to all trained networks, a maximum of networks outputs (the model output) is chosen. Since the trained neural network corresponds to the vehicle type, the maximum value of output determines the vehicle type to be assigned to meet a transportation request. The resulting choice of a vehicle type is then compared with the targeted or dispatcher's choice. The neural networks performances are measured as the deviation between the targeted output and the model output across all numerical examples of the dispatcher's decisions. This discrepancy or the error measure is considered as the objective function and the heuristic simulated annealing is used to minimize it. Since the application of a simulated annealing requires a large number of experiments, the training process was very long. However, the tuned fuzzy systems yield results superior to those obtained by the initial fuzzy controllers and can be used in real time.

In this application, the objective function that has to be minimized is calculated as a sum of differences between the model output (maximum of the networks outputs) and the targeted output over all training pairs. The heuristic simulated annealing is used to minimize the objective function.

The statistical training method such as the simulated annealing requires the definition of an energy function (objective function) depending on the parameters of the neural network. Whenever a new set of membership functions is generated randomly, the resulting energy is determined. If the obtained energy is improved, then a new set of membership functions is memorized, otherwise the acceptance or the rejection of the change is decided according to a given probability distribution. The possibility that the change that worsens (increases) the energy is retained implies that the algorithm would hardly be trapped in local energy minima. In this research, the fact that the algorithm would hardly be trapped in local minima that is, it will converge to a global minimum was the main reason to chose this method as a learning rule. The disadvantage of using the simulated annealing algorithm as a learning rule of the neural network is a very long training period.

The simulated annealing algorithm consists of the following steps:

Step 1: Develop a proper annealing schedule $\{t_1, t_2, \ldots, t_K\}$ consisting of a sequence of temperatures (control parameters), $t_1 > t_2 > \ldots > t_K$, and the amount of time required to reach the equilibrium at each temperature. Set $i = 1$.

Step 2: Generate a set of initial membership functions. For all the input vectors in a training set obtain the model output (maximum of the networks outputs) and calculate the objective function value (OF_{old}). The objective function minimizes the sum of errors between the targeted outputs and the model outputs.

Step 3: Generate a new set of membership functions by a small perturbation. The middle values (one in a case of a triangle or two in a case of a trapezoid) that are changed are uniformly distributed on arbitrarily defined intervals. Obtain the model output (maximum of the networks outputs) with the same input vectors (for the complete training set) and calculate the new objective function value (OF_{new}). Evaluate the change in the objective function ($\delta = OF_{\text{new}} - OF_{\text{old}}$). If $\delta < 0$, go to Step 5. Otherwise, go to Step 4.

Step 4: ($\delta \geq 0$) Compare a random variable, r, drawn from a uniform distribution on the [0, 1] interval, with the probability of accepting the new set of membership functions $P(\delta) = \exp(-\delta/t_i)$. If $r < P(\delta)$, go to Step 5. Otherwise, keep the old set of membership functions, and go to Step 3.

Step 5: ($\delta < 0$ or $r < P(\delta)$) Memorize the new set of membership functions and the new objective function value.

Step 6: If the thermal equilibrium has been reached at the temperature t_i, set $i = i+1$. Steady state or equilibrium is reached when we observe that an improvement of the objective function is highly unlikely. An epoch is an interval between checking if the equilibrium is reached. The epoch implies λ exchanges of all membership functions, where λ is a predefined number. The best solution that is, the least sum of errors between the model outputs and targeted outputs obtained through λ exchanges of the membership functions, represents the epoch. Consider the case where k epochs have already been generated. After the next epoch, the equilibrium is reached if

$$\frac{\left|OFV_{k+1} - OFV_p\right|}{OFV_{k+1}} < \varepsilon, \quad k = 1, \ldots, MEP - 1 \tag{5.8}$$

where OFV_{k+1} is objective function value that represents the k+1-st epoch, OFV_p is the least objective function value of all previous epochs' solutions, and ε is predefined constant. The maximum number of generated epochs at one temperature if the thermal equilibrium is not reached in the meantime, *MEP*, is set in advance. If $i > K$, the algorithm is completed.

The solutions obtained by a simulated annealing algorithm do not depend on the initial solution and usually approximate the optimal solution. However, the annealing schedule that is, the way the temperature gradually decreases and the initial temperature influence the performance of the algorithm. Initially, a temperature is given a high value; then it is slowly reduced until some small value, for which no deteriorations are accepted any more, is reached. Thus, the convergence of the obtained values for the membership functions is inherited from the convergence of the simulated annealing algorithm.

5.7.5. Numerical example

In the research done by Vukadinovic et al. (1998), the method used for tuning the fuzzy systems is mapping of a fuzzy system into a feedforward neural network trainable with supervised learning. The initial fuzzy systems developed by Milosavljevic et al. (1996) correspond to three different types of vehicle with capacities 4.4, 7, and 14 tons. Their fuzzy rule bases are shown in *Table 5.25*. The initial and the tuned membership functions for the input/output variables are shown in *Figure 5.28*, *Figure 5.29* and *Figure 5.30*.

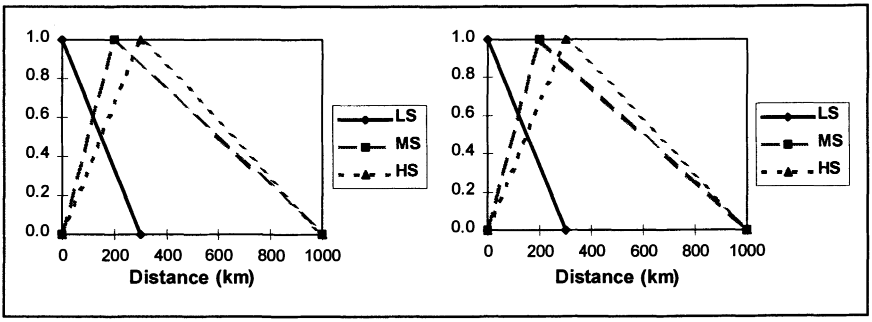

Figure 5.28. The initial and the final membership functions for the suitability of a vehicle with a capacity of 14 tons in terms of distance to be traveled

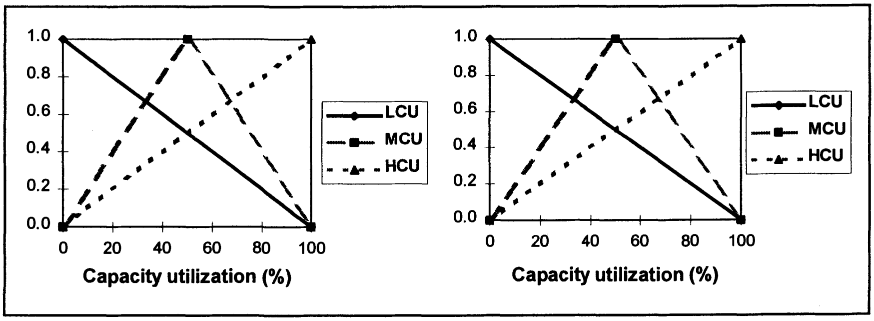

Figure 5.29. The initial and the final membership functions for the vehicle capacity utilization (for a vehicle with a capacity of 14 tons)

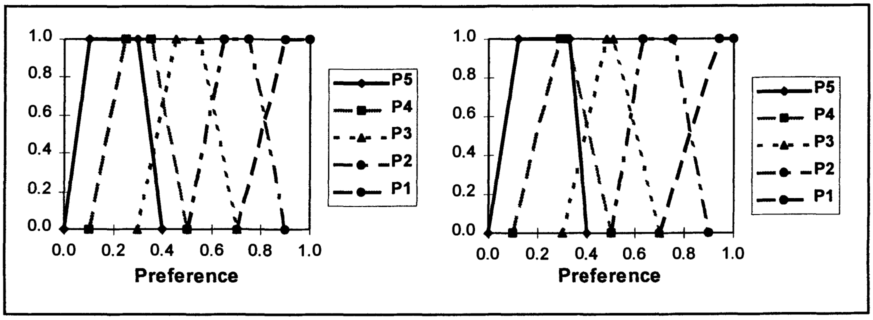

Figure 5.30. The initial and the final membership functions for the output variable that correspond to a vehicle with a capacity of 14 tons

The application of simulated annealing requires a large number of experiments. The initial temperature and the number of temperatures are varied during numerous experiments. The best result is obtained when an array of forty temperatures ($t_{i+1}=0.9t_i$) and an initial temperature $t_1 = 40$ are used (*Figure 5.31*).

The maximum number of generated epochs at one temperature $MEP = 30$. In other words, thirty epochs are generated if the thermal equilibrium is not reached in the meantime. The epoch implied twenty exchanges of all middle values of the membership functions. The value of ε used to check if the equilibrium is reached was 0.05.

The main difficulty in using a neurofuzzy modeling is in obtaining good training data. The reliance on the obtained result will be enhanced if sufficient and representative training data are available. Some authors suggest that as many as 100 data points is needed for each unit in a hidden layer (Hall and Smith, 1992). However, many researchers report quite acceptable results from relatively small training data set. In general, the more data the better the solution. The purpose of testing a neural network behavior on real (unseen) data is to provide an evidence about the adequacy of the network. As Dougherty (1995) claims: "It is important to remember that much more data are often required for training then testing; therefore where data are limited it may be better to reserve what real data are available for the testing phase, as this allows more credible results to be produced."

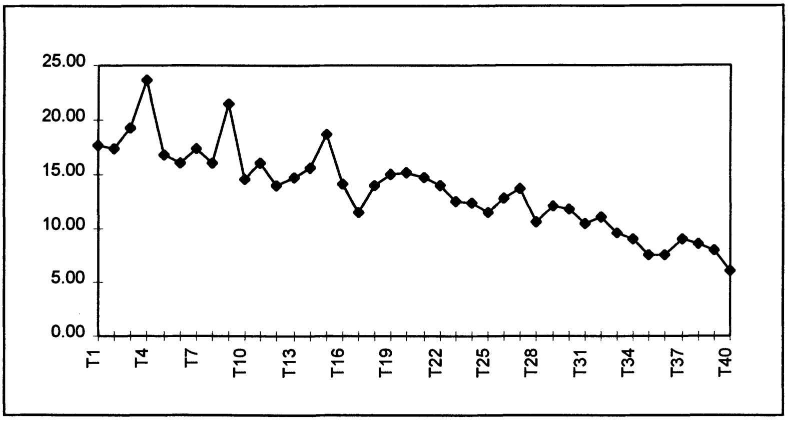

Figure 5.31. Change of the objective function

The three developed neural networks (corresponding to three vehicle types) are trained on 100 different examples of the dispatcher's decisions, and tested on sixty examples. The *Table 5.26* and *Table 5.27* show the characteristics of the transportation requests to be met as well as the decisions made by the dispatcher and the model. Incompatibility of the decisions is denoted by "***."

Table 5.26. Characteristics of 100 transportation requests to be met (training pairs)

Q (tons)	L (km)	Dispatcher's decision	Model output	Q (tons)	L (km)	Dispatcher's decision	Model output
1.8	101	1	2 ***	18	80	2	2
2.5	60	1	1	18.3	48	2	2
3	19	1	1	18.8	111	2	2
3.5	56	1	1	19	52	2	2
3.5	75	1	1	19.2	80	2	2
3.8	60	1	1	19.5	89	2	2
4	43	1	1	19.8	130	2	2
4.2	72	1	1	20	45	2	1 ***
4.4	75	1	1	20.3	130	2	2
4.9	84	2	2	21	101	2	2
5	74	2	2	21	104	2	2
5	125	2	2	21.4	36	1	1
5.7	106	2	2	22	52	1	1
6	41	2	1 ***	22	66	1	1
6	98	2	2	22.7	57	2	2
6.3	92	2	2	23	311	3	3
6.7	138	2	2	23	371	3	3
7	130	2	2	24	281	3	3
7	190	2	2	24	305	3	3
7.6	48	1	1	24.5	330	3	3
7.8	29	1	1	24.7	262	3	3
8	28	1	1	25	12	1	1
8	29	1	1	25	370	3	3
8	41	1	1	26	45	1	1
8.3	42	1	1	27	114	2	2
8.4	67	1	1	27	115	2	2
9.8	341	3	3	27	291	3	3
9.8	440	3	3	27	420	3	3
10	498	3	3	27.2	115	2	2
10.8	362	3	3	28	650	3	3
11	264	3	3	29	35	1	1
12	26	1	1	29.2	36	1	1
12	297	3	3	30	14	1	1
12	420	3	3	30	25	1	1
12.5	400	3	3	30.3	60	1	1
12.9	25	1	1	31	100	2	2
13	27	1	1	31.8	68	2	2
13.5	321	3	3	32.6	130	2	2
14	260	3	3	33	115	2	2
15	23	1	1	33.1	100	2	2
15.6	40	1	1	35	105	2	2
15.7	17	1	1	38	20	1	1

Table 5.26. Characteristics of 100 transportation requests to be met (training pairs) (continued)

Q (tons)	L (km)	Dispatcher's decision	Model output	Q (tons)	L (km)	Dispatcher's decision	Model output
16	28	1	1	38	360	3	3
16.2	17	1	1	39	11	1	1
16.5	28	1	1	40	280	3	3
16.8	52	1	1	41	371	3	3
16.9	26	1	1	44	12	1	1
17	31	1	1	51	355	3	3
17.2	62	1	1	54	257	3	3
18	62	2	2	56	125	2	2

Table 5.27. Characteristics of 60 transportation requests to be met (testing pairs)

Q (tons)	L (km)	Dispatcher's decision	Model output	Q (tons)	L (km)	Dispatcher's decision	Model output
3.6	66	1	1	20	255	2	2
4	36	1	1	20.3	220	2	2
4	76	1	1	21	165	2	2
4.2	80	1	1	21	245	2	2
4.2	105	2	1 ***	21.5	23	1	1
4.3	90	1	1	22	50	1	1
7	230	2	2	22.5	380	3	3
8.5	36	1	1	23	410	3	3
9.3	341	3	3	24	260	3	3
11	284	3	3	24	295	3	3
11	300	3	3	24	405	3	3
11.4	257	3	3	24.5	430	3	3
12	297	3	3	25	17	1	1
12.2	17	1	1	25	390	3	3
12.7	320	3	3	25.5	20	1	1
13	17	1	1	25.5	400	3	3
16	33	1	1	26	39	1	1
16.5	28	1	1	26	50	1	1
17	52	1	1	27	17	2	1 ***
17.3	63	1	1	27	160	2	2
17.8	60	2	2	27	180	2	2
17.8	164	2	2	27	415	3	3
18	40	2	1 ***	28	125	2	2
18	73	2	2	28	360	3	3
18	186	2	2	28	400	3	3
18.2	48	2	2	29	17	1	1
18.5	160	2	2	30	17	1	1
19	85	2	2	31.5	230	2	2
19	145	2	2	33	135	2	2
20	100	2	2	35	145	2	2

Regarding the training pairs, the tuned fuzzy systems predict the dispatcher's decision in 97% of the cases. The networks are tested on sixty examples and the following result is obtained: in 95% of the cases the tuned fuzzy systems predict the dispatcher's decision.

Based on the obtained results it may be concluded that the proposed neurofuzzy models can assist and guide the dispatcher to assign vehicle types to meet transportation requests. Once the networks have been trained on a large set of representative data, there is a justified belief that they (in the form of fuzzy logic systems) will perform reliably without frequent retraining when installed in the field.

References

Abe, S., and Lan, M.-S. (1995) "A Method for Fuzzy Rules Extraction Directly from Numerical Data and Its Application to Pattern Classification." *IEEE Transactions on Fuzzy Systems*, 3, 18-28.

Adamo, J.M. (1981) "Fuzzy Decision Trees." *Fuzzy Sets and Systems*, 4, 207-219.

Agard, M. (1970) "Monthly Assighment of Stewards by Computer." *Proceedings of the Tenth AGIFORS Symposium.* Terrigal, Australia: AGIFORS.

Akiyama, T., and Fu-Shao, C. (1993) "Fuzzy Mathematical Programming for Traffic Safety Planning on an Urban Expressway." *Transportation Planning and Technology*, 17, 179-190.

Akiyama, T., Fu-Shao, C., and Sasaki, T. (1994) "Traffic Flow on Urban Networks with Fuzzy Information." *Memorial Faculty of Engineering Kyoto University*, 56, 1-22.

Akiyama, T., Nakamura, K., and Sasaki, T. (1993) "Traffic Diversion Model on Urban Expressway by Fuzzy Reasoning." *Selected Proceedings of the Sixth World Conference of Transport Research, LYON 92*, 1011-1022 (vol. 2).

Akiyama, T., and Shao, C-F. (1993) "Fuzzy Mathematical Programming for Traffic Safety Planning on an Urban Expressway." *Transportation Planning and Technology*, 17, 179-190.

Akiyama, T., and Tsuboi, H. (1996) "Description of Route Choice Behaviour by Multi-Stage Fuzzy Reasoning." Paper presented at the Highways to the Next Century Conference, Hong Kong.

Akiyama, T., and Yamanishi, H. (1993) "Travel Time Information Service Device Based on Fuzzy Sets Theory." In B.M. Ayyub (ed.), *Proceedings of ISUMA '93, The Second International Symposium on Uncertainty Modeling and Analysis* (pp. 238-245), College Park, MD: IEEE Computer Press.

Al-Deek, H., and Kanafani, A., (1993) "Modeling the Benefits of Advanced Traveler Information System in Corridors with Incidents." *Transportation Research*, 1C, 303-324.

Alfa, A.S. (1986) "Scheduling of Vehicles for Transportation of the Elderly." *Transportation Planning and Technology*, 11, 203-212.

Alstrup, J., Boas, S., Madsen, O.B.G., and Vidal, R.V.V. (1986) "Booking Policy for Flights with Two Types of Passengers." *European Journal of Operational Research*, 27, 274-288.

Andreatta, G., and Romanin-Jacur, G. (1987) "Aircraft Flow Managment Under Congestion." *Transportation Science*, 21, 249-253.

Antosik, J.L. (1978) "Automatic Monthly Crew Assignment: A New Approach." *Proceedings of the Eighteenth AGIFORS Symposium.* Vancouver, Canada: AGIFORS.

Arabeyre, J.P., Fearnley, J., Steiger, F.C., and Teather, W. (1969) "The Airline Crew Scheduling Problem: A Survey." *Transportation Science*, 3, 140-163.

Baas, S.M., and Kwaakernaak, H. (1977) "Rating and Ranking of Multiple Aspect Alternative Using Fuzzy Sets." *Automatica*, 13, 47-58.

Baker, E.K., Bodin, L.D., and Fisher, M. (1985) "The Development and Implementation of a Heuristic Set Covering Based System for Air Crew Scheduling." *Transportation Policy Decision Making*, 3, 95-110.

Ball, M., and Roberts, A. (1985) “A Graph Partitioning Approach to Airline Crew Scheduling.” *Transportation Science*, 19, 107-125.

Bellman, R., and Zadeh, L. (1970) “Decision making in a fuzzy environment.” *Management Science*, 17, B144 - B164.

Belobaba, P.P. (1987a) “Airline Yield Management: An Overview of Seat Inventory Control.” *Transportation Science*, 21, 63–73.

Belobaba, P.P. (1987b) “Air Travel Demand and Airline Seat Inventory Management.” Ph.D. thesis, Flight Transportation Laboratory, Massachusetts Institute of Technology, Cambridge, MA.

Belobaba, P.P. (1989) “Application of a Probabilistic Decision Model to Airline Seat Inventory Control.” *Operations Research*, 37, 183–197.

Bezdek, J.C. (1993) “Fuzzy Models: What Are They and Why?” *IEEE Transactions on Fuzzy Systems*, 1, 1-6.

Bezdek, J.C. (1994) “The Thirsty Traveler Visits Gamont: A Rejoinder to ‘Comments on Fuzzy Sets - What are They and Why?’” *IEEE Transactions on Fuzzy Systems*, 2, 43-45.

Bezdek, J.C., and Pal, S.K. (1992) *Fuzzy Models for Pattern Recognition*. New York: IEEE Press.

Bhattacharya, U., Rao, J.R., and Tiwari, R.N. (1992) “Fuzzy Multi-Criteria Facility Location Problem.” *Fuzzy Sets and Systems*, 51, 277-287.

Bit, A.K., Biswal, M.P., and Alam, S.S. (1992) “Fuzzy Programming Approach to Multicriteria Decision Making Transportation Problem.” *Fuzzy Sets and Systems*, 50, 135-141.

Bodin, L., and Golden, B. (1981) “Classification in Vehicle Routing and Scheduling.” *Networks,* 11, 97-108.

Bodin, L., Golden, B., Assad, A., and Ball, M. (1983) “Routing and Scheduling of Vehicles and Crews: The State of the Art.” *Computers and Operations research*, 10, 63-211.

Bonissone, P.P. (1982) “A Fuzzy Sets Based Liguistic Approach:Theory and Applications.” In M.M. Gupta and E. Sanchez (eds.), *Approximate Reasoning in Decision Analysis* (pp. 329-339), Amsterdam: North-Holland.

Bortolan, G., and Degani, R. (1985) “A Review of Some Methods for Ranking Fuzzy Subsets.” *Fuzzy Sets and Systems*, 15, 1-19.

Brown, M., and Harris, C. (1994) *Neurofuzzy Adaptive Modelling and Control*. Englewood Cliffs, NJ: Prentice-Hall.

Buckley, J.J., and Hayashi, Y. (1994) “Fuzzy Neural Networks: A Survey.” *Fuzzy Sets and Systems*, 66, 1-13.

Buhr, J. (1978) “Four Methods for Monthly Crew Assignment - A Comparison of Efficiency.” *Proceedings of the Eighteenth AGIFORS Symposium*. Vancouver, Canada: AGIFORS.

Caplice, C., and Mahmassani, H. (1992) “Aspects of Commuting Behaviour: Preferred Arrival Time, Use of Information and Swithing Propensity.” *Transportation Research*, 26A, 409-418.

Carter, E., and Morlok, E. (1975) “Planning Air Transport Network in Appalachia.” *Transportation Engineering Journal of ASCE*, 101, 569-588.

Cerny,V. (1985) “Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm.” *Journal of Optimization, Theory and Applications*, 45, 41-55.

Chakroborthy, P. (1990) “Application of Fuzzy Set Theory to the Analysis of Capacity and Level of Service of Highways.” Master’s thesis, University of Delaware, Newark, DE.

Chakroborthy, P., and Kikuchi, S. (1990) “Application of Fuzzy Set Theory to the Analysis of Capacity and Level of Service of Highways.” In B.M. Ayyub (ed.) *Proceedings of*

ISUMA '90, The First International Symposium on Uncertainty Modeling and Analysis (pp. 146-150), College Park, MD: IEEE Computer Press.

Chan, Y., Regan, E.J., and Pan, W.M. (1986) "Inferring an Origin-Destination Matrix Directly from Network Flow Sampling." *Transportation Planning and Technology*, 11, 27-46.

Chanas, S., Delgado, M., Verdegay, J.I., and Vila, M.A. (1993) "Interval and Fuzzy Extensions of Classical Transportation Problems." *Transportation Planning and Technology*, 17, 203-218.

Chang, Y-H., and Shyu, T-H. (1993) "Traffic Signal Installation by the Expert System Using Fuzzy Set Theory for Inexact Reasoning." *Transportation Planning and Technology*, 17, 191-202.

Chao, C.-T., and Teng, C.-C. (1995) "Implementation of a Fuzzy Inference System Using a Normalized Fuzzy Neural Network." *Fuzzy Sets and Systems*, 75, 17-31.

Chen, L., May, A., and Auslander, D. (1990) "Freeway Ramp Control Using Fuzzy Set Theory for Inexact Reasoning." *Transportation Research*, 24A, 15-25.

Chen, S.H. (1985) "Ranking Fuzzy Numbers with Maximizing Set and Minimizing Set." *Fuzzy Sets and Systems*, 17, 113-129.

Chen, S.J., and Hwang, C.L. (1992) *Fuzzy Multiple Attribute Decision Making*. Berlin: Springer-Verlag.

Chin, S.M., Hwang, H.L., and Miaou, S.P. (1992) "Transportation Demand Forecasting with a Computer Simulated Neural Network Model ." In S.G. Ritchie and C. Hendrickson (eds.), *Proceedings of the International Conference on Artificial Intelligence Applications in Transportation Engineering*. San Buenaventura, CA: Institute of Transportation Studies, University of California, Irvine.

Chou, Y.H. (1990) "The Hierrarchical-Hub Model for Airline networks." *Transportation Planning and Technology*, 14, 243-258.

Christofides, N., and Korman, S. (1975) "A Computational Survey of Methods for the Set Covering Problem." *Management Science*, 21, 591-599.

Clarke, G., and Wright, J. (1964) "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points." *Operations Research*, 12, 568-581.

Croall, I.F., and Mason, J.P., (eds.) (1992) "*Industrial Applications of Neural Networks* ." ESPRIT Project 2092: Applications of Neural Networks for Industry in Europe (ANNIE Handbook). Berlin: Springer-Verlag.

Cullen,F., Jarvis, J., and Ratliff, H. (1981) "Set Partitioning Based Heuristics for Interactive Routing." *Networks*, 11, 125-144.

Curry, E.R. (1990) "Optimal Seat Allocation with Fare Classes Nested by Origins and Destinations." *Transportation Science*, 24, 193–204.

Cybenko, G. (1989) "Approximation by Superpositions of a Sigmoidal Function." *Mathematics of Control, Signals, and System*, 2, 303-314.

Czyzak, P., and Zak, J. (1995) "A Model of an Urban Transportation System Formulated as a Multiobjective Mathematical Programming Problem Under Uncertainty." *Journal of Advanced Transportation*, 29/1, 43-62.

Daganzo, C., and Sheffi, Y. (1977) "On Stochastic Models of Traffic Assignment." *Transportation Science*, 11, 253-274.

Darby-Dowman, K., and Mitra, G. (1985) "An Extension of Set Partitioning with Application to Scheduling Problems." *European Journal of Operational Research*, 21, 200-205.

De Ortuzar, J.D., and Willumsen, L.G. (1990) *Modelling Transport*. New York: Wiley.

Dombi, J. (1990) "Membership Function as an Evaluation." *Fuzzy Sets and Systems*, 35, 1-15.

Dougherty, M. (1995) "A Review of Neural Networks Applied to Transport." *Transportation Research C*, 1, 247-260.

Dougherty, M., and Joint, M. (1992) "A Behavioural Model of Driver Route Choice Using Neural Networks." In S.G. Ritchie and C. Hendrickson (eds.), *Proceedings of the International Conference on Artificial Intelligence Applications in Transportation Engineering.* San Buenaventura, CA: Institute of Transportation Studies, University of California, Irvine.

Dror, M., Laporte, G., and Trudeau, P. (1989) "Vehicle Routing with Stochastic Demands: Properties and Solution Frameworks." *Transportation Science*, 23, 166-176 .

Dror, M., and Powell, W. (1993) "Stochastic and Dynamic Models in Transportation." *Operations Research*, 41, 11-14.

Dror, M., and Trudeau, P. (1986) "Stochastic Vehicle Routing with Modified Savings Algorithm." *European Journal of Operational Research*, 23, 228-235.

Dubois, D., and Prade, H. (1983) "Ranking of Fuzzy Numbers in the Settings of Possibility Theory." *Information Science*, 30, 183-224.

Dubois, D., and Prade, H. (1984) "Criteria Agregation and Ranking of Alternatives in the Framework of Fuzzy Set Theory." In H.J. Zimmermann (ed.), *TIMS / Studies in the Management Science*, 20, (pp. 209-240), Amsterdam: Elsevier Science Publishers, North-Holland.

Dubois, D., and Prade, H. (1988) *Possibility Theory.* New York : Plenum.

Efstathiou, J., and Rajkovic, V. (1979) "Multiattribute Decision-making Using a Fuzzy Heuristic Approach." *IEEE Transactions on Systems, Man and Cybernetics*, SMC-9, 326-333.

Efstathiou, J., and Tong, R. (1982) "Ranking Fuzzy Sets: A Decision Theoretic Approach." *IEEE Transactions on Systems, Man and Cybernetics*, SMC-12, 655-659.

Er, M.C. (1988) "Decision Support Systems: A Summary, Problems, and Future Trends." *Decision Support Systems*, 4, 355-363.

Etcheberry, J. (1977) "The Set Covering Problem: A New Implicit Enumeration Algorithm." *Operations Research*, 25, 760-772.

Fincke, U. and Hamacher, H.W. (1987) "On the Expected Value of Stohastic Linear Programs and (Dynamic) Network Flow Problems." *European Journal of Operations Research*, 29, 307-330 .

Florian, M.A. (1976) *Traffic Equilibrium Methods. Lecture Notes in Economics and Mathematical Systems.* Vol. 118, Berlin: Springer-Verlag.

Flynn, J., and Ratick, S. (1988) "A Multiobjective Hierrarchical Covering model for the Essential Air Services Program." *Transportation Science*, 22, 139-147.

Fogel, L.J., Owens, A.J., and Walsh, M.J. (1996) *Artificial Intelligence through Simulated Evolution.* New York: Wiley.

Gamache, M., and Soumis. F. (1993) "A Method for Optimally Solving the Rostering Problem." Cahier du GERAD, G-90-40, Ecole des Hautes Etudes Commerciales, Montreal, Canada, H3T1V6 .

Gasco, J.L. (1977) "Reservations and Booking Control." *Proceedings of the Seventeenth AGIFORS Symposium.* Bad Kissingen, Germany: AGIFORS.

Gasco, J.L. (1980) "Report on the Reservations Study Group." *Proceedings of the twentieth AGIFORS Symposium.* New Delhi, India: AGIFORS.

Gerbracht, R. (1978) "A New Algorithm for Very Large Crew Pairing Problems." *Proceedings of the Eighteenth AGIFORS Symposium.* Vancouver, Canada: AGIFORS.

Gerbracht, R. (1979) "Optimum Overbooking Levels." *Proceedings of the Nineteenth AGIFORS Symposium.* Pugnochiuso, Italy: AGIFORS.

Ghafouri, M., and Lam, T. (1986) "Accessibility in the Deregulated Domestic Airline Network." *Transportation Research Record*, 1094, 10-17.

Ghobrial, A.A. (1983) "Analysis of the Air Network Structure: The Hubbing Phenomenon." Ph.D. thesis, University of California, Berkeley.

Giafferri, J., Hamon, J.P., and Lengline, S. (1982) "Automatic Monthly Assignment of Medium-Haul Cabin Crew." *Proceedings of the Twenty-second AGIFORS Symposium.* Bad Kissingen, Germany: AGIFORS.

Gillett, B., Miller, L. (1974) "A Heuristic Algorithm for the Vehicle Dispatch Problem" *Operations Research,* 22, 340-352.

Glaneert, W. (1984) "A Timetable Approach to the Assignment of Pilots to Rotations." *Proceedings of the Twenty-fourth AGIFORS Symposium.* Strasbourg, France: AGIFORS.

Glover, F. (1986) "Future Paths for Integer Programming and Links to Artificial Intelligence." *Computers and Operations Research,* 5, 533-549.

Glover, F. (1989) "Tabu Search, Part I." *ORSA Journal on Computing* 2(1), 4-32.

Glover, F. (1990) "Tabu Search, Part II." *ORSA Journal on Computing* 2(1), 4-32.

Glover, F. and Laguna, M. (1993) "Tabu Search" In C.R. Reeves (ed.), *Modern Heuristic Techniques for Combinatorial Problems.* (pp. 70-150), Oxford: Blackwell Scientific.

Glover, F., Glover, R., Lorenzo, J., and McMillan, C. (1982) "The Passenger Mix Problem in the Scheduled Airlines." *Interfaces,* 12, 73 – 79.

Goldberg, D. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning.* Reading, MA: Addison-Wesley.

Golden, B., and Assad, A., (eds.). (1988) *Vehicle Routing: Methods and Studies.* Amsterdam: North-Holland.

Golden, B., and Skicsim, C. (1985), "Using Simulated Annealing to Solve Routing and Locations Problems." Working paper MS/S 85-010, University of Maryland at College Park.

Gordon, S., and de Neufville, R. (1973) "Design of Air Transportation Networks." *Transportation Research,* 7, 207-222.

Grove, M.A. (1988) "A Surrogate for Linear Programs with Random Requirements." *European Journal of Operations Research,* 34, 399-402.

Gupta, M.M., and Rao, D.H. (1994), "Invited Review: On the Principles of Fuzzy Neural Networks." *Fuzzy Sets and Systems,* 61, 1-18.

Halgamuge, S.K., and Glesner, M. (1994), "Neural Networks in Designing Fuzzy Systems for Real World Applications." *Fuzzy Sets and Systems,* 65, 1-12.

Hall, C., and Smith, R. (1992), "Pitfalls in the application of neural networks for process control." In K.Warwick, G.W.Irwin, and K.J.Hunt (eds.), *Neural Networks for Control and Systems.* (pp. 243-256), IEE Control Engineering Series 46.

Hall, R. (1989) "Configuration of an Overnight Package Air Network." *Transportation Research,* 23A, 139-149.

Hall, R. (1993) "Non-Recurrent Congestion: How Big Is the Problem? Are Traveler Information Systems the Solution?" *Transportation Research* , 1C, 89-103.

Hamacher, H., Leberling, H., and Zimmermann, H.J. (1978) "Sensitivity Analysis in Fuzzy Linear Programming." *Fuzzy Sets and Systems,* 1, 269-281.

Hannan, E.L. (1981) "Linear Programming with Multiple Fuzzy Goals." *Fuzzy Sets and Systems,* 6, 235-248.

Hansen, P., "The Steepest Ascent Mildest Descent Heuristic for Combinatorial Programming." Paper presented at the Congress on Numerical Methods in Combinatorial Optimization, Capri, Italy.

Hassin, R., and Zemel, E. (1988) "Probabilistic Analysis of Capacitated Transportation Problems." *Mathematics of Operations Research,* 13, 80-89.

Holland, J. (1975) *Adaptation in Natural and Artificial Systems.* Ann Arbor: University of Michigan Press.

Homafair, A., and McCormick, V.E. (1992) "Full Design of Fuzzy Controllers Using Genetic Algorithms." In *SPIE Conference Neural Stochastic Methods Image Signal Process* 1766, 393-404.

Homafair, A., and McCormick, E. (1995) "Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controllers Using Genetic Algorithms." *IEEE Transactions on Fuzzy Systems*, 3, 129-139.

Hopfield, J. (1982) "Neural Networks and Physical Systems with Emergent Collective Computational Abilities." *Proceedings of the National Academy of Sciences of the United States of America*, 79, 2554-2558.

Horikawa, S., Furahashi, T., and Uchikawa, Y. (1992) "On Fuzzy Modeling Using Fuzzy Neural Networks with Backropagation Algorithm." *IEEE Transactions on Neural Networks*, 3, 801-806.

Hormann, A. (1987) "ATSAM (Air Traffic Simulation Analysis Model)." Paper presented at The III Mini Euro Conference, Herceg Novi, Yugoslavia.

Hornik, K., Stinchcombe, M., and White, H. (1989), "Multilayer Feedforward Networks are Universal Approximators." *Neural networks*, 2, 359-366.

Hsiao, C.-H., Lin, C.-T., and Cassidy, M. (1994) "Application of Fuzzy Logic and Neural Networks to Automatically Detect Freeway Traffic Incidents." *Journal of Transportation Engineering*, 120, 753-772.

Hsu, G.J.Y., and Tzeng, L.Y.-R. (1990) "A New Algorithm of Fuzzy Multiobjective Programming: The Compromise Factor Approach." *Proceedings of ISUMA '90, The First International Symposium on Uncertainty Modeling and Analysis* (pp. 689-693), College Park, MD: IEEE Computer Press.

Hua, J., and Faghri, A. (1993), "Dynamic Traffic Pattern Classification Using Artificial Neural Networks." *Transportation Research Record*, 1399, 14-19.

Hunt, J.G., and Lyons, G.D. (1994), "Modelling Dual Carriageway Lane Changing Using Neural Networks." *Transportation Research C*, 2C(4), 231-245.

Hwang, C.L., and Yoon, K. (1981) *Multiple Attribute Decision Making-Methods and Applications, A State-of-the-Art Survey*. New York: Springer Verlag.

Iida, Y., Akiyama, T., and Uchida, T. (1992) "Experimental Analysis of Dynamic Route Choice Behaviour." *Transportation Research*, 26B, 17-32.

Iniuguchi, M., Ichihashi, H. and Tanaka, H. (1990) "Fuzzy Programming: A Survey of Recent Developments." In R. Slowinski, and J. Teghem (eds.), *Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty.* (pp. 45-68), Kluwer Academic Publishers.

Jaikumar, R., and Solomon, M.M. (1987) "The Tug Fleet Size Problem for Barge Line Operations: A Polynomial Algorithm." *Transportation Science*, 21, 264-272.

Jain, R. (1976) "Decision Making in the Presence of Fuzzy Variables." *IEEE Transactions on Systems, Man and Cybernatics*, SMC-6, 698-703.

Jang, J.R.S. (1992) "Self-learning Fuzzy Controllers Based on Temporal Back-propagation." *IEEE Transactions on Neural Networks*, 3, 714-723.

Jang, J.R.S. (1997), *Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Englefood Cliffs, NJ: Prentice-Hall.

Jang, J.R.S., and Sun, C.T. (1995), "Neuro-Fuzzy Modeling and Control." *Proceedings of the IEEE*, 83, 378-406.

Jaw, J.J., Odoni, A.R., Psaraftis, H.N., and Wilson, H.N.M. (1986) "A Heuristic Algorithm for the Multi-vehicle Advance Request Dial-a-Ride Problem with Time-Windows." *Transportation Research*, 20B, 243-257.

Jia, L-M., and Zhang, X-D. (1994) "Distributed Intelligent Railway Traffic Control Based on Fuzzy Decisionmaking." *Fuzzy Sets and Systems*, 62, 255-265.

Johnson, D.S., Lenstra, J.K., and Rinnoy Kan, A.H.G. (1978) "The Complexity of the Network Design Problem." *Networks* , 8, 279-285.

Kabbara, G. (1982) "New Utilization of Fuzzy Optimization Method." In M. M. Gupta, and E. Sanchez (eds.), *Fuzzy Information and Decision Processes.* (pp. 239-246), Amsterdam: North-Holland.

Kagaya, S., Kikuchi, S., and Donelly, R. (1994) "Use of Fuzzy Theory Technique for Grouping of Trips in Vehicle Routing and Scheduling Problem." *European Journal of Operational Research,* 76, 143-154.

Kalic, M. (1994) "Designing an Airline Network of the Air Carrier." Master's thesis, Faculty of Transport and Traffic Engineering, University of Belgrade (in Serbian).

Kalic, M., Pavkovic, G., and Teodorovic, D. (1992) "Estimation of the Number of Telephone Impulses by Neural Networks." *Proceedings of the Twenty-first Yugoslav Conference on Operations Research* (pp. 212-216), Belgrade (in Serbian).

Kanafani, A. (1981) "Aircraft Technology and Network Structure in Short-Haul Air Transportation." *Transportation Research,* 15A, 305-314.

Kanafani, A. (1983) *Transportation Demand Analysis.* New York: McGraw-Hill.

Kanafani, A., and Al-Deek, H. (1991) "A Simple Model for Route Guidance Benefits." *Transportation Research,* 25B, 191-201.

Kanafani, A., and Chang.R. (1979) "A System Analysis of Short-Haul Air Transportation." Research Report, Institute of Transportation Studies, University of California, Berkeley.

Kanafani, A., and Ghobrial, A. (1982) "Aircraft Evaluation in Air Network Planning." *Transportation Engineering Journal of ASCE,* 108, 282-300.

Kanafani, A., and Ghobrial, A. (1985) "Airline Hubbing: Some Implications for Aircraft Economics." *Transportation Research,* 19A, 15 - 27.

Kasabov, N.K. (1996) *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering.* Cambridge, MA: MIT Press.

Kaufmann, A., and Gupta, M. (1985) *Introduction to Fuzzy Arithmetic.* New York: Van Nostrand Reinhold.

Kaufmann, A., and Gupta, M. (1988) *Fuzzy Mathematical Models in Engineering and Management Science.* New York: Elsevier Science.

Keller,J.M., Gray, M.R., and Givens, J.A. (1985) "A Fuzzy K-nearest Neighbor Algorithm." *IEEE Transactions on Systems, Man and Cybernetics,* 15, 580-585.

Kerre, E.E. (1982) "The use of fuzzy set theory in electrocardiological diagnostics" In M.M. Gupta and E. Sanchez (eds.), *Approximate Reasoning in Decision Analysis* (pp. 277-282), Amsterdam: North-Holland.

Khattak, A., Kanafani, A., and Le Colletter, E. (1994) "Stated and Reported Diversion Behaviour: Implication on the Benefits of ATIS." California PATH Research Report, UCB-ITS-PRR-94-13.

Kickert, W.J.M., and van Nauta Lemke, H.R. (1976) "Application of a Fuzzy Controller in a Warm Water Plant." *Automatica,* 12, 301-308.

Kikuchi, S. (1984) "Scheduling of Demand Responsive Transit Vehicles." *Journal of Transportation Engineering,* 110, 511-520.

Kikuchi, S. (1987) "Vehicle Routing and Scheduling Development for Transportation of Elderly and Handicapped Persons." *Transportation Quaterly,* 41, 207-227.

Kikuchi, S. (1988) "Development of Computerized Vehicle Scheduling System for Specialized Transportation: Delaware Case." *Transportation Planning and Technology,* 12, 151-170.

Kikuchi, S. (1992) "Scheduling Demand-Responsive Transportation Vehicles using Fuzzy-Set Theory." *Journal of Transportation Engineering,* 118, 391-409.

Kikuchi, S., Nanda, R., and Perincherry, V. (1993) "A Method to Estimate Trip O-D Patterns Using a Neural Network Approach." *Transportation Planning and Technology*, 17, 51-65.

Kikuchi, S., Nanda, R., and Perincherry, V. (1994) "Estimation of Trip Generation Using the Fuzzy Regression Method." Paper submitted for presentation at the 1994 Annual Meeting of the Transportation Research Board and for publication in *Transportation Research Record.*

Kikuchi, S., and Perincherry,V. (1990) "A Fuzzy Approach to the Transshipment Problem." In B.M. Ayyub (ed.) *Proceedings of ISUMA '90, The First International Symposium on Uncertainty Modeling and Analysis* (pp. 330-335), College Park, MD: IEEE Computer Press.

Kikuchi, S., Perincherry,V., Chakroborthy, P., and Takahashi, H. (1994) "Modeling of Driver Anxiety During Signal Change Intervals." *Transportation Research Record*, 1399, 27-35.

Kikuchi, S., and Rhee, J.H. (1989) "Scheduling Method for Demand Responsive Transportation System." *Journal of Transportation Engineering of ASCE*, 115, 630-645.

Kikuchi, S., and Vukadinovic, N. (1994) "Grouping Trips by Fuzzy Similarity for Scheduling of Demand-Responsive Transportation Vehicles." *Transportation Planning and Technology*, 18, 65-80 .

Kikuchi, S., Vukadinovic, N., and Easa, S.M. (1991) "Characteristics of the Fuzzy LP Transportation Problem for Civil Engineering Application." *Civil Engineering Systems*, 8, 134-144.

Kirby, M.G. (1980) "Optimal Pricing Policies for Air Transport Networks: Some Comments on Economics Regulation and the Australian Domestic Air Industry." *Transportation Research*, 14A, 275-278.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1983) "Optimization by Simulated Annealing." *Science*, 220, 671-680.

Klir, G., and Folger, T. (1988) *Fuzzy Sets, Uncertainty and Information.* Englewood Cliffs, NJ: Prentice-Hall.

Kohonen, T. (1982) "Self-organized Formation of Topologically Correct Feature Maps." *Biological Cybernetics 43*, 59-69.

Kong, S-G., and Kosko, B. (1992) "Adaptive Fuzzy Systems for Backing up a Truck-and-Trailer." *IEEE Transactions of on Neural Networks*, 3(2), 211-223.

Kosko, B. (1992a) "Fuzzy Systems as Universal Approximators." *Proceedings of the First IEEE International Conference on Fuzzy Systems* (March), (pp. 1153-1162).

Kosko, B. (1992b) *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Englefood Cliffs, NJ: Prentice-Hall.

Kosko, B. (1994) "The Probability Monopoly." *IEEE Transactions on Fuzzy Systems*." 2, 32-33.

Kosko, B. (1993) *Fuzzy Thinking*. New York: Hyperion.

Koutsopoulos, H., and Xu, H. (1993) "An Information Discounting Routing Strategy for Advanced Traveler Information System." *Transportation Research*, 1C, 249-264.

Kruse, R., and Meyer, K. (1987) *Statistics with Vague Data*. Dodrecht: D. Reidel.

Lai, Y-J., and Hwang, C-L. (1992) "Interactive Fuzzy Linear Programming." *Fuzzy Sets and Systems*, 45, 169-183.

Larkin, L. (1985) "A Fuzzy Logic Controller for Aircraft Flight Control." In M. Sugeno (ed.), *Industrial Applications of Fuzzy Control* (pp. 87-103), New York: Elsevier Science.

Larson, R., and Odoni, A. (1981) *Urban Operations Research*. Englefood Cliffs, NJ: Prentice-Hall.

Larson, R.C., Minkoff, A., and Gregory, P. (1988), "Fleet Sizing and Dispatching for the Marine Division of The New York City Department of Sanitation." In B.L. Golden and A.A. Assad (eds.) *Vehicle Routing Methods and Studies* New York: Elsevier Science.

Le Blanc, L.J. (1975) "An Algorithm for the Discrete Network Design Problem." *Transportation Science*, 9, 183-199.

Leberling, H. (1981) "On Finding Compromise Solutions in Multicriteria Problems Using the Fuzzy Min-Operator." *Fuzzy Sets and Systems*, 6, 105-118.

Le Cun, Y. (1985) "A Learning Procedure for Assymetric Threshold Network." *Proceedings of Cognitiva*, 85, 599-604.

Lee, A.O. (1990) "Airline Reservations Forecasting: Probabilistic and Statistical Models of the Booking Process." Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Lee, K.-M., Kwak, D.-H., and Leekwang, H. (1995) "Tuning of Fuzzy Models by Fuzzy Neural Networks." *Fuzzy Sets and Systems*, 76, 47-61.

Lee, M.A., and Tagaki, J. (1993) "Integrating Design Stages of Fuzzy Systems Using Genetic Algorithms". In *Proceedings IEEE International Conference on Fuzzy Systems FUZZ-IEEE '93* (pp. 612-617).

Lemke, C.E., Salkin, H.M., and Spielberg, K. (1971) "Set Covering by Single-Branch Enumeration with Linear-Programming Subproblems." *Operations Research*, 19, 998-1022.

Lin, C.-T., and Lee, C.S.G. (1991) "Neural-Network-Based Fuzzy Logic Control and Decision System." *IEEE Transactions on Computers*, 40, 1320-1336.

Lin, C.-T., and Lee, C.S.G. (1994) "Reinforcement Structure/Parameter Learning for Neural-Network-Based Fuzzy Logic Control Systems." *IEEE Transactions on Fuzzy Systems*, 2, 46-63.

Lin, C.T., and Lee, C.S.G. (1995) *Neural Fuzzy System: A Neuro-fuzzy Synergism to Intelligent Systems*. Englefood Cliffs, NJ: Prentice-Hall.

Lin, Y., and Cunningham III, G.A. (1995) "A New Approach to Fuzzy-Neural System Modeling." *IEEE Transactions on Fuzzy Systems*, 3, 190-198.

Lindley, D.V. (1994) "Comments on The Efficacy of Fuzzy Representations of Uncertainty." *IEEE Transactions on Fuzzy Systems*, 2, 37.

Lippmann, R.P. (1987) "An Introduction to Computing with Neural Nets." *IEEE ASSP Magazine* (April), 4-22.

Littlewood, K. (1970) "Aircrew Flight Sequencing Methods for Planning and Bid Line Construction." *Proceedings of the Tenth AGIFORS Symposium.* Terrigal, Australia: AGIFORS.

Littlewood, K. (1972) "Forecasting and Control of Passenger Bookings." *Proceedings of the Twelfth AGIFORS Symposium*, Nathanya, Israel: AGIFORS.

Lotan, T., and Koutsopoulos, H. (1993a) "Models for Route Choice Behaviour in the Presence of Information Using Concepts from Fuzzy Set Theory and Approximate Reasoning." *Transportation*, 20, 129-155.

Lotan, T., and Koutsopoulos, H. (1993b) "Route Choice in the Presence of Information Using Concepts from Fuzzy Control and Approximate Reasoning." *Transportation Planning and Technology*, 17, 113-126.

Magnanti, T.L., and Wong, R.T. (1984) "Network Design and Transportation Planning: Models and Algorithms." *Transportation Science*, 18, 1-55.

Mahmassani, H., and Shen-Te Chen, P. (1993) "An Investigation of the Reliability of Real-time Information for Route Choice Decisions in a Congested Traffic System." *Transportation*, 20, 157-178.

Mak, C. (1992) "Revenue Impacts of Airline Yield Management." M.S. thesis, Flight Transportation Laboratory, Massachusetts Institute of Technology, Cambridge, MA.

Mamdani, E. (1974) "Application of Fuzzy Algorithms for Control of Simple Dynamic Plant." *Proceedings IEE*, 212, 1585-1588.

Mamdani, E. (1981) "Advances in the Linguistic Synthesis of Fuzzy Controllers" In E.H. Mamdani and B.R. Gaines (eds.), *Fuzzy Reasoning and Applications.* London: Academic Press.

Mamdani, E. and Assilian, S. (1975) "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller." *International Journal of Man-Machine Studies*, 7, 1-13.

Marchettini, F. (1980) "Automatic Monthly Cabin Crew Rostering Procedure." *Proceedings of the Twentieth AGIFORS Symposium.* New Delhi, India: AGIFORS.

Mareš, M., and Horak, J. (1983) "Fuzzy Quantities in Networks." *Fuzzy sets and Systems*, 10, 13-22.

Marsten, R.E. (1974) "An Algorithm for Large Set Partitioning Problems." *Management Science*, 20, 774-787.

Marsten, R.E., and Shepardson, F. (1981) "Exact Solution of Crew Scheduling Problems Using the Set Partitioning Model: Recent Successful Applications." *Networks*, 11, 165-177.

Másson, E., and Wang, Y-J. (1990) "Introduction to Computation and Learning in Artificial Neural Networks." *European Journal of Operational Research*, 47, 1-28 .

McCulloch, W.S., and Pitts, W. (1943) "A Logical Calculus of the Ideas Immanent in Nervous Activity." *Bulletin of Mathematical Biophysics*, 5, 115-133.

Mendel, J.M. (1995) "Fuzzy Logic Systems for Engineering: A Tutorial." *Proceedings of the IEEE*, 83, 345-377.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., and Teller, A. (1953) "Equation of State Calculations by Fast Computing Machines." *Journal of Chemical Physics*, 21, 1087-1092.

Milosavljevic, N., Teodorovic, D., Papic, V., and Pavkovic, G. (1996) "A Fuzzy Approach to the Vehicle Assignment Problem." *Transportation Planning and Technology*, 20, 33-47.

Minkoff, A. (1985) "Real-time Dispatching of Delivery Vehicles." PhD Thesis in Operations Research, Massachusetts Institute of Technology, Cambridge, MA.

Minkoff, A. (1993) "A Markov Decision Model and Decomposition Heuristic for Dynamic Vehicle Dispatching." *Operations Research*, 41, 77-90.

Moore, R., .Evans, J., and .Noo, H. (1978) "Computerized Tailored Blocking." *Proceedings of the Eighteenth AGIFORS Symposium.* Vancouver, Canada: AGIFORS.

Nakatsuyama, M., Nagahashi, N., and Nishizuka, N. (1983) "Fuzzy Logic Phase Controller for Traffic Functions in the One-way Arterial Road." *Proceedings IFAC Ninth Triennial World Congress* (pp. 2865-2870), Oxford: Pergamon Press.

Nanda, R. (1991) "Application of Fuzzy Set Theory and Neural Networks to Regression and GMDH Analysis." Master of science thesis, University of Delaware, Newark.

Ndoh, N.N., and Ashford, N.J. (1994) "Evaluation of Transportation Level of Service Using Fuzzy Sets." Paper presented at the Seventy-third Annual Meeting, Transportation Research Board, Washington, D.C.

Nicoletti, B. (1975) "Automatic Crew Rostering." *Transportation Science*, 9, 33-42.

Odoni, A. (1987) "The Flow Management Problem in Air Traffic Control." In A. R. Odoni, L. Bianco, and G. Szego (eds.), *Flow Control of Congested Networks.* New York: Springer Verlag.

Østergaard, J.J. (1976) "Fuzzy Logic Control of a Heat Exchanger Process." Internal report, Electric Power Engineering Department, Technical University of Denmark, Lyngby, 318-362.

Pappis, C., and Mamdani, E. (1977) "A Fuzzy Controller for a Traffic Junction." *IEEE Transactions on Systems, Man and Cybernetics*, ASMC 7, 707-717.

Parker, D.B. (1985) *Learning-logic* (TR-47) Massachusetts Institute of Technology, Cambridge, MA.

Pavkovic, G. (1993) "Traffic Signal Coordination by Fuzzy Logic" Working paper, Faculty of Transport and Traffic Engineering, University of Belgrade (in Serbian).
Pavkovic,G. (1994) "Vehicle Routing Problem in the Case of Uncertain Demands at Nodes." Master of science thesis, Faculty of Transport and Traffic Engineering, University of Belgrade (in Serbian).
Pawlak, Z. (1991) *Rough Sets.* Boston: Kluwer.
Perincherry, V. (1990) "Application of Fuzzy Set Theory to Linear Programming." Master of science thesis, University of Delaware, Newark.
Perincherry, V., and Kikuchi, S. (1990) "A Fuzzy Approach to the Transshipment Problem" In B.M. Ayyub (ed.) *Proceedings of ISUMA '90, The First International Symposium on Uncertainty Modeling and Analysis.* College Park, MD: IEEE Computer Press.
Pierce, J.F. (1968) "Application of Combinatorial Programming to a Class of All-Zero-One Integer programming Problems." *Management Science*, 15, 91-209.
Pierce, J.F., and Lasly, J.S. (1973) "Improved Combinatorial Programming Algorithms for a Class of All-Zero-One Integer Programming Problems." *Management Science*, 19, 528-543.
Pollack, M. (1982) "Airline Route-Frequncy Planning: Some Design Trade-Offs." Transportation Research, 16A, 149-159.
Poorzahedy, H., and Turnquist, M.A. (1982) "Approximate Algorithms for the Discrete Network Design Problem." *Transportation Research*, 16B, 45-55.
Powell, W.B. (1987) "An Operational Planning Model for the Dynamic Vehicle Allocation Problem with Uncertain Demands." *Transportation Research*, 21B, 217-232.
Procyk, T.J., and Mamdani, E.H. (1979) "A Linguistic Self-organising Process Controller." *Automatica*, 15, 15-30.
Psaraftis, H. (1980) "A Dynamic Programming Solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem." *Transportation Science*, 2, 130-154.
Psaraftis, H. (1983) "An Exact Algorithm for the Single Vehicle Many-to Many Dial-A-Ride problem with Time Window." *Transportation Science*, 17, 351-357.
Putz, A., and Weber, R. (1996) "Automatic Adjustment of Weights in Fuzzy Rule Bases." In *EUFIT'96*, (September 2-5) (pp. 933-938).
Quadrado, J.C., and Quadrado, A.,F. (1996) "Fuzzy Modeling of Accessibility: Case Study Lisbon Metropolitan Area." *Proceedings of the Fourth European Congress on Intelligent Techniques and Soft Computing* (pp. 1307-1311), Aachen, Germany.
Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.
Rhee, J.H. (1987) "Vehicle Routing and Scheduling Strategies for Demand Responsive Transportation Systems." Ph.D. thesis, University of Delaware, Newark, Delaware.
Ritchie, S.G., and Cheu, R.L. (1993) "Simulation of Freeway Incident Detection Using Artificial Neural Networks." *Transportation Research C*, 1(3), 203-217.
Rumelhart, D.E., and McClelland, J.L. (1986) *Parallel Distributed Processing: Explorations in the microstructure of cognition.* (vols. 1 and 2). Cambridge, MA: MIT Press.
Rutenbar (1989) "Simulated Annealing Algorithms: An Overview." *IEEE Circuits and Devices*, 5, 19-26.
Rosenblatt, F. (1958) "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." *Psychology Review*, 65, 386-408.
Ross, T.J. (1995) *Fuzzy Logic with Engineering Applications.* New York: McGraw-Hill.
Rothstein, M. (1971a) "An Airline Overbooking Model." *Transportation Science*, 5, 180–192.
Rothstein, M. (1971b) "Airline Overbooking: The State of the Art." *Journal of Transport Economics and Policy*, 5, 96–99.

Rothstein, M. (1975) "Airline Overbooking: Fresh Approaches Are Needed." *Transportation Science*, 9, 169–173.
Rothstein, M. (1985) "OR and the Airline Overbooking Problem." *Operations Research*, **33**, 237 – 248.
Rothstein, M., and Stone,W. (1967) "Passenger Booking Levels." In *Proceedings of the Seventh Agifors Symposium.* Terrigal, Australia: AGIFORS.
Rubin, J. (1973) "Technique for the Solution of Massive Set Covering Problems with Application to Airline Crew Scheduling." *Transportation Science*, 7, 34-48.
Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986) "Learning Internal Representations by Error Propagation." In D.E. Rumelhart and J.L. McClelland (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (vol. 1).
Rutenbar, R. (1989) "Simulated Annealing Algorithms: An Overview." *IEEE Circuits and Devices*, 5, 19-26.
Rutherford, D., and Carter,G.A. (1976) "A Heuristic Adaptive Controller for a Sinter Plant." in *Proceedings of the Secind IFAC Symposium Automation in Mining, Mineral and Metal Processing*, Johannesburg.
Ryan, D.M. (1992) "The Solution of Massive Generalized Set Partitioning Problems in Air Crew Rostering." *Journal of the Operational Research Society*, 43, 459-467.
Sakawa, M. (1983) "Interactive Fuzzy Decision Making for Multiobjective Nonlinear Programming Problems." *Interactive Decision Analysis, Lecture Notes in Economics and Mathematical Systems. Proceedings*, Laxenburg, Austria, (vol. 229), 105-112.
Sakawa, M. (1993) "Fuzzy Sets and Interactive Multiobjective Optimization." In M.G. Singh (ed.), *Applied Information Technology.* New York: Plenum Press.
Sakawa, M., and Yano, H. (1986) "An Interactive Fuzzy Satisficing Method using Constraint Problems and its Application to Regional Planning." *Kybernetes*, 15, 121-129.
Sakawa, M. and Yano, H. (1990) "An Interactive Fuzzy Satisficing Method for Generalized Multiobjective Linear Programming Problems with Fuzzy Parameters." *Fuzzy Sets and Systems*, 35,125-142.
Sakawa, M., Yano, H., and Yumine, T. (1987) "An Interactive Fuzzy Satisficing Method for Multiobjective Linear-Programming Problems and its Application." *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-17, 654-661.
Sarra, D. (1988) "The automatic assignment model." In *Proceedings of the Twenty-eight AGIFORS Symposium.* Vancouver, Canada: AGIFORS.
Sasaki, T., and Akiyama, T. (1986) "Development of Fuzzy Traffic Control System on Urban Expressway." *Preprints of the Fifth IFAC/IFIP/IFORS International Conference in Transportation Systems* (pp. 333-338),Vienna, Austria: IFAC.
Sasaki, T., and Akiyama, T. (1987) "Fuzzy On-Ramp Control Model on Urban Expressway and Its Extension" In N.H. Gartner and N.H.M. Wilson (eds.), *Transportation and Traffic Theory* (pp. 377-395), New York: Elsevier Science.
Sasaki, T., and Akiyama, T. (1988) "Traffic Control Process of Expressway by Fuzzy Logic." *Fuzzy Sets and Systems*, 26, 165-178.
Sayed, T., Abdelwahab, W., and Navin, F. (1995) "Identifying Accident-Prone Locations Using Fuzzy Pattern Recognition." *Journal of Transportation Engineering*, 121, 352-358.
Sayers, T., and Bell, M.H.G. (1996) "Traffic Responsive Signal Control Using Fuzzy Logic-A Practical Modular Approach." *Proceedings of the Fourth European Congress on Intelligent Techniques and Soft Computing* (pp. 2159-2163), Aachen, Germany.
Schretter, N., and Hollatz, J. (1996) "A Fuzzy Logic Expert System for Determining the Required Waiting Period After Traffic Accidents." *Proceedings of the Fourth European Congress on Intelligent Techniques and Soft Computing* (pp. 2164-2170), Aachen, Germany.

Schwieterman, J., and Spencer, F. (1986) "Alternatives to the Hub: A Survey of Nonstop Air Service Opportunities." *Transportation Research Record*, 1094, 1-9.
Self, K. (1990) "Designing with Fuzzy Logic." *IEEE Spectrum*, 105, 42-44.
Sexton, T. (1979) "The Single Vehicle Many to Many Routing and Scheduling Problem." Ph.D. Ttesis, State University of New York at Sony Brook.
Sexton, T., and Bodin, L. (1985a) "Optimizing Single Vehicle Many-to-Many Operations with Desired Delivery Times I: Scheduling." *Transportation Science*, 19, 378-410.
Sexton, T., and Bodin, L. (1985b) "Optimizing Single Vehicle Many-to-Many Operations with Desired Delivery Times II: Routing." *Transportation Science*, 19, 411-435.
Sforza, A. (1989) "An Optimization Approach for Decision Support in Railway Traffic Control." in P. Korhonen, A. Lewandowski, and J. Wallenius (eds.), *Proceedings Multiple Criteria Decision Support*. Helsinki: Springer-Verlag.
Shafer, G. (1976) *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton University Press.
Sheffi, Y. (1985) *Urban Transportation Networks*. Englefood Cliffs, NJ: Prentice-Hall.
Shen, Y., Potvin, J.Y., Roy, S., and Rousseau, J.-M. (1993) "Integrating Operations Research and Artificial Intelligence Techniques for Vehicle Dispatching." Working paper CRT-939, University of Montreal, Canada.
Shlifer, E., and Vardi, Y (1975) "An Airline Overbooking Policy." *Transportation Science*, 9, 101 – 114.
Smith, B.C., Leimkuhler, J.F., and Darrow, R.M. (1992) "Yield Management at American Airlines." *Interfaces*, 22, 8 – 31.
Smith, P.N. (1993) "Fuzzy Evaluation of Potential Suburban railway Station Locations." *Journal of Advanced Transportation*, 27, 153-179.
Solomon, M., and Desrosiers, J. (1988) "Time Window Constrained Routing and Scheduling Problems." *Transportation Science*, 22, 1-13.
Soumis, F., Ferland, J.A., and Rousseau, J.M. (1981) "MAPUM: A Model for Assigning Passengers to a Flight Schedule." *Transportation Research*, 15A, 155–162.
Stein, D. (1978) "Scheduling Dial-a-Ride Transportation Systems." *Transportation Science*, 12, 232-249.
Sugeno, M. (ed.). (1985) *Industrial Applications of Fuzzy Control*. Amsterdam: North-Holland.
Sugeno, M., and Murakami, K. (1985) "An Experimental Study on Fuzzy Parking Control Using a Model Car" In M. Sugeno (ed.), *Industrial Applications of Fuzzy Control*. Amsterdam: Elsevier Science.
Sugeno, M., and Nishida, M. (1985) "Fuzzy Control of Model Car." *Fuzzy Sets and Systems*, 16, 103-113.
Sugeno, M., and Yasukawa, T. (1993) "A Fuzzy-Logic Based Approach to Qualitative Modeling." *IEEE Transactions on Fuzzy Systems*, 1, 7-31.
Swan, W.M. (1979) "A Systems Analysis of Scheduled Air Transportation Networks." Report FTL-R795, Massachusetts Institute of Technology, Cambridge, MA.
Tagaki, T., and Sugeno, M. (1985) "Fuzzy Identification of Systems and Its Applications to Modeling and Control." *IEEE Transaction on Systems, Man and Cybernetics*, SMC-15, 116-132.
Tanaka, H., and Asai, K. (1984) "Fuzzy Linear Programming Problems with Fuzzy Numbers." *Fuzzy Sets and System*, 13, 1–10.
Tanaka, H., Ichihashi, H., and Asai, K. (1986) "A Value of Information in FLP Problems via Sensitivity Analysis." *Fuzzy Sets and Systems*, 18, 119-129.
Teodorovic, D. (1983) "Flight Frequency Determination." *Journal of Transportation Engineering*, 109, 747-757.

Teodorovic, D. (1985) "Multicriteria Ranking of Air Shuttle Alternatives." *Transportation Research*, 19B, 63-72.
Teodorovic, D. (1988) *Airline Operations Research*. New York: Gordon and Breach Science.
Teodorovic, D. (1994) "Invited Review: Fuzzy Sets Theory Applications in Traffic and Transportation." *European Journal of Operational Research,* 74, 379-390.
Teodorovic, D. (1998) "Airline Network Seat Inventory Control: Fuzzy Set Theory Approach." In *Transportation Planning and Technology.*
Teodorovic, D., and Babic, O. (1993) "Fuzzy Inference Approach to the Flow Management Problem in Air Traffic Control." *Transportation Planning and Technology*, 17, 165-178.
Teodorovic, D., and Guberinic S. (1984) "Optimal Dispatching Strategy on an Airline Network After a Schedule Perturbation." *European Journal of Operational Research,* 15, 178-182.
Teodorovic, D., and Kalic, M. (1995) "A Fuzzy Route Choice Model for Air Transportation Networks." *Transportation Planning and Technology*, 19, 109-120.
Teodorovic, D., and Kalic, M. (1996) "Solving the Modal Split Problem by Fuzzy Rules Generated by Learning from Examples." In *Proceedings Information Technologies,* Zabljak (in Serbian).
Teodorovic, D., Kalic, M., and Pavkovic, G. (1994) "The Potential for Using Fuzzy Set Theory in Airline Network Design." *Transportation Research*, 28B, 103-121.
Teodorovic, D., and S. Kikuchi (1990) "Transportation Route Choice Model Using Fuzzy Inference Technique" In B.M. Ayyub (ed.) *Proceedings of ISUMA '90, The First International Symposium on Uncertainty Modeling and Analysis* (pp. 140-145), College Park, MD: IEEE Computer Press.
Teodorovic, D., and Kikuchi, S. (1991a) "Application of Fuzzy Sets Theory to the Saving Based Vehicle Routing Algorithm." *Civil Engineering Systems,* 8, 87-93.
Teodorovic, D., and Kikuchi, S. (1991b) *Fuzzy Sets with Transportation Engineering Applications*. Faculty of Transport and Traffic Engineering, University of Belgrade, Belgrade (in Serbian).
Teodorovic, D., and Krcmar-Nozic, E. (1989) "Milticriteria Model to Determine Flight Frequencies on an Airline Network Under Competetive Conditions." *Transportation Science,* 23, 14-25.
Teodorovic, D., Krcmar-Nozic, E., and Stojkovic, G. (1993) "Airline Seat Inventory Control: An Application of the Simulated Annealing Method." *Transportation Planning and Technology*, 17, 219–233.
Teodorovic, D., and Lucic, P. (1998) "A Fuzzy Set Theory Approach to the Aircrew Rostering Problem." *Fuzzy Sets and Systems*, 95, 261-271.
Teodorovic, D., and Pavkovic, G. (1992) "A Simulated Annealing Technique Approach to the Vehicle Routing Problem in the Case of Stochastic Demand." *Transportation Planning and Technology*, 16, 261-273.
Teodorovic, D., and Pavkovic, G. (1996) "The Fuzzy Set Theory Approach to the Vehicle Routing Problem when Demand at Nodes is Uncertain." *Fuzzy Sets and Systems*, 82, 307-317.
Teodorovic, D., and Stojkovic, G. (1990) "Model of Operational Daily Airline Scheduling." *Transportation Planning and Technology*, 14, 273-285.
Teodorovic, D., and Stojkovic, G. (1995) "Model to Reduce Airline Schedule Disturbances." *Journal of Transportation Engineering*, 121, 324-331.
Teodorovic, D., Vukanovic, S., and Obradovic, K. (1998) "Modeling Route Choice With Advanced Traveler Information by Fuzzy Logic." In *Transportation Planning and Technology.*

Terrab, M. (1990) "Ground Handling Strategies for Air Traffic Control." Ph. D. thesis, Department of Civil Engineering, Massachusetts Institute of Technology.

Tingley, G. A. (1979) "Still Another Solution Method for the Monthly Aircrew Assighment Problem." *Proceedings of the Nineteenth AGIFORS Symposium.* Pugnochiuso, Italy: AGIFORS.

Tong, R.M. (1976a) "Analysis of Fuzzy Control Algorithms Using the Relation Matrix." *International Journal of Man-machine Studies*, 8, 679-686.

Tong, R., M. (1976b) "Some Problems with the Design and Implementation of Fuzzy Controllers." Internal report CUED/F-CAMS/TR 127, Cambridge University Engineering Department.

Tong, R.M. (1977) "A Control Engineering Review of Fuzzy Systems." *Automatica*, 13, 559-569.

Tovey, G. (1988) "Simulated Annealing." *American Journal of Mathematical and Management Sciences*, 8, 389-407.

Tzeng, G.H., and Teng, J-Y. (1993) "Transportation Investment Project Selection with Fuzzy Multi-objectives." *Transportation Planning and Technology*, 17, 91-112.

Tzeng, G.-H., Teodorovic, D., and Hwang, M-J. (1996) "Fuzzy Bicriteria Multi-index Transportation Problems for Coal Allocation Planning of Taipower." *European Journal of Operational Research*, 95, 62-72 .

Tzeng, G-H., and Teng, J-Y. (1990) "Multiobjective Decision Making with Fuzzy Problems for Infrastracture Investment Planning: A Case of Transportation Investment." In B.M. Ayyub (ed.) *Proceedings of ISUMA '90, The First International Symposium on Uncertainty Modeling and Analysis* (pp. 635-640), College Park, MD: IEEE Computer Press.

Vukadinovic, N. (1992) "Fuzzy Relation Based Vehicle Scheduling Algorithm: Application to the Demand Responsive System." Master of science thesis, University of Delaware, Newark.

Vukadinovic, K. (1993) "Fuzzy Approach to the Inland Water Transport Process." Master of Science Thesis, Faculty of Transport and Traffic Engineering, University of Belgrade (in Serbian).

Vukadinovic, K. (1997) "A Neuro and Fuzzy Control of Inland Water Transportation with Application to Other Modes of Transportation." Ph.D. thesis, Faculty of Transport and Traffic Engineering, University of Belgrade (in Serbian).

Vukadinovic, K., and Teodorovic, D. (1994) "A Fuzzy Approach to the Vessel Dispatching Problem." *European Journal of Operational Research,* 76, 155-164.

Vukadinovic, K., Teodorovic, D., and Pavkovic, G. (1997) "A Neural Network Approach to the Vessel Dispatching Problem." *European Journal of Operational Research*, 102, 473-487.

Vukadinovic, K., Teodorovic, D. and Pavkovic, G. (1998) "An Application of Neurofuzzy Modeling: The Vehicle Assignment Problem." In *European Journal of Operational Research.*

Vukadinovic, K., Teodorovic, D., Pavkovic, G., and Rosic, S. (1996) "A Neural Network Approach to Mitigation of Vehicle Schedule Disturbances." *Transportation Planning and Technology*, 20, 93-102.

Vythoulkas, P.C., and Koutsopoulos, H.N. (1994) "Modeling Discrete Choice Behaviour Using Concepts from Fuzzy Set Theory, Approximate reasoning and Neural Networks." Research Report TSU 817, ESRC Transport Studies Unit, University of Oxford.

Waley, P. (1991) *Statistical reasoning with Imprecise Probabilities.* London: Chapman and Hall.

Wang, J. (1994) "A Neural Network Approach to Modeling Fuzzy Preference Relations for Multiple Criteria Decision Making." *Computers and Operations Research*, 21, 991-1000.

Wang, J., and Malakooti, B. (1992) "A Feedforward Neural Network for Multiple Criteria Decision Making." *Computers and Operations Research*, 19, 151-167.

Wang, K. (1983) "Optimum Seat Allocation for Multi-Leg Flights with Multiple Fare Types." *Proceedings of the Twenty-third AGIFORS Symposium.* Olive Branch, Mississippi: AGIFORS.

Wang, L-X., and Mendel., J. (1992a) "Back-Propagation of Fuzzy Systems as Nonlinear Dynamic System Identifiers." *Proceedings IEEE International Conference on Fuzzy Systems* (pp. 807-813), San Diego.

Wang, L.-X., and Mendel., J. (1992b) "Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least Squares Learning." *IEEE Transactions on Neural Networks*, 3, 807-813.

Wang, L-X. and Mendel, J. (1992c) "Generating Fuzzy Rules by Learning from Examples." *IEEE Transactions on systems, Man and Cybernetics*, 22, 1414-1427.

Wang, S., and Archer, N.P. (1994) "A Neural Network Technique in Modeling Multiple Criteria Multiple Person Decision Making." *Computers and Operations Research*, 21, 127-142.

Wang, Z., and Klir, G. (1992) *Fuzzy Measure Theory*. New York: Plenum.

Wasserman, P., D. (1989) *Neural Computing: Theory and Practice*. New York: Van Nostrand Reinhold.

Weatherfold, L.R., and Bodily, S.E. (1992) "A Taxonomy and Research Overview of Perishable - Asset Revenue Management: Yield Management, Overbooking and Pricing." *Operations Research*, 40, 831–844.

Wei, C.H., and Schonfeld, P.M. (1993) "An Artificial Neural Network Approach for Evaluating Transportation Network Improvements." *Journal of Advanced Transportation*, 27, 129-151.

Wenstop, F. (1976) "Fuzzy Set Simulation Models in a Systems Dynamics Perspective." *Kybernetes*, 6, 209-218.

Wong, J.T., Koppelman, F.S., and Daskin, M.S. (1993) "Flexible Assignment Approach to Itinerary Seat Allocation." *Transportation Research*, 27B, 33–48.

Xu, W., and Chan, Y. (1993a) "Estimating An Origin-Destination Matrix with Fuzzy Weights Part 1: Methodology." *Transportation Planning and Technology*, 17, 127-144.

Xu,W., and Chan, Y. (1993b) "Estimating An Origin-Destination Matrix with Fuzzy Weights Part 2: Case Studies." *Transportation Planning and Technology*, 17, 145-164.

Yager, R. (1981) "A Procedure for Ordering Fuzzy Subsets of the Unit Interval." *Information Science*, 24, 143-161.

Yager, R. (1982) "Some Procedures for Selecting Fuzzy Set-Theoretic Operators." *International Journal of General System*, 8, 115—124.

Yamazaki, T., and Sugeno, M. (1985) "A Microprocessor Based Fuzzy Controller for Industrial Purposes" In M. Sugeno (ed.), *Industrial Applications of Fuzzy Control*, New York: Elsevier Science.

Yang, H., Akiyama, T., and Sasaki, T. (1992) "A Neural Network Approach to the Identification of Real Time Origin-Destination Flows From Traffic Counts" In S.G. Ritchie and C. Hendrickson (eds.), *Proceedings of the International Conference on Artificial Intelligence Applications in Transportation Engineering* (pp. 253-269), San Buenaventura, CA: Institute of Transportation Studies, University of California, Irvine.

Yang, H., Kitamura, R., Jovanis, P., Vaughn, K., Abdel-Aty, M., and Reddy, P. (1993) "Exploration of Route Choice Behaviour with Advanced Traveler Information Using Neural Network Concept." *Transportation*, 20, 199-223.

Zadeh, L (1965) "Fuzzy Sets." *Information and Control*, 8, 338-353 .

Zadeh, L. (1972) "A Fuzzy-Set-Theoretic Interpretation of Linguistic Hedges." *Journal of Cybernetics*, 2, 4-34.

Zadeh, L. (1973) "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes." *IEEE Transactions on Systems, Man and Cybernetics*, SMC 3, 28-44.

Zadeh, L. (1975) " The Concept of a Linguistic Variable and its Application to Approximate Reasoning: I." *Information Science*, 8, 199-249

Zadeh, L. (1975) "The Concept of a Linguistic Variable and its Application to Approximate Reasoning: II." *Information Science*, 8, 301-357.

Zadeh, L. (1996) "Fuzzy Logic = Computing with Words." *IEEE Transactions on Fuzzy systems*, 4, 103-111.

Zimmermann, H.-J. (1976) "Description and Optimization of Fuzzy Systems." *International Journal of General Systems*, 2, 209-215.

Zimmermann, H.-J. (1978) "Fuzzy Programing and Linear Programing with Several Objective Functions." *Fuzzy Sets and Systems*, 1, 45-55.

Zimmermann, H.-J. (1987) *Fuzzy Sets and Decision Making and Expert Systems*. Boston: Kluwer.

Zimmermann, H.-J. (1986) "Fuzy Set Theory and Mathematical Programming." In A. Jones et al. (eds.), *Fuzzy Sets Theory and Applications* (pp. 99-114), Dodrecht: D. Reidel.

Zimmermann, H.-J. (1991) *Fuzzy Set Theory and Its Applications*. Boston: Kluwer.

Zimmermann, H.-J. (1992) "Methods and Applications of Fuzzy Mathematical Programming." In R.R. Yager and L. Zadeh (eds.), *An Introduction to Fuzzy Logic Applications in Intelligent Systems* (pp. 97-120), Boston: Kluwer.

Zionts, S. (1992) "Multiple Criteria decision making: The challenge that lies ahead." In G.H. Tzeng et al. (eds.), *Multiple Criteria Decision Making, Proceedings of the Tenth International Conference: Expand and Enrich the Domains of Thinking and Application* (pp. 17-26), Berlin: Springer-Verlag.

Index